



UNIVERSIDAD TÉCNICA DE BABAHOYO

FACULTAD DE ADMINISTRACIÓN, FINANZAS E INFORMÁTICA

NOVIEMBRE 2021 – ABRIL 2022

**EXAMEN COMPLEXIVO DE GRADO O DE FIN DE CARRERA PRUEBA
PRACTICA**

INGENIERIA EN SISTEMAS

PREVIO A LA OBTENCION DEL TITULO DE INGENIERIA EN SISTEMAS.

TEMA:

**ESTUDIO COMPARATIVO DE LAS HERRAMIENTAS DE METODOLOGÍAS
ÁGILES PARA EL APLICAR BUENAS PRÁCTICAS DE DESARROLLO EN
LA CALIDAD DE SOFTWARE.**

EGRESADA:

CINTHIA KARINA YÁNEZ TRIANA.

TUTOR:

ING. MIGUEL ÁNGEL ZÚÑIGA SÁNCHEZ.

RESUMEN

En la actualidad las empresas tienen la necesidad de automatizar sus procesos, las cuales optan por implementar sistemas de información en sus departamentos, con la finalidad de poder procesar eficientemente y en un tiempo óptimo. El software de mala calidad siempre representa riesgos. La mala calidad de software es resultado de malas prácticas en el desarrollo. A pesar de las ventajas que ofrece el desarrollo de software bajo un enfoque ágil, existen aspectos relacionados con el aseguramiento de la calidad que aún deben ser tratados. El presente documento se ha desarrollado utilizando un enfoque de investigación bibliográfico, en el cual se evidencia la consulta en bases de datos indexadas de internet, diferentes herramientas de metodologías ágiles para el desarrollo. En la industria del software moderna, la entrega de software de alta calidad es el principal objetivo para los desarrolladores de software. Este caso estudio proporciona una comparación entre las metodologías ágiles más populares, Scrum, Extreme Programan y Kanban. Existe una necesidad importante de que los desarrolladores sepan más sobre la calidad del software producido. Las empresas deben considerar que el principal problema durante el proceso de desarrollo de Software es el cambio en las especificaciones funcionales, seguido de la planificación muy optimista y que el uso de las metodologías ágiles les permitirá tener una mayor flexibilidad en los cambios requeridos.

Palabras clave

Comparativo, Herramientas, Ágiles, Calidad, Software

ABSTRACT

Nowadays, companies have the need to automate their processes, which choose to implement information systems in their departments, in order to be able to process efficiently and in optimal time. Poor quality software always poses risks. The causes that affect the quality of software are the result of bad practices in development. Despite the advantages offered by software development under an agile approach, there are aspects related to quality assurance that still need to be addressed. This document has been developed using a bibliographical research approach, in which the consultation in indexed internet databases, different tools of agile methodologies for development, is evidenced. In the modern software industry, delivering high quality software is the main goal for software developers. This case study provides a comparison between the most popular agile methodologies, Scrum, Extreme Programming and Kanban. There is a significant need for developers to know more about the quality of the software produced. Companies must consider that the main problem during the software development process is the change in functional specifications, followed by very optimistic planning and that the use of agile methodologies will allow them to have greater flexibility in the changes required.

Keywords

Comparative, tools, agile, quality, software

INTRODUCCIÓN

En la actualidad las empresas tienen la necesidad de automatizar sus procesos, las cuales optan por implementar sistemas de información en sus departamentos, con la finalidad de poder procesar eficientemente y en un tiempo óptimo. La principal problemática que tienen las empresas al adquirir o desarrollar software es la calidad del mismo. El software de mala calidad siempre representa riesgos. Las causas que afectan la calidad de software son resultado de malas prácticas en el desarrollo que aparecen desde la concepción del sistema.

Un problema principal del software de mala calidad son los costos que se derivan después de su implementación. Muchas veces estos costos son subestimados y se desconoce el impacto que pueden llegar a generar. Debido a esto en la creación de software se han implementado buenas prácticas de desarrollo aplicando las metodologías ágiles. Las metodologías ágiles de desarrollo de software están posicionándose fuertemente en las empresas más importantes del sector. Contemplando el desarrollo de software de manera integral, con un énfasis especial en la entrega de valor al cliente, en la generación de negocio y el retorno de la inversión.

El Aseguramiento de la Calidad de Software (ACS), es un conjunto de métodos, herramientas y técnicas que permiten gestionar la calidad en el desarrollo de un producto de software”. Las organizaciones hoy en día se ven abocadas al uso de metodologías de desarrollo ágiles, ya que nos permiten tener diferentes perspectivas y reaccionar ante cambios inesperados durante el desarrollo del proyecto, y brindar respuestas favorables y eficientes.

