



UNL

Universidad
Nacional
de Loja

Universidad Nacional de Loja

**Facultad de la Energía, las Industrias y los de Recursos Naturales
No Renovables**

Maestría en Ingeniería en Software

**Comparación de metodologías ágiles (XP y SCRUM) para la
estandarización del proceso de desarrollo de software en la
empresa CloudStudio Compañía Limitada**

**Trabajo de Titulación previo a la
obtención del título de Magíster en
Ingeniería en Software**

AUTORA:

Janneth Patricia Guamán Siguenza

DIRECTOR:

Ing. Edison Leonardo Coronel Romero, Mg. Sc

Loja - Ecuador

2023

Certificación

Loja, 24 de abril de 2023

Ing. Edison Leonardo Coronel Romero, Mg.Sc.

DIRECTOR DEL TRABAJO DE TITULACIÓN

CERTIFICO:

Que he revisado y orientado todo proceso de la elaboración del Trabajo de Titulación denominado: **Comparación de metodologías ágiles (XP y SCRUM) para la estandarización del proceso de desarrollo de software en la empresa “CloudStudio Compañía Limitada”**, previo a la obtención del título de **Magíster en Ingeniería en Software**, de autoría de la estudiante **Janneth Patricia Guamán Siguenza**, con cédula de identidad Nro. **0604056598**, una vez que el trabajo cumple con todos los requisitos exigidos por la Universidad Nacional de Loja para el efecto, autorizo la presentación para la respectiva sustentación y defensa.

Ing. Edison Leonardo Coronel Romero, Mg.Sc.

DIRECTOR DEL TRABAJO DE TITULACIÓN

Autoría

Yo, **Janneth Patricia Guamán Siguenza**, declaro ser autora del Trabajo de Titulación y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos y acciones legales, por el contenido del mismo. Adicionalmente acepto y autorizo a la Universidad Nacional de Loja la publicación del Trabajo de Titulación en el Repositorio Digital Institucional – Biblioteca Virtual.

Firma:

Cédula de Identidad: 0604056598

Fecha: 03/05/2023

Correo electrónico: janneth.guaman@unl.edu.ec

Teléfono: 0984027353

Carta de autorización por parte del autor, para consulta, reproducción parcial o total y/o publicación electrónica de texto completo, del Trabajo de Titulación

Yo, **Janneth Patricia Guamán Siguenza**, declaro ser autora del Trabajo de Titulación denominado: **Comparación de metodologías ágiles (XP y SCRUM) para la estandarización del proceso de desarrollo de software en la empresa “CloudStudio Compañía Limitada”** como requisito para optar el título de **Magíster en Ingeniería en Software**, autorizo al sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos muestre la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Institucional.

Los usuarios pueden consultar el contenido de este trabajo en el Repositorio Institucional, en las redes de información del país y del exterior con las cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia del Trabajo de Titulación que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja, a los tres días del mes de mayo de dos mil veintitrés.

Firma:

Autor: Janneth Patricia Guamán Siguenza

Cédula de identidad: 0604056598

Dirección: Coop. Santa Anita Norte, Calle H y Calle 3 - Riobamba

Correo electrónico: janneth.guaman@unl.edu.ec

Teléfono: 0984027353

DATOS COMPLEMENTARIOS:

Director del Trabajo de Titulación: Ing. Edison Leonardo Coronel Romero Mg. Sc.

Dedicatoria

Este Trabajo de Titulación está dedicado:

A Dios por su infinita bondad y amor, por ser guía y fortaleza en mi vida.

A mi papi Pedro que a pesar de la distancia que nos separa, siempre estás ahí apoyándome y enseñándome a afrontar las dificultades de la vida sin perder nunca la cabeza ni morir en el intento, sé que este momento es tan especial para ti como lo es para mí.

A mi mami Rosa por su amor, cariño y apoyo incondicional que me ha permitido ser una persona de bien.

A mis hermanas Rosita, Alexandra, Carolina, Lisset y Gissela quienes son mi inspiración y motivación para ser mejor cada día.

A mi compañero de vida, Oscar Vizuite por su amor y apoyo incondicional, por creer en mi capacidad, aunque hemos pasado momentos difíciles siempre has estado brindándome tu comprensión, cariño y amor.

A ti mi ángel que, aunque te sentí por poco tiempo sé que Dios tuvo sus consentidos y te eligió a ti.

A mis abuelitos allá en el cielo, que Dios los tiene en su gloria, especialmente a ti mamita Carmen, cuyo amor por tus nietas no conocía límites y quien inculco el valor del arduo trabajo, tus enseñanzas, virtudes y valores los mantendré conmigo el resto de mi vida.

A toda mi familia, por ser apoyo incondicional a lo largo de mi vida.

A todos los que me apoyaron para escribir y concluir esta tesis con éxito, quienes nos abrieron las puertas y compartieron sus conocimientos.

Con amor,

Janneth Patricia Guamán Siguenza

Agradecimiento

Agradecer a Dios por brindarme la salud, la vida y por permitirme hacer realidad este anhelado sueño.

A mis padres, Pedro y Rosita, mis hermanas Rosa, Alexandra, Carolina Lisset y Gissela, y mi compañero de vida Oscar Vizuite quienes me han brindado su apoyo incondicional y de corazón para que concrete con éxitos mis estudios.

Expreso mi agradecimiento a la Universidad Nacional de Loja, la cual me abrió sus puertas para culminar con éxito una etapa más de mi vida, preparándome para un futuro competitivo y poder servir a la sociedad con sólidos conocimientos para el progreso del país.

Un reconocimiento especial a mi director, Ing. Edison Coronel Romero por su calidad humana y todo el apoyo brindado al instruirme y guiarme para realizar el presente trabajo investigativo.

A los docentes de la maestría académica porque todos han aportado con sus conocimientos y consejos para mi formación profesional.

A mi empresa CloudStudio Cía. Ltda., y mi equipo de trabajo incondicional Marcos Guerrero, Jorge Duchi y Marcelo Espinosa, compañeros que sin duda me han facilitado los caminos para seguir, sin pedir nada a cambio y sin dudar de mi capacidad.

Son muchas las personas que han formado parte de mi vida a las que me encantaría agradecerles su amistad, consejos, apoyo, ánimo y compañía en los momentos más difíciles. Algunas están aquí con nosotros y otras en mis recuerdos y en mi corazón, sin importar en donde estén quiero darles las gracias por formar parte de mí, por todo lo que me han brindado y por todas sus bendiciones.

Para ustedes mi eterno agradecimiento y que Dios les bendiga

Janneth Patricia Guamán Siguenza

Índice de contenidos

Portada	i
Certificación.....	ii
Autoría	iii
Carta de autorización	iv
Dedicatoria	v
Agradecimiento	vi
Índice de contenidos	vii
Índice de tablas	x
Índice de Figuras	xi
Índice de anexos	xiii
1. Título	1
2. Resumen	2
2.1. Abstract	3
3. Introducción	4
4. Marco Teórico	6
4.1. METODOLOGÍA ÁGILE	6
4.1.1. Manifiesto Agile	6
4.2. SCRUM.....	8
4.2.1. Características	8
4.2.2. Roles	8
4.2.3. Ceremonias	9
4.2.4. Artefactos	10
4.2.5. Ventajas y Desventajas	12
4.3. EXTREME PROGRAMMING (XP)	12
4.3.1. Características	13
4.3.2. Roles	13
4.3.3. Herramientas	14
4.3.4. Fases.....	15
4.3.5. Prácticas XP	17
4.3.6. Ventajas y Desventajas	18

4.4. Buenas prácticas de programación.....	19
4.5. CloudStudio Compañía Limitada	22
4.5.1. Misión	22
4.5.2. Visión.....	22
4.5.3. Principales Productos.....	23
4.5.4. SIPOC Macro CloudStudio Cía. Ltda.....	23
5. Metodología	26
6. Resultados.....	27
Proceso de desarrollo de software en CloudStudio Cía. Ltda.....	29
• FASE 1: Levantamiento de requerimientos	29
• FASE 2: Diseño, desarrollo y pruebas	30
• FASE 3: Control de Calidad.....	33
• FASE 4: Liberación de versiones.....	33
PROPUESTA - MARCO DE TRABAJO PARA EL DESARROLLO DE SOFTWARE .	33
Personas	33
Roles	34
Cultura Organizacional	35
Equipo de Trabajo y Localización	36
Reuniones/Iteraciones	36
Artefactos	39
APLICACIÓN - MARCO DE TRABAJO PARA EL DESARROLLO DE SOFTWARE	40
Levantamiento de requerimientos	41
Miembros y Roles de los miembros.....	44
Pila del Producto (Product BackLog)	45
Lista de tareas de la iteración (Spring BackLog).....	46
Sprint / Iteraciones	47
Pruebas de aceptación	57
7. Discusión	63
8. Conclusiones	66
9. Recomendaciones	67
10. Bibliografía	68

11. Anexos	70
Formulario de encuesta - Procesos de Desarrollo de software en CloudStudio Cía. Ltda. .	70
Formulario de encuesta – Aplicación del Marco de Trabajo Propuesto	71
Certificado de Resumen (Abstract) - Ingles.....	72

Índice de tablas:

TABLA I. VENTAJAS Y DESVENTAJAS SCRUM	12
TABLA II. VENTAJAS Y DESVENTAJAS XP (EXTREME PROGRAMMING).....	18
TABLA III. PRÁCTICAS DE PROGRAMACIÓN A NIVEL GENERAL.....	19
TABLA IV. CUADRO COMPARATIVO SCRUM VS XP (EXTREME PROGRAMMING)	28
TABLA V. COLORES DEL TABLERO ACORDE A LA DIFICULTAD DEL REQUERIMIENTO.....	39
TABLA VI. REQUERIMIENTOS FUNCIONALES	42
TABLA VII. MIEMBROS Y ROLES DEL MARCO DE TRABAJO PROPUESTO	44
TABLA VIII. PILA DEL PRODUCTO (PRODUCT BACKLOG)	45
TABLA IX. LISTADO DE TAREAS DE ITERACIÓN (SPRINT BACKLOG).....	46
TABLA X. RESUMEN - REUNIÓN DIARIA SPRINT 1.....	47
TABLA XI. CASO DE USO - INGRESO DE SOLICITUD DE SEGUIMIENTO.....	49
TABLA XII. CASO DE USO: REVISIÓN DE SOLICITUD DE SEGUIMIENTO E INGRESO DE PLAN DE ACCIÓN.....	51
TABLA XIII. CASO DE USO: APROBACIÓN DEL PLAN DE ACCIÓN	53
TABLA XIV. CASO DE USO: SEGUIMIENTO DEL PLAN DE ACCIÓN	54
TABLA XV. CASO DE USO: CONSULTA DE SEGUIMIENTO	55
TABLA XVI. CASO DE PRUEBA.....	57

Índice de figuras:

Fig. 1. Flujo del proceso SCRUM	10
Fig. 2. Tablero Físico del backlog del Sprint.....	11
Fig. 3. Proceso de la programación extrema.....	15
Fig. 4. Principales productos CloudStudio Cía. Ltda.	23
Fig. 5. SIPOC Macro CloudStudio Cía. Ltda.	25
Fig. 6. Formato Revisión Requerimiento del Producto y/o Servicio.....	29
Fig. 7. Encuesta 1 – Gráfica porcentual acerca de la Pregunta 1.....	30
Fig. 8. Encuesta 1 – Gráfica porcentual acerca de la Pregunta 2.....	30
Fig. 9. Encuesta 1 – Gráfica porcentual acerca de la Pregunta 3.....	31
Fig. 10. Encuesta 1 – Gráfica porcentual acerca de la Pregunta 4.....	31
Fig. 11. Encuesta 1 – Gráfica porcentual acerca de la Pregunta 5.....	31
Fig. 12. Encuesta 1 – Gráfica porcentual acerca de la Pregunta 6.....	32
Fig. 13. Encuesta 1 – Gráfica porcentual acerca de la Pregunta 7.....	32
Fig. 14. Composición del equipo	34
Fig. 15. Sprint Iterativo.....	36
Fig. 16. Planificación	37
Fig. 17. Reunión Diaria del sprint	37
Fig. 18. Revisión del sprint.....	38
Fig. 19. Retrospectiva del sprint	39
Fig. 20. Resumen de Iteración del Sprint	40
Fig. 21. Levantamiento de Requerimientos acerca de la matriz de auditoría.....	41
Fig. 22. Caso de uso - Requerimientos Funcionales.....	43
Fig. 23. Tablero de actividades utilizando Jira	47
Fig. 24. Listado de solicitudes ingresadas por jefe de auditoría	50
Fig. 25. Pantalla de ingreso de información de solicitud de hallazgo de auditoría	51
Fig. 26. Pantalla Listado de solicitud de seguimiento del auditado.....	52
Fig. 27. Pantalla Revisión e ingreso de Plan de Acción	52
Fig. 28. Pantalla Listado de Solicitudes con Plan de acción para ser autorizados	53
Fig. 29. Pantalla de Autorización de Seguimiento de Auditoría	54

Fig. 30. Pantalla Ingreso de avance del Plan de acción por el auditado	55
Fig. 31. Pantalla Revisión del avance del plan de acción por jefe de auditoría.....	55
Fig. 32. Pantalla de consulta de solicitud de seguimiento con el plan de acción	56
Fig. 33. Reporte generado por Plan de Acción	57
Fig. 34. Acta de Certificación de Requerimientos.....	58
Fig. 35. Acta Entra-Recepción de Requerimiento/Proyecto.....	59
Fig. 36. Encuesta 2 – Gráfica porcentual acerca de la Pregunta 1.....	59
Fig. 37. Encuesta 2 – Gráfica porcentual acerca de la Pregunta 2.....	60
Fig. 38. Encuesta 2 – Gráfica porcentual acerca de la Pregunta 3.....	60
Fig. 39. Encuesta 2 – Gráfica porcentual acerca de la Pregunta 4.....	61
Fig. 40. Encuesta 2 – Gráfica porcentual acerca de la Pregunta 5.....	61
Fig. 41. Encuesta 2 – Gráfica porcentual acerca de la Pregunta 6.....	62
Fig. 42. Encuesta 2 – Gráfica porcentual acerca de la Pregunta 7.....	62

Índice de anexos:

Anexo 1. Formulario de Encuesta - Procesos de desarrollo de software.....70
Anexo 2. Formulario de Encuesta – Aplicación del marco de trabajo propuesto71
Anexo 3. Certificado de Resumen (Abstract) - Inglés.....72

1. Título

Comparación de metodologías ágiles (XP y SCRUM) para la estandarización del proceso de desarrollo de software en la empresa “CloudStudio Compañía Limitada”.

2. Resumen

El presente Trabajo de Titulación (TT) tiene como objetivo comparar las metodologías ágiles (XP y Scrum) para estandarizar el proceso de desarrollo de software dentro de la empresa CloudStudio Compañía Limitada.

Este TT se centra en que las metodologías ágiles como XP (eXtreme Programming) y Scrum que han fortalecido marcos de trabajo para varias organizaciones, priorizando la calidad del software en entregables y planificación adaptiva a los cambios que puedan surgir durante la etapa del desarrollo de software.

Para la realización del TT se aplicaron metodologías de investigación como: investigación documental para la elaboración de la fundamentación científica a través de la obtención de información fidedigna de revistas científicas y libros que contribuyan y sustente el tema de investigación. Además de encuestas para conocer el proceso de desarrollo de software dentro de la empresa CloudStudio Cía. Ltda.

El marco de trabajo propuesto en esta investigación es una combinación de las metodologías Scrum y XP, ya que, de acuerdo con los planteamientos de las características de cada metodología, se logra identificar que cada una de ellas posee diferentes orientaciones y propósitos; por una parte, Scrum maneja los aspectos administrativos de un proyecto de software, mientras que XP va enfocado a manejar los aspectos técnicos. De esta manera, definir correctamente los roles, reuniones y artefactos entregables ayudan a mantener organizados los proyectos o tareas a realizar dentro del equipo de trabajo de forma efectiva, obteniendo una planificación y control general de las tareas a fin de que los involucrados puedan visualizar el estado del proyecto.

La aplicación del marco de trabajo propuesto en este TT da como resultado una entera satisfacción a los miembros del equipo de desarrollo de la empresa CloudStudio Cía. Ltda., ya que en su implementación compartieron experiencias de trabajo en equipo.

Finalmente, cabe destacar que las metodologías ágiles Scrum o XP, puede llegar a traer grandes beneficios para las empresas, pero siempre dependerá del contexto en el que se apliquen y si sus principios son bien aplicados.

Palabras Claves: Metodologías Ágiles, Scrum, Programación Extrema, XP

2.1. Abstract

This Degree Project aims to compare agile methodologies (XP and Scrum) to standardize the software development process within CloudStudio Compañía Limitada.

This research focuses on agile methodologies such as XP (eXtreme Programming) and Scrum, which have strengthened frameworks for various organizations, prioritizing software quality in deliverables and adaptive planning for changes that may arise during the software development stage.

Besides, the research applied methodologies, such as documentary research, to develop the scientific foundation by obtaining reliable information from scientific journals and books that contribute to and support the research topic. In addition to surveys to learn about the software development process within CloudStudio Cía. Ltd.

The framework proposed in this research is a combination of the Scrum and XP methodologies since, according to the approaches of the characteristics of each methodology, it is possible to identify that each of them has different orientations and purposes; On the one hand, Scrum handles the administrative aspects of a software project, while XP is focused on handling the technical aspects. In this way, correctly defining the roles, meetings, and deliverable artifacts helps keep the projects or tasks to be carried out within the work team organized effectively, obtaining general planning and control of the tasks so that those involved can visualize the project status.

Applying the proposed framework in this research results in complete satisfaction for the development team members of the company CloudStudio Cía. Ltda., since in its implementation, they shared teamwork experiences.

Finally, it is necessary to emphasize; agile Scrum or XP methodologies can benefit companies significantly. However, it will always depend on the context in which they are applied and if their principles are well applied.

Keywords: Agile Methodologies, Scrum, Extreme Programming, XP

3. Introducción

Las metodologías ágiles se consideran indispensables, flexibles y adaptables en la ejecución de cualquier proyecto de software, ya que se basan en definir un marco de trabajo para su análisis, diseño, desarrollo e implementación.

El presente Trabajo de Titulación (TT) se enfoca en el estudio comparativo de las metodologías ágiles (XP eXtreme Programming y Scrum) dentro de la etapa del desarrollo de software para brindar características y ventajas en el ciclo de vida de un producto de software que optimice recursos basados en desarrollos rápidos y entregables de calidad que superen las expectativas de los clientes [1].

Es por ello, que se pretende realizar una comparación de metodologías ágiles (XP y SCRUM) para la estandarización del proceso de desarrollo de software en la empresa “CloudStudio Compañía Limitada”, ya que la constante evolución de las tecnologías de información han cambiado la forma de trabajar y planificar proyectos en la industria del software, lo que ha replanteado las características que sustenta el desarrollo de software tradicional con extensa documentación y desarrolladores que invierten mayor tiempo ocasionando pérdidas de recursos.

