UNIVERSIDAD DE COSTA RICA
SISTEMA DE ESTUDIOS DE POSGRADO

ELABORACIÓN DE UN PLAN DE IMPLEMENTACIÓN DE INTEGRACIÓN CONTINUA UTILIZANDO SOFTWARE LIBRE EN EMPRESAS CON PRESUPUESTO LIMITADO QUE UTILIZAN METODOLOGÍAS ÁGILES

Trabajo final de investigación aplicada sometido a la consideración de la Comisión del Programa de Estudios de Posgrado en Tecnologías de Información y Comunicación para la Gestión Organizacional, para optar al grado y título de Maestría Profesional en Tecnologías de Información y Comunicación para la gestión Organizacional

LUIS ALFREDO ROMERO ÁLVAREZ

Ciudad Universitaria Rodrigo Facio, Costa Rica

2019

Dedicatoria

Dedicado a mis padres y hermana, que con gran esfuerzo, me acompañaron en cada meta que he conseguido.

Agradezco especialmente a mi novia, por su apoyo y comprensión en este proceso, quien me ha ayudado a salir adelante en los momentos más difíciles.

Agradecimientos

Agradezco a Dios por darme la fuerza y sabiduría para afrontar el trayecto de este trabajo.

Además, agradezco a mis profesores de maestría, cuya dedicación ha permitido obtener los conocimientos para realizar de la mejor manera este proyecto.

Finalmente, un agradecimiento especial a todas las personas, cuyo aporte fue fundamental en el desarrollo de este proyecto.

"Este trabajo final de investigación aplicada fue aceptado por la Comisión del Programa de Estudios de Posgrado en Tecnologías de Información y Comunicación para la Gestión Organizacional de la Universidad de Costa Rica, como requisito parcial para optar al grado y título de Maestría Profesional en Tecnologías de Información y Comunicación para la Gestión Organizacional."

FRANCISCO JOSE BLANCO CHAVARRIA (FIRMA)
Firmado digitalmente por FRANCISCO JOSE BLANCO CHAVARRIA (FIRMA)
Fecha: 2020.05.13 17:57:25 -06'00'

M.Sc. Francisco Blanco Chavarría
Representante del Decano Sistema de Estudios de Posgrado

ALEJANDRO ULATE CAMPOS (FIRMA)
Digitally signed by ALEJANDRO ULATE CAMPOS (FIRMA)
Date: 2020.05.13 20:07:45 -06'00'

M.Sc. Alejandro Ulate Campos
Profesor Guía

VERNI GERARDO FERNANDEZ CASTRO (FIRMA)
Firmado digitalmente por VERNI GERARDO FERNANDEZ CASTRO (FIRMA)
Fecha: 2020.05.13 21:14:48 -06'00'

M.Sc. Verni Fernández Castro

Lector

WALTER FELIPE JENKINS CRUZ (FIRMA)
Firmado digitalmente por WALTER FELIPE JENKINS CRUZ (FIRMA)
Fecha: 2020.05.13 17:15:39 -06'00'

M.Sc. Felipe Jenkins Cruz
Lector

YORLENY SALAS ARAYA (FIRMA)
Firmado digitalmente por YORLENY SALAS ARAYA (FIRMA)
Fecha: 2020.05.13 22:37:24 -06'00'

M.Sc. Yorleny Salas Araya
Directora del Programa de Posgrado en Tecnologías de Información y Comunicación para la Gestión Organizacional

LUIS ALFREDO ROMERO ALVAREZ (FIRMA)
Firmado digitalmente por LUIS ALFREDO ROMERO ALVAREZ (FIRMA)
Fecha: 2020.05.12 17:16:58 -06'00'

Luis Alfredo Romero Álvarez
Sustentante

Tabla de contenido

Dedicatoria ii

Agradecimientos iii

Resumen vii

Abstract viii

Lista de cuadros ix

Lista de ilustraciones ix

Capítulo I: Planteamiento del tema 1

Antecedentes 1

Objetivos 1

Justificación 2

Capítulo II: Marco teórico 3

Objetivo de la investigación 3

Definiciones 3

Calidad de software 3

Integración continua 3

Automatización de pruebas unitarias 3

Metodologías ágiles 4

Integración continua en la organización 4

Características de la integración continua 5

Beneficios 6

Desventajas 7

Casos de éxito 7

Herramientas de integración continua 7

Automatización 8

Consideraciones 8
Cultura de integración 8
Capítulo III: Marco metodológico 9
Enfoque del proyecto 9
Naturaleza del proyecto 9
Alcance del proyecto 9
Diseño de investigación 10
Métodos y técnicas de recolección de información 10
Sujetos y fuentes de información 10
Técnicas e instrumentos de investigación 11
Procedimiento metodológico 11
Capítulo IV: Análisis y selección de herramientas 14
Sistema operativo 14
Base de datos 17
Repositorio de control de versiones 20
Aplicación de Integración Continua 23
Requerimientos generales 26
Capítulo V: Plan de implementación 27
Etapa 1: Preparación 28
Etapa 2: Despliegue de las herramientas 31
Etapa 3: Cierre 36
Capítulo VII: Conclusiones y Recomendaciones 43
Bibliografía 45
Anexos 50

Resumen

Este trabajo de investigación surge a partir de la problemática que se da en el ciclo de vida del desarrollo de software, especialmente, en la detección y corrección de errores en las etapas tempranas y, adicionalmente, por el proceso de integración de los subproductos de trabajo.

Ante dicha problemática, el objetivo de la investigación se centra en desarrollar una propuesta de mejora en el proceso de desarrollo de software, basado en herramientas gratuitas para la implementación de integración continua en organizaciones, con limitado presupuesto que utilizan metodologías ágiles, con el fin de mitigar y disminuir el impacto de la integración ayudando, así, a encontrar errores en las etapas tempranas.

Por tal motivo, es que surge la importancia de analizar este fenómeno, desde una perspectiva que permita descubrir, qué aspectos generan problemas a la hora de poder realizar entregas cada vez más depuradas y con un mínimo de correcciones posibles, que se traduzca en mejores tiempos de desarrollo y menos errores en la implementación.

Palabras clave:

Integración Continua, Metodologías ágiles, Desarrollo de Software

Abstract

This research work, arises from the problem that occurs in the life cycle of software development, especially in the detection and correction of errors in the early stages and, additionally, by the process of integration of work by-products.

Given this problem, the objective of the research is to develop a proposal for improvement in the software development process, based on free tools for the implementation of continuous integration in organizations with limited budgets that use agile methodologies, in order to mitigate and reduce the impact of integration thus helping to find errors in the early stages.

For this reason, it is that the importance of analyzing this phenomenon arises, from a perspective that allows us to discover, what aspects generate problems when it is possible to deliver more and more refined deliveries and with a minimum of possible corrections, which translates into better times of development and fewer errors in implementation.

Keywords:

Continuous Integration, Agile methodologies, Software development

Lista de cuadros

Cuadro 1. Guía de operacionalización de los objetivos 12
Cuadro 2. Características de los sistemas operativos 17
Cuadro 3. Características de las bases de datos 20
Cuadro 4. Características de los repositorios de control de versiones 22
Cuadro 5. Características de las aplicaciones de Integración Continua 25

Lista de ilustraciones

Ilustración 1: Esquema del plan de implementación 27

Lista de abreviaturas

A

ABP

Aprendizaje Basado en Problemas

AE

Arquitectura Empresarial

APIs

Application Programming Interface

Interfaz de Programación de Aplicaciones

B

BBVA

Banco Bilbao Vizcaya Argentaria

I

ISO

International Organization for Standardization

ISTQB

International Software Testing Qualifications Board

L

LTS

Long Term Support - Soporte de largo plazo

UNIVERSIDAD DE COSTA RICA

SEP Sistema de Estudios de Posgrado

Autorización para digitalización y comunicación pública de Trabajos Finales de Graduación del Sistema de Estudios de Posgrado en el Repositorio Institucional de la Universidad de Costa Rica.

Yo, Luis Alfredo Romero Álvarez, con cédula de identidad 3-0448-0670, en mi condición de autor del TFG titulado Elaboración de un plan de implementación de integración continua utilizando software libre en empresas con presupuesto limitado que utilizan metodologías ágiles

Autorizo a la Universidad de Costa Rica para digitalizar y hacer divulgación pública de forma gratuita de dicho TFG a través del Repositorio Institucional u otro medio electrónico, para ser puesto a disposición del público según lo que establezca el Sistema de Estudios de Posgrado. SI [X] NO *[ ]

***En caso de la negativa favor indicar el tiempo de restricción: _____ año (s).**

Este Trabajo Final de Graduación será publicado en formato PDF, o en el formato que en el momento se establezca, de tal forma que el acceso al mismo sea libre, con el fin de permitir la consulta e impresión, pero no su modificación.

Manifiesto que mi Trabajo Final de Graduación fue debidamente subido al sistema digital Kerwá y su contenido corresponde al documento original que sirvió para la obtención de mi título, y que su información no infringe ni violenta ningún derecho a terceros. El TFG además cuenta con el visto bueno de mi Director (a) de Tesis o Tutor (a) y cumplió con lo establecido en la revisión del Formato por parte del Sistema de Estudios de Posgrado.

INFORMACIÓN DEL ESTUDIANTE:

Nombre Completo: Luis Alfredo Romero Álvarez

Número de Carné: A85670 Número de cédula: 3-0448-0670

Correo Electrónico: luis_romero_a@outlook.com

Fecha: 08/05/2020 Número de teléfono: 8804-7185

Nombre del Director (a) de Tesis o Tutor (a): M.C.I. Alejandro Ulate Campos

LUIS ALFREDO ROMERO ALVAREZ (FIRMA)

Firmado digitalmente por LUIS ALFREDO ROMERO ALVAREZ (FIRMA)
Fecha: 2020.05.08 19:33:20 -06'00'

FIRMA ESTUDIANTE

Nota: El presente documento constituye una declaración jurada, cuyos alcances aseguran a la Universidad, que su contenido sea tomado como cierto. Su importancia radica en que permite abreviar procedimientos administrativos, y al mismo tiempo genera una responsabilidad legal para que quien declare contrario a la verdad de lo que manifiesta, puede como consecuencia, enfrentar un proceso penal por delito de perjurio, tipificado en el artículo 318 de nuestro Código Penal. Lo anterior implica que el estudiante se vea forzado a realizar su mayor esfuerzo para que no solo incluya información veraz en la Licencia de Publicación, sino que también realice diligentemente la gestión de subir el documento correcto en la plataforma digital Kerwá.

Capítulo I: Planteamiento del tema

Antecedentes

El proceso de desarrollo de software siempre ha estado cargado de problemas durante todo el ciclo. Desde los requerimientos cambiantes, hasta la atención de los problemas que surgen en la etapa de validación. La fase de integración es un asunto largo, que normalmente puede conllevar a introducir más problemas a un sistema en desarrollo (Fowler, 2006) (Duvall, Matyas, & Glover, 2007).

Esto aumenta cuando los desarrolladores trabajan de forma aislada, construyendo componentes por tarea, que al final del desarrollo, van agregando al producto final. Desde esta perspectiva, se puede tomar en cuenta si un proyecto tiene aproximadamente un año de vida de construcción, por ejemplo, su integración generará mucho trabajo tanto para los integradores, los expertos en validación y, posteriormente, a los desarrolladores que repararan los problemas hallados en el sistema.

No obstante, es importante recordar que hay empresas que implementan estrategias y no llevan un control completo de sus fuentes. Esto es un factor que provoca problemas a la hora de darle mantenimiento a las soluciones a través del tiempo, lo que genera gastos en la interpretación o la creación de una nueva versión.

Objetivos

General

Evaluar el proceso de desarrollo de software, en una empresa que utilice metodologías ágiles y que cuente con limitado presupuesto, con el fin de implementar la integración continua mediante herramientas de software libre.

Específicos

1. Seleccionar las herramientas de software libre más idóneas para implementar integración continua, en una empresa, mediante el análisis de requerimientos de hardware y procesos desarrollo de software de la empresa piloto.
2. Elaborar un plan de implementación, que permita a una empresa piloto desplegar las herramientas de software libre seleccionadas, con el fin de maximizar los procesos de desarrollo de software, mediante integración continua.
3. Elaborar instrumentos que permitan evaluar el nivel de satisfacción en el proceso de desarrollo de software en la empresa piloto, después de la implementación de la integración continua, mediante entrevistas y cuestionarios a los involucrados.

Justificación

La importancia de esta investigación radica en generar recomendaciones que brinden, a cualquier organización, la posibilidad de estructurar una solución de integración continua acorde a sus posibilidades financieras; apoyándose en opciones de software libre existentes, que le permitan adaptar el funcionamiento de las herramientas con el modelo organizacional.

Además, de promover algunos valores que incentiven la inclusión de los integrantes de un grupo de desarrolladores, con el fin de trabajar juntos en el beneficio de la organización, lo cual intentará provocar una cultura de integración en el personal, que permita llevar la administración de los productos de software y la rápida atención de los problemas que puedan surgir en el proceso de desarrollo.

En resumen, lo que se busca con esta investigación es proveer a una organización las herramientas suficientes para generar un entorno cultural en sus equipos de desarrollo, que permitan alinear el personal con los objetivos de la empresa, visualizados en la mejora del proceso de desarrollo apoyado con la integración continua, la atención más rápida de los problemas de las etapas de desarrollo de software y la creación más ágil de productos que la gerencia pueda visualizar.

Capítulo II: Marco teórico

En este capítulo se encuentran las definiciones técnicas necesarias para la comprensión de esta investigación.

