Formulación metodológica ágil para el desarrollo de software de sistema de información para la empresa Inchcape Digital Delivery Center

Jesús David Arias Estupiñán

Trabajo de grado para optar el título de Magíster en Dirección y Gestión de Proyectos

Director

Juan Antonio Hernández Estrada

Doctor en Gerencia de Proyectos

Universidad Santo Tomás, Bucaramanga

División de Ingenierías y Arquitecturas

Maestría en Dirección y Gestión de Proyectos

2025

ELABORADO POR:	REVISADO POR:	V° B° POR:
_____ AUTOR *Estudiante de Maestría*	_____ Director *Director del Trabajo de grado*	Comité de Trabajo de Grado

Dedicatoria

Quisiera dedicar el desarrollo de este proyecto principalmente a mi familia quienes han sido parte de este proceso de formación de maestría, a los docentes de la facultad de Ingeniería de Telecomunicaciones, fuente de inspiración y quienes sentaron las bases de quién soy actualmente como persona y como profesional. A mis compañeros de trabajo en Inchcape Digital Delivery Center quienes también conocen este proceso y me han aportado su valiosa experiencia y conocimiento para el desarrollo de este proyecto.

Agradecimientos

Agradezco a mis familiares quienes han estado desde un principio en este camino de maestría, por su apoyo y acompañamiento en este estudio y desarrollo de proyecto. Docentes de la facultad de Ingeniería de Telecomunicaciones y en especial de la Maestría en Dirección y Gestión de Proyectos, quienes formaron parte también de este proceso y son parte fundamental en la adquisición de conceptos importantes y de gran valor para mi formación profesional como Magister en Dirección y Gestión de Proyectos.

Un agradecimiento especial al docente Juan Antonio Hernández, mi director de tesis de maestría, pues me acompañó desde el planteamiento de mi proyecto y me brindó cada una de las herramientas y caminos importantes para el desarrollo de este proyecto.

Gracias a todos quienes también de alguna u otra forma han hecho parte de este proceso de formación.

Contenido

Lista de tablas

Lista de figuras

Resumen

El presente trabajo se planteó atender el problema evidenciado en la empresa Inchcape Digital Delivery Center sobre la gestión de información de un inventario de tecnologías implementadas por la empresa, que permita identificar los sistemas tecnológicos desactualizados, incluyendo máquinas virtuales y recursos implementados en los servicios en la nube; en tal sentido se planteó como objetivo general "Desarrollar una metodología ágil para el desarrollo de software de un sistema de información que recolecte y gestione el inventario de tecnologías implementadas en la empresa, facilitando la planificación estratégica de mejoras y actualizaciones tecnológicas para aumentar la eficiencia operativa y competitividad en el mercado". Como método se planteó la investigación y selección de diversos componentes de marcos y metodologías ágiles existentes, obteniendo como resultado una metodología ágil híbrida permitiendo establecer en las discusiones que, el uso de componentes de diversas metodologías ágiles permite la construcción de una sola metodología que se acople a las necesidades de la empresa y el producto final.

Palabras clave: sistemas tecnológicos, eficiencia operativa, metodologías ágiles, marcos ágiles

Abstract

The present work aimed to address the problem evidenced in the company Inchcape Digital Delivery Center on the information management of an inventory of technologies implemented by the company, which allows the identification of outdated technological systems, including virtual machines and resources implemented in cloud services; in this sense, the general objective was "To develop an agile methodology for the development of software for an information system that collects and manages the inventory of technologies implemented in the company, facilitating the strategic planning of technological improvements and updates to increase operational efficiency and competitiveness in the market". As a method, the research and selection of various components of existing agile frameworks and methodologies was proposed, obtaining as a result a hybrid agile methodology allowing to establish in the discussions that the use of components of various agile methodologies allows the construction of a single methodology that fits the needs of the company and the final product.

Keywords: technological systems, operational efficiency, agile methodologies, agile frameworks

Glosario

Eficiencia Operativa: es la capacidad de una organización para optimizar sus recursos, procesos y tecnología de manera que se maximicen los resultados mientras se minimizan los costos, el tiempo y los desperdicios. Es clave para mejorar la competitividad.

Marco Ágil: es un conjunto de prácticas y principios que promueven la flexibilidad, la colaboración y la entrega continua de valor en el desarrollo de proyectos, especialmente en software. Se enfoca en adaptarse a los cambios rápidos y en satisfacer las necesidades del cliente de manera iterativa.

Metodología Ágil: es un enfoque de gestión de proyectos que enfatiza la colaboración entre equipos, la adaptación continua y la entrega incremental de productos o servicios. Las metodologías como Scrum y Kanban son ejemplos populares dentro del marco ágil.

Sistemas Tecnológicos: son el conjunto de herramientas, software, hardware y redes que una organización utiliza para gestionar y procesar información. Estos sistemas permiten automatizar procesos y mejorar la eficiencia en diferentes áreas del negocio.

Software: es un conjunto de instrucciones o programas que permiten ejecutar tareas en una computadora o dispositivo. Existen diferentes tipos de software, como el de sistema (que gestiona el hardware), de aplicación (para tareas específicas) y de desarrollo (para crear otros programas).

Introducción

En la era digital, los avances tecnológicos continuos son cruciales para la eficiencia y competitividad de las empresas, especialmente cuando se ofrecen servicios a terceros. Muchas empresas utilizan sistemas tecnológicos anticuados, como máquinas virtuales y servicios en la nube, lo que aumenta los costes operativos y las vulnerabilidades de seguridad. Ello repercute negativamente en la eficiencia operativa y la capacidad de respuesta del mercado. Un estudio de Gartner predice que para 2025, el 45% de las organizaciones en todo el mundo experimentarán ataques de software, dañando su marca y la confianza de los clientes, lo que conducirá a pérdidas financieras y daños a la reputación.

La calidad de los servicios se ve comprometida por la falta de actualización, lo que plantea riesgos para las relaciones comerciales y la satisfacción del cliente. En un mercado competitivo, las tecnologías emergentes como la inteligencia artificial y el análisis de datos en tiempo real requieren infraestructuras robustas y modernas. Inchcape Digital Delivery Center, una empresa de la industria automotriz, está comprometida a proporcionar soporte tecnológico y soluciones.

El desarrollo de una metodología de software flexible basada en componentes existentes y metodologías ágiles puede permitir la gestión eficiente del sistema de inventario tecnológico y proporcionar una base sólida para la mejora estratégica y las actualizaciones tecnológicas. El resultado es una mayor eficiencia en el trabajo. Por lo tanto, surge la pregunta: ¿qué metodología se adapta mejor a las condiciones y necesidades de Inchcape Digital Delivery Center para gestionar y actualizar la infraestructura tecnológica?

El objetivo general es desarrollar una metodología ágil para el desarrollo de software en Inchcape Digital Delivery Center, que permita la planificación estratégica de mejoras tecnológicas y actualizaciones para mejorar la eficiencia operativa y la competitividad. Lo anterior se pretende

desarrollar a través de la selección de componentes y metodologías del marco ágil existentes, el diseño de un proceso híbrido para el desarrollo de software, el establecimiento de mecanismos de control para la gestión de riesgos y el establecimiento de métricas para evaluar el progreso del proceso ágil, en consonancia con las necesidades y los plazos de la empresa. Esto ayudará a mejorar la eficiencia y competitividad operacionales.

En el presente documento se encontrará la investigación respectiva sobre cada una de las metodologías ágiles existentes más indicadas para el desarrollo de la metodología deseada, adicionalmente se presenta información detallada sobre los aspectos más importantes de cada una de las metodologías, los cuales han sido clave para el desarrollo de esta nueva metodología. Finalmente se presenta la metodología de diseño que se implementó, dando como resultados unos principios claves, roles y responsabilidades de esta nueva metodología, las prácticas ágiles que se emplearán, métricas para la evaluación del rendimiento y calidad de los proyectos a desarrollar con esta metodología y finalmente unas estrategias para la mejora continua y adaptación de la metodología.