Los motivos para realizar este TT se centran en que las metodologías ágiles como XP (eXtreme Programming) y Scrum han fortalecido marcos de trabajo para varias organizaciones, priorizando la calidad del software en entregables y planificación adaptiva a los cambios que puedan surgir durante el desarrollo [2].

Según Schwaber [3], dirigir un negocio a partir del 2020, se ha convertido en un reto y en una diversidad tecnológica mayor para las empresas, por lo que esta investigación trata de aprovechar los conceptos y beneficios de cada una de ellas durante la etapa de desarrollo de software de un producto o proyecto de software, dividiendo grandes procesos en subproceso más pequeños, es decir, diseñar e implementar productos separados y combinarlos para abordar diferentes aplicaciones de forma organizada.

El presente TT inicia con un estudio comparativo entre las metodologías ágiles (XP y SCRUM) que establecen fundamentos teóricos relevantes en el proceso de desarrollo software, como características principales, roles, reuniones, artefactos, ventajas y desventajas de cada metodología, en la cual se establecerá y se identificará a la metodología que se ajuste a los requerimientos de la empresa CloudStudio Cía. Ltda., mediante una matriz relacional donde se estandarice los indicadores del proceso de desarrollo de software.

Finalmente se propondrá un marco de trabajo ágil basado en una de ellas, o en una combinación de estas metodologías para la etapa de desarrollo de software que contenga procesos estandarizados y buenas prácticas de programación para incrementar soluciones optimas de trabajo dentro de la empresa.

En base al marco de trabajo propuesto para la etapa de desarrollo de software, se aplicará en el proceso de desarrollo de la matriz de seguimiento de hallazgos de auditoría, requerimiento levantado para el cliente Cooperativa AT, incorporando artefactos, roles, iteraciones y documentos óptimos para obtener un entregable de calidad hacia el cliente.

4. Marco Teórico

En los años 90 surgieron movimientos y prácticas con el fin de mejorar los procesos de desarrollo de software, los mismos que desde su inicio han tenido que enfrentar problemas que surgen con el auge de la tecnología, desde cambios sociales hasta necesidades cambiantes que abordan proyectos más complejos y es necesario coordinar el trabajo conjunto de equipos en distintas disciplinas.

Para ese entonces era común escuchar un modelo lineal basado en tareas secuenciales para el desarrollo de software enfocado desde su análisis hasta la entrega de este [4]. En la actualidad, este enfoque no resulta adecuado en proyectos donde el entorno del sistema es cambiante y exige reducir los tiempos de desarrollo y una alta calidad en mantenimiento. En este contexto, las metodologías ágiles son una respuesta inmediata para llenar esa exigencia metodológica.

4.1. METODOLOGÍA ÁGILE

Son aquellas que permiten adaptar la forma de un trabajo a las condiciones de un proyecto, consiguiendo flexibilidad e inmediatez en la respuesta para amoldar el proyecto y su desarrollo a las circunstancias específicas del entorno. Dicho esto, las metodologías de desarrollo de software buscan proporcionar en poco tiempo pequeñas piezas de software en funcionamiento para aumentar la satisfacción del cliente. Estas metodologías utilizan enfoques flexibles y el trabajo en equipo para ofrecer mejores resultados [5].

4.1.1. Manifiesto Agile

A principios del 2001, con la cordillera Wasatch como telón de fondo, en Snowbird (Utah) [6], 17 personas se reunieron para debatir el futuro del desarrollo de software. Los miembros del grupo compartían la frustración sobre el estado actual de los acontecimientos, aunque no estaban de acuerdo en la forma de remediar la situación.

El problema, coincidieron, era que las empresas estaban tan centradas en planificar y documentar hasta el último detalle de los ciclos de desarrollo de software, que habían perdido de vista lo realmente importante: complacer a los clientes. Por ello, a través del manifiesto ágil se han descubierto mejores formas de desarrollar software tanto por experiencia propia como ayudando a terceros.

4.1.1.1. Valores

El manifiesto ágil presenta 4 valores que sobresalen de una metodología tradicional:

- Individuos e interacciones sobre procesos y herramientas
- Software funcionando sobre documentación extensiva
- Colaboración con el cliente sobre negociación contractual
- Respuesta ante el cambio sobre seguir un plan

4.1.1.2. Principios

El manifiesto ágil hace referencia a 12 principios detallados a continuación [7]:

1. Los proyectos se construyen alrededor de personas motivadas
2. Conversaciones directas es la mejor manera de comunicación
3. El producto funcionando es la mejor medida de progreso
4. Habilidad de mantener un ritmo de desarrollo sostenible
5. Atención continua a la excelencia técnica y al buen diseño
6. Simplicidad
7. Satisfacer al cliente entregándole valor rápidamente
8. Entregar frecuentemente un producto funcional
9. Los cambios a los requerimientos son bienvenidos
10. Cooperación diaria y cercana entre el equipo y el negocio
11. Equipos autoorganizados y autogestionados
12. Adaptación constante a circunstancias cambiante

El desarrollo ágil se centra en las iteraciones, flexibilidad, comunicación y adaptabilidad, en las iteraciones los equipos dividen el trabajo y lo realizan en periodos cortos, este será desarrollado por equipos multidisciplinarios en los que ellos deciden cómo van a llevarlas a cabo. Fomenta la comunicación entre los involucrados del proyecto, los miembros del equipo y el representante del cliente, esto puede prevenir errores que no se han detectado y lleven a retrasar la entrega del proyecto, así como la mejora de la toma de decisiones en beneficio del avance del proyecto [8].

Entre más metodologías más importantes se puede mencionar Kanban, Scrum, Lean Software Development(LSD), Feature Driven Development(FDD), Adaptive Software Development(ASD), Crystal, Extreme Programming(XP), entre otras.

4.2. SCRUM

Scrum es un marco de gestión de proyectos de metodología ágil que ayuda a los equipos a estructurar y gestionar el trabajo mediante un conjunto de valores, principios y prácticas [9].

Esta metodología se centra en las actividades de gerencia realizando principalmente una planificación adaptativa y el desarrollo incremental del software con entregas de productos funcionales en periodos de tiempo cortos, además representa una alternativa importante frente a escenarios de requerimientos cambiantes.

Scrum se enfoca en elevar y animar al máximo la productividad y eficacia de un equipo de trabajo para que puedan optar por autoorganizarse e interactuar entre ellos, aprendiendo a través de experiencias y reflexionando sobre cómo mejorar continuamente.

4.2.1. Características

Según Jeff Sutherland [10] sostiene que existe 3 pilares fundamentales en la metodología ágil Scrum:

Transparencia: Todos los implicados tienen conocimiento de qué ocurre en el proyecto y cómo ocurre, es decir, existe un entendimiento común y una visión general.

Inspección: Los miembros del equipo Scrum frecuentemente inspeccionan el progreso para detectar posibles problemas para saber que el trabajo fluye y que el equipo funciona de manera autoorganizada.

Adaptación: Cuando hay algo que cambiar, el equipo se ajusta para conseguir el objetivo del sprint. Esta es la clave para conseguir el éxito en proyectos complejos, donde los requisitos son cambiantes o poco definidos y en donde la adaptación, la innovación, la complejidad y flexibilidad son fundamentales.

4.2.2. Roles

El equipo tiene como foco entregar valor y ofrecer resultados de calidad que permitan cumplir los objetivos de negocio del cliente. Para ello, el equipo de trabajo Scrum puede estar conformado de 6 a 9 personas multifuncionales donde se definen los siguientes roles:

- **Product Owner/Dueño del producto:** es el responsable de maximizar el valor del producto resultante.
- **Scrum Master:** es el facilitador del equipo Scrum responsable de asegurar los procesos y ayudar a todos los miembros a entender y seguir las teorías, reglas y prácticas Scrum.
- **Equipo de desarrollo:** es el grupo de personas de realizar el trabajo y entregar el producto.

4.2.3. Ceremonias

Son reuniones que se realizan de forma iterativa en cada Sprint:

Planificación del sprint (Sprint Planning): el equipo scrum define las actividades que se van a abordar y el objetivo principal del sprint. Esta reunión puede durar 8 horas para sprint de un mes.

Reunión diaria (Daily Scrum): son conversaciones con una duración de máximo 15 minutos diarios donde se abordará temas como ¿qué hice ayer?, ¿qué debo hacer hoy?, además permite inspeccionar el trabajo y adaptarse en el caso de que exista algún cambio dentro del sprint.

Revisión (Sprint Review): el equipo presenta al dueño del producto las funcionalidades desarrolladas, el cliente valida los cambios realizados y además brinda un feedback sobre posibles mejoras, aprobarlas por completo o rechazarlas si no se ha cumplido con el objetivo.

Retrospectiva (Sprint Retrospective): el objetivo de esta reunión es mirar hacia atrás y realizar un análisis de lo que se ha hecho y sus resultados correspondientes, y decidir que medidas concretas emplear a fin de mejorar los resultados.

En la Fig. 1, se puede observar el proceso completo de la metodología Scrum.

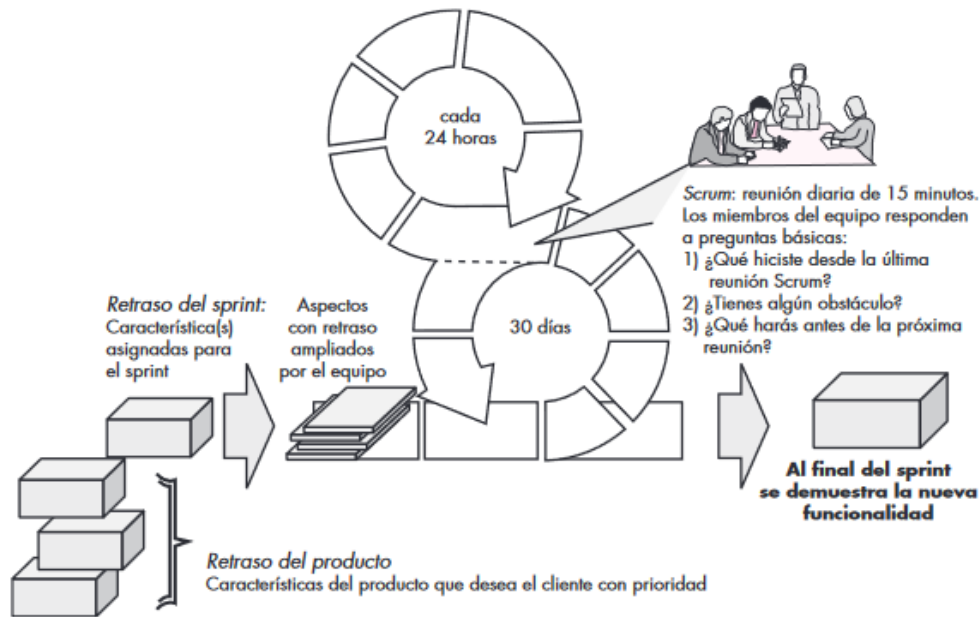


Fig. 1 Flujo del proceso SCRUM
Fuente: [11, p. 70]

4.2.4. Artefactos

La guía Scrum [3] propone 3 artefactos para mantener organizados los proyectos y ayudan a planificar y revisar cada uno de los sprints, aportando medios ineludibles para efectuar cada ceremonia. Entre los cuales se menciona:

Backlog de Producto (Product Backlog): listado dinámico de tareas que engloba todo un proyecto y es públicamente visible para todos los involucrados del proyecto. Esta lista representa “qué se pretende” pero sin mencionar “cómo hacerlo”. El propietario del producto es el responsable del backlog del producto, de su contenido, disponibilidad y priorización.

El backlog del producto es el listado de requerimientos funciones esperados del software, donde se deberán especificar de cada uno:

- El grado de prioridad
- Esfuerzo que demanda
- Granulidad
- Criterios de Aceptación

Cabe indicar, que cada requerimiento tiene un orden de prioridad que dependen de la funcionalidad, pérdidas o costo que demande posponer la implementación, riesgos y el valor diferencial respecto a otros productos de la competencia.

Backlog de Sprint (Sprint Backlog): es una lista reducida de ítems del backlog del producto que son negociados entre el dueño del producto y el scrum team durante la planificación del sprint.

Esta lista se genera al momento de iniciar cada sprint, ya que son las tareas o actividades que el equipo se compromete a desarrollar durante la iteración actual. Estas tareas no demandan una duración superior a un día de trabajo del miembro del equipo al que ha sido asignada y son actualizadas diariamente por el equipo y muestran:

- Las tareas pendientes, en curso y terminadas.
- La estimación del esfuerzo pendiente de cada tarea sin concluir.
- El nombre del miembro del equipo que se ha asignado dicha tarea.

El backlog del sprint se puede visualizar en la Fig. 2, mediante un tablero físico visto por todos los miembros del equipo de la siguiente manera:

PENDIENTES	EN CURSO	TERMINADAS
		
		
		

Fig. 2 Tablero Físico del backlog del Sprint

Fuente: <https://www.atlassian.com/es/agile/kanban/boards>

Incremento: Es el resultado de sumar todos los elementos del backlog del producto finalizado durante el sprint actual, es decir, el producto final utilizable de un sprint que se asemeja a un software funcional que puede ser implementado en un ambiente de producción de manera 100% efectiva.

4.2.5. Ventajas y Desventajas

A continuación, en la TABLA I , se muestra el resumen de las ventajas y desventajas sobre la metodología Scrum:

TABLA I
VENTAJAS Y DESVENTAJAS SCRUM

VENTAJAS	DESVENTAJAS
<ul style="list-style-type: none">- Establece metas cuantificables para el equipo garantizando la productividad.- La flexibilidad de Scrum permite realizar cambios en los hechos favoreciendo la resolución de obstáculos sobre la marcha.- Se observa el progreso del sprint o iteración de requerimientos- Entregan resultados rápidos en periodos muy cortos de tiempo.- El avance del proyecto es cercano y transparente hacia las partes interesadas, lo que aporta confianza en los clientes.- Prioriza tareas y distingue pendientes que pueden esperar.	<ul style="list-style-type: none">- Se debe conocer los roles Scrum para que no exista dispersión y choques en las funciones.- El éxito del proyecto dependerá de la preparación y experiencia del Scrum Master.- Las reuniones diarias y no tener una fecha de culminación del proyecto pueden frustrar a los miembros del equipo.- Si alguna tarea esta sin terminar, las tareas restantes empezaran a posponerse.- Puede ser difícil el manejo de Scrum en proyectos complejos.

4.3. EXTREME PROGRAMMING (XP)

Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico [12].

La ventaja básica de XP es que todo el proceso es visible y responsable. Los desarrolladores harán compromisos concretos sobre lo que lograrán, mostrarán un progreso concreto en forma de software implementable y, cuando se alcance un hito, describirán exactamente lo que hicieron y cómo y por qué diferían del plan. Esto permite a las personas orientadas a los negocios hacer sus propios compromisos comerciales con confianza, aprovechar las oportunidades a medida que surgen y eliminar los callejones sin salida de forma rápida y económica [13].

4.3.1. Características

La programación extrema se basa en cinco valores:

- **Comunicación:** Eficaz entre los ingenieros de software y los interesados, pone énfasis en la colaboración estrecha pero informal entre cliente y desarrolladores.
- **Simplicidad:** XP restringe a los desarrolladores para que diseñen sólo las necesidades inmediatas, en lugar de considerar las de futuro.
- **Retroalimentación:** El cliente debe familiarizarse con el software, por lo que deben ser presentados al finalizar cada módulo.
- **Respeto:** Miembros valiosos para el equipo, los desarrolladores respetan la experiencia de los clientes y viceversa.
- **Coraje:** Verdad sobre el progreso y las estimaciones, se adaptan a los cambios cuando ocurren.

4.3.2. Roles

Dentro de la metodología XP se puede distinguir los siguientes roles [14]:

Programador/Developer: es un desarrollador normal, que escribe el código y realiza la totalidad de las tareas del proyecto.

Cliente/Customer: es la persona responsable de escribir historias de usuarios, establecer prioridades y formular la cartera de productos

Encargado de pruebas/Tester: es el miembro del equipo responsable de la prueba del producto. La calidad del producto final depende en gran medida de su trabajo.

Rastreador/Tracker: es la persona cuya tarea principal es monitorear el progreso del desarrollo del software y detectar todos los problemas en él.

Entrenador/Coach: Se encarga de agendar las reuniones, se asegura de que el proceso de juntas sea seguido, registra los resultados de las reuniones para futuros reportes para el Tracker. Asiste a las reuniones y trae información importante, mantiene al equipo feliz y productivo.

Consultor: es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto ayuda al equipo a resolver un problema específico y puede que no siempre haya un consultor como parte del equipo.

Jefe del Proyecto: se encarga de agendar las reuniones, se asegura de que el proceso de juntas sea seguido, registra los resultados de las reuniones para futuros reportes para el Tracker. Asiste a las reuniones y trae información importante, mantiene al equipo feliz y productivo.

4.3.3. Herramientas

Historias de Usuario: representan una breve descripción acerca del comportamiento del sistema, se generan por cada característica principal y son utilizadas para cumplir estimaciones de tiempo y el plan de lanzamientos, así mismo reemplazan un gran documento de requisitos y presiden la creación de las pruebas de aceptación [15].

Cada historia de usuario debe ser comprensible y delimitada para que los programadores puedan implementarlas. El tiempo de desarrollo ideal para una historia de usuario es entre 1 y 3 semanas.

Release Planning: es una planificación donde los desarrolladores y clientes establecen los tiempos de implementación ideales de las historias de usuario, la prioridad con la que serán implementadas y las historias que serán implementadas en cada versión del programa, de la misma manera se considera como un plan de publicaciones donde se indican las historias de usuario para cada versión del programa y las fechas en las que se publicarán.

Tareas de Ingeniería (Task Card): son actividades que se vinculan al programador, ya que permite tener un acercamiento con el código.

Pruebas de Aceptación: Las Pruebas de aceptación son de vital importancia para el éxito de una iteración y el comienzo de la siguiente, con lo cual el cliente puede conocer el avance en

el desarrollo del sistema y a los programadores lo que les resta por hacer. Además, permite una retroalimentación para el desarrollo de las próximas historias de usuarios a ser entregadas. Estas son comúnmente llamadas pruebas del cliente, por lo que son realizadas por el encargado de verificar si las historias de usuarios de cada iteración cumplen con la funcionalidad esperada.

Tarjetas CRC(Clase-Responsabilidad-Colaboradores): Las Tarjetas CRC (Clase-Responsabilidades-Colaboradores), permiten conocer que clases componen el sistema y cuales interactúan entre sí. Se dividen en tres secciones: Nombre de la Clase, Responsabilidades y Colaboradores.