Objetivo de la investigación

La investigación se centrará en las ventajas que provee la implementación de la integración continua en los procesos de desarrollo de software. Para ello, se realizará un análisis de herramientas gratuitas que permitan a una organización implementar una opción de bajo costo, pero que brinde a su vez la retroalimentación suficiente para poder afirmar que cumplen un proceso de integración continua. Esto implica que no se abordará el proceso de puesta en marcha o lo que se define como despliegue continuo.

Definiciones

Calidad de software

Según se indica en la norma ISO (International Organization for Standardization, 2014) la calidad del software está dada por el grado de satisfacción de las necesidades explícitas o implícitas, cuando se usa en condiciones determinadas para lo cual fue hecho.

Integración continua

Como indica Fowler (2006), la integración continua es una práctica donde los desarrolladores integran su trabajo con frecuencia, mínimo una vez por día. Estas se verifican por compilaciones automáticas donde se detectarán errores.

Amazon (Amazon, s.f.), lo define como una práctica en la que los desarrolladores combinan sus cambios en un repositorio central, ejecutan versiones y pruebas automáticas, para encontrar errores con mayor rapidez, mejorar la calidad del software y reducir el tiempo de validación y liberación de nuevas versiones.

Automatización de pruebas unitarias

La ISTQB (International Software Testing Qualifications Board, s. f.), en su repertorio de significados, define las pruebas unitarias como el método que permite evaluar un componente o unidad de forma aislada y que cumple de forma satisfactoria el escenario de

la prueba y la automatización de pruebas, como la utilización de software para apoyar o realizar actividades de pruebas.

Metodologías ágiles

Graffius (2016) y Martel (2014) sintetizan la definición como un conjunto de procesos disciplinados, los cuales fomentan la adaptación, el liderazgo, la auto-organización y la responsabilidad de un equipo. Este permite tener un enfoque de negocios alineado a las necesidades del cliente y los objetivos de la organización.

Un estudio de CollabNet (2019) demuestra un incremento del uso de las metodologías ágiles a nivel mundial, siendo una de las más utilizadas SCRUM y algunas variantes híbridas de esta, Kanban y Extreme Programing le siguen en la lista. Este y otros autores, Project Management Institute (2018) y Priolo (2009), concuerdan en estas tres metodologías como las más usadas actualmente, mostrando el valor que aportan al proceso de desarrollo y la mejora organizacional.

Software libre

La Free Software Foundation (2019) define el software libre como todo aquel programa que los usuarios tienen la libertad de ejecutar, copiar, distribuir, estudiar, modificar y mejorar. Estos son desarrollados y mantenidos por una gran comunidad de personas que se encargan de preservar el ambiente de las mismas. Además, cuentan con bastante documentación, grupos que dan asesoría y ayudan a resolver problemas. La mayoría pueden ser accedidos sin costo y pueden ser modificados para adaptarse a las necesidades de la organización.

Integración continua en la organización

Fowler (2006) y Duvall, Matyas, & Glover (2007) plantean que la mejor forma de iniciar la incorporación de integración continua, es con un servidor que tenga el repositorio de código fuente, el cual debe mantenerlo de forma íntegra todo el tiempo y, debe permitir que cualquier desarrollador pueda realizar una copia del código fuente sobre la que realizará su trabajo.

Al finalizar su trabajo, el desarrollador realiza la compilación de su copia de fuente y ejecuta las pruebas unitarias. Las pruebas unitarias son creadas por el desarrollador y pueden ser supervisadas por el encargado de validación o por algún otro desarrollador ajeno al cambio para no sesgar las pruebas. Cuando la compilación y las pruebas se ejecutan de forma correcta, se actualiza la copia local y se vuelve a compilar para evitar conflictos con los cambios de otros desarrolladores. En caso de fallo, es responsabilidad del desarrollador trabajar esos problemas con el fin de generar un producto que se pueda integrar sin error.

Una vez que se logra obtener un producto estable, es posible integrar este producto con la versión que está en el repositorio. No obstante, una vez que se integra, se debe realizar nuevamente el proceso de compilación y ejecución de pruebas unitarias, para asegurar que el producto es estable y funcional, lo anterior, con el fin de verificar que no haya conflicto con los cambios de otros desarrolladores, pérdida de archivos o contenido a la hora de integrar.

Fitz (2009) y Humble & Farley (2010) agregan un par de cosas más que podrían ayudar a elevar la calidad y disminuir los errores que se pueden presentar en los sistemas. Entre ellas, menciona que las pruebas automatizadas ayudan bastante a encontrar problemas, aunque no habrá una prueba totalmente capaz de evaluar todas las posibilidades de escenarios con usuarios que pueden existir. Además de las revisiones de código, éstas aumentarán la calidad del software, evitarán defectos y educarán a los desarrolladores; no obstante, son humanos y podrían fallar en las revisiones.

Características de la integración continua

El proceso de integración continua se compone de los siguientes elementos:

- Un repositorio de código fuente.
- Una copia del código fuente en la máquina del desarrollador (es).
- Un conjunto de pruebas unitarias que respaldan los cambios realizados.
- Una actualización con la última versión del código fuente del repositorio.
- Una compilación en la máquina del desarrollador.
- Responsabilidad sobre los conflictos que puedan surgir.

- Protección de los cambios en el repositorio.
- Una compilación de la versión original con los cambios y todas las pruebas.
- Responsabilidad sobre los conflictos después de integrar con la copia original.

Asimismo, Pedro Galván (2017) indica que para que todo funcione, es necesario establecer algunas políticas que deben cumplir los integrantes del equipo, entre estas están:

- Registrar (commit) su código al menos una vez al día.
- No registrar código con errores.
- No registrar código sin probar.
- No generar nuevas versiones con código que no funcionan.
- No pueden terminar sus actividades del día hasta haber registrado su código y que el sistema se integre exitosamente.

Al integrar estas políticas en la jornada diaria, se va convirtiendo en parte de la cultura de los desarrolladores y, con ello, se genera un sentimiento de responsabilidad por el producto que se está creando en conjunto con el equipo de trabajo.

Beneficios

Pedro Galván (2017) define los beneficios de la integración continua como los siguientes:

- Reducir problemas de integración.
- Mejorar la visibilidad del estatus del producto de software.
- Acelerar la detección de fallas.
- Disminuir el tiempo dedicado a depurar errores.
- Evitar la espera para averiguar si un código funciona.

Los puntos anteriores son sumamente importantes porque la reducción de los problemas de integración en etapas tempranas, brinda la posibilidad de que se reduzcan los tiempos de entrega del producto, la capacidad de realizar pequeñas entregas funcionales que los usuarios puedan utilizar y, así, reducir la resistencia al cambio. Además, la reducción de los fallos que se pueden presentar por los posibles escenarios no cubiertos en las pruebas, del tiempo dedicado a la atención y búsqueda de la resolución de los problemas.

Desventajas

En el proceso de integración continua podemos identificar las siguientes desventajas:

- Es necesario un servidor dedicado a la integración continua.
- Ante el compromiso de no ingresar errores a la línea base, es posible que no se pueda integrar como es requerido por lo menos una vez al día.

Aunque parece poco, es posible que algunas de las organizaciones puedan considerarlo como una restricción para implementarlo debido al costo que pueda incurrir. Además, la adopción de la cultura de realizar integraciones de versiones estables, por lo menos una vez al día, puede ser al inicio un problema para los integrantes del equipo de trabajo, los cuales pueden experimentar cierta resistencia al proceso.

Casos de éxito

Amazon (Amazon, s.f.), una empresa dedicada a las ventas en línea, ha iniciado un proceso de desarrollo y venta de aplicaciones, con el principio de integración y entrega continua. Brinda el servicio que permite a otras organizaciones la implementación de la solución en sus negocios.

Asimismo, BBVA (2017) es una entidad financiera que ha cambiado la forma de los negocios bancarios al desarrollar APIs financieras con la ayuda de Docker, una aplicación que apoya la integración y despliegue continuo. Por el gran renombre que tiene esta entidad, se ha dedicado al desarrollo y comercialización de APIs financieras, con el fin de ayudar la conexión de otras empresas con las herramientas financieras.

Herramientas de integración continua

Un par de herramientas pioneras en la Integración Continua son:

- Jenkins (Jenkins, s.f.) que es un servidor autónomo de código abierto, que se puede usar para automatizar todo tipo de tareas relacionadas con la implementación de software. Soporta integración con varias herramientas de control de versiones.
- CruiseControl (Java, .Net) (CruiseControl, s.f.) es una aplicación de código abierto desarrollada por ThoughtWorks, que puede generar reportes en diferentes tipos de presentación.

Automatización

Pedro Galván (2017) resume los requisitos en cuatro puntos:

- Tener un repositorio maestro, donde esté disponible todo el código fuente y del que cualquier integrante del equipo pueda obtenerlo.
- Automatizar el proceso de integración para que cualquier persona pueda generar una versión ejecutable del sistema a partir del código fuente.
- Automatizar las pruebas para que sea posible ejecutar la matriz (suite) de pruebas en cualquier momento con un solo comando.
- Asegurar que cualquiera puede obtener el ejecutable más reciente y tener la confianza de que es la mejor versión hasta el momento.

A partir de estos puntos, y con la elección de alguna de las herramientas de integración continua, podría permitir iniciar la implementación de la solución para una organización.

Consideraciones

Para realizar la implementación, es necesaria la incorporación de un servidor de integración dedicado, con el fin de darle fiabilidad a este proceso. Es importante incluir algún sistema de notificaciones, que permita a los desarrolladores conocer el estado de las compilaciones y su oportuna retroalimentación del proceso.

Cultura de integración

Como lo indican Galván (2017) y Fowler (2006), los integrantes del equipo de trabajo necesitan sentir una responsabilidad con el trabajo que están realizando. La incorporación de las políticas que indica Galván puede provocar un sentimiento de unión con el equipo y, puede motivar la participación y colaboración de los integrantes en un esfuerzo conjunto por realizar las cosas de la mejor forma. Con esto, se puede incentivar la colaboración para la atención de los problemas y su resolución.

Capítulo III: Marco metodológico

Enfoque del proyecto

A continuación se expondrán la temática del proyecto, su alcance y el diseño que será usado para su realización.

Naturaleza del proyecto

Basado en los objetivos que se desean alcanzar en esta investigación, se utilizará un modelo cualitativo, que permita la recolección de datos a partir de escenarios o teorías propuestas, que permitan identificar las partes del proceso que se desea estudiar y, sirva como base para la implementación de nuevos escenarios que puedan ser replicados de forma sencilla por organizaciones.

Alcance del proyecto

El proyecto tiene como marco de acción la unidad de control de proyectos encargada de la etapa de desarrollo de los sistemas de información, en una empresa piloto de Costa Rica, que incorpore como metodología de trabajo el desarrollo ágil, con el fin de tener un enfoque en la recolección de resultados y no en el aprendizaje de metodologías ágiles.

El enfoque que se aplicará en esta investigación es descriptivo, ya que se busca determinar las partes del proceso de integración continua que se pueden degenerar durante la ejecución del desarrollo de software. Esto busca que las organizaciones puedan implementar, identificar y corregir la implementación con el fin de obtener el mejor beneficio de este paso.

Entre las limitaciones que se afrontan en este proyecto se pueden listar:

1. Encontrar una organización pequeña con un departamento de desarrollo de sistemas, que utilice una metodología ágil, que desee implementar o evaluar integración continua, dedicada al desarrollo propio o “in-house”.
2. Iniciativa por parte de una organización, para establecer horas de trabajo a la capacitación del personal en integración continua.
3. Iniciativa por parte del equipo para la práctica de los principios de integración continua.

4. Encontrar herramientas que se puedan integrar correctamente al ambiente de la organización o que se puedan personalizar, para la adecuada retroalimentación del equipo de trabajo.
5. Encontrar herramientas que puedan soportar la compilación de diferentes lenguajes, según las necesidades de la organización.

El resultado del trabajo aportará una guía, para que cualquier empresa pueda evaluar la forma en cómo implementar la integración continua en sus entornos de trabajo.

Diseño de investigación

La investigación se puede basar en dos etapas: una primera etapa de observación (diseño no experimental) y otra de aplicación (estudio de caso).

En la primera etapa, una observación con el fin de poder recolectar la información, que permita definir los puntos que se realizan mal en el proceso de Integración Continua en las organizaciones.

En la segunda etapa, después del levantamiento de resultados, se podrán realizar ajustes para la modificación del proceso, con el fin de mejorar la integración continua de la organización.

Métodos y técnicas de recolección de información

A continuación se argumentan las fuentes de información y las técnicas de recuperación de datos a utilizar en esta investigación.

Sujetos y fuentes de información

En la elaboración de la investigación se utilizarán varias fuentes para la recolección de información y documentación entre las cuales se encuentran:

Fuentes primarias

- **Escenarios**: datos que se obtendrán a partir de la observación de los escenarios de las organizaciones participantes.

- **Entrevistas**: al personal de la organización acerca de su relación con los procesos organizacionales alrededor del proceso de desarrollo de sistemas.

Fuentes Secundarias

- **Bibliografía**: dentro de las fuentes de referencia citadas se encuentran escritos de metodologías de desarrollo ágiles, así como del proceso de integración continua.
- **Documental**: los documentos de los procesos organizacionales que se aplican en la organización que participe de las observaciones.

Técnicas e instrumentos de investigación

En esta investigación se utilizarán las siguientes técnicas:

Observación

A partir de la observación se analizará los comportamientos y cultura de una organización, que permita determinar si el ambiente es apto para la implementación y desarrollo, y permitirá abordar los problemas que enfrenta el proceso de integración continua estratégicamente.