A pesar de las ventajas que ofrece el desarrollo de software bajo un enfoque ágil, existen aspectos relacionados con el aseguramiento de la calidad que aún deben ser tratados. Métodos como el Test Driven Development (TDD) y la Integración Continua, son empleados para tal fin; sin embargo, su uso no garantiza un desarrollo con calidad, ya que cada proyecto posee diferentes particularidades.

Acerca de las herramientas para garantizar el aseguramiento de la calidad no se encuentra documentada. Sin embargo, se logró recabar buenas prácticas que aseguran la calidad, en las cuales se centrará este trabajo de investigación. La idea central de este estudio de caso es analizar y estudiar de forma comparativa las herramientas de metodología ágiles que ayuden a la aplicación de buenas prácticas utilizadas para asegurar la calidad en los proyectos de software.

El presente documento se ha desarrollado utilizando un enfoque de investigación bibliográfico, en el cual se evidencia la consulta en bases de datos indexadas de internet, diferentes herramientas de metodologías ágiles para el desarrollo, y consecuentemente, con los resultados realizar una comparativa para determinar la opción más óptima, en la aplicación de buenas prácticas para asegurar la calidad de los sistemas de información.

El presente caso de estudio se relaciona con la línea de investigación referente a los sistemas de información y comunicación, emprendimiento e innovación, y a su vez está relacionada con la sublínea de las redes y tecnologías inteligentes de software y hardware.

Desarrollo

En un principio fueron apareciendo distintas técnicas para aplicar buen desarrollo en software, y a su vez lograr tener una metodología de trabajo con una buena organización. Luego aparecieron nuevas metodologías, como lo es el manifiesto ágil que puede llegar hacer aceptado por algunos expertos para solventar la necesidad de los usuarios y así tener una respuesta factible.

Un pequeño concepto de la metodología ágil nos dice que surgió debido a la necesidad de corregir los problemas existentes, los cuales fueron: la forma incorrecta durante la utilización de las practicas al momento de desarrollar un proyecto de software, dando como resultado un producto defectuoso y que no cumple los requerimientos que fueron solicitados por el usuario, y lograr una reducción significativa de la documentación y estructuración de las metodologías que antes conocidas como pesadas, debido a que requerían demasiado tiempo y esfuerzo, con lo que las entregas se realizaban con un retraso fuera de la fecha establecida. También generaban productos de mala calidad, los cuales ocasionaban problemas al momento de ser entregados, por otra parte, desarrollaban los proyectos de software de manera lineal y poco eficiente a los requerimientos iniciales del contrato o acuerdo pactado con el cliente.

Para el desarrollo de los proyectos de software, debemos seguir las siguientes características:

- Las interacciones y los individuos primero y proceso de herramientas.

- El producto trabajando frente a la gran cantidad de documentación
- El trabajo conjunto con el cliente ante la negación contractual.
- El resultado de los cambios frente al plan establecido.

Las características antes mencionadas permiten al usuario realizar el desarrollo de forma más rápida y optimizada con el mismo, y sabiendo que podemos trabajar por partes, pero en equipo. Contando los beneficios que este nos ofrece tanto para el desarrollo y para cliente.

Varias de las desventajas que tiene la metodología ágil son.

- Al no contar con la documentación adecuada podría existir una disminución en la reusabilidad del código.
- Cuando un proyecto ágil falla es muy probable volver a los métodos de trabajo tradicionales.
- Es muy fácil llegar a pensar que un proyecto ágil no requiere de documentación.
- Es normal una fuerte dependencia de los responsables. La persona encargada de dirigir el proyecto realiza decisiones centralizadas.

Varias de las metodologías ágiles son: scrum, Kanban, programación extrema, de las cuales la más empleada es la scrum debido a sus múltiples ideas para empresas. El cual es utilizado por su complejidad con el equipo de trabajo, siendo uno de los primeros en encontrar errores para posteriormente

resolverlo a tiempo y obtener como resultado un proyecto preciso y conciso tanto para el cliente y la empresa.

Kanban es parte de la metodología ágil, este software trabaja por medio de tableros el cual nos da la posibilidad de visualizar todos los flujos de trabajo para tener un equilibrio y disponibilidad entre equipo.

Actualmente la metodología ágil se adapta a los cambios, ya que es posible configurar los ajustes que cada equipo tiene, de acuerdo a su proyecto. Conforme al tipo de metodología que se esté utilizando, utiliza un desarrollo progresivo, ofreciendo un excepcional clima de trabajo y comunicación en equipo, está relacionado frecuentemente con el usuario. (ALEJANDRO, 2020).

El manifiesto ágil.