4.3.4. Fases

Pressman, en el libro de Ingeniería de Software [11], hace referencia a cuatro actividades estructurales como la planeación, diseño, codificación y pruebas, como se muestra en la Fig. 3:

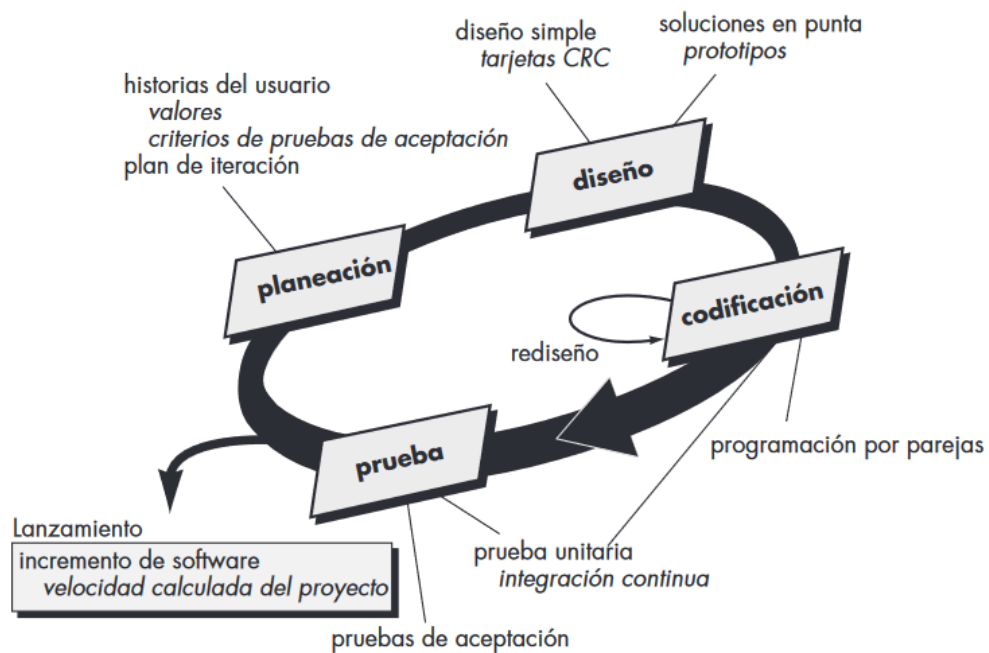


Fig. 3 Proceso de la programación extrema
Fuente: [11, p. 62]

Planeación: Según la identificación de las historias de usuario, se priorizan y se descomponen en mini-versiones. La planificación se va a ir revisando de poco. Cada dos o tres semanas aproximadamente de iteración, se debe obtener un software útil, funcional, listo para probar y lanzar.

En esta fase se debe tener en cuenta:

- **Iteraciones:** Todo proyecto que siga la metodología X.P. se divide en iteraciones de aproximadamente 3 semanas de duración. Al inicio de cada iteración los clientes deben seleccionar las historias de usuario definidas en el "Release planning" que serán implementadas. También se seleccionan las historias de usuario que no pasaron el test de aceptación que se realizó al terminar la iteración anterior. Las historias de usuario se dividen en tareas de entre 1 y 3 días de duración que se asignarán a los programadores.
- **Velocidad del proyecto:** representa la rapidez con la que se desarrolla el proyecto; para estimarla se debe contar el número de historias de usuario que se pueden implementar en una iteración. Usando la velocidad del proyecto se controla que todas las tareas se puedan desarrollar en el tiempo del que dispone la iteración. Es conveniente reevaluar esta medida cada 3 ó 4 iteraciones y si se aprecia que no es adecuada hay que negociar con el cliente un nuevo "Release Planning".
- **Programación en pareja:** el trabajo en pareja involucra a dos programadores trabajando en el mismo equipo; mientras uno codifica haciendo hincapié en la calidad de la función o método que está implementando, el otro analiza si ese método o función es adecuado y está bien diseñado. De esta forma se consigue un código y diseño con gran calidad.
- **Reuniones diarias.** los desarrolladores se reúnan diariamente y exponen sus problemas, soluciones e ideas de forma conjunta. Las reuniones tienen que ser fluidas y todos los interesados deben que tener voz y voto.

Diseño: en esta fase se intentará trabajar con un código sencillo, haciendo lo mínimo imprescindible para que funcione. Se obtendrá el prototipo y además, para el diseño del software orientado a objetos, se crearán:

- **Tarjetas CRC (Clase-Responsabilidad-Colaboración):** permiten al programador centrarse y apreciar el desarrollo orientado a objetos olvidándose de los malos hábitos de la programación procedural clásica.
- **Diseños simples:** implementar diseños simples y sencillos. Realizar lo menos complicado posible para conseguir un diseño fácilmente entendible e implementable que a la larga costará menos tiempo y esfuerzo desarrollar.

- **Glosarios de términos:** el uso de glosarios de términos y una correcta especificación de los nombres de métodos y clases ayudará a comprender el diseño y facilitará sus posteriores ampliaciones y la reutilización del código.
- **Riesgos:** Se sugiere utilizar una pareja de desarrolladores para que investiguen y reduzcan al máximo el riesgo que supone algún problema.
- **Funcionalidad extra:** No añadir funcionalidades extras al programa de acuerdo a lo especificado en las historias de usuario. Mas bien, tomarlas como una nueva historia de usuario e implementarla al finalizar las pendientes.
- **Refactorizar:** es mejorar y modificar la estructura y codificación de códigos ya creados sin alterar su funcionalidad para procurar optimizar su funcionamiento.

Codificación: en esta etapa, la programación se realiza “a dos manos”, en parejas en frente del mismo computador e inclusive a veces se intercambian las parejas. De esta forma, se asegura que se realice un código más universal, con el que cualquier programador podría trabajar y entender y así se conseguirá una programación organizada y planificada. La codificación debe hacerse ateniendo a estándares de codificación ya creados.

Pruebas: se realizan pruebas automáticas continuamente y al tratarse normalmente de proyectos a corto plazo, estas pruebas deben ser automatizadas y constante. Además, el propio cliente puede hacer pruebas, proponer nuevas pruebas e ir validando las mini-versiones generadas.

Lanzamiento: al llegar a esta fase significa que todas las historias de usuario o mini-versiones se han probado con éxito, ajustándose a los requerimientos del cliente. Se obtiene un software útil para incorporarlo al producto.

4.3.5. Prácticas XP

Para administrar de forma ágil y efectiva un proyecto de software debe adaptarse a cambios constantes y al ritmo de trabajo dinámico. Para ello se recomienda seguir las siguientes prácticas:

- Comunicación continua entre el equipo de desarrollo y el cliente para implementar sugeridos
- Integración continua para e entregar versiones operativas
- Diseñar lo más simple, pero con la funcionalidad requerida

- Realización de pruebas unitarias por parte de los programadores y pruebas de aceptación por parte del cliente
- Remover código duplicado para facilitar los posteriores cambios
- Programación en parejas para un mejor diseño y mayor satisfacción de los programadores
- El cliente debe estar presente y disponible para el equipo de desarrollo
- Normas y Estándares de programación definidos por los desarrolladores para tener un código legible

4.3.6. Ventajas y Desventajas

Dentro de las ventajas y desventajas de la metodología XP, se resumen en la TABLA II:

TABLA II
VENTAJAS Y DESVENTAJAS XP (EXTREME PROGRAMMING)

VENTAJAS	DESVENTAJAS
<ul style="list-style-type: none"> - Programación organizada y código de comprensión sencilla en todo momento - Menor tasa de errores gracias a la programación en pareja - Satisfacción del programador - El producto resultante es estable y productivo debido a pruebas continuas - Relación estrecha con el cliente - Gestión propia del tiempo 	<ul style="list-style-type: none"> - Se recomienda emplearlo solo en proyectos a corto plazo y simples - Dificultad para documentar ya que se realiza de manera rápida - Altos costos en caso de algún error o cambio en una iteración - Requiere control de versiones - Se requiere al cliente en todo el proceso

De acuerdo con los planteamientos de las características de cada metodología, se logra identificar que cada una de ellas posee diferentes orientaciones y propósitos; por una parte, Scrum maneja los aspectos administrativos de un proyecto de software, mientras que XP va enfocado a manejar los aspectos técnicos [16]. Por tanto, es posible combinar ambas metodologías de una manera efectiva, por una parte, es posible realizar las reuniones y prácticas propuestas por Scrum (el que) aplicando las prácticas y métodos definidos por XP (el cómo).

4.4. Buenas prácticas de programación

Conjunto de técnicas, principios, metodologías que se implementan en el software para que se vuelva fácil, rápido y seguro de desarrollar, mantener y desplegar el aplicativo.

Según Michael A. Jackson [17], *“La labor de un Ingeniero Informático o Ingeniero Software no es solamente crear aplicaciones software que funcionen correctamente, sino que además debe crear aplicaciones software robustas, eficientes, fáciles de mantener y un largo etcétera de propiedades extras. Muchas personas están capacitadas para codificar aplicaciones software, pero un número bastante más reducido de ellas están capacitadas para hacerlo conforme a los parámetros de calidad que se espera de un titulado superior en Informática”*.

Hoy en día es muy fácil aprender a programar por cuenta propia, dado la amplia cantidad de información disponible a través de internet y la multitud de cursos que se ofertan. Sin embargo, tanto el autoaprendizaje no debidamente supervisado como la mayoría de los cursos de programación que se imparten llevan a adquirir las habilidades necesarias para la creación de software artesanal, donde el programador codifica un algoritmo y prueba su correcto funcionamiento mediante un proceso de prueba y error tras largos periodos de depuración.

Por tal razón son de vital importancia su aplicación, y se mencionan los siguientes:

A nivel general [18]:

A continuación, en la TABLA III, se resume los indicadores principales de buenas prácticas de programación a nivel general:

TABLA III
PRÁCTICAS DE PROGRAMACIÓN A NIVEL GENERAL

PARADIGMA DE LA PROGRAMACIÓN ORIENTADA A OBJETOS	Conocer y entender el paradigma sobre el que se está trabajando, aplicarlos y entender el beneficio que aportan.
LEER CÓDIGO	Adquirir la capacidad de distinguir entre un código de calidad y uno no comprensible, para llegar a escribir un código legible para otros desarrolladores

SEGUIMIENTO ESTÁNDARES	DE	<ul style="list-style-type: none"> ✓ Nomenclatura de ficheros. ✓ Nomenclatura de funciones y módulos. ✓ Nomenclatura de variables. ✓ Comentarios. ✓ Lista de qué hacer y qué no hacer. ✓ Hacer que el código sea simple.
SIMPLICIDAD CÓDIGO	DE	<p>Escribir código de manera simple que sea sencillo de entender y mantenerlo para evitar reducir errores</p>
TESTING		<p>Analizar y corregir cada etapa antes de avanzar sobre nuevas partes del código, esto ayudará a evitar la acumulación de errores críticos a la hora de integrar el aplicativo de software</p>
REFACTORING		<p>Ayuda a los desarrolladores a comprender, mejorar, reescribir o incluso eliminar parte del código muerto, para llegar a un clean code, de un programa sin cambiar su comportamiento.</p> <ul style="list-style-type: none"> ✓ Clean Code: código limpio, simple y sencillo
METODOLOGÍAS ÁGILES		<p>A través de los lineamientos principales de cierta metodología se puede delegar a cada uno de los miembros del equipo las tareas necesarias para avanzar hasta alcanzar el objetivo final</p>
API REST		<p>Implementación de los servicios y métodos que el sistema de comunicación de datos puede ofrecer, por lo que integrarlo en el desarrollo del programa beneficiará en gran medida el funcionamiento de las aplicaciones externas</p>

SEGURIDAD	<ul style="list-style-type: none"> ✓ El acceso a los ficheros (tanto de forma física, como permisos). ✓ La posibilidad de modificar el código en la misma ejecución o la inyección sql. ✓ Desbordamiento de buffer al hacer uso de un array sin tamaño controlado. ✓ Formateo de los datos de entrada (en formularios). ✓ Actualización del IDE y las funciones. Una función obsoleta puede originar un agujero de seguridad en nuestro código. ✓ Uso de contraseñas en el código. Las contraseñas deben estar cifradas y en la base de datos.
DOCUMENTACIÓN	<p>Documentar todo lo necesario para facilitar el entendimiento del desarrollo y funcionamiento del software al personal del proyecto y futuras personas que trabajen en él</p>

A nivel de código [19]:

- ✓ Evitar el uso de variables globales
- ✓ Evitar el uso de variables globales
- ✓ Usar un único return por función, que se colocará como última sentencia de la función.
- ✓ Evitar escribir funciones y procedimientos demasiado largas.
- ✓ Evitar copiar y pegar trozos idénticos de código a lo largo de una aplicación software.
- ✓ Colocar cada clase o módulo en un fichero separado.
- ✓ Colocar la función main en una clase o módulo separado e independiente.
- ✓ Evitar el uso de elementos no habituales de un lenguaje.
- ✓ Usar siempre llaves ({}) en las estructuras de control.
- ✓ Colocar al lado de una llave que cierre un bloque de código un indicativo de que tipo de estructura cierra.
- ✓ Inicializar siempre las variables cuando se declaran.

- ✓ No declarar atributos de clases y registros como públicos, ni acceder a ellos directamente.
- ✓ No ser excesivamente estrictos con la ocultación de información.
- ✓ No usar caracteres propios del castellano en la codificación de programas.
- ✓ Controlar desbordamientos en vectores cuando se accede a ellos.
- ✓ Controlar que no se referencien punteros nulos.
- ✓ Comprobar consistencia semántica de los argumentos de una función.
- ✓ Acostumbrarse a depurar aplicaciones software sin usar depuradores.
- ✓ Expresar valores literales como constantes.
- ✓ La herencia nunca debe usarse para heredar atributos de una clase.

4.5. CloudStudio Compañía Limitada

Empresa ecuatoriana fundada en el año 2013, resultado de la integración de un grupo de especialistas que decidieron ofrecer sus servicios tecnológicos a Instituciones Financieras del mercado ecuatoriano. Durante los años de existencia de la empresa han aportado con productos, servicios y conocimientos a sus clientes principalmente a instituciones financieras como, Cooperativas de Ahorro y Crédito, Administradora de Fondos y Bancos.

CLOUDSTUDIO CIA. LTDA., es el socio estratégico encargado de proporcionar el recurso humano, tecnológico y metodológico ampliamente experimentado, que le permitirá a la institución cumplir sus objetivos de una manera organizada, sostenible y vanguardista.

4.5.1. Misión

Incrementar la competitividad, productividad, rentabilidad y eficiencia de nuestros clientes, ofreciendo productos y servicios innovadores, que se ajusten a los requerimientos de cada cliente y asumir como propias las necesidades de los clientes como premisa ineludible, en el diseño de soluciones eficaces.

4.5.2. Visión

Ser una empresa de reconocido prestigio nacional e internacional, en constante expansión y crecimiento; y convertirse en el socio estratégico de sus clientes, gracias a la calidad de productos y al cumplimiento de los compromisos asumidos.

4.5.3. Principales Productos

Los principales productos de la empresa se pueden observar en la Fig. 4, que resumen aplicaciones de forma conjunta o individual según requiera el cliente:



Fig. 4 Principales productos CloudStudio Cía. Ltda.

- Core financiero, SOFTBANK.
- Facturación electrónica, CS-I BILLINGWARE.
- Switch Transaccional, CS-I TRANSACTIONAL CORE
 - Para cajeros automáticos.
 - Para BCE, (Pago Bono de Desarrollo Humano).
 - Para integración con SERVIPAGOS (pago Ágil).
 - Seguridades OTP.
- Gestor de documentación, CS-I DOCUMENT.
- Web transaccional CSI- TRANSACTIONAL WEB Transferencias Interbancarias.
- Riesgo de Liquidez y Mercado CSI -RISKCONTROL.
- Capacitación Técnica (C#, .Net).

4.5.4. SIPOC Macro CloudStudio Cía. Ltda.

Como muestra la Fig. 5, SIPOC es una herramienta que representa procesos de una manera clara y concisa para la empresa, cada columna representa a los componentes de un proceso y su flujo de interacción en la ejecución de este. Los componentes se describen por sus siglas, las cuales son:

- ✓ Supplier / Proveedores
- ✓ Inputs / Entradas
- ✓ Process / Procesos
- ✓ Outputs / Salidas
- ✓ Customers / Clientes

El contexto de la investigación se centrará en el proceso de DESARROLLO Y PRUEBAS.

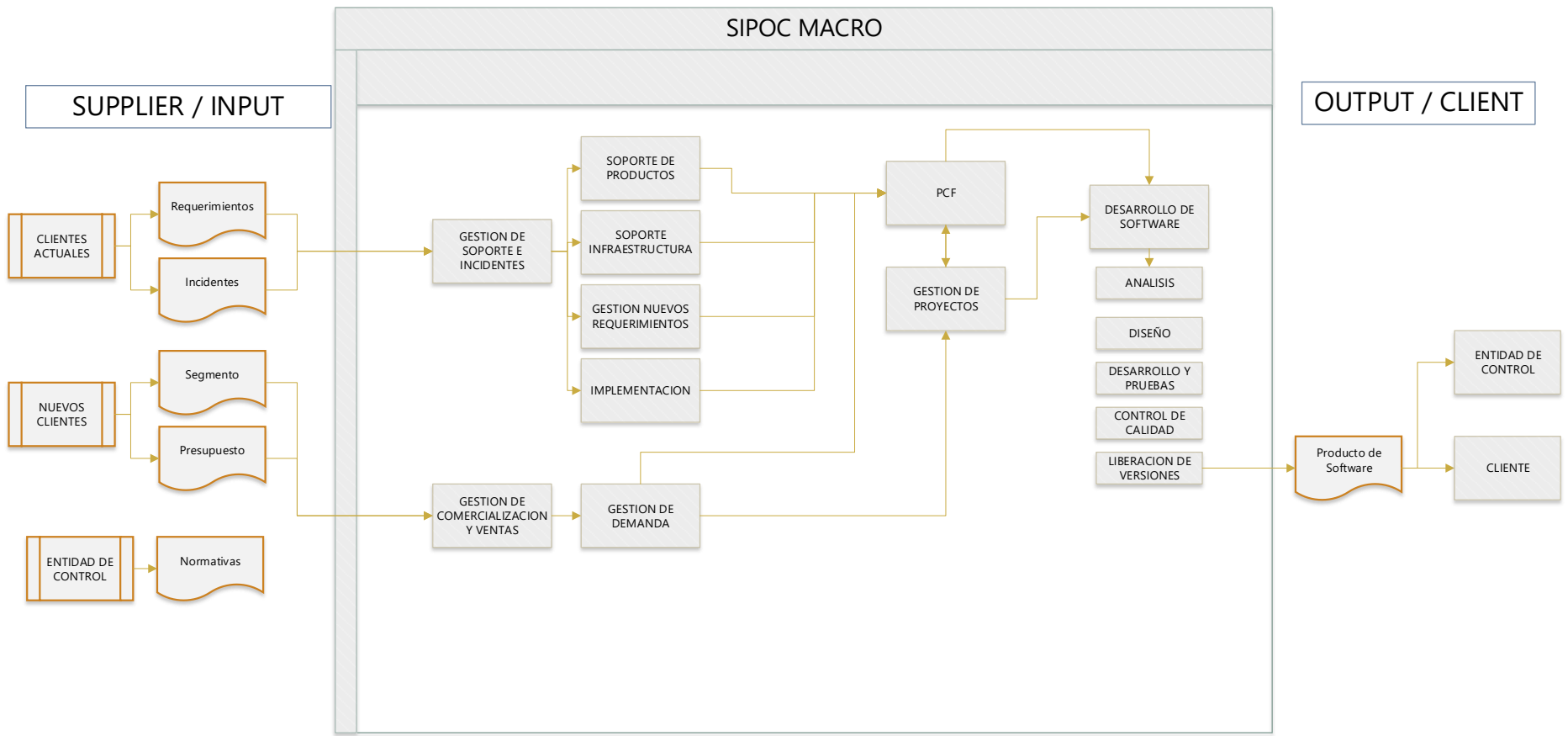


Fig. 5 SIPOC Macro CloudStudio Cía. Ltda.