1. Aspectos contextuales

En esta sección se aborda la problemática identificada en Inchcape Digital Delivery Center, relacionada con la gestión eficiente de su inventario tecnológico, dificultando la identificación de sistemas desactualizados, incluidas máquinas virtuales y recursos en la nube. Para resolverlo, se estableció como objetivo general desarrollar una metodología ágil para la creación de un sistema de información que permita recopilar y gestionar dicho inventario. Adicionalmente se presenta un breve marco institucional que permitirá tener un acercamiento a la empresa Inchcape Digital Delivery Center, conociendo más sobre su estructura organizacional, su misión, su visión, los servicios que ofrece y su distribución geográfica, lo cual ayuda en la identificación de las necesidades principales permitiendo un óptimo desarrollo del presente proyecto.

1.1 Planteamiento del problema

En la era digital actual, el avance tecnológico continuo es esencial para la eficacia corporativa y la competitividad, especialmente cuando se ofrecen servicios a terceros. Se ha evidenciado que muchas empresas hacen uso de sistemas tecnológicos desactualizados, incluyendo máquinas virtuales y recursos implementados en los servicios en la nube, así como sistemas que permitan la automatización de máquinas en las industrias; lo que conduce el aumento de los gastos operativos y vulnerabilidades de seguridad. Este problema repercute negativamente en la eficacia operacional y la capacidad de respuesta a las necesidades del mercado, observado en diversas industrias globales. De acuerdo con un estudio de Gartner, “para 2025, el 45% de las organizaciones de todo el mundo habrán sufrido ataques en el software de sus cadenas de suministro” (Moore, 2022), eventos que pueden dañar la marca de una organización y socavar la confianza del cliente, lo que lleva a pérdidas financieras y de reputación. La calidad de los servicios prestados se ve comprometida por la falta de actualización, lo que también pone en peligro los

vínculos comerciales y reduce la satisfacción del cliente. En un contexto actual donde las tecnologías emergentes, como la inteligencia artificial y el análisis de datos en tiempo real requieren infraestructuras sólidas y modernas, es imperativo que las empresas inviertan en mantener sus sistemas al día para no quedar rezagadas en un mercado competitivo.

Empresas del sector automotriz como lo es Inchcape Digital Delivery Center, que se dedican a brindar soporte tecnológico en la industria de la distribución automotriz, son empresas donde su fuerte a nivel tecnológico, con el soporte y desarrollo de soluciones de infraestructura en la nube, y no, el desarrollo de software de carácter adaptativo a las diferentes necesidades que pueda demandar cualquier otra compañía de cualquier otro sector productivo.

La formulación de una metodología basada en diversos componentes de marcos y metodologías ágiles ya existentes, que permita el desarrollo de un software para un sistema de información que recolecte el inventario de tecnologías implementados en la empresa Inchcape Digital Delivery Center, permitirá no solo la gestión eficiente del sistema de inventario tecnológico, sino que también proporcionará una base sólida para la planificación estratégicas de mejoras y actualizaciones tecnológicas en la empresa, otorgando así, mayor efectividad al desarrollo de un plan de trabajo.

Por lo tanto y teniendo en cuenta la importancia de generar una metodología ágil para el desarrollo de este software, se plantea la siguiente pregunta: ¿Cuál es la metodología que mejor se adapta a las condiciones y necesidades de la empresa Inchcape Digital Delivery Center para su gestión y actualización de infraestructura tecnológica?

1.2 Objetivos

1.2.1 Objetivo general

Desarrollar una metodología ágil para el desarrollo de software de un sistema de información que recolecte y gestione el inventario de tecnologías implementadas en la empresa Inchcape Digital Delivery Center, facilitando la planificación estratégica de mejoras y actualizaciones tecnológicas para aumentar la eficiencia operativa y competitividad.

1.2.2 Objetivos específicos

1. Seleccionar los componentes de marcos y metodologías ágiles existentes para formular la metodología apropiada a las necesidades de la empresa, mediante la investigación de diversas fuentes que proporcionen la información adecuada para cada marco investigado.
2. Diseñar el proceso ágil para la implementación del proyecto de Desarrollo del Software, integrando los diversos elementos obtenidos en la investigación de los marcos ágiles para la obtención de una metodología híbrida.
3. Establecer mecanismos de control que se implementarán en la metodología para el desarrollo del software que permitan tener una mayor gestión de los riesgos durante la ejecución del proyecto.
4. Fijar métricas que permitan evaluar el avance del cumplimiento del proceso ágil diseñado para el desarrollo de software, de acuerdo con las necesidades y tiempos previstos por la empresa.

1.3 Descripción institucional

Inchcape Digital Delivery Center (DDC) es parte de Inchcape PLC, líder mundial en distribución automática, que opera en más de 40 mercados. Los DDC se establecieron para proporcionar soluciones digitales, tecnológicas y de datos de vanguardia para respaldar las operaciones globales de Inchcape y sus socios fabricantes de equipos originales (OEM). Estos centros desempeñan un papel crucial en la remodelación de la industria automotriz a través de la innovación digital.

Inchcape lanzó sus Centros de Entrega Digital en 2020 con el objetivo de acelerar sus capacidades digitales y analíticas. En los últimos años, los DDC han crecido hasta incluir 1.400 empleados, lo que contribuye a la estrategia digital global y a la infraestructura de la empresa. Estos centros han ayudado a transformar el panorama digital del sector de la automoción mediante la creación de talento interno y la reducción de la necesidad de externalización (Communications, 2024).

Su misión es liderar la industria automotriz aprovechando la tecnología y los datos para brindar valor a los clientes y socios. Su visión es convertirse en un líder global en transformación digital para la industria automotriz, enfocándose en soluciones Unicanal y análisis de datos (Inchcape, 2024).

Inchcape Digital opera dos centros principales de entrega digital: uno en Bogotá, Colombia y el otro en Filipinas. Además, hay oficinas satélites en lugares como Cebú, Medellín y Santiago de Chile. Estos centros albergan diversos equipos, incluidos expertos en experiencia digital, análisis de datos y ciberseguridad. Los DDC sirven principalmente a las operaciones de Inchcape en varias regiones, incluidas las Américas, África y partes de Asia, lo que contribuye a la presencia de la compañía en más de 36 países (Communications, 2024).

Estos esfuerzos han consolidado la posición de Inchcape como actor global en la industria automotriz, impulsando la innovación digital y la eficiencia en todos sus mercados.

2. Marco referencial

2.1 Marco teórico

2.1.1 Metodologías Ágiles en el Desarrollo de Software

Las metodologías ágiles han ganado popularidad en el desarrollo de software debido a su enfoque en la flexibilidad y la adaptación a los cambios. Entre las metodologías más conocidas se encuentran SRUM, Kanban, XP (Extreme Programming), y Lean Software Development. Estas metodologías comparten principios comunes, como la entrega rápida de incrementos funcionales, la colaboración continua con el cliente, y la mejora constante del proceso.

El objetivo principal de estas metodologías es ofrecer soluciones que puedan adaptarse rápidamente a las necesidades cambiantes del cliente y del mercado. En el contexto de la empresa Inchcape Digital Delivery Center, la aplicación de un marco ágil permite una mayor adaptabilidad en la gestión del inventario de tecnologías, favoreciendo un entorno de mejora continua y actualización tecnológica (Schwaber J. S., 2020).