Entrevistas estructuradas

Las entrevistas estructuradas permitirán obtener las apreciaciones de los integrantes de los equipos con respecto a la integración continua, su cultura, afinación con el proceso y posibles mejoras que favorezca la personalización, para una mejor integración con la cultura organizacional.

Con estas técnicas, se espera recolectar algunos insumos que permitan encontrar la raíz de los problemas que enfrenta este proceso, para generar una estrategia que lo aborde de forma integral para una mejor conexión con el equipo y una mayor participación.

Procedimiento metodológico

En el siguiente cuadro se explican algunas de las actividades que se abordarán en la investigación para obtener los resultados, tanto positivos como negativos, que permitan

comprender los procesos de implementación de la integración continua en una organización.

Cuadro 1. Guía de operacionalización de los objetivos

Título:	Elaboración de un plan de implementación de integración continua utilizando software libre en empresas con presupuesto limitado, que utilizan metodologías ágiles		
Objetivo General:	Evaluar el proceso de desarrollo de software, en una empresa que utilice metodologías ágiles y que cuente con limitado presupuesto, con el fin de implementar la integración continua mediante herramientas de software libre.		
Objetivos Específicos	**Actividades a desarrollar**	**Técnicas o instrumentos a utilizar**	**Resultado o producto esperado**
Seleccionar las herramientas de software libre más idóneas para implementar integración continua, en una empresa, mediante el análisis de requerimientos de hardware y procesos desarrollo de software de la empresa piloto.	Revisar fuentes de información sobre las distintas herramientas, para la implementación de integración continua, su valor agregado y sus características. Generar un documento que recopile, por lo menos tres herramientas, que permitan una fácil implementación y valor agregado al proceso de desarrollo.	Revisión de documentos. Entrevista estructurada.	Documento resumen de las herramientas encontradas, con sus características y valor agregado.
Elaborar un plan de implementación,	Seleccionar la o las herramientas que mejor se adecuen a la organización y realizar un plan de	Ejecución de pruebas de las	Generar un documento de los pasos,

que permita a una empresa piloto desplegar las herramientas de software libre seleccionadas, con el fin de maximizar los procesos de desarrollo de software, mediante integración continua.	implementación, que brinde la posibilidad de desarrollar un pequeño piloto que permita observar la adaptación.	herramientas y observación del entorno.	implicaciones, adaptación y resultado del proceso.
Elaborar instrumentos, que permitan evaluar el nivel de satisfacción, en el proceso de desarrollo de software en la empresa piloto, después de la implementación de la integración continua, mediante entrevistas y cuestionarios a los involucrados.	Revisar los datos obtenidos de las etapas anteriores, con el fin identificar los pasos que generan conflictos a la hora de trabajar con soluciones de integración continua. Proponer los ajustes necesarios para la optimización del uso de integración continua. Generar un documento, que recopile los pasos y las propuestas de ajuste, que permitan a una organización adecuar su proceso de integración continua y crear una cultura en sus empleados.	Revisión de insumos generados durante la investigación. Generar propuestas, a partir de los insumos y el análisis de resultados. Entrevistas y cuestionarios a los participantes.	Documento resumen con las propuestas, el análisis obtenido de la investigación y ejemplificación.

Fuente: Elaboración propia

Capítulo IV: Análisis y selección de herramientas

Para realizar la instalación de un sistema de Integración Continua son necesarias varias herramientas como por ejemplo: contar con un sistema operativo para servidor, un repositorio de datos relacional, un control de versiones encargado de preservar el código fuente y una aplicación de Integración Continua, la que ayudará al equipo a descubrir problemas en etapas tempranas.

Cada una de las anteriores herramientas posee características específicas, lo que hace necesario definirlas y analizarlas para poder seleccionar el mejor hardware, que permita trabajar correctamente sin problemas y que no representen complicaciones a los equipos de trabajo.

A partir del análisis de las siguientes herramientas, se podrá definir los requerimientos mínimos de hardware para el funcionamiento adecuado de un entorno de Integración Continua y de cada herramienta seleccionada.

Sistema operativo

El sistema operativo es sumamente importante, ya que es la base que se utilizará para instalar y seleccionar las demás herramientas. Cada uno tiene sus diferencias y sus enfoques que se deben tomar en cuenta para realizar la selección.

Entre los sistemas operativos de software libre es común escuchar los nombres de Red Hat, Debian, Ubuntu y CentOS. Estos abarcan la mayoría de los servidores del mundo, según un informe de W3Techs (s.f.) y SecuritySpace (2019) para septiembre del 2019. En este informe, los primeros tres lugares de las distribuciones las lideran Debian, CentOS y Ubuntu, con lo cual, Linux lidera aproximadamente un 50% del mercado de los servidores web en el mundo.

Por su parte, Red Hat (s.f.) es una solución empresarial con un modelo de desarrollo de código abierto (Open Source), del cual se puede obtener una versión gratuita por 30 días o la compra de su licencia. De este se deriva CentOS (s.f.), una versión libre disponible para

su descarga en línea, soportada y actualizada con los paquetes que son liberadas al público por Red Hat. Cada versión de CentOS, es mantenida por 10 años y tiene actualizaciones cada 6 meses, lo que la convierte en un sistema muy estable. Entre los requerimientos de instalación que requiere CentOS están los siguientes: 512 MB de RAM, 2 GB de disco duro, Pentium I o superior.

Por otra parte, Debian (s.f.) es una solución libre, basada en varios núcleos que de igual forma son libres. Este tiene una gran comunidad de soporte y paquetes de aplicaciones. Cada versión tiene aproximadamente tres años de soporte y dos años extra para la versión de soporte de largo plazo (LTS). Entre los requerimientos de instalación que requiere Debian están los siguientes: 512 MB de RAM, 2GB de disco duro, Pentium 4 de 1GHz.

Asimismo, Ubuntu (s.f.) es una de las soluciones más comunes y fáciles de usar para usuarios principiantes, cuenta con una gran comunidad y documentación en línea. Las versiones LTS tienen soporte de cinco años y cada dos años se publica una versión nueva. Además, brinda un mayor soporte a los drivers y firmware, incluyendo la mayor cantidad posible, ya sean privativos o gratuitos. Entre los requerimientos de instalación que requiere Ubuntu Server están los siguientes: 1 GB de RAM, 10 GB de disco duro, Procesador de 1GHz con arquitectura x64 en adelante.

De las opciones descritas tanto CentOS como Debian son excelentes alternativas, si disponemos de un hardware con pocas prestaciones. No obstante, Ubuntu requiere un poco más de requerimientos, ya que la instalación trae algunas opciones que las otras no incluyen en su instalación, por lo que se deben instalar posteriormente (Tanenbaum, 2004).

En cuanto a madurez, Ubuntu tiene la comunidad y documentación más extensa de los tres, esto facilita que ante algún problema, sea posible encontrar la solución de forma más rápida. Las otras dos opciones, aunque cuentan con su comunidad y documentación, no es tan fácil encontrar soluciones a errores y, algunas veces se debe esperar hasta una nueva versión para solventar un problema (en el peor de los casos hasta 6 meses). Las diferencias entre los comandos también es importante considerarla, ya que se considera que tanto

CentOS como Debian son sistemas orientados hacia personal experto, por su parte, Ubuntu está enfocado hacia público principiante e intermedio.

Otro punto importante, son las actualizaciones, ya que Ubuntu permite realizar las actualizaciones de una manera sencilla y, los cambios de versión es posible hacerlos casi sin alteraciones. Este punto aplica tanto para los programas del sistema como para los programas instalados. Las otras dos opciones requieren que el administrador descargue los paquetes y aplique las actualizaciones de forma manual. En cuanto con la seguridad, Ubuntu recibe actualizaciones con mayor frecuencia e incluye los paquetes de actualizaciones que reciben las versiones de Debian, además de las propias mantenidas por el equipo de desarrollo.

Una ventaja que tiene Ubuntu sobre CentOS y Debian, es que este soporta una gran cantidad de drivers y firmware de diferentes proveedores. Esto permite tener una mayor compatibilidad con diversos dispositivos o equipos donde se pueda ejecutar.

De las alternativas presentadas, Ubuntu es la opción más apropiada como sistema operativo del servidor. Esto por su mayor compatibilidad, la orientación a usuarios principiantes, su capacidad de instalar actualizaciones con facilidad, la constancia en las actualizaciones y parches de seguridad, su extensa documentación y gran comunidad de soporte.

En el Cuadro 2, se presenta un resumen de las principales características de los sistemas operativos analizados para implementar una solución de Integración Continua.

Cuadro 2. Características de los sistemas operativos

	CentOS	Debian	Ubuntu
Modelo	Libre	Libre	Libre
Soporte	10 años	5 años	5 años
Requisitos	512 MB de RAM, 2 GB de disco duro Pentium I o superior	512 MB de RAM 2GB de disco duro Pentium 4 de 1GHz	1 GB de RAM 10 GB de disco duro Procesador de 1GHz con arquitectura x64 en adelante
Tipo de usuario	Experto	Experto	Principiante Intermedio
Fortaleza	Estabilidad Tiempo de soporte	Estabilidad	Comunidad Documentación Mayor soporte de drivers y firmware

Fuente: Elaboración propia

Base de datos

Entre las opciones de las bases de datos de software libre, hay tres nombres de peso que menciona Marín (2019): PostgreSQL, MySQL y MariaDB. Con más de veinte años de existir, PostgreSQL y MySQL han liderado todos estos años como las alternativas gratuitas de repositorios de datos en el mercado. Por su parte, con casi nueve años, MariaDB está en un proceso acelerado ganando terreno como una opción viable para las organizaciones.

Por su parte, MySQL (Oracle Corporation and/or its affiliates, s.f.), es una aplicación multiplataforma que se integra con muchas herramientas. Nació como una alternativa libre, hasta el momento en que fue adquirida por Oracle, momento en el cual algunas de sus opciones pasaron a ser de paga, pero aún existe una parte de este que se puede acceder de

forma gratuita. Entre los requerimientos de instalación que requiere MySQL están los siguientes: 512MB de RAM, 500MB de disco duro, procesador compatible con x86/x64.

Ante esa situación, surgió MariaDB (MariaDB Foundation, s.f.), el cual es considerado una bifurcación de ésta, al ser creada por los desarrolladores originales. En la actualidad, ha tomado mucha popularidad al ser potencialmente la sustitución para MySQL, al punto que algunas distribuciones de Linux ya la traen por defecto.

Sus creadores se toman el tema de la seguridad muy seriamente, por lo que aplican cada parche que es publicado por MySQL junto a los propios; además de esto, permite el acceso en tiempo real y es muy recomendada para sitios con mucho tráfico. Entre los requerimientos de instalación que requiere MariaDB están los siguientes: 1GB de RAM, 500MB de disco duro, procesador Pentium 4 en adelante compatible con x86/x64.

Asimismo, PostgreSQL (The PostgreSQL Global Development Group, s.f.), es una alternativa libre de base de datos ampliamente utilizada y con mucha documentación existente. Es soportada por la comunidad, quien se encarga dar el mantenimiento a sus características. Además, presenta una gran compatibilidad con diversas herramientas y plataformas por lo que se puede implementar fácilmente. Aunque es una herramienta con bastantes años de existir, es posible presenciar algunos problemas con el rendimiento de las tablas que tienen varias columnas y con bases de datos muy pequeñas. Entre los requerimientos de instalación que requiere PostgreSQL están los siguientes: 1GB de RAM, 512MB de disco duro, procesador 1GHz en adelante.

Las opciones presentadas comparten como características ser multiplataforma, su facilidad de instalación y configuración. MySQL presenta un buen rendimiento aunque puede reducir cuando las bases de datos superan determinado tamaño.

Por su parte, PostgreSQL presenta cierto problema con bases de datos pequeñas, ya que está optimizada para grandes volúmenes de datos. En cuanto a MariaDB, integra características relacionadas con bases de datos NoSQL (no relacional, presenta la capacidad de guardar registros tipo clave-valor), cuenta con una buena escalabilidad, flexibilidad,

rapidez en las transacciones realizadas y soporta una mayor cantidad de lenguajes (TablePlus, 2018).

De las alternativas presentadas, tanto PostgreSQL como MariaDB son dos buenas opciones, ya que la primera está orientada hacia bases de datos muy grandes y la otra hacia la rapidez. MySQL se ha dejado por fuera, anticipando su potencial reemplazo.

Analizando sus características, la mejor opción es MariaDB. Esto por equilibrio entre el rendimiento, compatibilidad con otras herramientas, popularidad entre otros sistemas operativos y grandes clientes que lo han adoptado como remplazo de MySQL; Y orientándolo hacia un ambiente de Integración Continua, anticipando la cantidad de histórico que se pueda desarrollar con los proyectos y, la velocidad en las inclusiones de métricas de código al repositorio, se considera que su rendimiento a futuro será más aceptable.

En el Cuadro 3, se presenta un resumen de las principales características de las bases de datos analizadas para implementar una solución de Integración Continua.

Cuadro 3. Características de las bases de datos

	PostgreSQL	MySQL	MariaDB
Modelo	Libre	Libre con algunas opciones de paga	Libre
Requisitos	1GB de RAM 512MB de disco duro procesador 1GHz en adelante	512MB de RAM 500MB de disco duro procesador compatible con x86/x64	1GB de RAM 500MB de disco duro procesador Pentium 4 en adelante compatible con x86/x64
Fortaleza	Mejor manejo de bases de datos grandes	Mejor rendimiento con bases de datos de determinado tamaño Características NoSQL	optimizada para sitios con mucho tráfico Características NoSQL Seguridad

Fuente: Elaboración propia

Repositorio de control de versiones

Entre las opciones de repositorios de datos libres, hay dos de peso e importancia: Git (s.f.) y Mercurial (s.f.). Estas dos opciones son sistemas distribuidos, por lo que cada persona tiene una copia completa de los archivos con todo su histórico, lo que permite tener un mejor control tanto para nuevas características como para regresiones, siendo casi imposible perder algún archivo.