5. Metodología

El TT tiene como objetivo desarrollar un marco de trabajo para estandarizar el proceso de desarrollo de software utilizando la combinación de las metodologías ágiles Scrum y XP, la misma que se basó en un enfoque cualitativo, evaluando las características principales de cada metodología y comparando los indicadores que aplican para proyectos en general.

Participantes:

Los participantes de la investigación eran el equipo de trabajo de la empresa CloudStudio, quienes proporcionaron información sobre el proceso de desarrollo de software que llevaban a cabo en la empresa.

Procedimiento:

Recopilación de información: Se basó en fuentes bibliográficas como revistas científicas y libros que sustentan la investigación en cuestión, utilizando la técnica documental.

Estudio de las metodologías ágiles Scrum y XP: Se llevó a cabo un estudio detallado de cada metodología ágil utilizando el enfoque deductivo.

Comparación de indicadores: Se compararon los indicadores que cada metodología aplica para proyectos en general.

Entrevista: Se realizó una entrevista con el equipo de trabajo de CloudStudio para determinar el proceso de desarrollo de software que llevaban a cabo y obtener información acerca del funcionamiento que debía ser plasmado en la matriz de seguimiento de hallazgos de auditoría con el marco de trabajo propuesto.

Desarrollo del marco de trabajo: Con base en la información recopilada y el análisis de las metodologías ágiles, se desarrolló el marco de trabajo para estandarizar el proceso de desarrollo de software utilizando la combinación de Scrum y XP.

Evaluación del marco de trabajo: Se evaluó el marco de trabajo propuesto a través de una encuesta al equipo de trabajo de CloudStudio Cía. Ltda., al finalizar su requerimiento para determinar su eficacia y eficiencia en el proceso de desarrollo de software.

6. Resultados

Las metodologías ágiles permiten adoptar una forma de trabajo de acuerdo con las condiciones del proyecto para que estos sean flexibles e inmediatos frente a una respuesta de entrega. Las empresas al utilizar una metodología ágil consiguen que sus proyectos sean gestionados de forma autónoma y eficaz, reduciendo así los costos e incrementando su productividad.

Las metodologías Scrum y XP (Extreme Programming) buscan el mismo objetivo que es crear una cultura de trabajo en equipo para que este alcance un alto nivel técnico y a la misma vez potencien su rendimiento cumpliendo así los objetivos del negocio.

Scrum catalogado como un marco iterativo e incremental que está enfocado a la gestión de procesos de desarrollo en equipos pequeños de software (6 – p personas multifuncionales) proporcionando entregas de software funcional en periodos cortos y fijos denominados sprint, que es la unidad básica de trabajo para un equipo Scrum. Esta metodología determina 3 roles principales como Dueño del producto, Scrum Master y Equipo de desarrollo a los que se asigna funciones y responsabilidad para que el trabajo sea colaborativo e iterativo para llegar al éxito del proyecto.

Extreme Programming (XP) hace hincapié a la satisfacción del cliente con entregas funcionales en un corto tiempo y permite a sus desarrolladores responder con confianza a los requisitos cambiantes de los clientes, incluso al final del ciclo de vida, es decir, todos los interesados del producto mantienen una comunicación constante. El equipo de trabajo (< 9 personas) está definido por clientes, programadores, testers, tracker, consultor, entrenador y jefe de proyecto que contribuyen con los conocimientos que tiene cada uno en función del objetivo principal que es realizar ciclos de entrega rápidos, continuos e incrementales mediante ciclos semanales, con reuniones periódicas entre el cliente y el equipo de desarrollo.

A través de la TABLA IV, se resumen las comparaciones entre Scrum y Extreme Programming (XP):

TABLA IV
CUADRO COMPARATIVO SCRUM VS XP (EXTREME PROGRAMMING)

INDICADOR	SCRUM	EXTREME PROGRAMMING (XP)
Roles	Roles definidos <ul style="list-style-type: none"> ✓ Product Owner ✓ Scrum Master ✓ Equipo de desarrollo 	Roles Principales <ul style="list-style-type: none"> ✓ Programador/Developer ✓ Cliente/Customer ✓ Encargado de pruebas/Tester ✓ Rastreador/Tracker ✓ Entrenador/Coach ✓ Consultor ✓ Jefe del Proyecto
Reuniones	Reuniones diarias, o por Sprint, retrospectiva	Planificación, lanzamientos pequeños, programación en pareja, refactorización, integración continua
Equipo de trabajo	Entre 6 y 9 personas multifuncionales	Grupos menores de 9 personas
Iteraciones	Sprint con una duración de 1 a 4 semanas	Iteraciones de aproximadamente 3 semanas de duración
Backlog	<ul style="list-style-type: none"> ✓ Product Backlog ✓ Backlog por sprint. 	<ul style="list-style-type: none"> ✓ Historias de Usuario ✓ Release Planning ✓ Tareas de Ingeniería (Task Card) ✓ Pruebas de Aceptación ✓ Tarjetas CRC (Clase-Responsabilidad-Colaboradores)
Pizarra	Scrum board en cada sprint.	Tarjetas CRC
Cambios	Limitados, espera al siguiente sprint	Implementan los cambios sugeridos


Como se puede observar en la Fig. 5, el equipo de trabajo en la empresa CloudStudio Cía. Ltda., se centra en el área de desarrollo de Software, donde abarca los nuevos requerimientos

e incidentes para soporte técnico. Tradicionalmente, manejan fases de una metodología en cascada por donde transita un proyecto para finalizar de manera exitosa, planeando y programando todas las actividades que conlleva un producto de software.

Proceso de desarrollo de software en CloudStudio Cía. Ltda.

- FASE 1: Levantamiento de requerimientos**

Proceso de analizar y documentar los requisitos de software que dan como resultado las especificaciones del software que debe cumplir. Para el levantamiento de requerimientos, la empresa genera reuniones directamente con el cliente y toma nota de las necesidades o mejoras que el usuario requiere dentro del producto. Se basa en una plantilla general donde se describe el requerimiento, tal como se muestra en la Fig. 6:

	REVISIÓN REQUERIMIENTO DEL PRODUCTO Y/O SERVICIO			
	Versión: 0	Fecha de Emisión: 16/06/2016	Pág: 1	Cod: REV01 Rev: 1

DATOS CLIENTE

Cliente:	
Contacto Cliente:	
Área / Módulo Contacto Cliente:	
Teléfono Cliente:	
Correo Cliente:	
Participantes Cliente:	
Fecha levantamiento:	

DATOS CLOUDSTUDIO

Analista Responsable:	
Participantes CloudStudio:	
Lugar del levantamiento	

Nombre responsable específico cliente:		
Teléfonos/correo responsable específico cliente:		
Dato general de requerimiento Marque con x	Detalle Requerimiento	Responsable
<input checked="" type="checkbox"/> Cambios de interfaz producto		
<input checked="" type="checkbox"/> Cambios de back producto		
Módulos adicionales		
Enlace con otros aplicativos		
Ajustes de seguridad		
Cambios en módulos producto		
Otro		
<input checked="" type="checkbox"/> Cambios de interfaz producto		
<input checked="" type="checkbox"/> Cambios de back producto		
Módulos adicionales		
Enlace con otros aplicativos		
Cambios en módulos producto		
Otro		

PARTICIPANTES REVISIÓN CLIENTE

Firma: _____

PARTICIPANTES LEVANTAMIENTO CLOUDSTUDIO

Firma: _____

Fig. 6 Formato Revisión Requerimiento del Producto y/o Servicio
Fuente: Empresa CloudStudio Cía. Ltda.

Levantado los requerimientos con el cliente, el jefe del proyecto analiza, prioriza y designa las actividades a los respectivos desarrolladores responsables de cada requerimiento, el presente

artefacto debe quedar plasmado con la firma correspondiente de los participantes, tanto de la empresa como del cliente.

- **FASE 2: Diseño, desarrollo y pruebas**

Para analizar la etapa de desarrollo de software se implementó una encuesta al equipo de trabajo de CloudStudio conformado por 15 desarrolladores para conocer de su experiencia el trabajo que llevan a cabo al desarrollar un requerimiento.

Por lo que se puede analizar de la siguiente manera:

Pregunta 1: ¿Ha tenido problemas en el desarrollo de algún requerimiento de software?

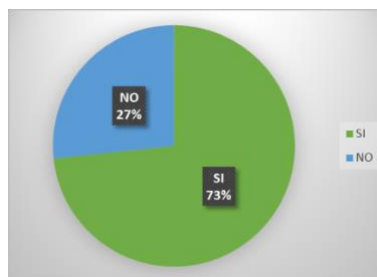


Fig. 7 Encuesta 1 – Gráfica porcentual acerca de la Pregunta 1

- En la Fig. 7, se puede determinar que el 73% de los desarrolladores han tenido problemas en el desarrollo de software dentro de la empresa, mientras que el 27% no ha presentado dificultad alguna

Pregunta 2: ¿Cree Ud., que al desarrollar un requerimiento cada desarrollador lo realiza por su propia cuenta?

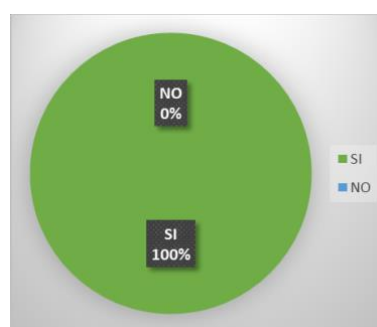


Fig. 8 Encuesta 1 – Gráfica porcentual acerca de la Pregunta 2

- El 100% de los desarrolladores realiza las actividades de desarrollo por su cuenta tal como se puede observar en la Fig. 8.

Pregunta 3: ¿Ha existido dificultades cuando un requerimiento que es prioritario para otro y no se complete en el tiempo estimado?

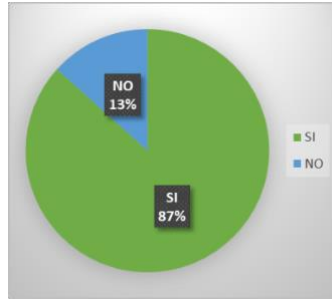


Fig. 9 Encuesta 1 – Gráfica porcentual acerca de la Pregunta 3

- En la Fig. 9, se determina que el 87% de los desarrolladores han tenido dificultades para desarrollar nuevas actividades por el retraso de tareas que no se han cumplido en el tiempo estimado, mientras que el 13% no han presentado estos inconvenientes.

Pregunta 4: Considera que se presentan retrasos en la entrega de requerimientos hacia el cliente porque no existen reuniones o presentaciones sobre los avances que se generan acerca del requerimiento

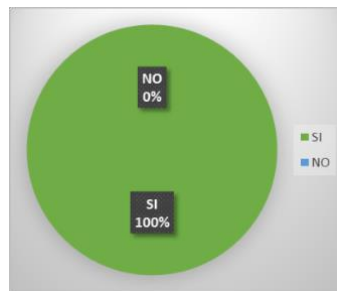


Fig. 10 Encuesta 1 – Gráfica porcentual acerca de la Pregunta 4

- En la Fig. 10, se puede visualizar que el 100% de los desarrolladores indica que tienen retrasos de entregas en requerimientos y hasta en proyectos enteros porque no existe una adecuada planificación ni control de actividades que cada miembro del equipo realiza.

Pregunta 5: Número de desarrolladores designados para el desarrollo de un proyecto

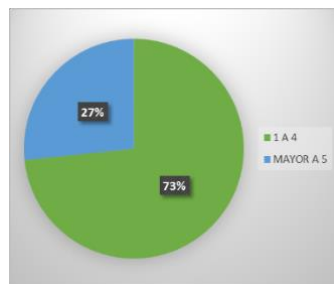


Fig. 11 Encuesta 1 – Gráfica porcentual acerca de la Pregunta 5

- La Fig. 11 muestra que el 73% de los desarrolladores indica que el número de desarrolladores para un proyecto se estima entre 1 a 4 siendo estos de menor impacto, mientras que se generan grupos mayor a 5 cuando el proyecto es extenso.

Pregunta 6: ¿Cree Ud., que es necesario establecer un marco de trabajo general para la empresa CloudStudio Cía. Ltda., en la etapa de desarrollo de software?

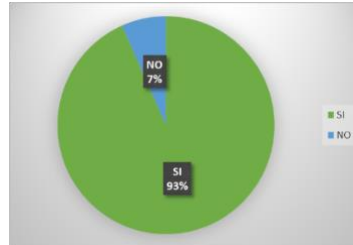


Fig. 12 Encuesta 1 – Gráfica porcentual acerca de la Pregunta 6

- Se observa en la Fig. 12, que el 93% de los desarrolladores indican que es necesario establecer un marco de trabajo para que todos concuerden en el desarrollo y conozcan de las actividades que se llevan a cabo dentro del mismo, mientras que 7% se encuentra conforme como realizan sus actividades.

Pregunta 7: ¿Cuál de las siguientes metodologías ágiles de desarrollo de software ha utilizado?

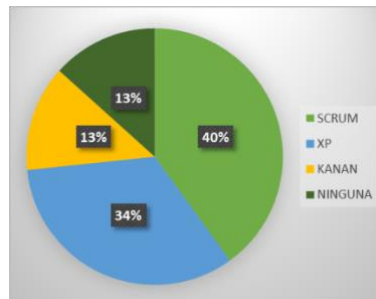


Fig. 13 Encuesta 1 – Gráfica porcentual acerca de la Pregunta 7

- En la Fig. 13, se aprecia que dentro del equipo de trabajo existe un 40% que conoce sobre la metodología Scrum, seguido del conocimiento de la metodología XP con un 34%, mientras que existe un conocimiento del 13% en metodología Kanban y de igual manera un 13% no ha utilizado ninguna metodología.

Con esto se puede establecer que en la fase de diseño y desarrollo de software cada desarrollador es responsable de su requerimiento, se centra básicamente en construir o modificar la funcionalidad del software. No existen tareas diarias ni reuniones de avances de sus actividades para que el jefe del proyecto tenga en conocimiento. Se manejan de manera

individual, no hay una comunicación con el cliente en el desarrollo del mismo hasta que exista una versión funcional. Si existen cambios dentro del desarrollo solicitados por el cliente se extienden los tiempos de desarrollo y muchas de las veces las fechas de entrega de proyectos se alarga por estos cambios y el costo del proyecto se eleva dejando perdidas a la empresa.

En esta etapa, el cliente no visualiza avances del producto hasta que se finalice el requerimiento.

Al finalizar el desarrollo del requerimiento, el responsable no entrega ninguna documentación donde se enfatice el desarrollo generado.

- **FASE 3: Control de Calidad**

Dentro de esta etapa, el usuario responsable del control de calidad genera pruebas conjuntamente con el cliente, donde se valida su funcionalidad y se verifica errores o mensajes de excepción en caso de algún error para corroborar el alcance del software y dar el visto bueno para su entrega al cliente. La empresa no tiene una herramienta para registrar la verificación y validación de requerimientos.

- **FASE 4: Liberación de versiones**

El jefe del área de desarrollo genera la publicación de los cambios generados por el desarrollador y envía hacia el cliente una versión funcional.

De acuerdo a lo analizado en la empresa CloudStudio Cía. Ltda., conjuntamente con las metodologías ágiles Scrum y eXtreme Programming(XP), se homologa las mismas para generar un marco de trabajo que estandarice el proceso de desarrollo de software en el que se relacione todo el equipo de trabajo de la empresa y el cliente para que los requerimientos solicitados sean los adecuados y esperados.

PROPUESTA - MARCO DE TRABAJO PARA EL DESARROLLO DE SOFTWARE

Personas

En las organizaciones las personas son fundamentales porque son aquellas que interactúan entre sí para determinar necesidades y alcances de los productos de software. Dentro de este grupo de interacciones se debe definir conceptos principales como roles y responsabilidades.

Roles

En este marco de trabajo el equipo se conformará con 2 roles más 1, es decir, líder del proyecto/Scrum Master y arquitecto desarrollador. Se dice “más 1” roles porque el rol Cliente participa constantemente durante todo el ciclo de vida del proyecto.

La composición del equipo se puede representar en la Fig. 14, de la siguiente manera:

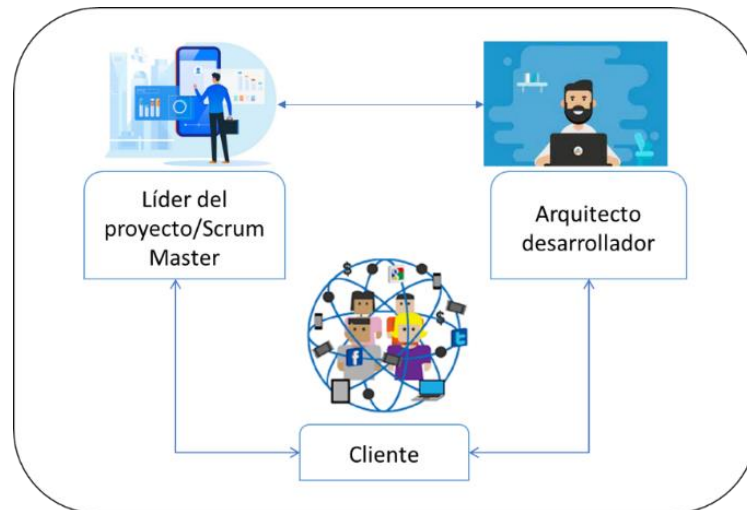


Fig. 14 Composición del equipo

Los roles que propone el marco de trabajo pueden subdividirse según las responsabilidades de cada rol para lograr una mejor escalabilidad.

Líder del proyecto/Scrum Master: cumple el rol de líder servicial que creará y mantendrá un ambiente laboral que permita cumplir con los objetivos planteados dentro del proyecto y optimizar la eficiencia grupal.