Este nos describe cuales son los principios sobre los que se basan los métodos conocidos como ágiles. Éste manifiesto nos indica un enfoque orientado a la participación de los usuarios y clientes, en lugar que, hacia los procesos y herramientas, haciendo más trabajando más en el software y menos en la documentación, cooperando de forma más activa con los clientes en vez de estar negociando y respondiendo a los cambios comprometiendo el plan de trabajo si es necesario. (Maida, 2015)

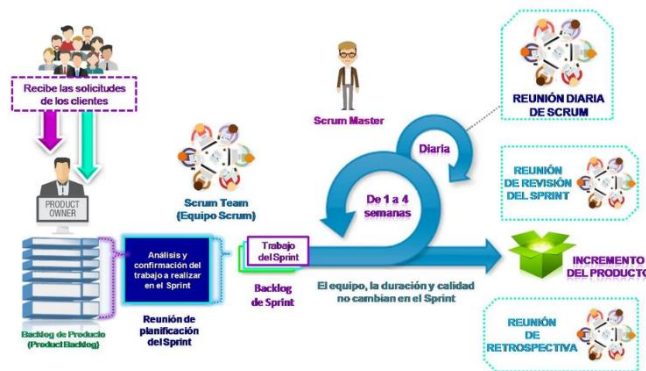
Tipos de metodologías ágiles.

- Scrum.
- Programación extrema (xp).
- Kanban.

Scrum

Esta metodología, en el ámbito de trabajo del desarrollo ágil, que trabaja con el ciclo de vida iterativo e incremental, donde se va liberando el producto por pares de forma periódica, aplicando las buenas prácticas de trabajo (en equipo), facilitando soluciones óptimas a los problemas que pueden ir apareciendo mediante el desarrollo del proyecto.

Scrum realiza entregas parciales (sprint) de producto final, esto nace de las necesidades que tiene el cliente. Este método está especializado para proyectos profundo, donde se necesitan obtener resultados de manera inmediata sin dejar de tomar en cuentas los aspectos: la innovación, la productividad, la flexibilidad y la



competitividad. (MURADAS, 2018)

Ilustración 1. Modelo Scrum.

Fuente: <https://openwebinars.net/blog/conoce-las-3-metodologias-agiles-mas-usadas/>

Programación extrema (Xp).

XP es una metodología ágil basada en el éxito en el desarrollo de software utilizando las relaciones interpersonales como clave, que sería uno de los primeros que tiene este método. Esta metodología trabaja en equipo, preocupándose por que los desarrolladores tengan un buen aprendizaje, y proporcionando un buen clima de trabajo. XP se basa en la comunicación entre el cliente y el equipo de desarrollo, existe una continua comunicación entre todos los participantes, simplicidad en las soluciones implementadas y valor para enfrentar. (LEÓN, 2016)



Ilustración 2 Método XP

Fuente: <https://openwebinars.net/blog/conoce-las-3-metodologias-agiles-mas-usadas/>

Kanban

Kanban es un método visual que controla la producción que está formada por un sistema de señales que a lo largo de toda la cadena de

producción dirige el proceso de reabastecimiento y empieza con el conocimiento de lo que el cliente demanda, hasta que obtener el producto final. El sistema Kanban se encarga de controlar las piezas o componentes que están dentro de la cadena de producción que sean cantidades suficientes para reemplazar las que ya se han utilizado, consiguiendo así una producción sin existencias.

Kanban es el método común entre las empresas y organizaciones que buscan aumentar la flexibilidad de los negocios, de este modo mejorar los servicios que brindan a sus clientes, sin tener que realizar cambios relativamente extensos. (Lendínez, 2019).



Ilustración 3 Método Kanban.

Fuente: <https://www.egestionaconsultores.es/el-metodo-kanban-para-mejorar-la-productividad/>

Cuando hablamos de calidad de software es hablar de un rol importante para el desarrollador y la aceptación del mismo para esto se necesitan varias características que deben cumplir y garantizar al cliente que su función será eficiente ya que esta mide las expectativas y necesidad del mismo, sin dejar de

tomar en cuenta los requisitos y parámetros que el usuario desee para la medida de calidad establecidas.

De acuerdo a los modelos de calidad de software para obtener una mejor práctica debemos de tomar en cuenta los documentos que proporciona para tener un énfasis en la práctica y poder avanzar, tener una mejora al momento de brindar un servicio de buena calidad.

Esto significa que los modelos de software son una construcción que nos ayudan a evaluar y medir las necesidades de los clientes, para así poder implementar estrategias de las cuales nos permitirá mejorar las fases de análisis, diseño, desarrollo y prueba de software.