Como primera opción, Git (s.f.) es uno de los programas de control de versiones más popular del mercado. Permite la integración con diferentes herramientas tanto para ver las

diferencias de código o el editor que se va a utilizar. La configuración de los usuarios es a partir del nombre de usuario y un correo.

Asimismo, es una alternativa que por defecto no cuenta con una interfaz de usuario; su funcionamiento es a través de una consola de comandos, que permite realizar toda la funcionalidad que brinda. Existen algunos complementos que pueden instalarse en algún editor o por aparte para poder hacer la interacción de una forma más natural para el usuario. Para un equipo con poca experiencia, lo ideal es utilizar alguno de los complementos para facilitar su uso y aprendizaje.

En el sitio web de Git, los usuarios pueden encontrar documentación bastante extensa del uso de la aplicación, incluso con algunos ejemplos sencillos para explicar los comandos y poder aplicarlo. Aun así, la documentación de los comandos es un poco compleja, por lo que sí requiere una pequeña curva de aprendizaje, ya que cada comando cuenta con varias opciones lo que aumenta la dificultad. Entre los requerimientos de instalación que requiere Git están los siguientes: 256MB de RAM, 150MB de disco duro, procesador compatible con x86.

Por otra parte, Mercurial, en su sitio web, muestra una gran cantidad de extensiones para todo tipo de tareas, lo que permite facilitar su integración con otras herramientas. Desde el color de la fuente, el editor y hasta la seguridad se pueden manejar con los complementos que ofrecen. No cuenta con una interfaz de usuario, por lo que su acceso es a través de una consola de comandos, que permite ejecutar fácilmente las opciones que brinda. Aun así, los comandos son sencillos y fáciles de seguir, por lo que no debería de existir ningún problema si un usuario con poca experiencia los utiliza. Si por alguna razón, se requiere utilizar una interfaz gráfica para los usuarios, puede utilizar la herramienta que provee TortoiseHg (s.f.), la cual es muy sencilla y de fácil configuración.

Asimismo, Mercurial cuenta con una documentación muy fácil y rápida de leer; se enfoca en la sencillez de las acciones y muestra gráficamente con algunos ejemplos lo que la aplicación realiza. Con pocos minutos, un usuario puede crear un pequeño ejemplo y aprender a utilizar los comandos que ofrece esta herramienta. Entre los requerimientos de

instalación que requiere Mercurial están los siguientes: 256MB de RAM, 100MB de disco duro, procesador compatible con x64.

No obstante, una diferencia entre estas dos herramientas es su uso. Actualmente, Git ha ganado mucha popularidad, apoyada por grandes proyectos y clientes que lo han comenzado a utilizar y a proveer apoyo. De esta forma, se ha convertido casi en un estándar para los desarrolladores y, ha desplazado a otras opciones incluyendo a Mercurial.

De las opciones presentadas, la mejor opción tiende a ser Git. Esto por su popularidad en el mercado, el soporte que le están dando grandes proyectos y patrocinadores, su integración con diferentes herramientas, su extensa documentación que se encuentra en varios idiomas y su comunidad de soporte.

En el Cuadro 4, se presenta un resumen de las principales características de los repositorios de control de versiones analizados para implementar una solución de Integración Continua.

Cuadro 4. Características de los repositorios de control de versiones

	Git	**Mercurial**
Modelo	Libre	Libre
Orientación	Distribuido	Distribuido
Tipo de usuario	Avanzado	Principiante Intermedio
Requisitos	256MB de RAM, 150MB de disco duro, procesador compatible con x86	256MB de RAM, 100MB de disco duro, procesador compatible con x64
Fortaleza	Mayor popularidad y documentación	facilidad de uso

Fuente: Elaboración propia

Aplicación de Integración Continua

Existen muchas aplicaciones de Integración Continua disponibles en el mercado, entre ellas podemos encontrar Jenkins, GoCD y CruiseControl. De las anteriores, Jenkins es más popular y más usada, pero implementa más características que pueden hacer un poco más compleja su utilización.

Por un lado, CruiseControl (s.f.) es una aplicación pionera creada en Java, la cual nació para proyectos desarrollados en ese lenguaje. No obstante, su utilidad fue tal, que también se desarrolló una versión para .Net (CruiseControl.NET, s.f.) y, actualmente, una para el lenguaje Ruby. Su importancia creció tanto, que en la actualidad la empresa creadora ThoughtWorks inició un nuevo proyecto al que llamó Cruise y después renombraron como GoCD. Entre los requerimientos de instalación que requiere CruiseControl están los siguientes: 2GB de RAM, 500MB de disco duro.

Asimismo, CruiseControl es una herramienta con altas prestaciones, dedicada totalmente a la Integración Continua. La principal desventaja que contiene es que no soporta varios lenguajes en una misma aplicación. De esta forma, si se desarrollan soluciones de software en Java y C#, por ejemplo, sería necesario instalar tanto CruiseControl para Java y CruiseControl.net para C#.

De igual modo, GoCD (ThoughtWorks Inc., s.f.) es una aplicación moderna, desarrollada hacia un enfoque de entrega continua, para apoyar a las organizaciones, pero que aporta la parte de Integración Continua. Es multiplataforma, con una versión que es gratuita y se integra con varias herramientas de control de versiones. Entre los requerimientos de instalación que requiere GoCD están los siguientes: 2GB de RAM, 1GB de disco duro, Procesador de 2GHz y dos núcleos mínimo.

Además, GoCD es una excelente opción con muy buena documentación detallada. Es un poco pesada en cuanto a consumo de recursos de hardware, pero su soporte para varios lenguajes y la orientación hacia el despliegue continuo, le dan un plus que puede potenciar la estrategia de un negocio. Cuenta con bastante información en línea en su sitio oficial y tiene el respaldo de una organización que es pionera en el tema de integración y despliegue

continuo. Además de esto, incluye dentro de su configuración, una sección para el envío de correos y notificaciones a los usuarios, con la información que cada uno desee, por ejemplo, reportarle al equipo cuando la ejecución de pruebas ha fallado por la realización de un cambio.

Por otra parte, Jenkins (Jenkins, s.f.), es un servidor de automatización multiplataforma enfocado en la Integración Continua, pero apoya también ciertos aspectos de la entrega continua. En caso de que una organización desee incorporarlo posteriormente, es una buena opción que se puede contemplar. Este aunque no tiene muchos años, proviene de una bifurcación de Hudson, el cual es un proyecto que tiene bastantes años de ser usado.

También, Jenkins es una opción muy completa y es compatible con varios lenguajes. Mantiene un equilibrio entre los requerimientos de hardware y las opciones que brinda tanto de integración como despliegue continuo. Cuenta con una muy buena documentación y una gran comunidad de soporte en línea la cual puede ayudar a la resolución de problemas. Para facilitar su uso requiere que se instalen varios complementos, pero todos están disponibles desde su administrador donde se pueden buscar para agregarlos. Entre los requerimientos de instalación, que requiere Jenkins, están los siguientes: 1GB de RAM, 50 GB de disco duro.

Tanto GoCD, como Jenkins, son dos perfectas opciones que se pueden utilizar para una solución de Integración Continua. Estas dos herramientas le dan la posibilidad a la empresa de poder seguir creciendo en el aspecto de la agilidad de las soluciones de software. Ambas tienen una excelente documentación, muy buenas bases, son robustas y compatibilidad con varios lenguajes.

En el Cuadro 5, se presenta un resumen de las principales características de las aplicaciones analizadas para facilitar su comparación.

Cuadro 5. Características de las aplicaciones de Integración Continua

	Jenkins	**GoCD**	**CruiseControl**
Modelo	Integración Continua Despliegue Continuo	Integración Continua Despliegue Continuo	Integración Continua
Tipo de usuario	Intermedio	Intermedio	Principiante Intermedio
Requisitos	1GB de RAM 50 GB de disco duro	2GB de RAM 1GB de disco duro Procesador de 2GHz y dos núcleos mínimo	2GB de RAM 500MB de disco duro
Fortaleza	Mayor popularidad	Respaldo de una empresa con más de 18 años de experiencia en Integración Continua y Despliegue Continuo	Documentación detallada

Fuente: Elaboración propia

Asimismo, es recomendable agregar una herramienta más llamada SonarQube (SonarSource S.A, 2008), la cual es gratuita y se puede integrar con Jenkins y GoCD. Además, permite realizar análisis de código fuente, genera métricas para evaluar y mejorar la calidad de los sistemas. El potencial de esta herramienta está en la posibilidad de poder realizar análisis de código duplicado, la aplicación de estándares y la evaluación de la cobertura de las pruebas unitarias, entre otras. Aunque requiere un poco más de requisitos de hardware (2GB de

RAM) es una excelente opción que puede dotar de mucha información importarte a la organización, con el fin de hacer seguimiento del ciclo de vida de los productos de software, aportar una retroalimentación más pronta a los equipos de trabajo y apoyo a la toma de decisiones.

En su sitio web es posible encontrar documentación detallada del proceso de instalación y de las características con las que cuenta esta herramienta, así como de los lenguajes que soporta y como realizar su integración con otras herramientas.

Requerimientos generales

Para realizar una implementación de Integración Continua, se pueden utilizar las siguientes herramientas de las ya mencionadas: Ubuntu Server como sistema operativo, MariaDB como base de datos, Git como repositorio de control de versiones, Jenkins como encargado de Integración Continua y SonarQube para el análisis del código fuente.

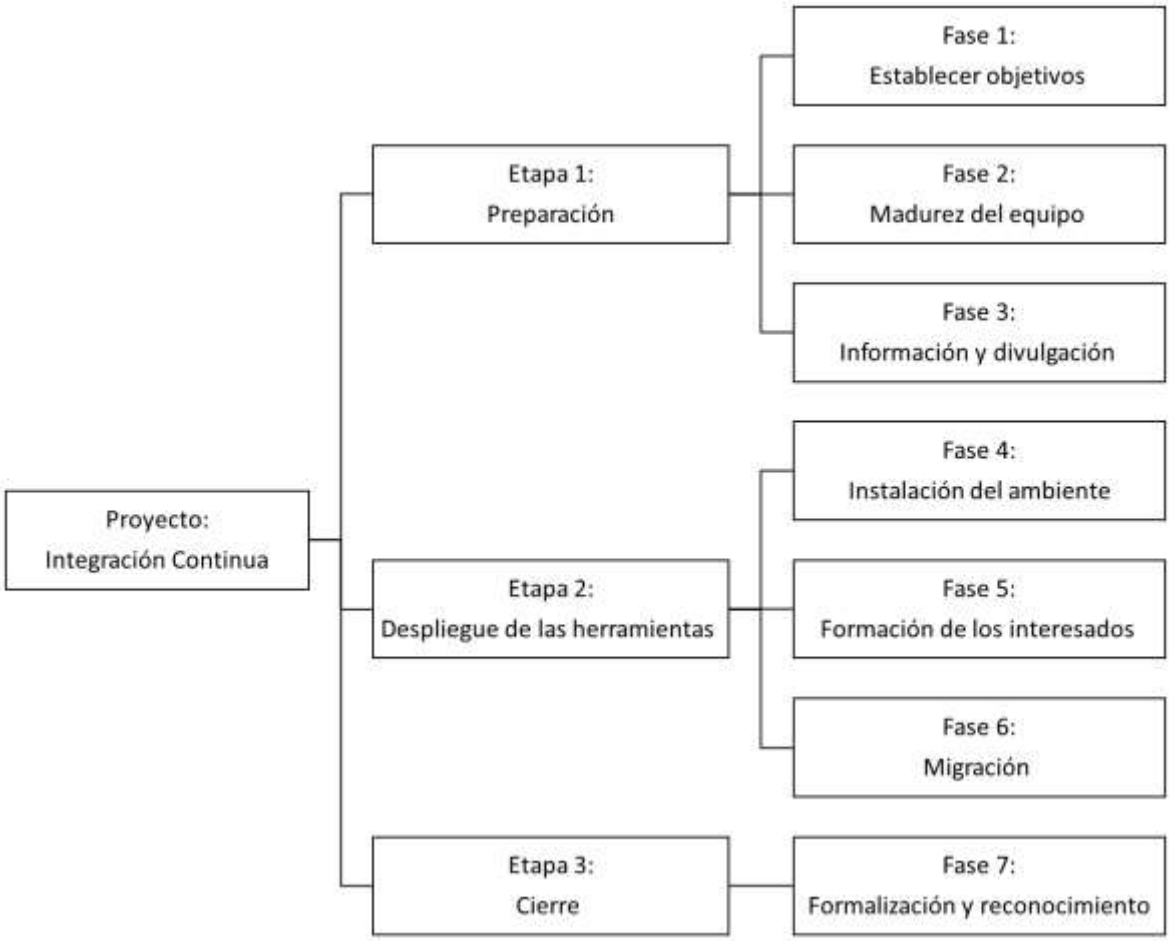
A partir de esta selección de herramientas, se requiere contar con un equipo que tenga las siguientes características mínimas: 1GB de memoria RAM, 50GB de disco duro, un procesador de 1GHz, doble núcleo compatible con la arquitectura x86/x64. Con esta configuración se puede establecer un ambiente adecuado para la implementación y buen funcionamiento de las herramientas seleccionadas.