2.1.2 Componentes de los Marcos Ágiles

La selección de los componentes adecuados de los marcos ágiles es un paso esencial para adaptar una metodología a las necesidades específicas de una organización. SCRUM, por ejemplo, se enfoca en equipo autogestionados y ciclos de trabajo cortos (sprint), mientras que Kanban prioriza la visualización del flujo de trabajo para mejorar la eficiencia (Anderson, 2010).

La formulación de una metodología ágil a partir de metodologías existentes para Inchcape requerirá una combinación de estas prácticas, integrando la estructura de trabajo iterativa de SCRUM con la capacidad de Kanban para gestionar flujos y cuellos de botella. Esto permitirá que el sistema de información que recolecta y gestiona el inventario de tecnologías se desarrolle de manera ordenada y ajustable a las prioridades del proyecto (Dingsøyr, 2012).

2.1.3 Diseño de un Proceso Ágil para el Desarrollo de Software

El diseño de un proceso ágil para el desarrollo de software implica la integración de prácticas adaptadas a la realidad del proyecto y a los recursos disponibles. En este caso, se busca desarrollar una metodología que maximice la entrega de valor y minimice los tiempos de desarrollo, manteniendo la calidad del producto final (Highsmith, 2002).

Este proceso debe considerar elementos como la definición de historias de usuario, la planificación de iteraciones, y la realización de reuniones de revisión y retrospectiva para ajustar el enfoque del proyecto. Esto no solo facilita la entrega continua de valor, sino que también mejora la planificación estratégica de mejoras y actualizaciones tecnológicas en la empresa (Cohn, 2004).

2.1.4 Mecanismos de Control y Gestión de Riesgos

La gestión de riesgos es un aspecto fundamental en la ejecución de proyectos de software. En un entorno ágil, los mecanismos de control permiten identificar y mitigar riesgos de manera temprana, a través de la retroalimentación constante y la evaluación continua del progreso del proyecto (Boehm, 1988).

Entre los mecanismos más efectivos en el contexto ágil se encuentran las reuniones diarias de seguimientos y las revisiones al final de cada iteración. Estas prácticas permiten ajustar las

estrategias y adaptar el proceso a los nuevos riesgos identificados, mejorando la capacidad de respuesta del equipo (Conforto, 2016).

2.1.5 Métricas para Evaluar el Proceso Ágil

Las métricas en el desarrollo ágil son esenciales para evaluar el progreso del proyecto y asegurar que se cumplen los objetivos en el tiempo estipulado. Algunas de las métricas más utilizadas incluyen la velocidad del equipo, el lead time (tiempo de entrega), y la cantidad de trabajo en progreso (WIP, por sus siglas en inglés) (Kerzner, 2019).

Para el desarrollo del sistema de información en Inchcape Digital Delivery Center, se recomienda establecer métricas que reflejen tanto el avance de cada iteración como el cumplimiento de las historias de usuarios. Esto permitirá una evaluación objetiva del proceso y facilitará la toma de decisiones para ajustes y mejoras (Leffingwell, 2011).

2.1.6 Implementación de Metodologías Híbridas en proyectos de Tecnología

2.1.6.1 eXScrum – Una combinación de Scrum y Extreme Programming

El estudio “Empirical Evalution of the Proposed eXScrum Model” de (Qureshi, 2012) describe la implementación de eXScrum un híbrido que integra prácticas de Scrcum y XP en un equipo de 6 desarrolladores asignados a un proyecto de nómina. El marco conserva los componentes de Scrum (planificación, daily stand-ups, retrospectivas) y las prácticas de XP (TDD, programación por parejas e integración continua).

El proyecto se implementó en un contexto de proyecto industrial real para reemplazar varios sistemas legados de nómina. Integrando prácticas como Sprint Planning y daily stand-ups

para planificar y monitorear avances, programación en parejas para mejorar la calidad y compartir conocimiento en el desarrollo del proyecto, y adicionalmente se integró Test-driven development (TDD) e integración continua para asegurar robustez y detectar fallos prematuramente.

La integración de estas metodologías permitió la reducción significativa de defectos, la mejora en la calidad del código y la mitigación de las limitaciones separadas de Scrum (falta de ingeniería) y XP (falta de estructura). Con esta integración, se aprovechó la estructura de gestión de proyectos de Scrum y fortalece el rigor técnico de XP, equilibrando gobernanza y calidad.

2.1.6.2 Xcrum (Scrum + XP)

El artículo “Xcrum: A Synergistic Approach Integrating Extreme Programming with Scrum” de (Hosseini, 2023) presenta Xcrum una metodología híbrida que combina de forma integral prácticas de XP dentro del flujo de Scrum.

El objetivo principal de esta integración de metodologías se basó en mantener la agilidad de Scrum (flexibilidad, entregas iterativas) mientras se refuerzan las prácticas técnicas de XP. EN el proyecto se integraron elementos de Scrum como el planning, daily reviews y retrospectivas y las prácticas XP integradas, como TDD, pair programming integración continua y refactorización.

Esta integración trajo consigo el aumento de velocidad de entrega sin sacrificar la calidad del producto, mejorando la detección temprana de errores y la satisfacción del equipo y stakeholders debido a la solidez técnica y gestión estructurada.

2.2 Marco conceptual

2.2.1 Metodología Ágil

La metodología ágil es un enfoque de gestión de proyectos que se centra en la entrega incremental de valor mediante iteraciones cortas y frecuentes. Se caracteriza por su adaptabilidad al cambio, su énfasis en la colaboración entre equipo y el cliente, y la entrega continua de productos funcionales. En el contexto de desarrollo de software, permite responder a los cambios de manera eficiente, mejorando la calidad y reduciendo los tiempos de entrega (Highsmith, 2002).

2.2.2 Sistema de Información

Un sistema de información es un conjunto organizado de componentes que recopila, procesa, almacena y distribuye información para apoyar la toma de decisiones, la coordinación y el control en una organización (Stair, 2015). En el caso de la empresa Inchcape Digital Delivery Center, el sistema de información desarrollado tiene como objetivo gestionar de manera eficiente el inventario de tecnologías, facilitando la planificación estratégica y la toma de decisiones sobre actualizaciones y mejoras tecnológicas.

2.2.3 Inventario de Tecnologías

El inventario de tecnologías se refiere a la recopilación y registro de los activos tecnológicos disponibles en una organización, como software, hardware y redes. Este inventario permite una mejor gestión de los recursos tecnológicos, identificando necesidades de actualización o mejora, y asegurando un uso eficiente de la infraestructura tecnológica (Laudon, 2016). En el

proyecto, el sistema de información se enfocará en la gestión automatizada de este inventario para optimizar la eficiencia operativa de la empresa.

2.2.4 Gestión de Riesgos

La gestión de riesgos en proyectos de software es el proceso de identificación, análisis y respuesta a los riesgos que pueden afectar el desarrollo del proyecto ((PMI)., 2021). En un entorno ágil, la gestión de riesgos es continua y se integra en las iteraciones del proyecto, permitiendo ajustes rápidos en las estrategias de desarrollo. Esto ayuda a minimizar los impactos negativos y maximizar las oportunidades que puedan surgir durante la ejecución del proyecto.

2.2.5 Métricas en Metodologías Ágiles

Las métricas en el contexto de metodologías ágiles son herramientas para medir el rendimiento y el avance de un proyecto. Entre las métricas más comunes se encuentran la velocidad del equipo, que mide la cantidad de trabajo completado en cada iteración, el lead time, que indica el tiempo que transcurre desde que una tarea es planificada hasta que se completa (Kerzner, 2019). Estas métricas son fundamentales para evaluar el cumplimiento del proceso ágil y ajustar las estrategias en función de los resultados obtenidos.