Los modelos de calidad de software según su clasificación son.

- Calidad a nivel de proceso
- Calidad a nivel de producto
- Calidad en uso

Calidad a nivel de proceso significa que tiene que ser programada desde su inicio y casi hasta su final llevando un seguimiento para evitar riesgo, pudiendo así garantizar el debido cumplimiento con el mismo. No olvidemos que durante este proceso no podemos dejar de pasar la fase de verificación ya que el nivel de calidad disminuiría tanto para el proceso inicial y el producto final.

Calidad a nivel de producto se trata de evaluar los diferentes criterios para poder alcanzar la satisfacción del usuario dependiendo de la calidad del producto el cual puede ser: interna, externa y en uso. De esto depende alcanzar la satisfacción del cliente.

Calidad en uso este hace énfasis a la usabilidad y está basada en brindar seguridad y satisfacción y aceptación por parte del usuario final. (Álvarez-Carreño, Modelos de calidad del software, un estado del arte., 2016)

Metodologías de desarrollo de software.

Las metodologías de desarrollo surgen a partir de las múltiples necesidades de las empresas con el fin de obtener buena calidad de los productos de software desarrollados; por tal motivo decidieron diseñar un entorno de trabajo con el fin de mejorar los pasos del desarrollo de las aplicaciones en sus diferentes etapas que tiene el ciclo de vida del software. (ALEJANDRO, 2020)

Las metodologías se encuentran divididas en tradicionales y ágiles, también existen las híbridas en estos últimos años dependiendo de las necesidades del proyecto. (DEMERA, 2021).

Factores de calidad de software.

Concepto de calidad.

La calidad es la capacidad de cumplir con los requisitos o expectativas que tenga el usuario a cerca del producto. Y así cumpla con las necesidades que él requiera. Se puede utilizar herramientas o indicadores de calidad con los cuales podrían descubrir si los productos serian útiles para las personas o no. (MARIANA, 2021).

Modelo de calidad de software

Calidad al nivel de proceso.

Es un desarrollo que puede ser planificada desde el principio y regularmente en las fases que se va desarrollando, también se debe llevar un control o seguimiento de los aspectos de cualidad y poder evitar riesgos e ofrecer soporte continuo, se garantiza un nivel de cumplimiento optimo en los factores de calidad, sin dejar de tomar en cuenta los factores o criterios que se pueden presentar en el mismo que puede decrecer el nivel de calidad no solo del proceso, sino del desarrollo. (Álvarez-Carreño, Modelos de calidad del software, un estado del arte, 2016)

Calidad al nivel de producto.

El nivel de producto es un servicio que definen la apreciación que tiene el cliente sobre su valor. Si la clientela considera que lo que brinda el nivel le puede resolver ciertos requisitos, lo consideraría de una mayor función de manera en que recibe una serie de diversos asociados a ese resultado. (Quality, 2019).

Calidad en uso.

Es importante recalcar algunos escenarios que utilizan los términos de facilidad y calidad en uso, la definición de calidad en uso es más amplia y abarca un poco más de componentes que la facilidad y es una de la ultimas características de producto en software. El nivel de uso es el conjunto de atributos que se relaciona con la aceptación por parte del usuario y seguridad, y se encuentra basada en la eficacia, productividad, seguridad y satisfacción. (Álvarez-Carreño, 2016)

Ingeniería de requerimientos.

Es un desarrollo que comprende las actividades de crear mantener un documento de requerimientos del sistema. Se deben estructurar un método, el cual estudie las obligaciones del sistema y desarrollar un conjunto de guía gráficos del mismo que actúan en ciertas una especificación.

Actividades genéricas:

Que el sistema sea factible, los requerimientos tengan un buen análisis, que sean específico los requerimientos, Verificación.

las actividades es el resultado de algún tipo de modelo que describe el proceder del sistema al que se le añade un avance con información extra, que detallan el cumplimiento o confianza requerida. (Sommerville)

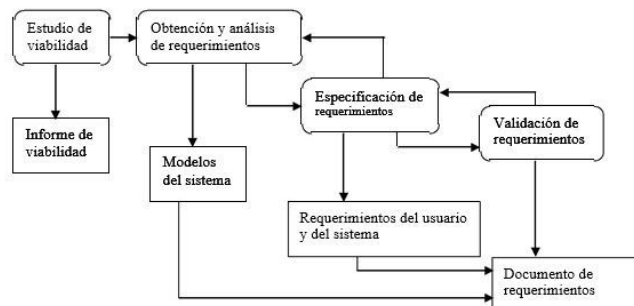


Ilustración 4 Ingeniería de requerimiento

Fuente: <http://requerimientosadsi118.blogspot.com/2016/08/ingenieria-de-requerimientos.html>.