Capítulo V: Plan de implementación

En este capítulo se tratarán los aspectos que se deben abordar para la implementación de Integración Continua en una organización.

Se ha organizado el siguiente plan de implementación en tres etapas y siete fases, que permitirán ir paso a paso al ritmo que la organización pueda asimilar, controlar y monitorear cada una de las entregas y sus resultados.

Ilustración 1: Esquema del plan de implementación



Fuente: Elaboración propia

Etapa 1: Preparación

Según Romero (2016), dos de las causas más comunes para el fracaso de un proyecto son la urgencia de finalizar el proyecto y los problemas de comunicación entre las partes. Por esos motivos, la implementación debe ser gradual, para que los participantes que se ven involucrados mantengan el interés durante todo el ciclo del proyecto. Su colaboración en el proceso puede ayudar a obtener una retroalimentación pronta a problemas o posibles mejoras que se puedan agregar con el paso del tiempo.

Además, hacer un análisis del ambiente de la organización, así como de los objetivos, permitirá evaluar si se debe realizar un ajuste en la dirección de la misión y visión. Si la organización cuenta con una arquitectura empresarial (AE) orientada al proceso de adaptación y actualización constante, resultará más viable el éxito del proyecto, generando una cultura de cambio entre los colaboradores. Cada empresa cuenta con su propio modelo AE, pero debe incluir entre sus objetivos la mejora y la adaptación tecnológica, así como sus planes de mejora constante, según indica Trejos (2018).

Preparar pautas informativas, hacer reuniones con el personal y hacer un análisis del equipo son decisivos para saber su madurez antes de realizar el cambio. Si el equipo es maduro, puede buscar su apoyo para iniciar la implementación y si no lo es, debe buscar la mejor manera de motivarlo y obtener su apoyo, de modo que desarrollen la madurez de equipo, que los colaboradores generen un vínculo fuerte de forma que aporte beneficios a la organización y al desarrollo de todos los proyectos futuros.

A partir de lo descrito, se puede iniciar esta fase con los siguientes objetivos:

General:

- Analizar el ambiente organizacional, con el fin de identificar el nivel de madurez para la implementación de una solución de Integración Continua.

Específicos

- Identificar el grado de madurez de los equipos de trabajo y su respuesta ante los cambios.

- Identificar la cultura organizacional que influye sobre los equipos de trabajo.
- Desarrollar y aplicar estrategias que permitan favorecer una cultura organizacional entorno a los objetivos de crecimiento del negocio.

En los anexos se encuentra una serie de cuestionarios que se pueden aplicar a los equipos de trabajo, según su rol, con el fin de poder evaluar su integración con la organización y la madurez que tienen los equipos. Esto ayudará a establecer una base con la que podrá iniciar el proceso y encaminar la incorporación de Integración Continua. Estos artefactos se pueden suministrar a los colaboradores cuantas veces sea conveniente, de modo que permita evaluar la estabilidad y madurez del equipo, así como para identificar deficiencias o problemas que estén afectando el desarrollo de las labores de los integrantes.

Fase 1: Establecer objetivos

Dos aspectos mencionados por Rodríguez, García y Lamarca (2011), donde listan las causas frecuentes de fracaso de un proyecto son las siguientes: periodos de tiempo poco realista y mala definición de los objetivos y metas. Es esencial fijar los objetivos y las metas en el orden que se desean lograr, tomando en cuenta tiempos realistas que apoyen la labor del equipo y, de la curva de aprendizaje que puede requerir la implementación.

Asimismo, es posible realizar algunas tareas de forma paralela, pero debe tomar en cuenta la carga que puede significar en el equipo de trabajo o colaboradores. Par ello, es esencial llevar el control de la carga de trabajo para que el equipo no se abrume y no decaiga su ánimo con el proyecto.

Además, es obligatorio realizar un listado de las metas que se desean alcanzar, de los recursos con los que ya cuenta la organización, de la inversión económica inicial que debe realizar, si se requiere del conocimiento que tiene el equipo de trabajo, de las cargas de trabajo y la disponibilidad con la que cuentan los colaboradores. Todo esto para establecer la base del proyecto y los posibles riesgos que pueden surgir en el arranque del proyecto.

Esta fase debe tener como único fin, establecer un panorama inicial del proyecto, que permita conocer las metas reales que se desea alcanzar, preferiblemente a corto, mediano y

largo plazo, de modo que se pueda incorporar poco a poco la solución permitiendo a los colaboradores la asimilación de cada herramienta, un monitoreo o evaluación de los resultados según el avance.

Fase 2: Madurez del equipo

Contar con un equipo maduro ayuda a que los proyectos puedan ir con buen ritmo. La comunicación permanente permite un mejor crecimiento del equipo y favorece la integración de los colaboradores. La mala comunicación o la pérdida de esta pueden significar el fracaso de un proyecto según Rodríguez et al. (2011).

En este punto, es preciso evaluar las habilidades, conocimiento y madurez del equipo, tanto del entorno de trabajo como en los avances de las tecnologías. A partir de los cuestionarios (del uno al tres) de los Anexos, se podrá dar una idea de la madurez del equipo y la comprensión del ambiente que lo rodea.

Además, es posible implementar el cuestionario cuatro, para conocer y evaluar la implementación de las metodologías de trabajo. Esto con el fin de identificar si se está realizando de la mejor manera posible o si debe agregar o modificar alguna práctica.

También, es ideal ejercer un liderazgo activo e incluyente sobre el equipo, de forma que permita elevar la motivación con acciones y no solamente con recompensas, para así evitar el síndrome del fracaso que describe Cortés (2004) en su artículo, ya que este será un proceso de aprendizaje para ambas partes y puede provocar la desmotivación de algún colaborador.

Esta fase debe tener como único fin, lograr la mayor participación y aceptación por parte de los colaboradores y de los cambios que se desean llevar a cabo. Además, obtener una evaluación de la metodología de trabajo, para identificar las prácticas que se deben modificar o agregar para maximizar el rendimiento de los equipos de trabajo.

Fase 3: Información y divulgación

Para lograr el apoyo, la participación de los miembros del equipo y asegurar el éxito del proyecto, debe informar y comunicar con antelación los diferentes componentes que se van a incorporar en el proceso de desarrollo. Todo el personal debe conocer los cambios que se desean realizar para que se identifiquen con las metas que se quieren alcanzar, enfocado en la mejora de la comunicación organizacional, según indica Lemus (2000) .

Realizar reuniones para la integración con los participantes, una vez se tenga los resultados de los cuestionarios aplicados, puede influir positivamente. De esta forma, será posible darles a conocer el plan del proyecto, ayudando así a la integración del equipo. Es posible ganar el apoyo de los integrantes del equipo de trabajo dedicando no más de cinco minutos diarios con pequeñas campañas de videos, que demuestren los beneficios de estas herramientas y las tendencias actuales. Para esto, se pueden utilizar las estructuras con las que cuenta la empresa, ya sea por correo electrónico, por boletines mensuales o semanales, resúmenes de noticias, bases de datos de conocimiento, entre otras.

En caso de no tener algún medio como los citados anteriormente, la organización podría optar por la implementación de alguna solución, que le permita reunir en un solo lugar el conocimiento que se va generando de las labores diarias. Tener una base de conocimiento, permitirá a los colaboradores poder encontrar con mayor rapidez la solución a los problemas que puedan surgir. Además, pueden actualizar y agregar nuevos datos convirtiéndose en un punto de referencia y capacitación para el nuevo personal.

Lo principal en esta fase es comunicar progresivamente los cambios y la información de las herramientas, de esta forma, los colaboradores podrán asimilar esta etapa y sentirse más integrados.

Etapa 2: Despliegue de las herramientas

El despliegue o implementación es una de las fases más críticas de un proyecto, en la cual se puede fracasar, si no se tiene el control y monitoreo adecuado de los objetivos y las metas. Factores internos como externos de la organización pueden influir, por lo que el encargado del proyecto debe mantener una revisión constante y eficiente, así como cuidar el ánimo del equipo en todo momento.

Es esencial llegar a este punto con un equipo motivado, consciente de lo que se desea realizar y cómo se desea hacer. Repasar los objetivos y las metas que se están tratando de alcanzar ayuda a que los colaboradores tengan siempre presente lo que se quiere lograr, de modo que no se pierda la ruta a seguir establecida en el plan del proyecto.

A partir de lo descrito, se puede iniciar esta fase con los siguientes objetivos:

General:

- Ejecutar la implementación de las herramientas de Integración Continua, con el fin de mejorar el proceso de desarrollo de software en la organización.

Específicos

- Realizar capacitaciones a los colaboradores, con el fin de dar a conocer las ventajas que aportan al ciclo de vida de desarrollo de software.
- Instalar y configurar las herramientas de Integración Continua, que permita crear un ambiente para preservar el código fuente y facilite su validación con pruebas unitarias.
- Realizar la migración del código fuente al nuevo repositorio, con el fin de permitir a los equipos de trabajo realizar y practicar la Integración Continua.

Fase 4: Instalación del ambiente

Antes de iniciar la implementación, es preciso suplir las necesidades propias del hardware. Esto implica acondicionar un espacio físico que cuente con una temperatura adecuada y, un suministro de energía que le permita estar en línea ante un eventual corte eléctrico. Es crucial que la organización cuente con un plan de recuperación ante desastres (DRP), como lo indican Mendoza (2014) y Matos, Beriguete & Peña (2015), o un plan de continuidad del negocio (BCP), como lo describe Cobb (2014), y que pueda ser evaluado para garantizar el funcionamiento ante alguna emergencia.

En cuanto a la instalación de las herramientas, estas no presentan gran complejidad durante su instalación, aunque algunas si requieren algunos elementos de configuración o complementos para su ejecución. A continuación, se listan algunos de los puntos que se deben tomar en cuenta a la hora de instalar las herramientas:

- Para Ubuntu Server, activar el firewall, así como OpenSSH para la seguridad de las conexiones de red y del servidor. Además, actualizar o instalar la última versión de Java, ya que la mayoría de las aplicaciones lo utilizan o están desarrolladas en esta plataforma.
- Para GIT (Sistema de control de versiones distribuido para el rastreo de cambios en el código fuente durante el desarrollo de sistemas), configurar los usuarios de los colaboradores. Además, es posible cambiar algunas otras características como el visor de diferencias de código, por ejemplo.
- Para MariaDB, agregar a la carpeta de instalación el complemento de conexión a la base de datos (JDBC).
- Asimismo, SonarQube, debe agregar a la carpeta de instalación el complemento de conexión a la base de datos (JDBC). Además, requiere que se configure la base de datos a la cual se va a conectar. Estas dos configuraciones se realizan a través de los archivos de configuración de la aplicación.
- Para la herramienta Jenkins, debe indicarle la ruta donde está instalado el JDK de java; esto se realiza desde la interfaz gráfica en la configuración del sistema. Además, requiere instalar el complemento para GIT y SonarQube. Para este último, debe ingresar nuevamente a la configuración de Jenkins, localizar la sección de la configuración llamada SonarQube y completar la información que solicita, la cual ya se ha aplicado anteriormente en la instalación. También, configurar las notificaciones de las compilaciones y ejecuciones de las pruebas unitarias con el fin de retroalimentar al equipo de trabajo.

Toda la información necesaria, para el proceso de instalación de las herramientas, se encuentra disponible en la página oficial de cada una. Estas son guías paso a paso de como configurar y utilizar cada una de las aplicaciones y las opciones deseadas.

Al finalizar la instalación de todas las herramientas, se debe realizar un ejercicio para verificar que la configuración realizada y el proceso funcionan con la menor intervención humana. Si es posible documentar este ejercicio, podrá utilizarlo en el proceso de formación del equipo, de modo que sea más gráfico y comprensible.

El fin de esta fase es lograr una implementación exitosa del ambiente de Integración Continua, sin perder de vista el ánimo del equipo de trabajo. Aunque la instalación no debe presentar mucho problema, aunque podría generar un nivel de estrés entre los colaboradores que, en algunos casos, puede ser perjudicial para la moral del equipo.

Fase 5: Formación de los interesados

El proceso de formación de los interesados se realiza desde la etapa uno, con la divulgación del proyecto y las campañas, donde se promueve el uso de las herramientas que están por utilizar los colaboradores y, sigue, incluso, después de finalizar el proyecto, esto porque el conocimiento adquirido de este proceso se transmitirá a los nuevos colaboradores.

Para mantener y mejorar la unión del equipo es posible utilizar la técnica de Aprendizaje Basado en Problemas (ABP), como lo describe Guevara (2011), de esta forma se incentiva el trabajo colaborativo, se puede centrar en la resolución de problemas de la vida real y permite la participación de todos los integrantes de forma activa.

Para este punto, se puede realizar un pequeño ejercicio de laboratorio donde participe el equipo, construyendo una pequeña demostración basado en problemas donde todos participen, por ejemplo, concurrencia en la protección (commit) de cambio de código. De esta forma, podrán aprender con la práctica, reforzar la comunicación, mantener su motivación y desarrollar el autoaprendizaje.

Esta fase no debe representar mayor problema, será más bien un punto de equilibrio o impulso para el equipo. La atención, en este momento, debe dirigirse a la motivación del equipo, mantenerlo concentrado en lo que se está realizando y la fase de migración que está por iniciar.

Fase 6: Migración

Una vez completadas las fases anteriores, llega el momento de migrar el repositorio de código fuente al nuevo sistema. Lo ideal es hacerlo gradualmente, debido a que podrían presentarse algunos errores de dependencias de tecnologías de terceros, que puede requerir un poco de tiempo extra en investigación para los involucrados.