2.2.6 Planificación Estratégica

La planificación estratégica es el proceso mediante el cual una organización define su dirección y toma decisiones sobre la asignación de sus recursos para alcanzar sus objetivos a largo plazo (Bryson, 2018). En el caso de la empresa Inchcape, la planificación estratégica de mejoras y

actualizaciones tecnológicas busca mantener la competitividad y la eficiencia operativa mediante el uso adecuado del inventario de tecnologías.

2.3 Estado del arte

El desarrollo de software se ha transformado gracias a la implementación de metodologías con enfoque ágil, proporcionando así flexibilidad, adaptabilidad y un proceso iterativo que mejora la entrega de valor. Si bien es cierto, el uso de estrategias convencionales aporta valor a estos desarrollos tecnológicos, se es necesario mencionar las deficiencias de estas estrategias, como el modelo en cascada, que resultó ineficaz en entornos con requisitos cambiantes (Beck, 2000). Es por esto, por lo que se da paso a las metodologías ágiles.

En la actualidad, se cuentan con diversas metodologías ágiles permitiendo a las compañías el desarrollo optimo de proyectos de software, agregando mejora de la calidad del producto, mayor satisfacción del cliente, mayor motivación de los trabajadores, trabajo colaborativo, uso de métricas más relevantes (Sotomayor, 2024); entre otros aspectos que resaltan su implementación frente a metodologías convencionales. Para el desarrollo de este proyecto se estudian metodologías que, según investigación propia, son las más utilizadas y proporcionan gran valor y excelentes prácticas para el desarrollo de nuestra metodología ágil; siendo estas, SCRUM, Extreme Programming (XP) y Lean Software Development.

2.3.1 SCRUM

Un marco de trabajo basado en iteraciones llamadas “sprints”, con roles definidos y eventos que fomentan la inspección y adaptación continua (Schwaber K. &., 2020)

Se distingue porque su estructura es basada en un desarrollo incremental. Esto significa que cada ciclo de desarrollo de un producto o servicio se divide en “pequeños proyectos” que se subdividen más a fondo en las fases de análisis, desarrollo y prueba. La entrega regular y parcial del producto terminado se denomina interacciones o aceleraciones de proceso durante el ciclo de desarrollo. Con esta metodología se pueden llevar a cabo proyectos complejos que requieren flexibilidad y rapidez para aplicar los resultados. Mediante reuniones periódicas para garantizar el cumplimiento de los objetivos establecidos, se tratará de gestionar y normalizar los errores que puedan surgir en procesos excesivamente largos.

El principal objetivo de la metodología es la realización de reuniones, que pueden ser de varios tipos: reuniones de planificación, diarias, de revisión y retrospectivas. Lo más importante es que el proyecto se completa tras un sprint para evaluar y sugerir formas de mejorar su desarrollo. La innovación, la adaptabilidad, la competitividad y la productividad son los pilares principales de Scrum (Sotomayor, 2024).

Henrik Kniberg, entrenador de Scrum@Scale, describe cómo los patrones clave del marco de trabajo de Scrum At Scale ayudaron para transformar la cultura de ingeniería en Spotify, donde los equipos trabajan en squads con procesos SCRUM adaptados a su cultura organizacional, logrando una entrega continua y revolucionando la industria del streaming musical (SCRUM@SCALE, s.f.).

La metodología incremental e iterativa de SCRUM organiza el trabajo en periodos breves conocidos como Sprints. Sus principales beneficios incluyen mejorar la rendición de cuentas y la transparencia del equipo mediante funciones y rituales bien definidos, así como facilitar la gestión del cambio mediante entregas frecuentes. Para que los equipos puedan auto gestionarse y completar tareas determinadas, SCRUM requiere un alto grado de madurez organizativa. Si no se

adapta adecuadamente, la cultura ágil puede ser vista como inflexible o burocrática en entornos donde no está solidificada. Este marco es perfecto para desarrollar productos dinámicos y complejos con requisitos que cambian rápidamente (Schwaber K. &., 2020).

2.3.2 Extreme Programming (XP)

Dado que su objetivo principal es mejorara las relaciones entre empleados y consumidores, esta tecnología es muy útil, especialmente para empresas en fase de creación o de consolidación. Mejorar las relaciones interpersonales a través del trabajo en equipo, la comunicación y la eliminación de tiempo de inactividad es el secreto del éxito de Extreme Programming XP (Sotomayor, 2024).

Revisando las fases que componen esta metodología, encontramos:

- Planificación del proyecto con el cliente
- Diseño del proyecto
- Codificación, donde los programadores trabajan en pareja para obtener resultados más eficientes y de calidad.
- Pruebas para comprobar que funcionan los códigos que se van implementando

La Corporación Chrysler emprendió un proyecto denominado Sistema de Remuneración Integral de Chrysler (comúnmente llamado C3) para sustituir múltiples aplicaciones de nómina por un solo sistema. Numerosos libros sobre el enfoque de Extreme Programming (XP) han hecho referencia a C3. De esta manera, Chrysler Corporation intentó sustituir muchos sistemas de nómina por uno solo mediante el proyecto Chrysler Comprehensive Compensation System (C3). Los métodos de desarrollo de software se utilizaron a lo largo de todo el proyecto y posteriormente se codificaron como Extreme Programming (XP).

Mediante el uso de técnicas como el desarrollo basado en pruebas (TDD), la integración continua, la programación de pares y la retroalimentación continua, XP pretende maximizar la calidad técnica del software. Su ventaja es que produce un código claro organizado y libre de errores en una fase temprana del proyecto, lo que facilita el aprendizaje rápido y la adaptación al cambio. Sin embargo, su puesta en práctica requiere mucha disciplina, habilidades técnicas altamente cualificadas y un trabajo de equipo continuo. Si una coordinación rigurosa, el escalamiento en equipos grandes también puede ser un desafío. Los equipo pequeños o medianos que trabajan en proyectos con mucha ambigüedad técnica o necesidades cambiantes son los más adecuados para XP (Beck, 2000).

2.3.3 Lean Software Development

El objetivo de la metodología Lean es mejorar continuamente los procedimientos y resultados de su equipo. Cuando se implementa correctamente esta metodología, ayuda a los equipos a proporcionar valor a los clientes de manera eficaz (Poppendieck, 2003).

La mejora continua y el respeto por las personas son los dos pilares de la metodología Lean que sirven como marco para todos los proyectos Lean (Atlassian, s.f.).

- *Mejora Continua:* los equipos pueden hacer ajustes graduales a las personas, productos y procesos para mejorar constantemente los sistemas con el apoyo de un ciclo continuo de retroalimentación. Mediante el proceso de identificación, evaluación y modificación gradual de los sistemas o procesos actuales, los equipos pueden reducir los desperdicios e incrementar la productividad general.
- *Respeto:* Los supervisores se toman esas opiniones e ideas muy en serio y valoran las contribuciones del equipo y la retroalimentación de los consumidores. Los gerentes Lean

asignan el trabajo a lo largo del flujo de trabajo de la manera más eficiente posible para promover un trabajo en equipo cercano y proporcionar a los consumidores el mayor valor.

Los conceptos del método Lean de Toyota se modifican para la industria del desarrollo de software mediante el uso del desarrollo de Lean Software. Esta estrategia tiene como objetivo agilizar los procesos, empoderar a los equipos, promover la toma de decisiones basada en pruebas y deshacerse de cualquier tarea que no proporcione valor. Sus beneficios incluyen un énfasis en la reducción de residuos y la eficiencia operativa lo que conduce a tiempos de entrega más rápidos y clientes más felices. Sin embargo, si la organización no es capaz de adoptar una cultura de cambio y mejora continua, su adopción puede resultar difícil. Los proyectos que requieren agilidad operativa y optimización de procesos, como los de sectores industriales o con numerosas relaciones interdepartamentales, son los más adecuados para Lean Software Development (Poppendieck, 2003).