Para el desarrollo de la presente investigación se utilizó la metodología de investigación bibliográfica y la técnica de la observación de acuerdo a ella pudimos obtener cierta información. para reforzar el caso de estudio utilizamos como instrumento recolección de información de conceptos científicos y datos importantes, por medio de análisis profundos de revistas universitarias, Bibliotecas en líneas, repositorios universitarios y artículos científicos, para el

desarrollo de la comparación entre las metodologías. La investigación bibliográfica fue desarrollada en las siguientes fases:

- Investigaciones bibliográficas por medios del uso de buenas prácticas y herramientas para asegurar la calidad del software.
- Establecer la comparativa entre las diferentes herramientas previamente consultadas.
- Análisis de los resultados obtenidos.
- Elaboración de las conclusiones obtenidas luego del desarrollo del trabajo de investigación.

Estudio comparativo entre la metodología Scrum y Xp.

Parámetros	Desarrollo basado en Scrum	Programación extrema (XP)
Principio de diseño	Diseño complejo	Código simplificado y adecuación de cambios inesperados por medio de la refactorización
Naturaleza de la interacción con el cliente	No necesariamente presencial	Interacción con el cliente en el sitio
Complejidad del diseño	Diseño profundo	Diseño sencillo
Coordinador del proyecto	Maestro Scrum	Entrenador XP

Funciones asignadas	3 roles predefinidos: dueño del producto, Scrum Master y equipo de desarrollo	Sin roles preestablecidos
Propiedad del proceso	Maestro Scrum	Propiedad del equipo
Propiedad del producto	El dueño del producto es responsable del producto.	Responsabilidad del grupo del producto
Colaboración en equipo	Equipos multifuncionales	Equipos autoorganizados
Enfoque de flujo de trabajo	Iteraciones (Sprint)	Sin iteraciones. Desarrollo del flujo de tareas
Gestión de requerimientos	Requisitos gestionados por medio de artefactos a través de Sprint Backlog y Product Backlog	Gestionado por medio de Story Cards
Entrega del producto	Entrega conforme sprints en caja de tiempo	Entrega continua
Estándares de codificación	Sin estándares de codificación	Se utilizan estándares de codificación.
Enfoque de prueba	No se utiliza un enfoque formal para las pruebas.	Desarrollo basado en pruebas, incluidas las pruebas de aceptación
Acomodación de	No se permite cambio en	Apto al cambio incluso en

Cambios	Sprint.	etapas posteriores de desarrollo.
---------	---------	-----------------------------------

Tabla 1 Tabla comparativa entre Scrum y Xp

Elaborado por: Cinthia Yáñez

En relación a la comparativa entre Scrum y XP, En Scrum, los equipos y las reuniones están completamente definidos por equipo de trabajo designados, mientras que la cuestión de cómo se hace el trabajo se la deja a juicio de cada uno de los equipos para y ellos son lo que deciden como se realizara. XP, por otro lado, viene con un conjunto de prácticas básicas que podrían parecer bastante difíciles y complicadas para los miembros principiantes. Podríamos pensar que Scrum es una metodología más orientada a la productividad mientras que xp se inclina más por la ingeniería.

Estudio comparativo entre las metodologías Scrum y Kanban.

	Scrum	Kanban
Funciones	Depende del propietario del producto, equipo de desarrollo y el maestro de scrum es opcional.	No tiene funciones definidas y todavía consta con un director de proyecto.
Cronograma	La duración depende de los Sprint (más u menos 2 semanas)	Este depende de las necesidades del proyecto hasta que esté terminado

Metodología de lanzamiento	Se entrega al final de cada sprint.	Su entrega es continua
Métricas primarias	Su valor al momento de ser creado y su velocidad.	Plazo de entrega, tiempo del ciclo, cantidad de trabajo de proceso,
Modificaciones	No permite cambios o modificación durante la ejecución de los sprint.	Los cambios se pueden hacer en cualquier momento de ejecución.

Tabla 2 Tabla comparativa entre Scrum y Kanban

Elaborado por Cinthia Yáñez

En relación a la comparativa entre Scrum y Kanban, la primera metodología es más prescriptiva que la segunda, debido a que evita determinar roles y equipos y que no tiene una estructura formal de reuniones. Kanban tampoco prescribe iteraciones, aunque es posible incorporar si se desea. Las técnicas de visualización de procesos de Kanban lo convierten en ideal para equipos ubicados en el lugar de trabajo en una acumulación de elementos los cuales están sujetos a cambios frecuentes.

Estudio comparativo entre la metodología Xp y Kanban.