Identificar la forma más viable para poder realizar la migración de cada proyecto de código fuente al nuevo repositorio es crucial. Cada empresa tiene diferentes necesidades, por lo que es vital seleccionar la mejor forma para no afectar su continuidad. La idea es poder atender la necesidad del negocio y de la migración de forma paralela, sin ofrecer mucho recargo al equipo de trabajo.

Entre las estrategias que se pueden seguir están las siguientes:

- Migrar cada proyecto que se va a utilizar, pensando en resolver los problemas que puedan surgir en cada uno según se requiera.
- Migrar los proyectos de mayor soporte o críticos, similar a lo anterior, pero en este caso se abordan todos los proyectos que tienen una mayor frecuencia de uso según sea el caso.
- Migrar los proyectos de menor soporte, similar a lo anterior, pero en este caso se atienden los proyectos que tienen una menor demanda, lo que ayuda a tener experiencia para cuando se requiera migrar los proyectos más críticos.
- Migrar los proyectos de forma hibrida, en este caso, se selecciona un proyecto crítico o de mayor soporte junto con sus dependencias, se evalúa el riesgo siendo el proyecto principal el de mayor riesgo, y se inicia la migración por la dependencia de menor riesgo.

Esta fase tiene un nivel crítico, tanto por mantener estables las necesidades del negocio y por atender la migración del código fuente al nuevo repositorio; hacerlo gradualmente, para no sobrecargar al equipo y mantener su motivación al máximo.

Etapa 3: Cierre

Esta etapa es la culminación del proyecto, en la cual se debe evaluar los objetivos planteados por la organización y los resultados obtenidos. Se podrán definir procedimientos a seguir para cada herramienta o proceso nuevo creado, a partir de la implementación de Integración Continua. Lo ideal es que todo quede formalmente establecido, para que se fomenten en las nuevas incorporaciones de personal, así como al aplicar mejoras en etapas futuras del proceso.

A manera de seguimiento, es posible aplicar cíclicamente los cuestionarios de los anexos con el fin de identificar desafíos o nuevas mejoras que puedan tener los colaboradores, lo que permitirá desarrollar un cronograma de mejoras, impulsando la modernización del proceso de desarrollo y de Integración Continua. El tiempo para aplicar los cuestionarios puede ser cada cuatro meses, aunque este tiempo puede variar según sea conveniente.

A este punto se debe considerar el trabajo del equipo, la dedicación aportada al éxito del proyecto y la actualización de las tecnologías de trabajo para el equipo. Establecer un mecanismo de reconocimiento ayudará a que los colaboradores se vean incluidos, puedan mantener su motivación con las metas y objetivos del negocio, aportando valor y el desarrollo de una cultura organizacional.

A partir de lo descrito, se puede iniciar esta etapa con los siguientes objetivos:

General:

- Fomentar la cultura organizacional en los colaboradores, que permita promocionar el uso de la Integración Continua por medio de procedimientos a los nuevos colaboradores y equipos de trabajo.

Específicos

- Desarrollar y aplicar los procedimientos, que permitan extender la utilización de las herramientas de Integración Continua a los nuevos colaboradores.
- Incentivar el desarrollo de la cultura organizacional en los equipos de trabajo, con el fin de elevar el ánimo de los colaboradores.

- Desarrollar y aplicar estrategias de reconocimiento a los colaboradores que permitan ampliar la cultura organizacional del equipo del trabajo en beneficio del negocio.

Fase 7: Formalización y reconocimiento

Una vez realizado todo el trabajo y terminadas con éxito cada una de las fases, es hora de establecer por escrito los procedimientos y manuales, que se encarguen de dar a conocer al personal nuevo tanto la metodología de trabajo como las herramientas y procesos que se realizan en cada tarea. Aunque los involucrados pueden sentirlo burocrático, ofrece un buen punto de partida a la hora de capacitar al personal nuevo, incluso en algunos casos para refrescar las tareas. Además, puede utilizar los procesos de capacitación de los nuevos colaboradores para exaltar la cultura organizacional, las nuevas prácticas y el apoyo a la mejora continua de los procesos organizacionales, de esta forma, los nuevos colaboradores apoyarán el proceso y aportarán nuevas ideas de modernización.

Algunas personas valoran mucho el reconocimiento, por lo que dar mérito al equipo de trabajo, al finalizar el proceso, ayudará a continuar creando una cultura organizacional, orientada hacia el cumplimiento de las metas de la organización, con el fin de que siga creciendo y mejorando continuamente, para lograr obtener el apoyo de los colaboradores en los futuros proyectos que se vayan a implementar.

Esta fase debe dejar bien definido, y para conocimiento de todos los colaboradores, las normas o procedimientos que se utilizarán en el ciclo de vida del desarrollo de software. De esta forma, tanto los actuales como nuevos colaboradores, podrán acudir a estos documentos cuando lo requieran. Además, mantener una constante interacción con el equipo de trabajo para mantener el ánimo creciente y constante

Capítulo VI: Evaluación del nivel de satisfacción de los interesados

En este capítulo se abordarán los aspectos para continuar evaluando la implementación del proyecto, a partir de la retroalimentación que se pueda obtener directamente de los colaboradores y de la metodología; en cuanto a la satisfacción del trabajo realizado y su funcionamiento con el paso del tiempo.

La comunicación, la integración y motivación de los involucrados es fundamental para lograr que los proyectos empresariales salgan adelante; con apoyo y compromiso de todas las partes es posible lograr el cumplimiento de las metas y objetivos establecidos (Organización Internacional del Trabajo, 2016). Pero no es un asunto de una sola vez, sino que es preciso evaluar constantemente el ambiente organizacional para medir el nivel de motivación y apoyo.

Como parte del proceso de cierre se dio la sugerencia de aplicar cuestionarios cada cierto tiempo (los cuales se encuentran como anexos); no obstante, es posible que algunos datos queden por fuera como, por ejemplo, ideas de mejora o incluso lo que sienten por trabajo realizado. Por esto, es importante realizar una evaluación para determinar qué tan satisfechos están las partes involucradas con el desempeño alcanzado por el equipo de trabajo, por los productos desarrollados, por el clima organizacional y el cumplimiento de los objetivos.

Hay tres partes directamente ligadas al proceso de desarrollo: el equipo de trabajo, los encargados en la toma de decisiones (administrador, líder técnico o encargado del equipo) y el cliente o dueño de negocio. Estos interactúan en las fases del ciclo para definir los elementos que se van a crear, así como las reglas que deben seguir y evaluar, esto para asegurar que el producto es lo deseado y funcionará como se espera.

El equipo de trabajo está directamente relacionado con las herramientas de implementación, los que experimentan su rendimiento y usabilidad. Por eso, es esencial obtener su apreciación, de modo que se pueda intervenir el proceso y hacer las mejoras pertinentes

para reducir los inconvenientes que puedan afectar el proceso. Por ejemplo, ante alguna negativa de los colaboradores como no llegan las notificaciones del resultado de la compilación del código, o tardan mucho en ejecutarse las pruebas unitarias; es necesario actuar pronto, ya que esto se traduce en que el equipo no tiene una retroalimentación rápida y podría aumentar el riesgo de arrastrar algún error.

Para obtener mayor información de los colaboradores, es posible utilizar las reuniones de retrospectiva, de modo que puedan expresar, de forma sincera, lo que piensan, además que puedan externar qué está bien o lo que pueda necesitar una mejora. Estas reuniones pueden ir cambiando conforme va madurando el equipo, o según el tipo de público con el que cuenta, por lo que no hay una única forma de hacerlo ni una mejor que otra. Hay que probar, desde lo más simple como dar su opinión en un papel sin nombre, hasta lo más dinámico como actividades de integración, o incluso mezclarlos, con el fin de que todos participen de forma activa. El fin de esta actividad, es obtener información de cada integrante del equipo, que ayude a mejorar el ambiente, la participación, la metodología y las herramientas (Foladori & Guerrero, 2017).

Para evaluar el ánimo de los colaboradores, se puede utilizar el cuestionario número cinco, incluido en los anexos, para obtener los sentimientos de cada persona, su motivación y su satisfacción con el ambiente que lo rodea, de modo que se pueda anticipar los problemas que puedan surgir. Este cuestionario debe usarse para indagar y ayudar a mejorar el ambiente, si no se usa adecuadamente, en el peor de los casos, puede fomentar la separación del equipo y podrían evadir las preguntas con respuestas genéricas y sin valor. Toda respuesta sincera, por parte de los colaboradores, aportará puntos de mejora y retroalimentación a los ciclos de trabajo, de forma que se puede llegar a optimizar tanto los modos de trabajo como los tiempos de entrega.

Los encargados de la toma de decisiones, tienen a mano las métricas del equipo, y pueden identificar cuánto tiempo o recursos le consume a cada persona resolver los inconvenientes, que surgen del uso de las herramientas. Las métricas son un punto de referencia para evaluar si se están cumpliendo los objetivos, el aprendizaje del equipo, las técnicas y el desarrollo. Estas normalmente son comunicadas a los colaboradores para que tengan

presente que se evalúa y de qué forma se realizará. Entre más puntos de valor se cubran con las métricas, se podrá crear un panorama que permita predecir el comportamiento del equipo, de modo que se pueda actuar a tiempo ante los problemas (Avendaño, 2019).

Los datos de las métricas se pueden contrastar con la información que obtienen de las retrospectivas, de modo que ayude a categorizar cada problema y, así, atenderlos según su criticidad. La idea es lograr un análisis integral del comportamiento en el periodo laboral, la perspectiva obtenida de cada colaborador y sus estados de ánimo. Esto podría ayudar a buscar soluciones integrales a los problemas tanto laborales como personales de los colaboradores, logrando que causen un menor impacto en su desempeño laboral, una mejor relación con la empresa y reforzar la cultura organizacional.

El cliente o dueño de negocio es quien tiene más peso, porque es quien influye en la continuación o cese de un proyecto y, en el peor de los casos, es quien decide si mantiene la relación de negocios o busca otro proveedor. Es importante entenderlos, orientar la función del negocio hacia ellos y evaluar su satisfacción, con el fin de obtener su punto de vista, poder atender sus sugerencias para fortalecer la relación con el mismo, el apoyo y la motivación (González, Carmona, & Rivas, 2007).

La retroalimentación oportuna, por parte de los clientes, aporta una ventaja en la mejora continua del negocio, de modo que la organización puede enfocar sus estrategias al valor agregado de sus productos y servicios, alcanzando una mejora sobre los competidores.

Para evaluar la satisfacción del cliente se utiliza el cuestionario número seis, incluido en los anexos, con el propósito de descubrir cómo es la relación que se mantiene, su estado de ánimo, la posibilidad de obtener una buena crítica ante futuros clientes, favorecer la mejora de los productos, así como la atención que se brinda; logrando atender los comentarios negativos que pueda proporcionar de forma eficiente y eficaz.

A partir de los resultados que se obtienen de la implementación de dichos cuestionarios, es posible evaluar las medidas que se deben ejecutar para minimizar el impacto de lo que puede ocurrir. Para la evaluación se presenta a continuación una serie de medidas que

pueden tomarse en cuenta, para realizar una estimación de la satisfacción contrastando los puntos de vista de los colaboradores contra los clientes.

Considerando el cuestionario # 5, asigne un punto por cada una de las preguntas que son contestadas de forma positiva, totalice los puntos a partir de la cantidad de cuestionarios aplicados y obtenga el promedio de satisfacción de los colaboradores. Las respuestas neutrales o negativas deben ser atendidas, según consideren los encargados de la toma de decisiones y suman cero a dicho promedio.

Para el cuestionario # 6 se aplicará la siguiente clasificación que afecta el promedio de satisfacción de los colaboradores:

- Para la pregunta 1, si el cliente da una respuesta negativa se resta dos puntos, si la respuesta es neutral resta un punto, ya que un comentario reservado o negativo puede influir en la decisión de continuar o no con un proyecto.
- Para la pregunta 2, si el cliente aporta aspectos negativos se resta dos puntos, ya que esto puede llegar a afectar la imagen de la empresa.
- Para la pregunta 3, lo que se busca es evaluar la percepción del cliente sobre el equipo de trabajo. Para ello se utilizará una escala de uno a diez de la siguiente forma:
    - 0 – 6: Ineficiente: es necesario reunirse con el equipo para identificar los problemas y corregirlos inmediatamente. Esta escala restará dos puntos.
    - 7 – 8: Neutro: requiere mejoras que se deben analizar con el equipo para mejorar la imagen con el cliente. Esta escala restará un punto
    - 9 – 10: Eficiente: un equipo trabajador, esforzado y aporta valor al cliente.
- Para la pregunta 4, lo que busca es evaluar la satisfacción del cliente con respecto a los productos desarrollados y entregados en cada iteración. Para ello, se utilizará una estala de uno a diez de la siguiente forma:
    - 0 – 6: Detractores: un cliente que puede perder el interés en la empresa y sus servicios, lo cual requiere una respuesta rápida para mejorar la imagen y recobrar su confianza. Esta escala restará dos puntos.

    - 7 – 8: Pasivos: un cliente que se mantiene en una posición neutral o de indecisión, lo cual requiere acciones que permitan ganar su confianza. Esta escala restará un punto.
    - 9 – 10: Promotores: un cliente satisfecho con el trabajo realizado, es potencial a promocionar la imagen de la empresa y sus servicios.
- Para la pregunta 5, lo que se busca es apelar a la sinceridad del cliente y evaluar su satisfacción real con los servicios prestados. En caso de que la respuesta sea negativa, se requiere una respuesta oportuna y un cambio de estrategia en el equipo, ya que puede provocar que el cliente no quiera continuar con los servicios ofrecidos por la empresa.