2.4 Comparación de Metodologías Ágiles

Tabla 1. *Comparación de Metodologías Ágiles*

Metodología	Enfoque Principal	Ventajas Clave	Limitaciones	Contexto Ideal
SCRUM	Gestión de proyectos iterativa	Transparencia, colaboración, adaptación	Requiere disciplina	Proyectos con requisitos cambiantes
XP	Calidad del código y feedback rápido	Código limpio, pruebas constantes	Alta demanda técnica	Equipos pequeños, requisitos técnicos complejos

Lean	Eficiencia y eliminación de desperdicios	Flujo optimizado entrega rápida	Difícil sin cultura ágil	Proyectos con foco en eficiencia operativa

3. Metodología de Diseño

Digital.ai realizó una encuesta en 2020 en donde se demuestra la eficiencia de combinar metodologías ágiles: las organizaciones que adoptaron prácticas ágiles reportaron una mejora del 60% en la eficiencia del equipo y un 55% en la calidad del producto (Digital.ai, 2020).

Como ejemplo de cómo se pueden adaptar los marcos ágiles, una de las combinaciones más populares que se utilizan actualmente combina aspectos del enfoque eXtreme Programming (XP), el marco SCRUM y el método Kanban. SCRUM ofrece directrices para el uso de un equipo de desarrollo diverso, un Product Owner, un Scrum Máster y un Backlog relacionada con el producto. También cubre la planificación de sprint, el Daily Scrum, las revisiones de sprint y las sesiones retrospectivas de sprint.

Al mostrar el flujo de trabajo, hacer que los obstáculos sean evidentes y permitir la gestión del flujo mediante el ajuste del límite de trabajo en proceso, un tablero Kanban ayuda al equipo a ser aún más exitoso. La eficacia de los equipos ágiles se ve reforzada por técnicas de ingeniería inspiradas en XP, como pruebas automatizadas, refactorización, integración continua, uso de tarjetas (story cards) y desarrollo basado en pruebas (Project Management Institute, 2017). En conclusión, la combinación de técnicas de varias fuentes diferentes da como resultado un efecto sinérgico que supera a cada elemento por sí solo.

3.1 ¿Por qué combinar SCRUM, XP y Lean Software Development?

Integrar metodologías ágiles como SCRUM, Extreme Programming (XP) y Lean Software Development, se justifica con la sinergia de cada una de sus fortalezas propias de cada metodología que ayudan a abordar de manera integral los diversos desafíos presentes en un desarrollo de software. En esta sección se detallan los beneficios específicos de cada metodología que han sido estudiados para el desarrollo de esta tecnología, producto final del presente proyecto.

La combinación de SCRUM, XP y Lean permite crear una metodología balanceada que aborda tanto la planificación y organización del trabajo (SCRUM), como la calidad del producto (XP) y la eficiencia operacional (Lean). Esta sinergia es especialmente valiosa para el desarrollo de un sistema de gestión de actualizaciones donde los procesos deben ser ágiles confiables y sostenibles.

Al integrar lo mejor de cada enfoque, se logra una metodología que:

- Mantiene ciclos iterativos y predecibles.
- Asegura la calidad técnica del software desde etapas tempranas.
- Optimiza el flujo de trabajo y reduce desperdicios.

Este análisis comparativo fortalece el sustento teórico y práctico de la metodología propuesta, alineándola con las mejores prácticas y realidades del desarrollo ágil contemporáneo.

“El estudio independiente realizado en una muestra de más de 80000 proyectos ha demostrado que los equipos ágiles son, por término medio, un 25% más productivos que sus homólogos de la industria.” (Kremic, s.f.)

3.2 Aplicabilidad de la Metodología en Inchcape Digital Delivery Center

Debido a que responde inmediatamente a las necesidades de Inchcape Digital Delivery Center para la eficiencia operativa, calidad técnica y adaptabilidad, la combinación de SCRUM, XP y Lean Software Development es una solución muy eficaz y estratégica. Con el fin de facilitar la entrega iterativa de las características del sistema, una clara gestión de prioridades y una comunicación fluida entre los equipos técnicos y las partes interesadas, SCRUM organizará el trabajo en ciclos breves y regulados. Por su parte, XP ofrecerá técnicas como programación emparejada, pruebas automatizadas e integración continua que mejorarán en gran medida la calidad del software. Esto reducirá los fallos críticos de liberación y garantizará un progreso técnico sólido y fiable. Por último, pero no menos importante, Lean ayudará a racionalizar los procesos y a eliminar tareas inútiles, lo que conducirá a un uso más eficaz de los recursos y a una mayor flexibilidad para adaptarse a los cambios en el entorno tecnológico. La planificación estratégica de las actualizaciones depende la capacidad de la organización para gestionar eficazmente la información sobre el estado de los sistemas y la infraestructura, que se obtendrá con estos enfoques.

Considerando todo, este enfoque ágil será un potente instrumento para aumentar la competitividad tecnológica de Inchcape, reducir el riesgo operativo y permitir decisiones basadas en datos precisos y actuales. Esta elección responde a las necesidades específicas de Inchcape Digital Delivery Center, presentando así la elección de ciertos componentes puntuales de cada una de las metodologías estudiadas:

3.2.1 SCRUM

La integración de la metodología SCRUM proporciona una estructura organizativa más clara, producto de los ciclos de trabajo cortos denominados “sprints”, logrando así una facilidad en la planificación y adaptación continua.

Se adoptan roles como Product Owner y Scrum Máster, así como la planificación por sprints y las reuniones diarias. Estas prácticas son cruciales para mantener una coordinación fluida entre áreas técnicas, proyectos y operaciones, propias de un entorno empresarial global y complejo como el de Inchcape.

3.2.2 Extreme Programming (XP)

XP centra sus bases en prácticas técnicas rigurosas como la programación en parejas y el desarrollo guiado mediante pruebas, obteniendo así un alta en la calidad del código y logrando reducir posibles defectos en el desarrollo de este.

De XP se integran prácticas como la programación en parejas, la integración continua y el desarrollo guiado por pruebas (TDD), esenciales para garantizar un software confiable en entornos donde los errores pueden traducirse en pérdidas operativas importantes.

3.2.3 Lean Software Development

Lean se enfoca en la eliminación de actividades innecesarias y la optimización del flujo de trabajo, de esta manera se presenta una mejora en la eficiencia y reduce tiempos de entrega para el proyecto.

Aprovechando las cualidades de Lean se implementan principios como la eliminación de desperdicios la entrega de valor y la toma de decisiones basada en datos, ajustándose a la necesidad

de optimizar los procesos internos del centro digital y responder de forma eficiente a los cambios tecnológicos y de negocio.

3.3 Criterios de Diseño

Para el desarrollo de esta metodología, se definen una serie de criterios que ayudan a la implementación de conceptos y métodos claves y asertivos, permitiendo así una metodología adaptable a las necesidades de la empresa. A continuación, se exponen los criterios:

3.3.1 Flexibilidad y Adaptabilidad

La metodología debe permitir ajustes según las necesidades cambiantes del proyecto, garantizando así que la metodología pueda evolucionar con el tiempo.

3.3.2 Calidad del Software

La incorporación de pruebas automatizadas, integración continua y revisión de código que permita garantizar robustez y minimizar defectos en el producto final.

3.3.3 Eficiencia en la Gestión del Trabajo

La implementación de SCRUM permitirá estructurar el desarrollo en iteraciones manejables y garantizar la entrega incremental de valor.