	Xp	Kanban
--	-----------	---------------

Principio del diseño	Reducción del código y adaptación de cambios inesperados por medio de la refactorización	Reduce la cantidad de trabajo en curso y asegura la reducción de residuos
Naturaleza de la interacción con el cliente	Interacción con el cliente en el lugar.	No necesariamente presencial
Complejidad del diseño	Diseño sencillo.	Diseño visual sencillo
Coordinador del proyecto	Entrenador XP	Trabajo en equipo
Colaboración en equipo	Equipos autoorganizados	El equipo se integra de recursos especializados.
Entrega del producto	Entrega constante.	Entrega constante.
Enfoque de prueba	Desarrollo por medio de pruebas, incluidas las pruebas de aceptación.	Pruebas realizadas después de la implementación de cada producto de trabajo.

Tabla 3 Tabla comparativa entre Xp y Kanban

Elaborado por Cinthia Yáñez

En relación a la comparativa entre XP y Kanban, la primera divide proyecto en varias etapas y en cada una de estas realiza un ciclo entero de planificación, diseño, desarrollo y prueba. Mientras que la segunda está basada en el trabajo en equipo y en el flujo de tarea de forma permanente, orientándose al ahorro de tiempo durante la planificación. Donde podemos ver que la metodología XP hace énfasis a la relación entre el cliente y el equipo de desarrollo, mientras que la metodología Kanban se limita al número máximo de tarea que puede realizarse, impidiendo que se abra una nueva tarea mientras no se termine una anterior.

Parámetros	Desarrollo basado en Scrum	Programación extrema (XP)	Metodología Kanban
Principio de diseño	Diseño complejo	Simplificación del código y acomodación de cambios inesperados a través de la refactorización	Limita la cantidad de trabajo en curso y garantiza la reducción de residuos
Naturaleza de la interacción con el cliente	No obligatoriamente presencial	Interacción con el cliente en el sitio	No obligatoriamente presencial
Complejidad del diseño	Diseño complejo	Diseño simple	Diseño visual sencillo
Coordinador del proyecto	Maestro Scrum	Entrenador XP	Trabajo en equipo
Funciones asignadas	3 roles predefinidos : propietario del producto, Scrum Master y equipo de desarrollo	Sin roles prescritos	Sin roles prescritos
Propiedad del proceso	Maestro Scrum	Propiedad del equipo	Propiedad del equipo
Propiedad del producto	El propietario del producto es responsable del producto.	Responsabilidad del grupo del producto	Responsabilidad del grupo del producto
Colaboración en equipo	Equipos multifuncionales	Equipos autoorganizados	El equipo se compone de recursos especializados.
Enfoque de flujo de trabajo	Iteraciones (sprints)	Sin iteraciones. Desarrollo del flujo de tareas	iteraciones cortas
Gestión de requerimientos	Requisitos gestionados en forma de artefactos a través de Sprint Backlog y Product Backlog	Gestionado en forma de Story Cards	Gestionado mediante tableros Kanban
Entrega del producto	Entrega según sprints en caja de tiempo	Entrega continua	Entrega continua
Estándares de codificación	Sin estándares de codificación	Se utilizan estándares de codificación.	Sin estándares de codificación
Enfoque de prueba	No se utiliza un enfoque formal para las pruebas.	Desarrollo basado en pruebas, incluidas las pruebas de aceptación	Pruebas realizadas después de la implementación de cada producto de trabajo
Acomodación de Cambios	Cambios no permitidos en sprints	Susceptible de cambiar incluso en etapas posteriores de desarrollo.	Cambios permitidos en cualquier momento

En relación a la comparativa entre las tres metodologías podemos decir que Scrum se aplica en proyecto donde los resultados a corto plazo son necesario con el plan de poder trabajar de manera colaborativa fomentando buen equipo de trabajo. Kanban por otro lado dice que dispone de un tablero dependiendo de las tareas que se les asignen a los miembros de los equipos, pero individual. Xp brinda adaptación a la hora de realizar software a la hora de brindar un producto de software se puede escoger cualquiera de las tres opciones dependiendo de las necesidades de los usuarios y empresas.

Después del estudio comparativo entre las tres metodologías ágiles y luego de haber analizado cada una de las ventajas y desventaja que cada una de estas poseen, y cuyo principal objetivo es agregar un gran valor posible durante el desarrollo de un proyecto de software tanto para los desarrolladores como para el cliente podemos notar las siguientes diferencias:

- Scrum y Kanban son herramientas que nos permite trabajar de forma mas eficaz estableciendo que hacer en cada una de las etapas.
- Scrum, Kanban y XP son adaptable.
- Scrum y Kanban está basado en el desarrollo incremental.
- Kanban siempre busca la calidad del producto o servicio que corresponda con los objetivos plasmados.
- XP predomina la retroalimentación entre el cliente y el desarrollador generando proyectos dinámicos y cambiantes.
- En Scrum predomina la división de trabajo en interacciones y revisión continua de resultado parciales.