Una vez definido el promedio de satisfacción de los colaboradores final, podemos categorizarlo en la siguiente escala:

- 0 – 5: Desenfocado: indica que se requiere trabajar con el equipo, identificar los problemas que están afectando el estado anímico de los colaboradores, lograr un crecimiento y fomento de una cultura organizacional.
- 6 – 8: En crecimiento: indica que el equipo busca mejorar y lograr los objetivos y metas propuestas.
- 9 – 10: Enfocado: indica que el equipo trabaja orientado hacia los principios agiles, está motivado y concentrado en el logro de los objetivos y metas de la organización.

Es importante recordar, que el equipo ágil debe crecer enfocado en la atención continua y la satisfacción del cliente, que forman parte de los principios del manifiesto ágil (Beck, y otros, 2001). Por tal motivo, entre más rápida sea la retroalimentación, mejor será la respuesta a los cambios que sean necesarios, logrando fomentar la relación con los clientes, de modo que logre perfeccionar la relación con el equipo y no impacte la imagen de la organización.

Capítulo VII: Conclusiones y Recomendaciones

Conclusiones

A partir de lo desarrollado en esta investigación se presentan las siguientes conclusiones:

En el mercado hay una gran cantidad de herramientas para Integración Continua, pero se deben evaluar las características y los objetivos de la organización, para identificar cuáles se adaptan mejor y le permitirán obtener un mejor resultado.

La comunicación y la capacitación son dos elementos muy importantes, que una organización puede utilizar para apoyar la toma de decisiones y la ejecución de proyectos, con esto puede buscar el éxito de cada proyecto y minimizar los riesgos a los que se puede exponer.

Las organizaciones deben apoyar la formación de una cultura organizacional, de modo que los colaboradores puedan desarrollarse creyendo en los objetivos y las metas propuestas por el negocio. Esto para facilitar la integración y la participación de los empleados en los futuros proyectos.

Los colaboradores deben mantener siempre una moral alta, de modo que la organización pueda contar con todos los individuos a la hora de iniciar nuevos proyectos, y es deber de los encargados estar monitoreando que el ánimo de los equipos de trabajo se mantenga todo el tiempo, y contrastarlo con las métricas obtenidas del ciclo de desarrollo.

Además, es importante que el administrador de proyectos lleve un constante monitoreo, que le permita evaluar el cumplimiento de las metas y objetivos propuestos, incluyendo, el estado de ánimo de los colaboradores, con el fin de disminuir los riesgos del proyecto.

Asimismo, es necesario realizar evaluaciones periódicas para determinar el nivel de satisfacción tanto de colaboradores como de clientes, con el fin de mejorar el servicio que se ofrece. De esta forma, se podrá aportar un valor agregado a cada producto desarrollado por la organización, así como en los servicios que brinda. La retroalimentación que se pueda obtener de forma oportuna, podrá ser utilizada en beneficio de la organización para

apoyar las decisiones y marcar el rumbo que debe tomar en cuanto a la ejecución de proyectos.

Recomendaciones

Tomando como base lo planteado, en esta investigación, se podrá desarrollar la implementación de una solución de Integración Continua en una organización pequeña con bajo presupuesto. Asimismo, es importante tomar en consideración la madurez y el conocimiento de los colaboradores en este ámbito, de modo que se pueda anticiparse con capacitaciones de las herramientas antes de iniciar la ejecución del proyecto.

También, es posible aplicarlo en una empresa mediana, haciendo los ajustes apropiados de hardware, que maneje estructuras de equipos de trabajo pequeños no mayores a ocho integrantes. De esta forma, podrá implementar un ambiente con bajo presupuesto y adaptable al entorno organizacional.

Ante cualquier escenario, es muy importante poder realizar la evaluación de la satisfacción de las partes interesadas, con el fin de favorecer la mejora de todo producto y servicio brindado por la organización, aportando de esta forma valor agregado que permita sobresalir la competencia.

Bibliografía

Amazon. (s.f.). *Integración continua del software | Pruebas automatizadas | AWS*. Recuperado el 5 de Junio de 2018, de Amazon Web Services, Inc.: https://is.gd/zQ8Apb

Avendaño, M. (23 de Agosto de 2019). *Métricas para equipos ágiles*. Recuperado el 01 de Noviembre de 2019, de LinkedIn: https://is.gd/hfMe2S

BBVA. (16 de Febrero de 2017). *"Docker está cambiando la forma de hacer desarrollo de software"*. Recuperado el 10 de Junio de 2018, de BBVAOpen4U: https://is.gd/Zb4JS2

Beck, K., Beedle, M., Bennekum, A. v., Cockburn, A., Cunningham, W., Fowler, M., . . . Thomas, D. (2001). *Doce principios del software Ágil*. Recuperado el 08 de Octubre de 2019, de AgileManifesto.org: https://is.gd/Rk1s03

Cobb, S. (1 de Mayo de 2014). *Business continuity management: key to securing your digital future*. Recuperado el 08 de Octubre de 2019, de welivesecurity by ESET: https://is.gd/voTlGM

CollabNet, Inc. (7 de Mayo de 2019). *13th Annual State of Agile Report*. Recuperado el 12 de Junio de 2019, de CollabNet VersionOne: https://is.gd/tH59RI

Cortés Mejía, A. (2004). *Estilos de liderazgo y motivación laboral en el ambiente educativo.* Recuperado el 08 de Octubre de 2019, de Sistema de Información Científica Redalyc: https://is.gd/2P2Bnq

CruiseControl. (s.f.). *CruiseControl*. Recuperado el 08 de Octubre de 2019, de CruiseControl: https://is.gd/V64s2C

CruiseControl. (s.f.). *CruiseControl Home*. Recuperado el 5 de Agosto de 2018, de CruiseControl Home: https://is.gd/V64s2C

CruiseControl.NET. (s.f.). *Welcome to the CruiseControl.NET project*. Recuperado el 08 de Octubre de 2019, de CruiseControl.NET: https://is.gd/L3ivY8

Debian. (s.f.). *Acerca: Debian*. Recuperado el 08 de Octubre de 2019, de Debian: https://is.gd/cVwd7o

Duvall, P. M., Matyas, S., & Glover, A. (2007). *Continuous Integration.* Addison-Wesley .

Fitz, T. (08 de Febrero de 2009). *Continuous Deployment*. Recuperado el 10 de Junio de 2018, de https://is.gd/tey65G

Foladori, H., & Guerrero, P. (Edits.). (2017). *Malestar en el trabajo: Desarrollo e intervención.* (M. V. Urrutia, & P. Guerrero Morales, Trads.) LOM Ediciones.

Fowler, M. (1 de Mayo de 2006). *Continuous Integration*. Recuperado el 3 de Junio de 2018, de martinfowler.com: https://is.gd/yP1mMp

Free Software Foundation. (21 de Marzo de 2019). *¿Qué es el software libre?* Recuperado el 13 de Junio de 2019, de Free Software Foundation | gnu.org: https://is.gd/HFplRa

Galván Kondo, P. (1 de Mayo de 2017). *Integración Continua | SG Buzz*, Edición 54. (Software Gurú) Recuperado el 7 de Junio de 2018, de SG: Integración Continua: https://is.gd/EqJBUJ

Git. (s.f.). *Acerca: Git*. Recuperado el 08 de Octubre de 2019, de Git: https://is.gd/0Fd5D0

Git. (s.f.). *Documentación: Git*. Recuperado el 08 de Octubre de 2019, de Git: https://is.gd/NLeuwu

González, L., Carmona, M. Á., & Rivas, M. Á. (2007). *Guía para la medición directa de la satisfacción de los clientes.* Madrid, España: EGONDI ARTES GRÁFICAS S.A.

Graffius, S. M. (2016). *Agile Scrum: Your Quick Start Guide with Step-by-Step Instructions* (9781533370242 ed.). CreateSpace Independent Publishing Platform.

Guevara Mora, G. (05 de Septiembre de 2011). *Aprendizaje basado en problemas como técnica didáctica para la enseñanza del tema de la recursividad.* Recuperado el 08 de Octubre de 2019, de Portal de Revistas Académicas - UCR: https://is.gd/1Zew1K

Humble, J., & Farley, D. (2010). *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation* (Primera ed.). Addison-Wesley Professional.

International Organization for Standardization. (Marzo de 2014). *ISO/IEC 25000:2014 - Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - Guide to SQuaRE*. Recuperado el 4 de Junio de 2018, de ISO: International Organization for Standardization: https://is.gd/1QYaZJ

International Software Testing Qualifications Board. (s. f.). *Certifying Software Testers Worldwide - ISTQB® International Software Testing Qualifications Board*.

Recuperado el 5 de Junio de 2018, de ISTQB® International Software Testing Qualifications Board | Glossary: https://is.gd/gqVy01

Jenkins. (s.f.). *Jenkins*. Recuperado el 4 de Agosto de 2018, de Jenkins: https://is.gd/JxVl6O

Jenkins. (s.f.). *Jenkins*. Recuperado el 08 de Octubre de 2019, de Jenkins: https://is.gd/B7cqbx

Lemus Hernández, R. (2000). *La investigación en la comunicación organizacional.* (A. M. Palma, Ed.) Recuperado el 08 de Octubre de 2018, de Universidad Rafael Landívar: https://is.gd/UBRQbc

Letelier, P. (24 de Febrero de 2013). *Agility at work*. Recuperado el 08 de Agosto de 2018, de Carta de Prácticas Ágiles: Arma tu propio menú ágil: https://bit.ly/2OS8SHj

Letelier, P. (2014). *Agile Roadmap+*. Recuperado el 08 de Agosto de 2019, de Prácticas Ágiles de Agile Roadmap+: https://bit.ly/2QYV290

MariaDB Foundation. (s.f.). *About MariaDB.* Recuperado el 08 de Octubre de 2019, de The MariaDB Foundation: https://is.gd/x7bvNb

Marín, R. (16 de Abril de 2019). *Los gestores de bases de datos más usados en la actualidad*. Recuperado el 08 de Octubre de 2019, de Revista digital INESEM: https://is.gd/4hFPai

Martel, A. (2014). *Gestión práctica de proyectos con Scrum: Desarrollo de software ágil para el Scrum Master* (Tercera ed.). United States: CreateSpace Independent Publishing Platform.

Matos, M., Beriguete, M., & Peña, R. (2015). *Diseño de un plan de recuperación ante desastre (DRP): Plan de recuperación ante desastres (DRP) naturales o humanos, aplicado a una institución educativa.* Editorial Académica Española.

Mendoza, M. Á. (14 de Octubre de 2014). *¿En qué consiste un Plan de Recuperación ante Desastres (DRP)?* Recuperado el 08 de Octubre de 2019, de welivesecurity by ESET: https://is.gd/uFBFGa

Mercurial SCM. (s.f.). *Acerca: Mercurial SCM*. Recuperado el 08 de Octubre de 2019, de Mercurial SCM: https://is.gd/pW8VKG

Oracle Corporation and/or its affiliates. (s.f.). *Acerca: MySQL*. Recuperado el 08 de Octubre de 2019, de MySQL: https://is.gd/zbwgvU

Organización Internacional del Trabajo. (2016). *Mejore su negocio: el recurso humano y la productividad* (I ed.). Ginebra, Suiza: International Labour Office. Enterprises Dept. Obtenido de https://is.gd/fISg3j

Priolo, S. (2009). *METODOS AGILES: Espanol, Manual Users, Manuales Users.* Creative Andina Corp.

Project Management Institute. (2018). *Agile Practice Guide (Spanish)* (Primera ed.). (P. M. Institute, Ed.) United States.

Red Hat. (s.f.). *Productos*. Recuperado el 08 de Octubre de 2019, de Red Hat: https://is.gd/uioFxN

Rodríguez, J. R., García Mínguez, J., & Lamarca Orozco, I. (Junio de 2011). *Gestión de proyectos informáticos: métodos, herramientas y casos Ebook.* Barcelona: Editorial UOC, S.L. Recuperado el 08 de Octubre de 2019

Romero P., F. (05 de Octubre de 2016). *Estrategia y Gestión de Contratos en Proyectos.* Recuperado el 20 de Octubre de 2019, de Deloitte.com: https://is.gd/MrxJlP

SecuritySpace. (1 de Septiembre de 2019). *Research Reports: OS/Linux Distributions using Apache*. Recuperado el 08 de Octubre de 2019, de SecuritySpace: https://is.gd/BpYv24

SonarSource S.A. (2008). *SonarQube*. Recuperado el 08 de Octubre de 2019, de SonarQube: https://is.gd/HdW16k

TablePlus. (11 de Septiembre de 2018). *MariaDB vs PostgreSQL - A quick comparison*. Recuperado el 08 de Octubre de 2019, de TablePlus: https://is.gd/k2deOZ

Tanenbaum, A. S. (2004). *Sistemas Operativos Modernos* (Segunda ed.). Prentice Hall.