Dentro de las principales responsabilidades se puede mencionar:

- Mantener de comunicación y cooperación entre todos los roles, enfocado siempre en las metas
- Asistir y facilitar la resolución de problemas y consultas que tenga en el equipo o con el cliente y eliminar impedimentos
- Crear un ambiente participativo, empoderando al equipo, compartiendo el poder y la toma de decisiones
- Realizar y participar de reuniones que se presenten dentro del proyecto

Equipo de trabajo/Arquitecto desarrollador: cumple la tarea de analizar, diseñar, programar y evaluar la solución propuesta del proyecto. Para esto, el rol debe liderar la definición y evolución de la estructura técnica y del diseño de la solución. Entre otras responsabilidades tiene:

- Identificar tareas
- Realizar una estimación sobre el tiempo y esfuerzo (VD) que llevará completar una Tarea o Ítems.
- Mantener actualizado al equipo sobre el estado de sus actividades, en las reuniones diarias y en el tablero de control.
- Colaborar para definir y/o modificar la arquitectura.
- Estar dispuesto a aprender habilidades fuera de su campo de experticia.
- Ser proactivo para detectar conflictos o problemas e identificar soluciones a los mismos.
- Mantener un ambiente de comunicación constante dentro del equipo.

Ciente: responsable de transmitir al equipo de software los factores políticos del proyecto, y los objetivos del negocio. Es el representa de todas las partes interesadas de la organización.

Sus principales responsabilidades son:

- Colaborar con el equipo para planificar, revisar y dar detalle a los objetivos de cada iteración
- Actuar como interlocutor único ante el equipo, con autoridad para tomar decisiones
- Mantener una comunicación constante con el líder del proyecto
- Asistir a las reuniones planificadas
- Resolver dudas que puedan surgir con respecto a los requerimientos de la solución
- Participar activamente en la evaluación de aceptación del requerimiento

Cultura Organizacional

La clave para lograr un clima de aprendizaje, comunicación, y responsabilidad es la confianza y el respeto. Como consecuencia del respeto y la confianza los roles estarán dispuestos a compartir información y ayudarse mutuamente, a ver errores, al autoaprendizaje, a confiar en que cada una de las personas realizó un esfuerzo perfecto en sus responsabilidades y cumplió el objetivo con el que el equipo se comprometió. Este marco de trabajo propone eliminar los

roles jerárquicos, es decir, cada rol tiene un rango igual a otro, lo que le hace diferente son las tareas que cumple.

Equipo de Trabajo y Localización

El equipo de trabajo para esta propuesta de estandarización de software se provee entre 4 a 5 personas multifuncionales,

Además, se propone manejar de manera híbrida la localización de los integrantes del equipo:

- **Presencial:** el equipo se encuentra en las instalaciones de la organización donde realiza las tareas designadas, mejorando las relaciones interpersonales entre sus miembros, así como, reduciendo tiempos de respuestas ante problemas y mejorando la comunicación y el aprendizaje.
- **Remoto:** el equipo trabajará en sus hogares y de acuerdo a sus dudas habrá una comunicación directa con otros integrantes del equipo.

Es posible lograr buenos resultados en un equipo presencial o remoto, en estos casos las tecnologías de comunicación y gestión son clave para un trabajo exitoso

Reuniones/Iteraciones

Todo proyecto será dividido en sprint (evento o producto central) de acuerdo a un objetivo. Su duración es máximo 1 mes. En la Fig. 15, se puede apreciar un Sprint iterativo desde su planificación hasta la retrospectiva.

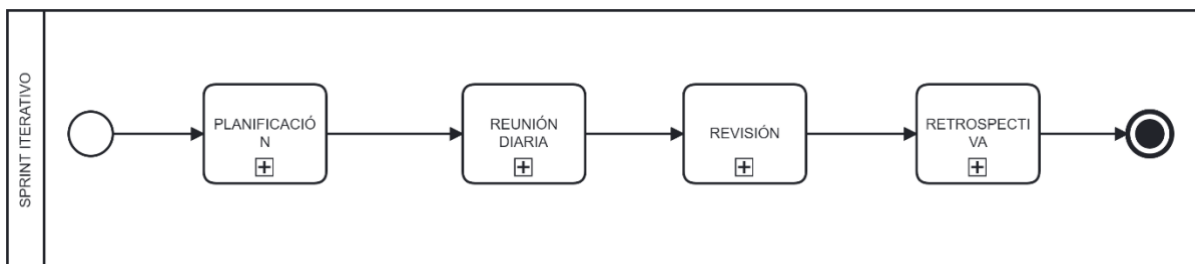


Fig. 15 Sprint Iterativo

Planificación: Planificar el trabajo a realizarse durante el sprint, ver Fig. 16, su duración no puede ser mayor a 8 horas y se establecerá las tareas para obtener el entregable al finalizar el sprint.

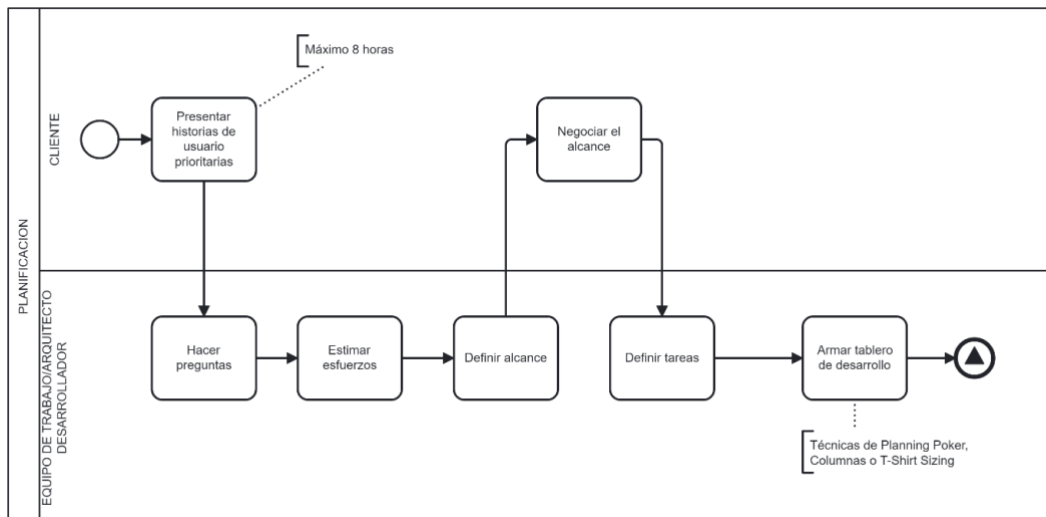


Fig. 16 Planificación

Reunión Diaria: Reunión diaria máximo de 20 minutos para conocer que se realizará en las próximas 24 horas, y se realizará como mecanismo de inspección del trabajo realizado y medición del progreso. Como se muestra en la Fig. 17, en la planificación se contemplará 3 preguntas a cada desarrollador:

- ¿Qué se trabajó ayer?
- ¿Qué realizará el día de hoy?
- ¿Se presento algún inconveniente?

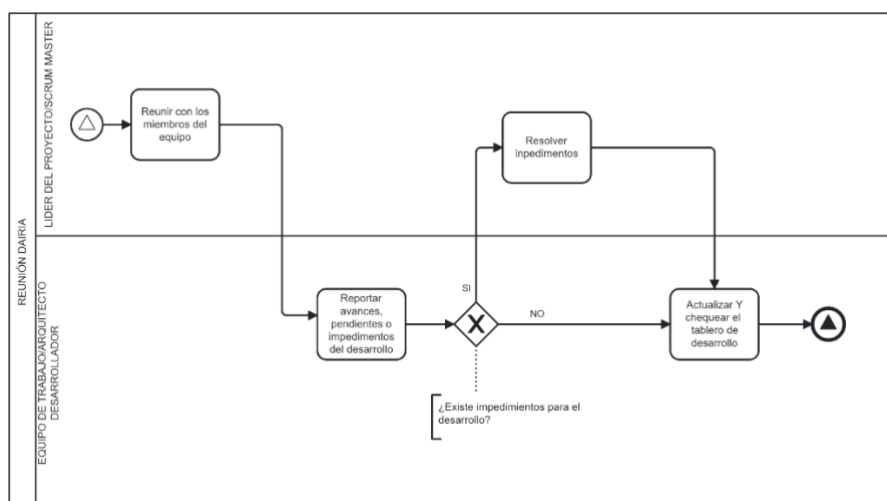


Fig. 17 Reunión Diaria del sprint

Esta reunión diaria tiene un fin tácito que es la de generar un compromiso en el desarrollador. La falta de trabajo o compromiso saldrá a la luz cuando no existan respuestas inmediatas a la primer y segunda pregunta propuestas para esta reunión.

Revisión de sprint: Tiene como finalidad inspeccionar el trabajo realizado durante el sprint. Como se aprecia en la Fig. 18, la reunión tendrá un máximo de 5 horas donde intervendrán las partes interesadas (equipo de trabajo, cliente y líder del proyecto). Se revisará el estado y los posibles cambios que sugiera el cliente para optimizar el mismo.

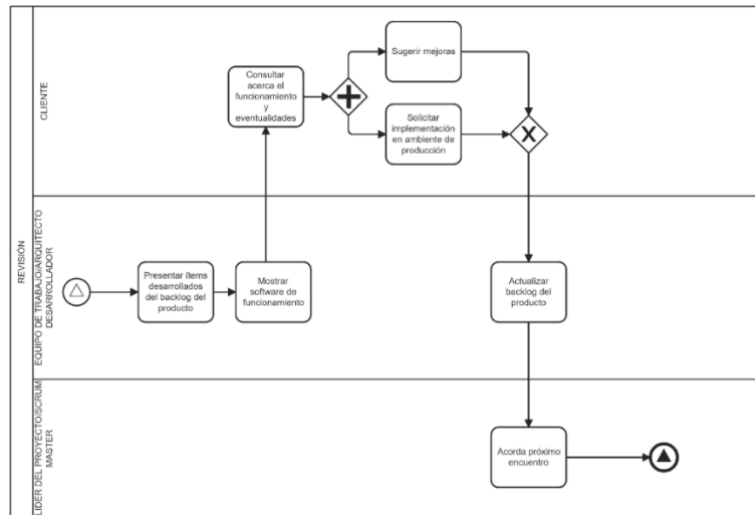


Fig. 18 Revisión del sprint

Mediante el entorno de desarrollo/Testing se puede presentar el trabajo realizado, presentando las funcionalidades a través de un flujo consecutivo de funcionalidades, con ejemplos concretos. En este apartado, se puede recibir feedback sobre ciertos incrementos o mejoras donde se podrá contemplar o se tomará en cuenta para un nuevo sprint.

Retrospectiva del sprint: Reunión para realizar una revisión e inspección interna a los involucrados del proyecto y crear un plan de mejoras para el próximo sprint, durará no más de 4 horas y se evaluará al sprint actual. Como muestra la Fig. 19, la reunión de retrospectiva mira atrás en el tiempo y permitirá conocer qué cosas se hicieron bien, qué cosas se podrían haber hecho de mejor manera, qué cosas se debe dejar de hacer, qué cosas se debería implementar y qué cosas resultaron frustrantes. Todas las respuestas contempladas en esta reunión se convierten en una retroalimentación y el equipo decidirá sobre qué elementos contemplará en el plan de mejoras para el próximo sprint.

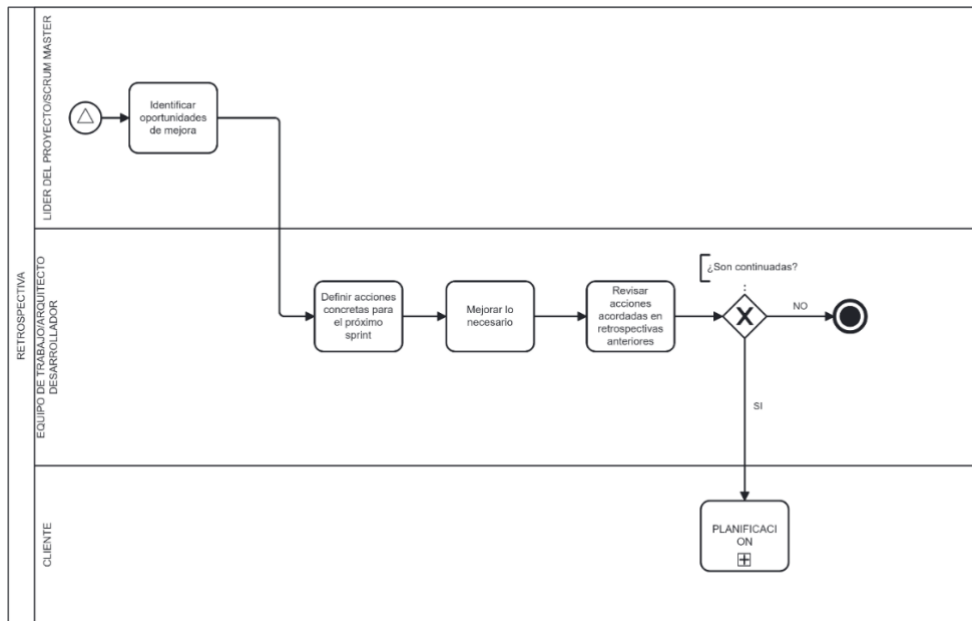


Fig. 19 Retrospectiva del sprint

Se utilizará un tablero online para determinar las actividades del equipo de trabajo donde se distingan las tareas “POR HACER”, “EN PROGRESO”, “REALIZADAS”, en el cual se distinguirán tarjetas CRC (Clase, Responsabilidad, Colaborador) de colores diferentes según el nivel de dificultad de la tarea. En la TABLA V, se muestra los colores por dificultad del requerimiento según se lo identifica dentro de la empresa.

TABLA V
COLORES DEL TABLERO ACORDE A LA DIFICULTAD DEL REQUERIMIENTO

DIFICULTAD	COLOR
Alta	Rojo
Media	Amarillo
Baja	Verde

A este tablero tendrán acceso todas las partes involucradas del proyecto para información de avance general del mismo.

Artefactos

Los artefactos son las representaciones del trabajo que se realiza y proporcionan transparencia hacia quienes soliciten información del proyecto.

- **Product Backlog:** listado donde se enumeran los requisitos funcionales y no funcionales del requerimiento.

- **Lista de productos:** es la lista de todos los requerimientos del producto, esta lista será realizada y generada en la etapa de levantamientos de requerimientos.
- **Historias de usuario:** se trata de describir los requerimientos el cliente desde su perspectiva, incluyendo un título y las principales funciones.

Por cada reunión de revisión o retrospectiva se debe contemplar un documento de revisión, aprobación o entrega del cumplimiento del objetivo durante el sprint.

En la Fig. 20, se puede observar el resumen de las reuniones propuestas para el marco de trabajo para la etapa de desarrollo de software.

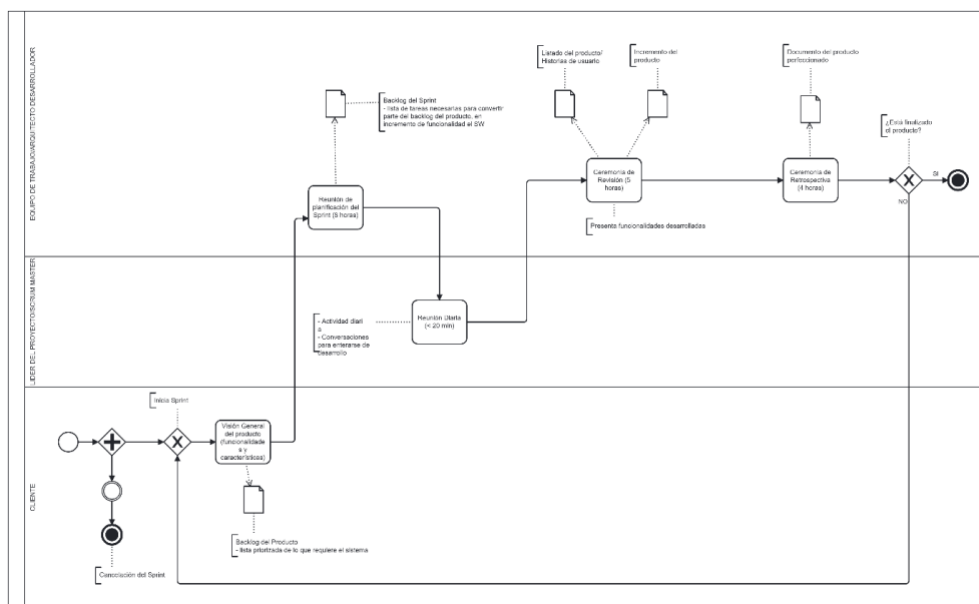


Fig. 20 Resumen de Iteración del Sprint

APLICACIÓN - MARCO DE TRABAJO PARA EL DESARROLLO DE SOFTWARE

Se aplica en la empresa CloudStudio Cía. Ltda., el marco de trabajo descrito anteriormente para el desarrollo de software enfocado en el requerimiento de la matriz de seguimiento de auditoría para cooperativas de ahorro y crédito.

El marco de trabajo propuesto se caracteriza por segmentar las actividades en pequeños grupos y priorizar entregas parciales al cliente para su revisión ofreciendo así soluciones optimas y oportunas en tiempos cortos.