3.3.4 Eliminación de Actividades Innecesarias

Aplicar principios Lean para optimizar tiempos, reducir tareas innecesarias y maximizar la entrega de valor.

3.3.5 Evaluación y Mejora Continua

Medir el rendimiento del proyecto con indicadores clave (KPIs) como lo pueden ser la velocidad del equipo, defectos detectados en el desarrollo y la satisfacción del usuario para optimizar el proceso progresivamente.

3.3.6 Colaboración y Comunicación Efectiva

Fomentar la interacción constante entre los miembros del equipo y los stakeholders a través de reuniones diarias, retrospectivas y revisiones periódicas.

3.3.7 Automatización de Procesos

Implementar herramientas que faciliten la gestión ágil, la integración y la entrega para mejorar la eficiencia operativa.

3.4 Proceso de Construcción

Posterior a la investigación y documentación sobre las metodologías ágiles óptimas para el desarrollo de esta metodología (SCRUM, XP y Lean Software Development), se llevaron a cabo las siguientes fases siguiendo un enfoque iterativo:

3.4.1 Análisis de Requerimientos

En esta fase se identificaron las necesidades específicas del desarrollo de software para la gestión de actualizaciones, asegurando que la metodología se adapte al contexto.1

3.4.2 Selección de Prácticas Ágiles

Se analizaron las mejores prácticas de cada metodología y se definieron aquellas que serían integradas en la nueva metodología.

3.4.3 Definición de Roles y Responsabilidades

Se establecieron roles específicos dentro del equipo de desarrollo, asegurando una clara distribución de tareas y responsabilidades.

3.4.4 Diseño de Procesos y Flujos de Trabajo

Se estructuraron los ciclos de desarrollo, definiendo cómo se integrarían las iteraciones SCRUM, las prácticas técnicas de XP y la optimización de Lean.

3.4.5 Definición de Métricas de Evaluación

Se establecieron indicadores clave de rendimiento (KPIs) para evaluar la efectividad de la metodología una vez implementada, incluyendo la cantidad de información recolectada y el estado de las versiones de los sistemas e infraestructuras.

4. Desarrollo de la nueva metodología

El desarrollo de esta nueva metodología ágil surge de la necesidad de contar con un enfoque optimizado para la gestión de actualizaciones de software y hardware en entornos tecnológicos dinámicos con lo presenta la empresa Inchcape Digital Delivery Center. Para ello, se combinan como se menciona en el inciso anterior, principios de SCRUM, Extreme Programming (XP) y Lean Software Development, con el objetivo de mejorar la organización, la calidad del software y

la eficiencia operativa. A continuación, se expone el proceso de construcción de la metodología y los criterios que guiaron su desarrollo.

El desarrollo de esta nueva metodología se llevó a cabo en varias etapas, basadas en análisis detallados de las metodologías existentes y en la adaptación de sus mejores prácticas a las necesidades específicas del proyecto.

4.1 Investigación y Revisión de Literatura

Inicialmente se contó con la revisión y análisis de las metodologías ágiles más utilizadas en la industria del software, evaluando sus beneficios y limitaciones. La literatura y bibliografía revisada nos presentó una amplia gama de libros, artículos científicos y estudios de casos que documentaban la aplicación de SCRUM, XP y Lean Software Development en distintos contextos.

4.2 Análisis de Requerimientos

Se fue necesaria la identificación de los principales desafíos en la gestión de actualizaciones de software y hardware, considerando aspectos como la trazabilidad de versiones, la automatización de procesos y la minimización de riesgos en cada actualización. De acuerdo con esto, se pudo identificar los objetivos principales y principios esenciales de la metodología.

4.3 Selección de Prácticas Ágiles

De acuerdo con los hallazgos y estudios de la revisión de literatura y análisis de requerimientos, se seleccionaron prácticas específicas de cada una de las metodologías ágiles mencionadas anteriormente:

4.3.1 SCRUM

- Organización del trabajo en sprints
- Reuniones diarias de seguimiento
- Revisiones periódicas y retrospectivas

4.3.2 XP

- Desarrollo basado en pruebas (TDD)
- Integración continua
- Programación en pares para mejorar la calidad del software

4.3.3 Lean Software Development

- Eliminación de actividades innecesarias en procesos.
- Optimización del flujo de trabajo
- Enfoque en la entrega de valor

4.4 Definición de Roles y Responsabilidades

Se establecieron los roles clave dentro del equipo de desarrollo, obteniendo como resultado los siguientes roles y sus responsabilidades:

- *Product Owner:* Define las prioridades del producto y representa a los interesados.
- *Scrum Máster:* Facilita los procesos ágiles y remueve impedimentos.
- *Development Team:* Encargado de la implementación técnica y aseguramientos de la calidad.
- *Tech Lead / Arquitecto:* Garantiza la integridad técnica del sistema.

- *Quality Assurance:* Encargado de la validación de calidad del software.
- *Risk Manager:* Se encarga de la identificación y mitigación de riesgos durante el desarrollo del software.

4.5 Diseño de Procesos y Flujos de Trabajo

La implementación de esta nueva metodología permitirá un modelo de trabajo iterativo que permite la entrega progresiva de valor, con ciclos de desarrollo bien definidos y métricas para la evaluación del desempeño del equipo. Este modelo está basado en los siguientes principios operativos:

- Planificación de sprint con objetivos definidos.
- Desarrollo incremental con revisiones constantes.
- Evaluación de cada entrega mediante métricas de rendimiento.
- Mejor continúa basada en el aprendizaje obtenido de cada iteración.

4.6 Definición de Métricas de Evaluación

Por último, para la medición de rendimiento y efectividad de la metodología, se definieron los siguientes indicadores clave de desempeño (KPIs):

4.6.1 Cantidad de Información Recolectada

Esto permitirá evaluar la capacidad del sistema para capturar datos relevantes sobre versiones y actualizaciones.

4.6.2 Frecuencia y calidad de actualizaciones

Este índice mide la estabilidad y efectividad de los cambios implementados.

4.6.3 Tiempo de Respuesta a Incidentes

Permite evaluar la agilidad del equipo en la resolución de problemas críticos.

5. Resultados

Esta sección está destinada a la presentación de los resultados del desarrollo de la nueva metodología ágil propuesta para el diseño e implementación de un software de gestión de actualizaciones. Se encontrará, en primer lugar, una descripción general de la metodología y sus fundamentos. Posteriormente, se detallarán cada uno de los *principios clave* que guían su aplicación, seguidos por la definición de *roles y responsabilidades* dentro del equipo de trabajo. Se explicarán también las *prácticas ágiles seleccionadas e implementadas*, especificando su utilidad y cómo contribuyen al cumplimiento de los objetivos del proyecto. Asimismo, se presentará el *modelo de evaluación del rendimiento y la calidad de software*, fundamentado en métricas clave que permiten monitorear el avance y detectar oportunidades de mejora. Finalmente, se expondrán las estrategias adoptadas para garantizar la *mejora continua y la adaptación progresiva* de la metodología en función del aprendizaje organizacional y los cambios en el entorno tecnológico.

Tras un proceso de investigación, análisis e identificación de aspectos importantes para el desarrollo de una óptima metodología que permita el desarrollo de un software de sistema de información, se obtiene una metodología que se fundamenta en los principios de SCRUM, Extreme Programming (XP) y Lean Software Development, que permitirá garantizar la eficiencia operativa,

organización clara, seguimiento adecuado, gestión de riesgos, evaluación con métrica y planeación estratégica.

5.1 Resultados obtenidos bajo la premisa del primer objetivo

5.1.1 Metodología Ágil Integrada para la Gestión de Actualizaciones Tecnológicas

Un marco ágil que combina componentes clave de SCUM, Extreme Programming (XP) y Lean Software Development para el desarrollo eficiente, ordenado y adaptable de software orientado a la gestión de actualizaciones de sistemas software y hardware. Esta metodología propone una estructura iterativa e incremental que favorece la planificación estratégica, la calidad técnica, la eliminación de desperdicios y la mejora continua.