The CentOS Project. (s.f.). *Acerca: CentOS*. Recuperado el 08 de Octubre de 2019, de The CentOS Project: https://is.gd/Pck7Uw

The PostgreSQL Global Development Group. (s.f.). *The PostgreSQL Global Development Group*. Recuperado el 08 de Octubre de 2019, de PostgreSQL: https://is.gd/BD9fU6

ThoughtWorks Inc. (s.f.). *GoCD*. Recuperado el 08 de Octubre de 2019, de GoCD: https://is.gd/9rDM91

TortoiseHg. (s.f.). *Acerca: TortoiseHg*. Recuperado el 08 de Octubre de 2019, de TortoiseHg: https://is.gd/HSfH0E

Trejos Zelaya, I. (26 de Junio de 2018). *Arquitectura empresarial y TIC.* Recuperado el Octubre de 2019, de Cámara de Tecnologías de Información y Comunicación | CAMTIC: https://is.gd/DRoVqY

Ubuntu. (s.f.). *Acerca: Ubuntu*. Recuperado el 08 de Octubre de 2019, de Ubuntu: https://is.gd/5x5kI9

W3Techs. (s.f.). *Usage statistics of Linux for websites*. Recuperado el 08 de Octubre de 2019, de W3Techs: https://is.gd/Ia3Ywq

Anexos

Instrumentos para recolección de datos

Cuestionario #1

El siguiente cuestionario está dirigido al encargado del equipo de trabajo (Administrador de proyecto, Líder técnico), con el fin de conocer el ambiente de trabajo, la madurez del equipo y la interacción de los participantes con las metodologías ágiles. Al ser de tipo abierto, se da la posibilidad de obtener más información de los encuestados y de poder formular más preguntas, acordes al tema, según las respuestas de los mismos.

Lugar: Fecha:

Encargado del equipo (Administrador de proyecto, Líder técnico)

1. ¿Cuánto tiempo tiene el equipo de formado?
2. ¿Cuánto tiempo tienen los integrantes de trabajar juntos?
3. ¿Cuánto tiempo tiene el equipo de trabajar con metodologías ágiles?
4. ¿Cuáles son los medios de comunicación que utiliza el equipo?
5. ¿Cómo es la relación de los integrantes del equipo?
6. ¿Han tenido enfrentamientos entre sí? ¿Cómo los han resuelto?
7. ¿Considera que el equipo es arriesgado? ¿Podría el equipo aceptar un cambio en su forma de trabajar?
8. ¿Cómo considera que es el desempeño de los integrantes del equipo?
9. ¿Cómo actúa el equipo frente a los problemas? ¿Debaten entre ellos o buscan la solución por su cuenta?
10. ¿Cómo actúan los integrantes ante la supervisión de su trabajo? ¿Cómo actúan ante la crítica de su trabajo?
11. ¿Cuánto tiempo aproximadamente se requiere para tener una versión beta funcional? *
12. ¿Cuántos cambios se le deben hacer a esa versión después de ser evaluada por el encargado de validación?
13. ¿Cuántas veces no se ha cumplido la fecha de entrega de un producto por retrasos en los tiempos de entrega? ¿Por qué motivo no se logró?
14. De acuerdo a la respuesta de la pregunta 11, ¿Cómo actuó el equipo ante esa situación?

15. De acuerdo a la respuesta de la pregunta 11, ¿De qué manera cree usted que se pudo minimizar el impacto de esos motivos?
16. Ante la situación descrita en las preguntas anteriores, ¿Considera usted que el equipo de trabajo ha logrado un buen nivel de madurez?
17. ¿Cómo es la participación del cliente/dueño del producto durante el proceso de desarrollo?

Notas:

* Entiéndase *versión beta funcional*, como un producto con el cual se puede interactuar, que puede ser liberado para obtener retroalimentación por parte del cliente o para ser probado por el departamento de validación.

Cuestionario #2

El siguiente cuestionario está dirigido los desarrolladores, con el fin de conocer el ambiente de trabajo, su interacción con la metodología de trabajo, los retos a los que se enfrentas y su punto de vista de esta situación. Al ser de tipo abierto, se da la posibilidad de obtener más información de los encuestados y de poder formular más preguntas, acordes al tema, según las respuestas de los mismos.

Lugar: Fecha:

Desarrollador

1. ¿Cuál es su opinión de la actual metodología de trabajo? ¿Cree que necesita algún ajuste?
2. ¿Cómo ven los demás integrantes del equipo la actual metodología de trabajo?
3. ¿Cuáles considera usted que son los problemas más comunes en el desarrollo de sus tareas diarias?
4. ¿Cómo interactúan los integrantes del equipo ante esos problemas?
5. ¿Cómo cree usted que se podría resolver esos problemas?
6. ¿Quién considera usted que debe encargarse de resolver esos problemas?
7. ¿Considera usted que las tareas están bien equilibradas por persona?
8. ¿Qué tan frecuente el departamento de validación reporta problemas en un producto? ¿Cuáles han sido los motivos?
9. ¿Cree usted que es necesario agregar pruebas para validar el proceso de desarrollo? ¿Cree usted que podría ser de utilidad en las tareas que realiza?
10. ¿Hay guías o prácticas a seguir al desarrollar productos de software? ¿Considera que son útiles? ¿Sabe usted si están actualizadas?

Cuestionario #3

El siguiente cuestionario está dirigido los encargados de validación, con el fin de conocer el ambiente de trabajo, su interacción con la metodología de trabajo, los retos a los que se enfrentas y su punto de vista de esta situación. Al ser de tipo abierto, se da la posibilidad de obtener más información de los encuestados y de poder formular más preguntas, acordes al tema, según las respuestas de los mismos.

Lugar: Fecha:

Encargado de validación

1. ¿Cuál es su opinión de la actual metodología de trabajo?
2. ¿Cómo ven los demás integrantes del equipo la actual metodología de trabajo?
3. ¿Cuáles son los errores más frecuentes encontrados en los diferentes productos de software? ¿Sabe si se aplica alguna categorización para esos errores?
4. ¿Cómo considera usted que se podría minimizar esos errores frecuentes?
5. ¿Considera usted pertinente que se desarrollen bases de datos de conocimiento para que puedan ser consultados y evitar esos errores comunes?
6. ¿Qué tan complejo es evaluar un componente de software? ¿Cuánto tiempo requiere aproximadamente?*
7. ¿Considera usted que sería de apoyo automatizar algunas de las pruebas sobre los productos de software?
8. ¿Quién considera usted que debería encargarse de desarrollar esas pruebas?

Nota:

*Entiéndase *componente de software*, como una unidad de software que puede ser aislada y testeada la cual es independiente del producto final.

Se considera incluir la siguiente lista de prácticas desarrollada por Patricio Letelier, (2013) y (2014), con el fin de evaluar la implementación de la metodología agile y aplicar recomendaciones que permitan mejorar el proceso.

Cuestionario #4

El siguiente cuestionario está dirigido a la metodología de trabajo utilizada por el equipo, con el fin de realizar ajustes y/o recomendaciones que permitan mejorar el rendimiento del equipo.

Evaluación de la práctica agile en la organización

	Práctica	Metodología	Aplica
PRA01	Promover la sencillez en todos los aspectos. Ofrecer la solución más simple y mínima que pueda ser satisfactoria para el cliente.	Lean, XP	
PRA02	Abordar y entregar trabajo terminado de forma incremental.	Kanban, XP, Scrum	
PRA03	Realizar entregas frecuentes de unidades de trabajo terminadas.	Kanban, XP, Scrum	
PRA04	Realizar reuniones de planificación frecuentemente (frecuencia de pocas semanas, no meses).	XP, Scrum	
PRA05	Acotar el trabajo previsto para un período en base a su estimación y la correspondiente coherencia con la capacidad del equipo.	XP, Scrum	
PRA06	Organizar el trabajo en iteraciones que agrupan unidades de trabajo que son entregadas en una fecha prevista.	XP, Scrum	
PRA07	Evitar invertir esfuerzo en adelantar trabajo que no esté comprometido y/o no esté cercano a su entrega.	Lean	
PRA08	Organizar el trabajo del equipo con el foco en la generación de un buen flujo de trabajo terminado.	Kanban	

PRA09	Gestión continua y multicriterio del trabajo pendiente para que esté siempre debidamente priorizado.	Scrum	
PRA10	Limitar el trabajo en proceso (WIP), es decir, la cantidad de unidades de trabajo que tiene el equipo en una determinada actividad.	Kanban	
PRA11	Formar equipos pequeños y procurar que mantengan sus integrantes.	XP, Scrum	
PRA12	Acotar el ámbito de trabajo de cada equipo.	---	
PRA13	Seguimiento continuo (frecuencia de días, no semanas).	Kanban, XP, Scrum	
PRA14	Realizar una reunión diaria del equipo al completo, cara a cara y muy breve.	XP, Scrum	
PRA15	Visualización de todo el trabajo pendiente encargado al equipo.	Kanban	
PRA16	Gestión integrada de todo el trabajo asignado, tanto a nivel del equipo como a nivel de cada miembro.	---	
PRA17	Cliente en estrecho contacto con el equipo y altamente disponible, incluso si es posible, que esté in-situ	XP, Scrum	
PRA18	Que exista una única persona que tome las decisiones respecto de las prioridades del trabajo del equipo y que sea un buen representante de la parte cliente	XP, Scrum	
PRA19	Realizar reuniones de revisión del trabajo entregado	Scrum	
PRA20	El equipo se auto-organiza y toma las decisiones técnicas	Scrum	
PRA21	Jefe de carácter líder y facilitador en lugar de actitud del jefe autoritario y controlador	XP, Scrum	
PRA22	Co-localización de los miembros del equipo, todo el equipo trabajando en el mismo espacio físico.	XP, Scrum	
PRA23	Contar con un espacio físico de trabajo que favorezca	XP	

	la interacción entre los miembros del equipo.		
PRA24	Establecer y comunicar al equipo la visión del producto o servicio, y reforzarla regularmente.	XP	
PRA25	Que el equipo sume entre sus miembros las habilidades para abordar todas las actividades necesarias para terminar el trabajo.	Scrum	
PRA26	Que los integrantes del equipo puedan encargarse de diferentes tipos de actividades (ojalá de todas), aunque puedan ser especialistas en alguna(s) de ellas.	Scrum	
PRA27	Trabajo centrado en satisfacer pruebas de aceptación acordadas con el cliente.	XP	
PRA28	Documentar, pero solo lo estrictamente necesario. Que sea rentable el aprovechamiento de la documentación respecto del esfuerzo asociado a elaborarla.	Lean	
PRA29	Establecer pautas para gestionar convenientemente el re-trabajo.	Lean	
PRA30	Que exista un líder de mejora de proceso disponible para el equipo.	XP, Scrum	
PRA31	Establecimiento de estándares para el trabajo técnico del equipo	XP	
PRA32	Realizar reuniones de retrospectiva para evaluar el desempeño del equipo y sus formas de trabajo. Mejora continua del proceso.	Scrum	
PRA33	Acordar y definir qué se entiende por trabajo terminado, tanto para las actividades realizadas por el equipo como respecto de las entregas al cliente	Scrum	
PRA34	Trabajo o actividades realizadas en conjunto por dos o más integrantes	XP	
PRA35	No abusar de las horas extras, negociar y re-planificar	XP	

	oportunamente para evitarlo		
PRA36	Reducir las interrupciones o cambios de contexto que afectan en su trabajo a los miembros del equipo	Lean	
PRA37	Establecer una disciplina de aprovechamiento de las reuniones	---	
PRA38	Automatizar las pruebas para poder garantizar que el producto mantiene el comportamiento deseado cuando se realizan cambios	XP	
PRA39	Postergar hasta último momento la asignación del encargado de realizar una actividad.	---	
PRA40	Integrar de forma continua en el producto el trabajo terminado	XP	
PRA41	Promover que los miembros del equipo en su trabajo lleguen a conocer todas las partes del producto o servicio que han sido encargadas al equipo.	XP	
PRA42	Mejorar continuamente la organización interna del producto para facilitar su mantenimiento.	XP	

Cuestionario # 5

El siguiente cuestionario está dirigido cada colaborador, con el fin de conocer su satisfacción y sentimientos ante el ambiente en el que desarrolla su trabajo. Al ser de tipo abierto, se da la posibilidad de obtener más información de los encuestados y de poder formular más preguntas, acordes al tema, según las respuestas de los mismos. Puede considerar hacer estas preguntas de forma digital si es oportuno o si la persona esta geográficamente distante.

Lugar: Fecha:

Colaborador

¿Cómo considera que fue la iteración que concluyó?

¿Qué aspectos considera que influyeron más en la iteración?

¿Cómo influyeron sus compañeros en la iteración?

¿Qué aspectos considera que pueden mejorarse para las próximas iteraciones?

¿Qué considera que podría aportar usted para lograr esas mejoras?

¿Considera que se cumplieron los objetivos de la iteración de la mejor forma?

¿Cómo se siente con el producto entregado? ¿Le gustaría cambiar o mejorar algo?

¿Cómo se siente trabajando en el proyecto actual? ¿Le gustaría cambiar por un tiempo?

¿Cómo se siente trabajando con su equipo? ¿Le gustaría cambiar por un tiempo?

¿Cómo se siente con la carga de trabajo?

Cuestionario # 6

El siguiente cuestionario está dirigido cada cliente, con el fin de conocer su satisfacción con el producto que se le ofrece. Al ser de tipo abierto, se da la posibilidad de obtener más información de los encuestados y de poder formular más preguntas, acordes al tema, según las respuestas de los mismos. Puede considerar hacer estas preguntas de forma digital si es oportuno o si la persona esta geográficamente distante.

Lugar: Fecha:

Cliente

¿Considera que el equipo resuelve sus dudas de forma oportuna durante la iteración?

¿Qué aspectos considera que pueden mejorar en la próxima iteración?

¿Cómo considera que fue el rendimiento del equipo durante la iteración?

¿Qué tan satisfecho esta con el producto presentado en la iteración?

¿Recomendaría usted los servicios ofrecidos?