Levantamiento de requerimientos

En la Fig. 21, se muestra el formato establecido por la institución para la etapa de levantamiento de requerimientos, misma que contiene la siguiente información:

	REVISIÓN REQUERIMIENTO DEL PRODUCTO Y/O SERVICIO		
	Versión: 0	Fecha de Emisión: 16/06/2016	Pág: 1

DATOS CLIENTE

Cliente:	COOPERATIVA AT
Contacto Cliente:	Pintag Llanganate
Área / Módulo Contacto Cliente:	AUDITORIA
Fecha levantamiento:	01/12/2022

DATOS CLOUDSTUDIO

Analista Responsable:	Ing. Marcos Guerrero
Participantes CloudStudio:	Ing. Janneth Guamán
Lugar del levantamiento	Instalaciones de la Cooperativa

Nombre responsable específico cliente:		
Teléfonos/correo responsable específico cliente:		
Dato general de requerimiento Marque con X	Detalle Requerimiento	Responsable
<input type="checkbox"/>	Cambios de interfaz producto	
<input checked="" type="checkbox"/>	Cambios de back producto	
<input type="checkbox"/>	Módulos adicionales	
<input type="checkbox"/>	Enlace con otros aplicativos	
<input type="checkbox"/>	Ajustes de seguridad	
<input type="checkbox"/>	Cambios en módulos producto	
<input type="checkbox"/>	Otro	<p>Nueva funcionalidad. Crear un módulo de seguimiento de hallazgos de auditoría, donde se podrá realizar lo siguiente:</p> <ol style="list-style-type: none"> 1) Ingreso la solicitud de seguimiento, la información a ingresar debe estar de acuerdo al adjunto enviado al correo electrónica. Se asigna y envía por correo electrónico al usuario auditado 2) Aceptación de un usuario auditado al que se delegó la acción. Acepta e ingresa las posibles estrategias o plan de acción para mitigar el riesgo sobre el hallazgo indicado. 3) Aprobación de estrategias o plan de acción del auditado por el jefe de auditoría. 4) Seguimiento d estrategias o plan de acción por el auditado y jefe de auditoría. 5) Finalización del seguimiento de auditoría incorporando un reporte de responsabilidad

Fig. 21 Levantamiento de Requerimientos acerca de la matriz de auditoría

A partir de estos requerimientos se distinguen los requerimientos funcionales en la TABLA VI con los que se trabajará durante el TT:

TABLA VI
REQUERIMIENTOS FUNCIONALES

USUARIO	N°	REQUISITO	DESCRIPCION
Jefe de Auditoría	REF01	Ingreso de solicitud de seguimiento	<p>El sistema permitirá el ingreso de la información que se contempla en el archivo Excel, catálogos definidos por el área de auditoría donde solo permita seleccionar opciones de una lista desplegable, además permitirá ingresar un comentario y seleccionar el usuario auditado.</p> <p>La información debe ser enviada a las notificaciones del sistema y al correo electrónico del usuario auditado</p>
Jefe de Auditoría	REF02	Aprobación de estrategias y plan de acción del auditado	<p>El jefe de auditoría podrá revisar el plan de acción ingresado por el auditado y tomará la decisión de aprobar, negar o retornar de etapa al plan de acción para ser modificado. Notificará al usuario auditado a través del correo electrónico</p>
Auditado	REF03	Revisión de solicitud de seguimiento e ingreso de plan de acción	<p>El auditado podrá visualizar la información de la solicitud de seguimiento e ingresar y enviar el respectivo plan de acción (Actividades, documentos de respaldos, fecha inicio, fecha fin) o estrategias para mitigar el hallazgo descrito</p>

Auditado	REF04	Seguimiento del plan de Acción o estrategias	El sistema permitirá al auditado ingresar la información correspondiente a cada actividad de su plan de acción
Jefe de Auditoría	REF05	Seguimiento del plan de Acción o estrategias	El sistema permitirá al jefe de auditoría revisar las actividades realizadas por el auditado de acuerdo al plan de acción aprobado. Si es el caso, el sistema debe permitir dar por terminado el seguimiento
Auditado	REF06	Consulta de Seguimiento	El sistema permitirá revisar el historial de seguimientos del auditado, permitirá también imprimir un respaldo físico del seguimiento
Jefe de auditoría	REF07	Consulta de Seguimiento	El sistema permitirá revisar el historial de seguimientos por auditado e imprimir un respaldo físico general, por auditado o por detalle de seguimiento según convenga

Los requerimientos funcionales, ver Fig. 22, se pueden resumir en los casos de uso siguientes:

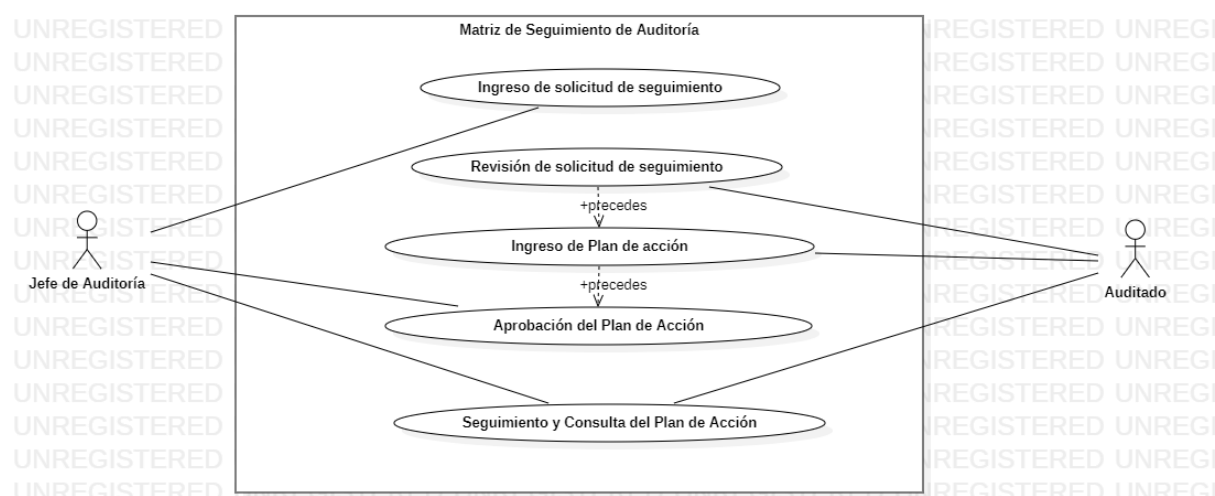


Fig. 22 Caso de uso - Requerimientos Funcionales

A partir de esta información, se lista las historias de usuario a contemplar en la aplicación del marco de trabajo para el desarrollo de software:

- Ingreso de solicitud de seguimiento
- Revisión de solicitud de seguimiento e ingreso de plan de acción
- Aprobación del plan de acción
- Seguimiento del plan de Acción
- Consulta de Seguimiento

Miembros y Roles de los miembros

En el marco de trabajo propuesto se describen tres roles imprescindibles para la etapa del desarrollo de software:

- Cliente/Product Owner
- Equipo de desarrollo/Arquitecto o Desarrollador
- Líder del proyecto/Scrum Master

Según el marco de trabajo propuesto, en la TABLA VII, se detalla el rol y la persona responsable dentro de la empresa:

TABLA VII
MIEMBROS Y ROLES DEL MARCO DE TRABAJO PROPUESTO

ROL	MIEMBRO	CONTACTO
Cliente/Product Owner	Cooperativa AT representada por Pintag Llanganate	Contacto: 0989755778 Email: pllanganate@coacat.com.ec
Líder del proyecto/Scrum Master	Ing. Marcos Guerrero	Contacto: 0983309625 Email: mguerrero@cloudstudio.com.ec
	Ing. Janneth Guamán	Contacto: 0984027353

Equipo de desarrollo/Arquitecto o Desarrollador		Email: jguaman@cloudstudio.com.ec
	Ing. Jorge Duchi	Contacto: 0983421045 Email: jduchi@cloudstudio.com.ec
	Ing. Marcelo Espinoza	Contacto: 0994036590 Email: mespinosa@cloudstudio.com.ec

Pila del Producto (Product BackLog)

Es la colección de requisitos establecidos por los miembros del equipo, aquí se numeran las funcionalidades que debe presentar la matriz de seguimientos de hallazgos dentro del Core financiero Softbank para determinar el número de sprint que conformarán el desarrollo de esta funcionalidad. En la TABLA VIII, se detalla la pila del producto que se trabajará en el TT:

TABLA VIII
PILA DEL PRODUCTO (PRODUCT BACKLOG)

N°	DETALLE DE TAREAS	ESTIMACIÓN EN HORAS	PRIORIDAD	RESPONSABLE EQUIPO DE TRABAJO
HU-001	Ingreso de solicitud de seguimiento	24	ALTA	Janneth Guamán
HU-002	Revisión de solicitud de seguimiento e ingreso de plan de acción	24	ALTA	Jorge Duchi
HU-003	Aprobación del plan de acción	24	ALTA	Marcelo Espinoza
HU-004	Seguimiento del plan de Acción	16	MEDIA	Janneth Guamán

HU-005	Consulta de Seguimiento	16	BAJA	Jorge Duchi Marcelo Espinosa
--------	-------------------------	----	------	---------------------------------

Con la definición de requerimientos sobre la matriz de seguimientos de hallazgos, se obtiene la complejidad y el esfuerzo requerido de cada una determinando que los requerimientos son realizables y se procede a agrupar las diferentes tareas en iteraciones o sprint.

Lista de tareas de la iteración (Spring BackLog)

En la TABLA IX, se describe las tareas que agruparan los sprint a realizar dentro del marco de trabajo propuesto, tareas que deben ser cumplidas en un determinado periodo de tiempo, específicamente 3 semanas para este desarrollo.

TABLA IX
LISTADO DE TAREAS DE ITERACIÓN (SPRINT BACKLOG)

Sprint (Iteración)	Tareas	Responsable	Fecha		Esfuerzo (horas)
			Inicio	Fin	
1	HU-001	Equipo de desarrollo	06-02-2023	09-02-2023	24
	HU-002				
	HU-003				
2	HU-004	Equipo de desarrollo	24-02-2023	26-02-2023	16
	HU-005				

Cabe indicar que en la iteración 1, los responsables de cada tarea trabajaran a la par en cada una de ellas, ya que la información que relaciona las tres tareas son importantes para llevar a cabo el flujo de la solicitud de seguimientos de hallazgos.

Para asignar y llevar a cabo el control de las tareas se ha generado un tablero de actividades dentro del aplicativo Jira, ver Fig. 23, donde cada responsable podrá revisar el avance de las actividades y del proyecto en general:

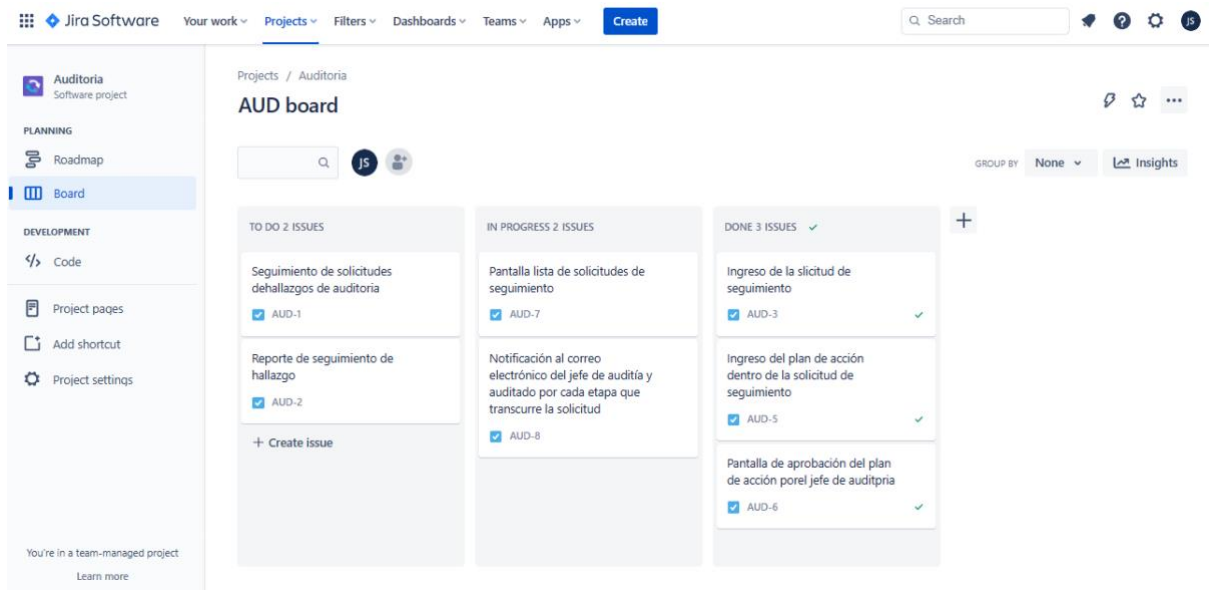


Fig. 23 Tablero de actividades utilizando Jira

Sprint / Iteraciones

A continuación, en la TABLA X, se describe el resumen de las reuniones diarias mantenidas entre el Líder del Proyecto y el equipo de desarrollo:

TABLA X
RESUMEN - REUNIÓN DIARIA SPRINT 1

FECHA	SPRINT	RESPONSABLE	OBSERVACIONES
			<p>A. ¿Qué se trabajó ayer?</p> <p>B. ¿Qué realizará el día de hoy?</p> <p>C. ¿Se presento algún inconveniente?</p>
07-02-2023	1	Janneth Guamán	<p>A. Creación de tablas dentro de SQL de las clases a utilizar</p> <p>B. Pantalla para listar las solicitudes ingresadas</p> <p>C. Ninguna</p>
	1	Jorge Duchi	<p>A. Creación de clases y mantenimientos simples</p> <p>B. Pantalla para listar las solicitudes enviadas al usuario auditado</p> <p>C. Ninguna</p>
	1	Marcelo Espinosa	<p>A. Creación de objetos y métodos a utilizar</p>

			<p>B. Pantalla para listar las solicitudes con el plan de acción para su autorización</p> <p>C. Estado de la solicitud antes de ser autorizada</p>
08-07-2023	1	Janneth Guamán	<p>A. Pantalla para listar las solicitudes ingresadas</p> <p>B. Pantalla de ingreso de información de la solicitud de hallazgo que permita seleccionar el usuario auditado</p> <p>C. Ninguna</p>
	1	Jorge Duchi	<p>A. Pantalla para listar las solicitudes enviadas al usuario auditado</p> <p>B. Pantalla para mostrar la información de la solicitud y que el usuario auditado pueda ingresar el plan de acción</p> <p>C. Ninguna</p>
	1	Marcelo Espinosa	<p>A. Pantalla para listar las solicitudes con el plan de acción para su autorización</p> <p>B. Pantalla para aprobar o negar el plan de acción enviado por el usuario auditado</p> <p>C. Ninguna</p>
09-02-2023	1	Janneth Guamán	<p>A. Pantalla de ingreso de información de la solicitud de hallazgo que permita seleccionar el usuario auditado</p> <p>B. Revisar el flujo de la solicitud que se ingresa y enviar al usuario notificado</p> <p>C. Errores que son corregidos de inmediato</p>
	1	Jorge Duchi	<p>A. Pantalla para mostrar la información de la solicitud y que el usuario auditado pueda ingresar el plan de acción</p>

			<p>B. Revisar que la solicitud llegue a la bandeja del usuario auditado y este pueda ingresar el plan de acción</p> <p>C. Errores que son corregidos de inmediato</p>
	1	Marcelo Espinosa	<p>A. Pantalla para aprobar o negar el plan de acción enviado por el usuario auditado</p> <p>B. Revisar que el plan de acción enviado por el usuario auditado llegue a la bandeja del jefe de auditoría</p> <p>C. Errores que son corregidos de inmediato</p>

Como resultado de la integración de las tres primeras tareas realizadas se procede a la presentación del avance al cliente, con fecha 10-02-2023 en las instalaciones de la cooperativa, en donde se procede a realizar la prueba conjuntamente con el representante de la cooperativa, jefe de auditoría y un usuario auditado.

Dentro de la reunión de revisión, se valida que el flujo de la solicitud de seguimientos de hallazgos de auditoría inicie y finalice sin errores, validando el ingreso de información, campos en blanco y mensajes de alerta. Se valida por el cliente sin presentar ningún argumento de corrección. Además, se genera un acta de validación y aceptación para continuar con el siguiente Sprint.

Cabe indicar que para el siguiente sprint se trabajó de igual manera tanto en las reuniones diarias como en la reunión de revisión.

En las reuniones de retrospectiva, los miembros del equipo comparten las experiencias en la utilización de reuniones diarias, revisiones y la forma de compartir conocimientos en medio del aprendizaje.

A continuación, se muestran las historias de usuarios descritas, las cuales fueron utilizadas para llevar a cabo el desarrollo del requerimiento con sus respectivas pantallas.

En la TABLA XI, se resume el flujo de ingreso de la solicitud de seguimiento de hallazgos de auditoría:

TABLA XI
CASO DE USO - INGRESO DE SOLICITUD DE SEGUIMIENTO

Caso de Uso:	Ingreso de solicitud de seguimiento	Actor(es):	Jefe de Auditoría
Código	HU-001	Pantalla(s):	

Ref. Req:	REF01
Resumen:	Permite al jefe de auditoría ingresar la información sobre la solicitud de seguimiento de hallazgos
Objetivos:	<ul style="list-style-type: none"> ▪ Generar solicitud de seguimiento ▪ Enviar al usuario auditado
Flujo de actividades:	<ol style="list-style-type: none"> 1. Mostrar en una pantalla el listado de todas las solicitudes de hallazgos ingresadas por el jefe de auditoría, mostrando 2 botones Nuevo – Editar 2. Al pulsar el botón nuevo, permitirá ingresar una nueva solicitud seleccionando los datos de acuerdo a las listas desplegables y además deberá seleccionar al usuario auditado. 3. Al pulsar el botón Editar podrá modificar cierta información de la solicitud de hallazgos antes de ser enviada al usuario auditado. 4. Una vez lista la solicitud, el jefe de auditoría puede pulsar el botón enviar y se notificará en la bandeja del sistema y correo del auditado.

En la Fig. 24, se muestra la pantalla para el rol jefe de auditoría donde se lista las solicitudes en estado ingresada.

Seguimiento de Auditoría

Fecha Inicio: 20-03-2023 Fecha Fin: 20-03-2023 Nivel de Riesgo: TODOS Buscar

Nuevo

Fecha	Responsable de Emisión	Nº Informe	Tipo de Proceso	Subcomponente	Nº Hallazgo	Observación por Componente/ Subcomponente	Impacto	Probabilidad	Riesgo	Recomendación	
Recuento=0											

Página 1 de 1

Fig. 24 Listado de solicitudes ingresadas por jefe de auditoría

Seguido se muestra la Fig. 25, pantalla donde el rol jefe de auditoría podrá ingresar los datos de la solicitud hacia el rol auditado.

Seguimiento de Auditoría

Detalle del Seguimiento de Auditoría

Responsable de Emisión: SEPS

N° de informe: 001 Fecha de revisión: 01-02-2023

Tipo de Proceso: Operativo

SubComponente: RIESGO OPERATIVO

Observaciones por Componente/Subcomponente: Observaciones descritas por el Jefe de Auditoría

Impacto: 5 - Moderado Probabilidad: 2 - Muy Baja

10 **RIESGO:** **RIESGO MEDIO**

Recomendación: Recomendación para mitigar el hallazgo por el jefe de auditoría

Área: CREDITO

Responsable: JEFE DE CREDITO

Usuario: ADMINISTRADOR

Fig. 25 Pantalla de ingreso de información de solicitud de hallazgo de auditoría

En la TABLA XII, se describe el caso de uso para la revisión de la solicitud por parte del rol auditado, en donde también puede ingresar el plan de acción para mitigar el riesgo del hallazgo:

TABLA XII
CASO DE USO: REVISIÓN DE SOLICITUD DE SEGUIMIENTO E INGRESO DE PLAN DE ACCIÓN

Caso de Uso:	Revisión de solicitud de seguimiento e ingreso de plan de acción	Actor(es):	Auditado
Código	HU-002	Pantalla(s):	
Ref. Req:	REV02		
Resumen:	Permite revisar al auditado la solicitud de hallazgo recibida e ingresar el plan de acción correspondiente frente a las recomendaciones del jefe de auditoría		
Objetivos:	<ul style="list-style-type: none"> ▪ Revisar solicitud de seguimiento de hallazgo ▪ Ingresar y enviar el plan de acción 		
Flujo de actividades:	1. El sistema mostrará una pantalla donde se listará todas las solicitudes de hallazgos enviadas al usuario auditado en estado		

2. Mostrará un botón Editar, el cual debe seleccionar la solicitud y pulsar para revisar e ingresar el plan de acción correspondiente
3. En la pantalla de edición, podrá guardar el plan de acción o procesar etapa para enviar una vez culminado el mismo al jefe de auditoría.
4. También incluirá un botón Comentarios, donde se listará las observaciones que ha enviado el jefe de auditoría al procesar la etapa.