- La metodología Scrum es mucho más rígida y define mucho más las partes de trabajo quitándole libertad a los miembros del equipo.
- La metodología Kanban identifica los flujos de trabajos además de que los mide y los optimiza, entregándole a los miembros del equipo mucha más libertad ya que define menos partes del trabajo.
- La metodología XP se centra en la adaptabilidad, definiendo al menos el rol de los desarrolladores y el cliente, además de utilizar buenas prácticas.

Conclusiones

Luego de realizar el presente caso de estudio, se pudo concluir que la calidad de software depende de muchos factores, entre ellos, la complejidad, personas que intervienen o tienen la necesidad de adquirir un proyecto.

Al momento de utilizar metodologías ágiles en el desarrollo de software estas nos ayudan a tener una respuesta rápida y segura en la forma de trabajar para obtener una buena calidad de software.

Se puede utilizar herramientas que facilita el potencial para el desarrollo de la empresa nos permita ahorrar tiempo también dinero para facilitar la interacción con los usuarios ayudando a mejorar la relación con ellos.

Permite a los equipos de desarrollo enfocarse en las tareas a la hora de construir software fomentado buena prácticas para que los proyecto tenga un entorno ordenado y los proyectos finalicen de manera exitosa.

Anexos

	Metodologías ágiles	Enfoque tradicional (Project Management)
Centrado en	Personas	Procesos
Forma de mantener al cliente al tanto	Entregas parciales funcionales	Documentación
Estilo de desarrollo	Iterativo	Lineal
Planeación	Relajada, abierta al cambio en todo momento	Estricta y hecha por adelantado
Priorización basada en	Aportar valor al negocio	Cumplir con el plan establecido
Forma de trabajo del equipo	Auto organizada	Controlada y supervisada por la dirección de proyectos
Estilo de gestión	Descentralizada	Centralizada
Adaptabilidad al cambio	Alta, constantemente se va actualizando la planeación	Baja, se debe pasar por un sistema formal de control de cambios
Tipo de liderazgo	Colaborativo, busca ser un facilitador para el equipo	Autoritario, dicta y dirige la forma de trabajo que deberá cumplir el equipo
Involucración del cliente	Alta durante todo el proyecto	Varía de acuerdo con el ciclo de vida del proyecto
Tipo de entrega	Parciales durante todo el proyecto	Única al finalizar el proyecto

Ilustración 5 Tabla Comparativa entre metodología ágiles y tradicionales

Fuente:

https://books.google.com.ec/books?id=u8pFDwAAQBAJ&pg=PA116&dq=tablas+comparativas+de+metodologias+agiles&hl=es-419&sa=X&ved=2ahUKEwjpu_GHl-D2AhUiSzABHZzDCQgQ6AF6BAgLEAI#v=onepage&q=tablas%20comparativas%20de%20metodologias%20agiles&f=false

+

Metodología	XP	SCRUM	Kanban	Agile Inception
Característica	Lleva de 5 a 10 iteraciones Se define el alcance del proyecto mediante la redacción de "historias de usuario" del cliente. Se tienen reuniones diarias de seguimiento para comunicación y compartir problemas y soluciones. El cliente ayuda al desarrollo del proyecto. Programación en parejas.	Los proyectos se ejecutan en bloques temporales (iteraciones-sprints) de un mes natural (pueden ser de 2 o 3 semanas si así se necesita)	es un sistema de información que controla de modo armónico la fabricación de los productos necesarios en la cantidad y tiempo necesarios en cada uno de los procesos que tienen lugar tanto en el interior de la fábrica, como entre distintas empresas.	es un conjunto de dinámicas orientadas a enfocar a todas las personas involucradas en un proyecto hacia un mismo objetivo, reduciendo muchas de las incertidumbres, ayudando a explicitar los riesgos más evidentes y poniendo en común las expectativas de todos
Procesos	La programación del software se hace siempre en pareja, lo que se llama programar a dos manos. Se asegura con este método que al menos un programador conoce y controla la labor de otro y queda revisado. La ventaja es que se produce mejor código que en base a un programador, aunque la dificultad de la misma sea mayor.	En Scrum un proyecto se ejecuta en ciclos temporales cortos y de duración fija (iteraciones que normalmente son de 2 semanas, aunque en algunos equipos son de 3 y hasta 4 semanas, límite máximo de feedback de producto real y reflexión). El proceso parte de la lista de objetivos/requisitos priorizada del	Existe una serie de principios básicos con el fin de obtener el máximo rendimiento de su flujo de trabajo. Visualice lo que hace (su flujo de trabajo) Limite la cantidad de Trabajo en Proceso (límites del TEP) Realice un seguimiento de su tiempo Lectura fácil de indicadores visuales Identifique los cuellos de botella y elimine lo que	Establece el Objetivo del producto que hemos de crear. Para qué? Cuál es el beneficio esperado? La herramienta que buscas se llama Elevator Pitch:

		producto, que actúa como plan del proyecto. En esta lista el cliente (Product Owner) prioriza los objetivos balanceando el valor que le aportan respecto a su coste (que el equipo estima considerando la Definición de Hecho) y quedan repartidos en iteraciones y entregas.	resulta descartable	
Herramientas	<p>Historias de Usuario: descripciones cortas de lo que el sistema debe realizar.</p> <ul style="list-style-type: none"> Plan de entregas: Cronograma de entrega y orden de las historias de usuario. Plan de iteraciones: se prueban y desarrollan las historias de usuario de acuerdo al orden establecido del plan. Kanban: Técnicas visuales para ver la 	<p>Historias de Usuario: descripciones cortas de lo que el sistema debe realizar.</p> <ul style="list-style-type: none"> Plan de entregas: Cronograma de entrega y orden de las historias de usuario. Plan de iteraciones: se prueban y desarrollan las historias de usuario de acuerdo al orden establecido del plan. Kanban: Técnicas visuales para ver la situación de cada tarea 	Hay docenas de herramientas Kanban excelentes por ahí. En esta reseña de las herramientas Kanban, analizamos detalladamente los siguientes programas: monday.com, Kanbanize, MeisterTask, Clarizen, ProjectManager.com, ZOHO Projects, Smartsheet, Trello	<p>Mucho de los ejemplos se han construido con una herramienta online Realtime Board (https://realtimeboard.com) que nos permite construir los distintos tipos escenarios simulando el uso de post-its.</p> <p>Why are we here? Elevator Pitch Product Box NOT List</p> <p>Descarga</p>

	situación de cada tarea y se representa en pizarra llenas de post-it.	y se representa en pizarra llenas de post-it.	ZenHub Breeze	
Roles	Los equipos de un proyecto de esta tipología y magnitud tienen normalmente las siguientes figuras y roles: Clientes, Programadores, Testers, Manager,	<p>Cliente: responsable de conducir el proyecto.</p> <p>Programador: Estiman el tiempo y transfieren las historias de usuario a código.</p> <p>Tester: encargado de las pruebas.</p> <p>Encargado de seguimiento (Tracker): Hace el seguimiento de acuerdo a la planificación, interés en la velocidad.</p> <p>Entrenador (Coach): Guiar y orientar al equipo. Que comprenda las directrices xp.</p> <p>Gestor (big boss): Gerente del proyecto.</p>	Service request Manager Service Delivery Manager Reuniones en Kanban Replenishment Meeting Daily Kanban Service delivery meeting Delivery Planning meeting Strategy Review Operations Review Risk Review	En este tipo de metodología se necesita desarrollar distintos tipos de roles como lo son: Gerente del proyecto (project manager) Patrocinador del proyecto (project sponsor) Arquitecto de solución (solution architect) Actores clave (key stakeholders) Propietario de la aplicación (Application owner) Arquitecto empresarial (Enterprise architect)

Ilustración 6 Cuadro Comparativo de metodología ágiles

Fuente: <https://www.studocu.com/co/document/universidad-cooperativa-de-colombia/ingenieria-de-requisitos/cuadro-comparativo-metodologias-agiles/10231316>




Similitud del Urkund



Document Information

Analyzed document	Trabajo Final 1.pdf (D131685525)
Submitted	2022-03-27T17:20:00.0000000
Submitted by	
Submitter email	cyanez@fafi.utb.edu.ec
Similarity	5%
Analysis address	mzuniga.utb@analysis.orkund.com

Sources included in the report

W	URL: https://openwebinars.net/blog/conoce-las-3-metodologias-agiles-mas-usadas/ Fetched: 2019-11-27T11:05:55.9930000	 2
W	URL: https://www.researchgate.net/publication/317840767_Analisis_comparativo_de_las_metodologias_agiles_en_el_desarrollo_de_software_aplicadas_en_Colombia Fetched: 2019-10-28T17:16:41.2770000	 1
W	URL: https://enjoylife.com.ar/novedades/sistemas/metodologias-desarrollo-agile-kanban-vs/ Fetched: 2022-03-27T17:20:37.4630000	 4