Su propósito es optimizar la recolección, análisis y gestión de datos sobre versiones de infraestructura y software garantizando entregas constantes de valor, minimización de errores y capacidad de respuesta ante cambios tecnológicos. Está orientada especialmente a entornos corporativos como Inchcape Digital Delivery Center, que requieren soluciones ágiles pero robustas con un enfoque fuerte en calidad, eficiencia operativa, trazabilidad y adaptación continua.

5.1.2 Principios clave

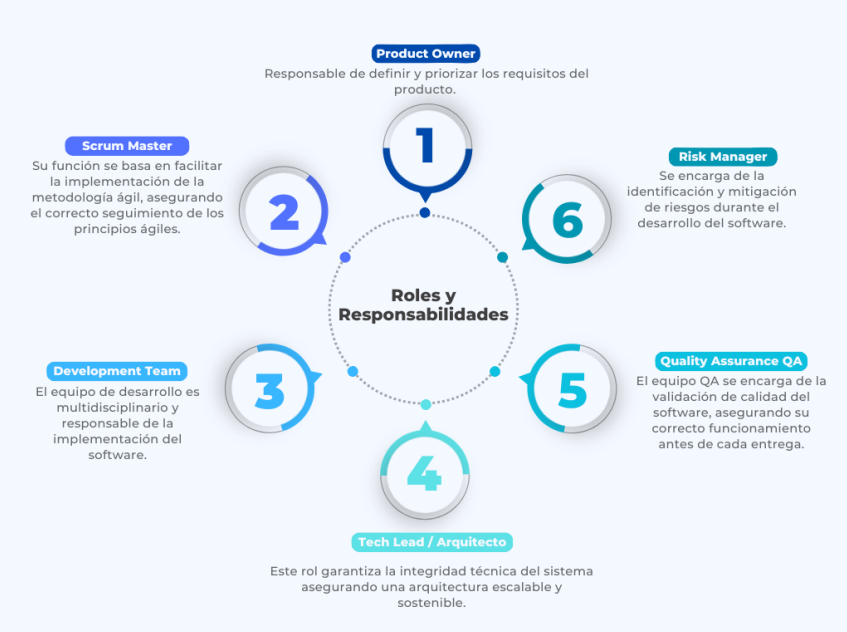
Una metodología ágil basada en los siguientes principios clave:

Figura 1. *Principios clave*

Entrega de Valor Continuo	• La metodología permitirá priorizar entregas incrementales que aporten valor a los usuarios.
Colaboración y Comunicación	• Mantener una comunicación fluida entre equipo, stakeholders y usuarios finales.
Adaptabilidad y Mejora Continua	• Ajustar procesos y prácticas con base en la retroalimentación y métricas.
Automatización y Calidad	• Integrar pruebas automatizadas y refactorización continua para garantizar calidad.
Minimización del Desperdicio	• Aplicar principios Lean para reducir tareas innecesarias y optimizar tiempos.
Gestión Proactiva de Riesgos	• Identificar y mitigar riesgos en cada fase del desarrollo.

5.1.3 Roles y Responsabilidades

Figura 2. *Roles y responsabilidades*



5.1.3.1 Product Owner:

El Product Owner es responsable de definir y priorizar los requisitos del producto, asegurando que las funcionalidades desarrolladas aporten valor al usuario final.

Dentro de sus funciones se encuentran:

- Recoger y gestionar requisitos de los stakeholders
- Priorizar el backlog del producto
- Aprobar entregables al final de cada sprint.

5.1.3.2 Scrum Máster / Agile Coach

El Scrum Máster o Agile Coach facilita la implementación de la metodología ágil, asegurando el correcto seguimiento de los principios ágiles.

Sus funciones incluyen:

- Eliminar impedimentos para el equipo.
- Facilitar reuniones ágiles (Daily Stand-up, Sprint Planning, Sprint Review, Retrospective).
- Fomentar la mejora continua.

5.1.3.3 Development Team

El equipo de Desarrollo es multidisciplinario y responsable de la implementación del software.

EL equipo presenta las siguientes funciones:

- Diseñar, desarrollar y realizar pruebas al software.
- Implementar integraciones y mejoras continuas.
- Mantener buenas prácticas de desarrollo.

5.1.3.4 Tech Lead / Arquitecto

Este rol garantiza la integridad técnica del sistema, asegurando una arquitectura escalable y sostenible.

El arquitecto tiene las siguientes funciones:

- Diseñar y supervisar la arquitectura del software.
- Brindar guía técnica al equipo de desarrollo.
- Evaluar tecnologías y herramientas.

5.1.3.5 Quality Assurance (QA)

El equipo de QA se encarga de la validación de calidad de software, asegurando su correcto funcionamiento antes de cada entrega.

Sus funciones incluyen:

- Diseñar e implementar pruebas automatizadas y manuales.
- Validar los entregables de cada sprint.
- Reportar defectos y asegurar su resolución.

5.1.3.6 Risk Manager

Este rol se encarga de la identificación y mitigación de riesgos durante el desarrollo del software.

El Risk Manager cuenta con las siguientes funciones:

- Identificar riesgos en cada fase del desarrollo.
- Implementar estrategias de mitigación.
- Evaluar la efectividad de las acciones preventivas.

5.2 Resultados obtenidos bajo la premisa del segundo objetivo

5.2.1 Prácticas Ágiles Implementadas

5.2.1.1 Sprints

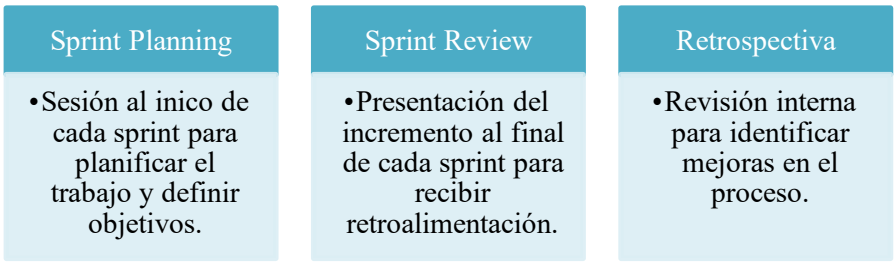
Esta práctica se basa en ciclos cortos de desarrollo (2-3 semanas) donde se entrega un incremento funcional del producto.

5.2.1.2 Daily Stand-up

Reuniones diarias de 15 minutos para revisar avances, impedimentos y coordinar el día.

5.2.1.3 Sprint Planning, Sprint Review y Retrospectivas

Figura 3. *Sprint Planning, Sprint Review y Retrospectivas*



5.2.1.4 Backlog Priorizado

Consiste en una lista ordenada de requisitos gestionada continuamente por el Product Owner.

5.2.1.5 Integración Continua

En esta práctica, cada cambio es integrado y validado frecuentemente para evitar acumulación de errores en el desarrollo del proyecto.

5.2.1.6 Desarrollo Basado en Pruebas (TDD)

Se basa en escribir pruebas antes del código para asegurar que cada función cumple con su objetivo.

5.2.1.7 Programación en Parejas

Dos desarrolladores trabajan juntos en el mismo código, mejorando la calidad y compartiendo conocimiento. Esto permite una mayor sinergia en el proceso de construcción del software.

5.2.1.8 Refactorización Continua

Esta práctica permite mejorar constantemente el código para mantenerlo limpio y sostenible.