La Fig. 26 muestra la pantalla para listar las solicitudes en estado pendiente de aprobar, donde se visualizará las solicitudes respondidas por el rol auditado hacia el rol jefe de auditoría.

Seguimiento de Auditoría

Fecha Inicio: 20-02-2022 Fecha Fin: 20-03-2023 Nivel de Riesgo: TODOS

Fecha	Responsable de Emisión	Nº Informe	Tipo de Proceso	Subcomponente	Nº Hallazgo	Observación por Componente/ Subcomponente	Impacto	Probabilidad	Riesgo	Recomendación
01-02-2023	SEPS	001	Operativo	RIESGO OPERA...	1	Observaciones descritas por...	5	2	10	Recomendación para mitigar... CREDI

Recuento=1

Página 1 de 1

Fig. 26 Pantalla Listado de solicitud de seguimiento del auditado

La Fig. 27, muestra la pantalla para el ingreso del plan de acción de la solicitud de hallazgos.

Seguimiento de Auditoría

Detalle del Seguimiento de Auditoría

Nº de informe: 1

Responsable de Emisión: SEPS

Nº de informe: 001 Fecha de revisión: 01-02-2023

Tipo de Proceso: Operativo

SubComponente: RIESGO OPERATIVO

Observaciones por Componente/Subcomponente: Observaciones descritas por el Jefe de Auditoría

Impacto: 5 - Moderado Probabilidad: 2 - Muy Baja

RIESGO: 10 RIESGO MEDIO

Recomendación: Recomendación para mitigar el hallazgo por el jefe de auditoría

Área: CREDITO

Responsable: JEFE DE CREDITO

Usuario: ADMINISTRADOR

DESCRIPCION DE LA ESTRATEGIA O ACCIONES CORRECTIVAS

Descripción de la estrategia del área	Fecha Inicio	Fecha Fin	Documento Entregable
Descripción de estrategia 1	20-03-2023	20-03-2023	documento entregable 1
Actividad de plan de acción	22-03-2023	26-03-2023	entregable 2

Fig. 27 Pantalla Revisión e ingreso de Plan de Acción

En la TABLA XIII, se describe el caso de uso para la aprobación del plan de acción por el rol de jefe de auditoría.

TABLA XIII
CASO DE USO: APROBACIÓN DEL PLAN DE ACCIÓN

Caso de Uso:	Aprobación del plan de acción	Actor(es):	Jefe de Auditoría
Código	HU-003	Pantalla(s):	
Ref. Req:	REV03		
Resumen:	Permite aprobar el plan de acción enviado por el auditado		
Objetivos:	<ul style="list-style-type: none"> ▪ Aprobar plan de acción 		
Flujo de actividades:	<ol style="list-style-type: none"> 1. El sistema mostrará una pantalla donde se listará todas las solicitudes de hallazgos recibidas por el auditado donde se detalla el plan de acción. 2. El jefe de auditoría podrá pulsar el botón ver y decidir si retorna etapa, autoriza o anula el plan de acción descrito por el auditado. 		

La Fig. 28, muestra la pantalla donde se listen las solicitudes en estado por aprobar plan de acción para el rol jefe de auditoría:

Autorización Seguimiento de Auditoría

Fecha Inicio: 20-03-2022 Fecha Fin: 20-03-2023 Nivel de Riesgo: TODOS

Fecha	Responsable de Emisión	Nº Informe	Tipo de Proceso	Subcomponente	Nº Hallazgo	Observación por Componente/ Subcomponente	Impacto	Probabilidad	Riesgo	Recomendación
01-02-2023	SEPS	001	Operativo	RIESGO OPERA...	1	Observaciones descritas por...	5	2	10	Recomendación para mitigar... CREDI

Recuento=1

Página 1 de 1

Fig. 28 Pantalla Listado de Solicitudes con Plan de acción para ser autorizados

En la Fig. 29, se muestra la pantalla para retornar etapa, aprobar o negar el plan de acción.

Autorización Seguimiento de Auditoría

Fig. 29 Pantalla de Autorización de Seguimiento de Auditoría

A continuación, en la TABLA XIV, se describe las actividades para dar seguimiento al plan de acción ejecutado por el rol auditado.

TABLA XIV
CASO DE USO: SEGUIMIENTO DEL PLAN DE ACCIÓN

Caso de Uso:	Seguimiento del plan de Acción	Actor(es):	Jefe de Auditoría Auditado
Código	HU-004	Pantalla(s):	
Ref. Req:	REV04 – REV05		
Resumen:	Permite revisar el avance del plan de acción de cada solicitud		
Objetivos:	<ul style="list-style-type: none"> Ingresar y revisar el avance del plan de acción 		
Flujo de actividades:	<ol style="list-style-type: none"> El sistema permitirá al auditado ingresar el avance del plan de acción El sistema permitirá al jefe de auditoría revisar el avance del plan de acción y darlo por terminado al seguimiento pulsando el botón terminar seguimiento 		

En la Fig. 30, se muestra la pantalla del rol auditado para revisar el avance de la solicitud de seguimiento de hallazgos.

Fig. 30 Pantalla Ingreso de avance del Plan de acción por el auditado

En la Fig. 31, se muestra la pantalla del rol jefe de auditoria para revisar el avance de la solicitud de seguimiento de hallazgos, dar por terminado el mismo.

Aprobación Seguimiento de Auditoría

Fig. 31 Pantalla Revisión del avance del plan de acción por jefe de auditoría

A continuación, en la TABLA XV, se describe las actividades de consulta de solicitudes terminadas.

TABLA XV
CASO DE USO: CONSULTA DE SEGUIMIENTO

Caso de Uso:	Consulta de Seguimiento	Actor(es):	Jefe de Auditoria Auditado
Código	HU-005	Pantalla(s):	
Ref. Req:	REV06 – REV07		

Resumen:	Permite consulta el detalle de la solicitud de seguimiento de hallazgos del auditado y su plan de acción finalizado
Objetivos:	<ul style="list-style-type: none"> ▪ Consultar información sobre la solicitud y su plan de acción
Flujo de actividades:	<ol style="list-style-type: none"> 1. El sistema permitirá al jefe de auditoría y auditado consultar el histórico de solicitudes de hallazgos y los planes de acción ejecutados. 2. Puede imprimir reportes de acuerdo a su necesidad

La Fig. 32 muestra la pantalla de consulta de la solicitud de seguimiento una vez finalizada.

Seguimiento de Auditoría Finalizada

Detalle del Seguimiento de Auditoría

Cerrar Comentarios

Nº de informe: 1

Responsable de Emisión: SEPS

Nº de informe: 001 Fecha de revisión: 01-02-2023

Tipo de Proceso: Operativo

SubComponente: RIESGO OPERATIVO

Observaciones por Componente/Subcomponente: Observaciones descritas por el Jefe de Auditoría

Impacto: 5 - Moderado Probabilidad: 2 - Muy Baja

10 RIESGO: RIESGO MEDIO

Recomendación: Recomendación para mitigar el hallazgo por el jefe de auditoría

Área: CREDITO

Responsable: JEFE DE CREDITO

Usuario: ADMINISTRADOR

DESCRIPCIÓN DE LA ESTRATEGIA O ACCIONES CORRECTIVAS

Imprimir Cambios de Solicitud

Descripción de la estrategia del área	Fecha Inicio	Fecha Fin	Documento Entregable	Tiene Imagen	Descargar	Estado	Imprimir
Descripción de estrategia 1	20-03-2023	20-03-2023	documento entregable 1	<input type="checkbox"/>	Archivo	EN PROCESO	Imprimir
Actividad de plan de acción	22-03-2023	26-03-2023	entregable 2	<input type="checkbox"/>	Archivo	EN PROCESO	Imprimir

Fig. 32 Pantalla de consulta de solicitud de seguimiento con el plan de acción

Además, dentro de esta pantalla se muestra un reporte sobre la solicitud de seguimiento y su plan de acción, como se puede observar en la Fig. 33:

HALLAZGOS DE AUDITORIA INTERNA Y PLANTEAMIENTO DE ESTRATEGIAS

Usuario:ADMIN - Agencia:MATRIZ - Fecha Sistema: 20/03/2023 - Fecha/Hora: 15/04/2023 12:05:39

Nº HALLAZGO: 1

FECHA ACTUAL: 01/01/2019
 TIPO DE PROCESO: Operativo
 RESPONSABLE DE LA ESTRATEGIA: JEFE DE CREDITO
 USUARIO: ADMINISTRADOR
 Nº INFORME: 001
 SUBCOMPONENTE: RIESGO OPERATIVO

ÁREA: CREDITO
 FECHA REVISIÓN: 01/02/2023

IMPACTO: 5 - Moderado
 PROBABILIDAD: 2 - Muy Baja
 NIVEL DE EXPOSICIÓN AL RIESGO: 10 RIESGO MEDIO

DESCRIPCION DEL HALLAZGO	RECOMENDACIÓN	DESCRIPCION DE LA ESTRATEGIA O ACCIONES CORRECTIVAS	DOCUMENTO ENTREGABLE (Documento que debe presentar para justificar el cumplimiento de la estrategia)
Descripción de estrategia 1	Recomendación para mitigar el hallazgo por el jefe de auditoria	Descripción de estrategia 1	documento entreable 1
		FECHA INICIO: 20/03/2023	FECHA FIN: 20/03/2023

SEGUIMIENTO DE LAS RECOMENDACIONES / ACCIONES CORRECTIVAS			
	FECHA DE REVISIÓN	PORCENTAJE	COMENTARIO
1ra. Revisión			
2da. Revisión			
3ra. Revisión			

AUDITOR INTERNO

AUDITADO
ADMINISTRADOR

AUDITOR JR.

Fig. 33 Reporte generado por Plan de Acción

Pruebas de aceptación

Se define de manera general la prueba de aceptación y se describe en la TABLA XVI, donde contempla el flujo general de la solicitud de seguimiento de hallazgos de auditoría, y se describe las tareas realizadas en su iteración con el cliente:

TABLA XVI
CASO DE PRUEBA

CASO DE PRUEBA			
Código: CP1	Nº HU: 001-002-003-004-005		
Descripción	Aceptación	SI	NO
Ingreso de la solicitud		X	
Listar la solicitud en la pantalla del auditado		X	
Crear el plan de acción y enviar al jefe de auditoria		X	
Dar seguimiento al plan de acción el auditado		X	

Consultar la solicitud de seguimiento	X	
Resultado Esperado:	Flujo de solicitud de seguimiento correctamente	
Evaluación de la prueba:	Las pruebas se concluyeron satisfactoriamente	

Una vez generada las reuniones respectivas, debe existir un acta de certificación de los requerimientos entre los involucrados del proyecto, para dar paso a su liberación en ambiente de producción del cliente.

A continuación, en la Fig. 34 muestra la plantilla utilizada para el presente trabajo, que será plasmado como Acta oficial de Certificación de requerimientos de la empresa:



Gestión de Proyectos
Control de Calidad

COOPERATIVA DE AHORRO Y CRÉDITO AT

ACTA DE CONFORMIDAD No 8	
PROYECTO:	Implementación de Core Financiero SoftBank
RESPONSABLE:	Janneth Guamán
FECHA:	05/03/2023

DETALLE DE ENTREGABLES				
CODIGO HISTORIA	DESCRIPCIÓN HU	DETALLE	COMENTARIOS	CRITERIO DE ACEPTACIÓN

PARTICIPANTES	
POR EL CLIENTE	POR CLOUDSTUDIO
RESUMEN Revisión y Certificación del módulo de Auditoría, por parte de los usuarios responsables de la cooperativa AT, de acuerdo a los requerimientos levantados. Se presenta el sistema, donde se muestra la funcionalidad de cada requerimiento desarrollado en la versión de SoftBank para la cooperativa.	
Firma del representante del Cliente	Firma del representante de CloudStudio

RELACION DE ANEXOS	
Anexo 1:	

Fig. 34 Acta de Certificación de Requerimientos

De igual manera, en la Fig. 35 se presenta el documento de entrega-recepción de los requerimientos para finalizar con el requerimiento/proyecto.

ACTA ENTREGA - RECEPCIÓN		
1.- Datos Generales		
Módulo		
Cliente		
Fecha Inicio	Fecha Finalización	Acta N°
Gerente del Proyecto		
Revisión CloudStudio		
Revisión COAC		

Razón Entrega-Recepción Módulo

A los xx días del mes del xxxx, se da por presentado y revisado el presente módulo cumpliendo las siguientes etapas:

--	--

En el presente resumen se da por entendido que:

- El personal responsable de la Cooperativa revisa la funcionalidad del proceso que genera el módulo en Softbank.
- Se da por presentado y entregado el módulo en el proyecto de implementación del Core financiero SoftBank en la COAC AT

Comentarios

En caso de existir requerimientos nuevos, el proveedor se encargará de evaluar los tiempos de desarrollo y costos adicionales en caso de que lo requiera.

Aprobaciones

NOMBRE	INSTITUCIÓN	FIRMA

Fig. 35 Acta Entra-Recepción de Requerimiento/Proyecto

Para finalizar se ha realizado una encuesta a los desarrolladores de la empresa para conocer sus experiencias y expectativas una vez aplicado el marco de trabajo propuesto.

A continuación, se detalle lo obtenido en las encuestas realizados:

Pregunta 1: ¿Cree que es necesario llevar a cabo un control y planificación del proceso evolutivo del desarrollo de software enfocado en las tareas asignadas a cada miembro del equipo?

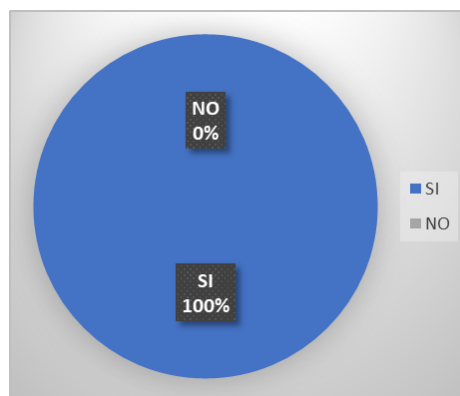


Fig. 36 Encuesta 2 – Gráfica porcentual acerca de la Pregunta 1

- En la Fig. 36 se observa que el 100% de los miembros del equipo indican que es necesario establecer un control y planificación de las tareas que debe realizar para tener un enfoque general del avance del proyecto y que este sea visible para todos.

Pregunta 2: ¿Cree Ud., que al aplicar el marco de trabajo propuesto los equipos de trabajo se autoorganizan para cumplir con sus tareas en los tiempos definidos?

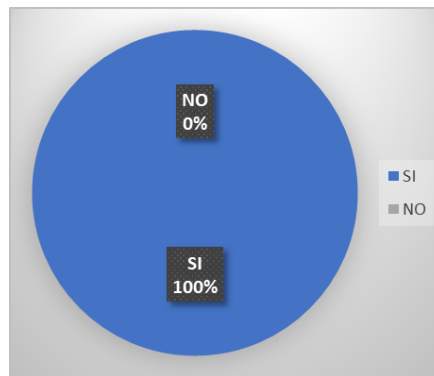


Fig. 37 Encuesta 2 – Gráfica porcentual acerca de la Pregunta 2

- Al aplicar el marco de trabajo propuesto en la empresa CloudStudio Cía. Ltda., el 100% del equipo de trabajo indica que estos se autoorganizaron para cumplir con las tareas en los tiempos definidos como se divide en la Fig. 37.

Pregunta 3: ¿El avance del proyecto o de las tareas del equipo esta visible para todos los involucrados?

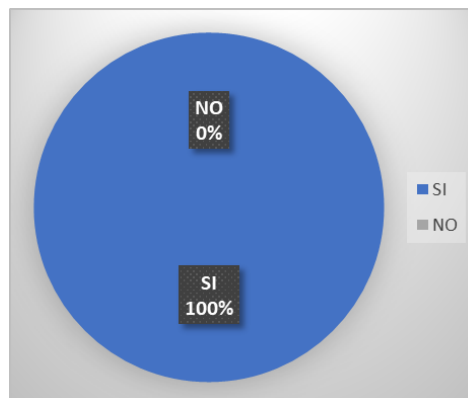


Fig. 38 Encuesta 2 – Gráfica porcentual acerca de la Pregunta 3

- La Fig. 38 indica que el 100% de los desarrolladores mencionan que las tareas y su avance fueron visibles para todos los involucrados, ya que utilizaron el tablero de tareas donde distinguían las tareas POR HACER, EN EJECUCIÓN Y TERMINADAS.

Pregunta 4: Considera que las reuniones diarias permiten conocer el estado de las tareas de cada miembro del equipo y su compromiso con su finalización

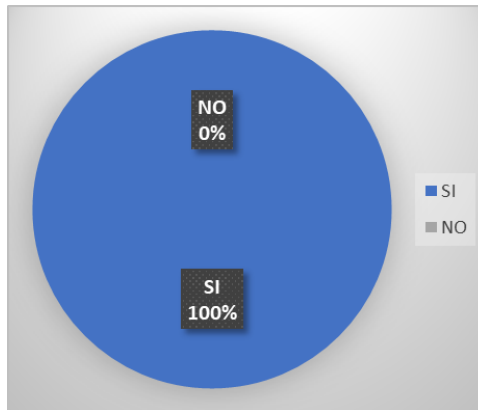


Fig. 39 Encuesta 2 – Gráfica porcentual acerca de la Pregunta 4

- Como se observa en la Fig. 39, el 100% del equipo indica que en las reuniones diarias se resumían las tareas que fueron ejecutadas y que se ejecutan durante el día, y que el líder del proyecto tuvo un control permanente de las tareas asignadas a cada uno.

Pregunta 5: Considera que la reunión de revisión con el cliente garantiza la continuidad del proyecto

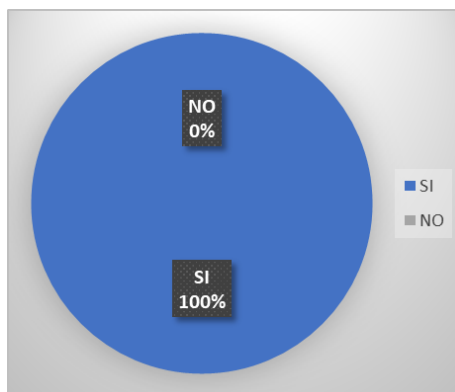


Fig. 40 Encuesta 2 – Gráfica porcentual acerca de la Pregunta 5

- La Fig. 40 determina que la reunión de revisión permitió al 100% del equipo de trabajo presentar las funcionalidades desarrolladas al cliente, para que valide los cambios y si existe algún cambio rápido poder realizarlo, caso contrario planificarlo para una nueva iteración o sprint.

Pregunta 6: Considera que, en la reunión de retrospectiva, los miembros del equipo comparten la experiencia obtenida y esto permite mejorar las próximas iteraciones de los requerimientos

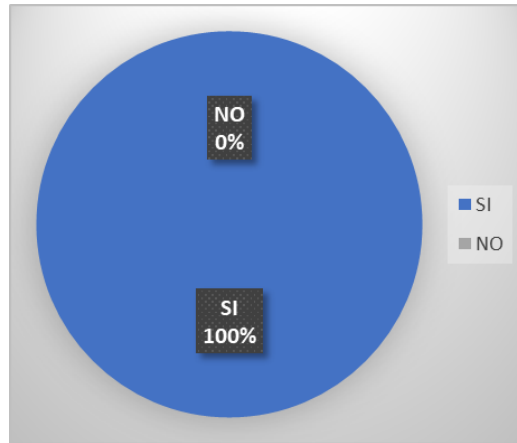


Fig. 41 Encuesta 2 – Gráfica porcentual acerca de la Pregunta 6

- En la Fig. 41 se observa que el 100% de los desarrolladores compartió experiencias y enseñanzas que les dejó las iteraciones realizadas para el desarrollo del requerimiento. Cabe indicar que se realizó un análisis interno de los resultados obtenidos y crearon un plan de mejoras para aplicar en el desarrollo posterior.

Pregunta 7: ¿Cuál cree Ud., que es la principal razón para aplicar un marco de trabajo en la etapa de desarrollo de software?

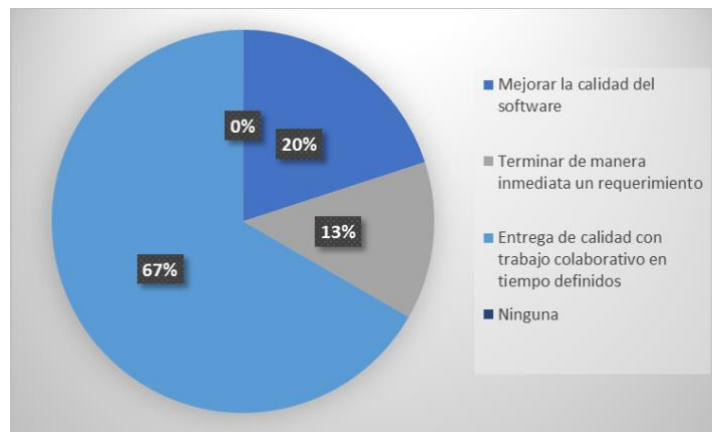


Fig. 42 Encuesta 2 – Gráfica porcentual acerca de la Pregunta 7

- En la Fig. 42, se observa que para el 67% de los desarrolladores, aplicar un marco de trabajo dentro de la empresa genera una entrega de calidad con trabajo colaborativo en tiempos definidos, el 20% indica que solo mejora la calidad del software y para el 13% solo ayuda a terminar de manera inmediata un requerimiento.

7. Discusión

En este apartado se presenta el resumen de las actividades que se llevaron a cabo para la ejecución del TT. Se destacará el estudio realizado de cada metodología y su caso práctico en tres etapas detalladas a continuación:

Etapas 1: Realizar un estudio comparativo acerca de las metodologías ágiles (XP y SCRUM) que permitan establecer fundamentos teóricos relevantes en el proceso de desarrollo software.

La realización de un estudio comparativo acerca de las metodologías ágiles (XP y Scrum) para establecer fundamentos teóricos relevantes en el proceso de desarrollo de software es una tarea importante y desafiante. Para alcanzar este objetivo, se realizó una serie de actividades que permitieron obtener información valiosa y relevante acerca de estas metodologías ágiles.

- En primer lugar, se realizó un marco teórico acerca de las metodologías ágiles en general. Esto implicó investigar acerca de los principios y valores que rigen estas metodologías, así como conocer las ventajas y desventajas de su aplicación en el desarrollo de software.
- Posteriormente, se estudió la metodología Scrum, sus principios, prácticas y técnicas específicas, determinando ventajas y desventajas, así como los casos de éxito y fracaso en su aplicación.
- De manera similar, se estudió la metodología XP, lo que implica comprender sus valores, principios y técnicas específicas. Se analizó las ventajas y desventajas, así como los casos de éxito y fracaso en su aplicación.
- Además, se investigó acerca de las buenas prácticas de programación para obtener soluciones óptimas. Esto permitirá establecer las bases necesarias para comparar y evaluar las metodologías ágiles en cuestión.
- Finalmente, se compararon las características y diferencias entre las metodologías XP y Scrum, en donde se abordó los principios de las metodologías ágiles, las técnicas específicas que utilizan, y cómo estas metodologías se ajustan a diferentes contextos de desarrollo de software.

Para lograr el objetivo de identificar la metodología ágil que se ajuste a los requerimientos de la empresa CloudStudio a través de una matriz relacional para de esta manera estandarizar los procesos de desarrollo de software, se llevaron a cabo las siguientes actividades:

Etapa 2: Identificar la metodología ágil que se ajuste a los requerimientos de la empresa mediante una matriz relacional para estandarizar los procesos de desarrollo de software.

- En primer lugar, se realizó un estudio exhaustivo del proceso de desarrollo de software de la empresa CloudStudio. Se identificaron las etapas del proceso, los roles y responsabilidades de cada uno de los miembros del equipo, y se establecieron las principales fortalezas y debilidades del proceso actual. Este análisis permitió tener una comprensión clara de los requerimientos específicos de la empresa.
- Con el estudio de estas metodologías XP y Scrum, se identificaron los principios, prácticas y técnicas específicas que las caracterizan. Asimismo, se evaluaron las ventajas y desventajas de su aplicación y se analizó cómo se ajustan estas metodologías a diferentes contextos de desarrollo de software.
- Luego de este estudio, se procedió a identificar la metodología que mejor se adaptaba a las necesidades de la empresa CloudStudio. Para esto, se elaboró una matriz relacional que permitió comparar las diferentes metodologías en función de los requerimientos específicos de la empresa, tales como el tamaño del equipo, artefactos, reuniones, interacciones, backlog del producto, el grado de incertidumbre y la necesidad de flexibilidad en el proceso de desarrollo.
- Tras la elaboración de la matriz relacional, se pudo identificar la metodología que mejor se adaptaba a las necesidades de la empresa CloudStudio. En este caso, se utilizó la metodología Scrum, debido a su capacidad para manejar proyectos complejos, su enfoque en la colaboración y comunicación constante entre los miembros del equipo, y su flexibilidad para adaptarse a los cambios en el proceso de desarrollo, sin dejar de lado, la metodología XP que se utilizó para proponer el cómo llevar a cabo el proyecto, generando insumos de trabajo que satisfaga la calidad del producto.
- Finalmente, se desarrolló una matriz relacional para estandarizar los procesos de desarrollo de software en la empresa CloudStudio, basada en la metodología Scrum y XP identificada como la más adecuada para las necesidades de la empresa.

Etapa 3: Proponer un marco de trabajo ágil para el desarrollo de software basado en el estudio de XP y SCRUM que contenga procesos estandarizados y buenas prácticas de programación para incrementar soluciones óptimas de trabajo dentro de la empresa “CloudStudio Compañía Limitada”

Para lograr el objetivo de proponer un marco de trabajo ágil para el desarrollo de software basado en el estudio de XP y SCRUM que contenga procesos estandarizados y buenas prácticas de programación para incrementar soluciones óptimas de trabajo dentro de la empresa "CloudStudio Compañía Limitada", se llevaron a cabo las siguientes actividades:

- Se propuso un marco de trabajo basado en los principios de XP y SCRUM, que permitiera estandarizar las actividades del proceso de desarrollo de software en la empresa. Este marco de trabajo incluyó las principales etapas del proceso de desarrollo, tales como la planificación, el diseño, la codificación, la integración y las pruebas. Asimismo, se incluyeron buenas prácticas de programación, tales como el uso de pruebas unitarias, la integración continua y el código limpio.
- De inmediato, se aplicó el marco de trabajo propuesto en las etapas del proceso de desarrollo de la matriz de seguimiento de hallazgos de auditoría como caso práctico. Se realizaron evaluaciones continuas del proceso de desarrollo y se registraron las observaciones relevantes, a fin de identificar oportunidades de mejora y optimización.
- Finalmente, se refinó el marco de trabajo con las observaciones que se presentaron con el caso práctico. Se incorporaron mejoras en el proceso de desarrollo, tales como la definición de criterios claros de aceptación de los requisitos, la implementación de pruebas de aceptación automatizadas, y la integración de herramientas de seguimiento de problemas y tareas. Asimismo, se reforzaron las buenas prácticas de programación, tales como la utilización de patrones de diseño y la revisión continua del código fuente.

8. Conclusiones

- La aplicación de las metodologías Scrum y XP en el desarrollo de proyectos de software permite la introducción de un conjunto de buenas prácticas y métodos que mejoran el rendimiento del equipo de trabajo, permitiendo mostrar la evolución del producto de software de forma ordenada y de calidad. Estas metodologías son útiles en el desarrollo de software en empresas, siempre y cuando se apliquen adecuadamente y se adapten a las necesidades de la empresa.
- El estudio comparativo de las metodologías ágiles Scrum y XP permitió identificar sus similitudes y diferencias, así como las buenas prácticas y métodos que se pueden aplicar en el proceso de desarrollo de software. El conocimiento adquirido de las metodologías ágiles proporciona fundamentos teóricos relevantes para el desarrollo de software en empresas.
- La matriz relacional permitió identificar las similitudes entre Scrum y XP y seleccionar la metodología que mejor se ajusta a los requerimientos de la empresa CloudStudio Compañía Limitada. La utilización de la matriz relacional para estandarizar los procesos de desarrollo de software mejora la calidad y eficiencia del proceso.
- El marco de trabajo propuesto contiene procesos estandarizados y buenas prácticas de programación basados en el estudio de Scrum y XP, Scrum para conocer que se va a realizar y XP para determinar el cómo realizarlo. Este marco de trabajo ha permitido a CloudStudio Compañía Limitada incrementar soluciones óptimas de trabajo en el proceso de desarrollo de software, mejorar la calidad del producto final, tiempos de desarrollo y entrega.

9. Recomendaciones

Se recomienda que:

- Antes de seleccionar una metodología ágil, se analicen cuidadosamente los indicadores que contiene cada una de ellas. Esto permitirá la elección de la metodología más adecuada para la empresa, evitando pérdidas de tiempo, dinero y aumentará la satisfacción del equipo de trabajo al entregar un sistema robusto.
- Se identifique el proceso de desarrollo de software en cada empresa para poder implementar la metodología ágil que mejor se adapte a las situaciones específicas de la empresa. Esto ayudará a mejorar la calidad y eficiencia del proceso de desarrollo de software.
- Se ordenen las fases, tareas y entregas del proceso de desarrollo de software para estar preparados frente a los cambios que se puedan dar. Además, es recomendable utilizar un tablero de tareas para que el avance del proyecto sea visible para todos los miembros del equipo de trabajo. Esto permitirá una mejor gestión del proyecto y una mayor eficiencia en el proceso de desarrollo de software.

10. Bibliografía

- [1] E. - E. & B. School, «Enfoque Tradicional VS Enfoque Agile,» SCRUMMBI, Madrid, 2020.
- [2] k. Schwaber, «Agile Project Management,» Microsoft Press, 2004.
- [3] K. Schwaber y J. Sutherland, «Guía de Scrum,» La Guía Definitiva de Scrum: Las Reglas del Juego, Noviembre 2020. [En línea]. Available: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-European.pdf>. [Último acceso: 15 Enero 2023].
- [4] M. y. S. M. Alaimo, «Proyectos ágiles con Scrum: flexibilidad, aprendizaje, innovación y colaboración en contextos complejos,» Buenos Aires, Kleer, 2013.
- [5] R. Hat, «¿Qué es la metodología ágil?,» Red Hat, Inc., 2022 Julio 2022. [En línea]. Available: <https://www.redhat.com/es/devops/what-is-agile-methodology#resumen>. [Último acceso: 12 Enero 2023].
- [6] «Manifiesto por el Desarrollo Ágil de Software,» Ward Cunningham, 2001. [En línea]. Available: <https://agilemanifesto.org/iso/es/manifesto.html>. [Último acceso: 2023 Enero 12].
- [7] J. Highsmith, «Principios del Manifiesto Ágil,» Ward Cunningham, 2001. [En línea]. Available: <https://agilemanifesto.org/iso/es/principles.html>. [Último acceso: 12 Enero 2022].
- [8] V. H. Mercado Ramos, J. Zapata y Y. F. Cevallos, «Herramientas y buenas prácticas para el aseguramiento de calidad de software con metodologías ágiles,» Revista de Investigación Desarrollo e Innovación RIDI, 2015. [En línea]. Available: <https://dialnet.unirioja.es/servlet/articulo?codigo=6763102>. [Último acceso: 15 Diciembre 2022].
- [9] C. DRUMOND, «SCRUM - Guía de la metodología scrum: qué es, cómo funciona y cómo empezar,» Atlassian, [En línea]. Available: <https://www.atlassian.com/es/agile/scrum>. [Último acceso: 22 Enero 2023].
- [10] J. Sutherland, «Tecnología de objetos Jeff Sutherland,» [En línea]. Available: <http://jeffsutherland.com/>. [Último acceso: 15 Enero 2023].
- [11] R. S. Pressman, «Proceso XP,» de *Ingeniería del software - Un enfoque práctico*, México, MCGRAW-HILL INTERAMERICANA EDITORES, 2010, p. 47.

- [12] D. P. Pendidikan, «Conference on eXtreme Programming and Agile Processes in Software Engineering,» www.xp2004.org, [En línea]. Available: www.agileuniverse.com.
- [13] K. Beck, «Planning Extreme Programming,» ISBN 0201710919, 10 Noviembre 2014. [En línea]. Available: <http://wiki.c2.com/?ExtremeProgramming>.
- [14] E. Bahit, Scrum y Extreme Programming para programadores, Buenos Aires, Argentina: Safe Creative, 2012.
- [15] O. Castillo, D. Figueroa y H. Sevilla, «Fases de la Programación Extrema,» [En línea]. Available: <https://programacionextrema.tripod.com/fases.htm>. [Último acceso: 16 Enero 2023].
- [16] B. M, «Is Scrum and XP suitable for CSL development?,» de *Procedio Computer Science*, 2010, pp. 1511-1517.
- [17] M. A. Jackson, Principles of program design, ACADEMIC PRESS, 1975.
- [18] R. C. Martin, Agile Software Development, Principles, Patterns, and Practices, United States of America: Pearson Education Limited - www.EBooks.World.ir, Junio 2011.
- [19] P. Sanchez, «Consejos y Buenas Prácticas en Programación,» Facultad de Ciencias, Universidad de Cantabria, 22 Septiembre 2010. [En línea]. Available: <https://personales.unican.es/sanchezbp/teaching/faqs/programming.html>. [Último acceso: 16 Enero 2023].
- [20] E. Evans, «Domain Driven Design Reference. Definitions and Pattern Summaries,» 2015. [En línea].

11. Anexos

Anexo 1. Formulario de encuesta - Procesos de Desarrollo de software en CloudStudio Cía. Ltda.

Datos Referenciales:

Empresa: CloudStudio Cía. Ltda.

Objetivo: Obtener información acerca de los procesos de desarrollo de software

Instrucciones: Favor sea veraz en la respuesta, ya que los resultados serán de uso exclusivo para motivo de la investigación.

1. ¿Ha tenido problemas en el desarrollo de algún requerimiento de software?
SI () NO ()
2. ¿Cree Ud., que al desarrollar un requerimiento cada desarrollador lo realiza por su propia cuenta?
SI () NO ()
3. ¿Ha existido dificultades cuando un requerimiento que es prioritario para otro no se complete en el tiempo estimado?
SI () NO ()
4. Considera que se presentan retrasos en la entrega de requerimientos hacia el cliente porque no existen reuniones o presentaciones sobre los avances que se generan acerca del requerimiento
SI () NO ()
5. Número de desarrolladores designados para el desarrollo de un proyecto
1-4 () Mayor a 5 ()
6. ¿Cree Ud., que es necesario establecer un marco de trabajo general para la empresa CloudStudio Cía. Ltda., en la etapa de desarrollo de software?
SI () NO ()
7. ¿Cuál de las siguientes metodologías ágiles de desarrollo de software ha utilizado?
SCRUM () XP eXtreme Programmig ()
KANBAN () NINGUNA ()

Anexo 2. Formulario de encuesta – Aplicación del Marco de Trabajo Propuesto

Datos Referenciales:

Empresa: CloudStudio Cía. Ltda.

Objetivo: Obtener información acerca de la experiencia del equipo de trabajo al aplicar el marco de trabajo propuesto en la etapa de desarrollo de software

Instrucciones: Favor sea veraz en la respuesta, ya que los resultados serán de uso exclusivo para motivo de la investigación.

1. ¿Cree que es necesario llevar a cabo un control y planificación del proceso evolutivo del desarrollo de software enfocado en las tareas asignadas a cada miembro del equipo?
SI () NO ()
2. ¿Cree Ud., que al aplicar el marco de trabajo propuesto los equipos de trabajo se autoorganizan para cumplir con sus tareas en los tiempos definidos?
SI () NO ()
3. ¿El avance del proyecto o de las tareas del equipo esta visible para todos los involucrados?
SI () NO ()
4. Considera que las reuniones diarias permiten conocer el estado de las tareas de cada miembro del equipo y su compromiso con su finalización
SI () NO ()
5. Considera que la reunión de revisión con el cliente garantiza la continuidad del proyecto
SI () NO ()
6. Considera que en la reunión de retrospectiva, los miembros del equipo comparte la experiencia obtenida y esto permite mejorar las próximas iteraciones de los requerimientos
SI () NO ()
7. ¿Cuál cree Ud., que es la principal razón para aplicar un marco de trabajo en la etapa de desarrollo de software?
Mejorar la calidad del software ()
Terminar de manera inmediata un requerimiento ()
Entrega de calidad con trabajo colaborativo en tiempo definidos ()
NINGUNA ()

Anexo 3. Certificado de Resumen (Abstract) - Inglés

Riobamba, 24 de abril del 2023

Yo, **YESENIA CECILIA MERINO UQUILLAS** con cédula de identidad **0603819871**, en calidad de **LICENCIADA EN CIENCIAS DE LA EDUCACION PROFESORA DE IDIOMAS: INGLES** con número de registro SENESCYT 1019-2017-1841305,

CERTIFICO

Que en la ciudad de Riobamba, a los 24 días del mes de abril del dos mil veintitrés, se ha procedido a realizar la revisión de la traducción al idioma Inglés del apartado RESUMEN del Trabajo de Titulación denominado **Comparación de metodologías ágiles (XP y SCRUM) para la estandarización del proceso de desarrollo de software en la empresa “CloudStudio Compañía Limitada”**, previo a la obtención del título de **Magíster en Ingeniería en Software**, de la autoría de **Janneth Patricia Guamán Siguenza**, con cédula de identidad Nro. **0604056598**, pudiendo manifestar que se encuentra traducido en su integridad, manteniendo el mismo mensaje de su originalidad.

Es cuanto puedo certificar en honor a la verdad.



Escaneado digitalmente por:
**YESENIA CECILIA
MERINO UQUILLAS**

Yessenia Merino Uquillas

LICENCIADA EN CIENCIAS DE LA EDUCACION