5.2.1.9 Eliminación de Procesos Innecesarios

Evitar tareas innecesarias, documentación excesiva o procesos burocráticos, esto permitirá un proceso más lineal y limpio que da como resultado una entrega precisa y concreta con gran valor. Sin embargo, este proceso se aplica cuando así el desarrollo del proyecto lo demande, es importante tener presente que no siempre será necesario la eliminación de algún proceso o tarea.

5.2.1.10 Flujo de Valor

El equipo de proyecto se enfoca en priorizar las tareas que aportan valor directo al cliente.

5.2.1.11 Aprendizaje Constante

Adaptar el proceso constantemente según la retroalimentación y resultados. Esto permite una mejora gradual y en ocasiones exponencial, sobre los procesos desarrollados en la ejecución del proyecto y desarrollo del software.

5.3 Resultados obtenidos bajo la premisa del tercer objetivo

5.3.1 Evaluación del Rendimiento y Calidad

Para asegurar la eficiencia operativa y la calidad del software desarrollado, la metodología define un conjunto de métricas clave. Estas métricas no solo permiten evaluar el progreso y desempeño del equipo, sino que también proporcionan información relevante sobre el impacto del software en la gestión de actualizaciones. A continuación, en el siguiente punto se presenta una descripción más detallada de las métricas seleccionadas, su forma de cálculo y el valor que aportan.

5.3.1.1 Métricas definidas

Tabla 2. *Métricas definidas*

Métrica	Descripción	Forma de Cálculo	Utilidad
Cantidad de Información Recolectada	Mide cuántos datos sobre el estado actual de versiones de software y hardware han sido capturados por el sistema.	$\frac{Total\ de\ Registros}{Documentos\ creados\ durante\ cada\ sprint}$	Permite evaluar si el software cumple su función principal de capturar información clave para la gestión de actualizaciones.

Métrica	Descripción	Forma de Cálculo	Utilidad
Tiempo de Ciclo (Cycle Time)	Tiempo que transcurre desde que una tarea entra al backlog hasta que es completada y entregada.	$Fecha\ de\ Finalización - Fecha\ de\ Inicio\ de\ cada\ tarea$	Permite evaluar la agilidad operativa del equipo y detectar cuellos de botella.
Velocidad del Equipo	Cuántos Story Points completó el equipo durante cada sprint.	Suma de Story Points terminados al final de cada sprint.	Evalúa la capacidad de entrega del equipo y sirve para estimar tiempos futuros.
Tasa de Defectos	Número de defectos detectados por iteración.	$\frac{Defectos\ Reportados}{Funcionalidades\ entregadas\ por\ Sprint}$	Indica el nivel de calidad de software producido en cada ciclo.
Lead Time	Tiempo total desde que surge una necesidad de actualización hasta que se implementa en el software.	$Fecha\ de\ Implementación - Fecha\ de\ Solicitud$	Mide la eficiencia general del proceso y permite evaluar tiempos de respuesta.

5.3.1.2 ¿Cómo se halla cada métrica?

- *Cantidad de Información Recolectada:* Se obtiene directamente desde la base de datos del software desarrollado, registrando automáticamente cada nuevo dato agregado sobre versiones, sistemas y equipos tecnológicos.
- *Tiempo de Ciclo:* Se registra desde herramientas de gestión ágil (jira, Trello o similares) que permiten seguir el movimiento de cada ítem desde el estado “Pendiente” hasta “Completado”. Esta información se puede extraer al cierre de cada sprint.
- *Velocidad del Equipo:* Se calcula al final de cada sprint, sumando los Story Points completados. La velocidad se promedia cada tres sprints para identificar la tendencia.
- *Tasa de Defectos:* Se obtiene al registrar los defectos detectados por el equipo de QA y se relaciona con la cantidad de funcionalidades entregadas en el sprint.

- *Lead Time:* Se mide desde la solicitud inicial (que queda registrada en el backlog) hasta que la funcionalidad relacionada es puesta en producción o validad por el Product Owner.

5.3.1.3 Uso de gráficos y reportes

El seguimiento continuo de estas métricas se complementa con gráficos que permiten visualizar tendencias y anomalía, tales como:

- *Burndown Charts:* Para evaluar si el equipo completa el trabajo planeado dentro del sprint.
- *Gráficos de Tendencia de Defectos:* Para observar si la calidad mejora o empeora con el tiempo.
- *Histogramas de Lead Time:* Estos permiten identificar patrones y posibles bloqueos en el flujo de trabajo.
- *Gráficos de Velocidad:* Para evaluar si la productividad del equipo es estable, creciente o decreciente.

5.3.1.4 ¿Qué permiten identificar estas métricas?

- *Cumplimiento de Objetivos:* Identificar si la funcionalidad clave de recolectar y gestionar información de versiones es efectiva.
- *Rendimiento Operativo:* Si el equipo es capaz de entregar valor de forma constante y sin sobrecargas.
- *Calidad Técnica:* Si el software cumple estándares de calidad y estabilidad.
- *Identificación de Cuellos de Botella:* Si existen etapas del proceso que generan demoras recurrentes.

- *Evolución del Equipo:* Las métricas permitirán identificar si las prácticas ágiles y la mejora continua están teniendo un impacto positivo en la eficiencia y calidad del proyecto.

5.4 Resultados obtenidos bajo la premisa del cuarto objetivo

5.4.1 Mejora Continua y Adaptación de la Metodología

Para mantener la metodología alineada con las necesidades del equipo y del negocio, se proponen las siguientes estrategias de mejora continua:

5.4.1.1 Realización periódica de retrospectivas

Las retrospectivas son reuniones clave al final de cada sprint, donde el equipo reflexiona sobre el proceso seguido. Durante estas sesiones, se analizan:

- *¿Qué salió bien?* – Identificar prácticas que funcionaron correctamente y que deben mantenerse.
- *¿Qué salió mal?* – Detectar problemas, impedimentos o prácticas que no aportaron valor.
- *¿Qué mejora?* – Definir acciones concretas para optimizar el proceso en el siguiente sprint.

La retrospección permite que el equipo sea dueño de su proceso y fomenta una cultura de mejora incremental, clave para adaptarse a contextos cambiantes. Documentar y hacer seguimiento a las acciones propuestas asegura que las mejoras no se pierdan.

5.4.1.2 Monitoreo constante de KPIs y comparación de métricas a lo largo del tiempo.

Para saber si el proceso está mejorando o deteriorándose, es necesario medir y comparar indicaciones clave (KPIs) de forma constante. Los principales KPIs que se monitorean son:

- *Velocidad del equipo* – Evalúa la capacidad de entrega por sprint.
- *Tasa de defectos* – Mide la calidad del producto.
- *Lead Time* – Muestra la eficiencia del flujo de trabajo.
- *Cantidad de Información Recolectada* – Evalúa la funcionalidad clave del software.

Al graficar y comparar estos indicadores a lo largo del tiempo, se pueden identificar tendencias positivas o negativas, detectar desviaciones y tomar decisiones basadas en datos para ajustar la metodología.

5.4.1.3 Feedback continuo de usuarios y stakeholders

La metodología fomenta una relación cercana con los stakeholders y usuarios finales, quienes aportan un feedback continuo sobre las funcionalidades entregadas y la utilidad del producto. Este feedback se recoge durante:

- Revisiones de Sprint
- Entrevistas periódicas con stakeholders
- Encuestas breves para recoger impresiones del usuario.

El feedback realimenta el backlog y permite adaptar el producto y el proceso en función de las necesidades reales, no solo de supuestos. Estos aseguran que el desarrollo siga alineado con los objetivos estratégicos del negocio.

5.4.1.4 Mantenimiento de un backlog de mejoras

En paralelo al backlog de producto, se mantiene un backlog de mejoras para la propia metodología. En cada retrospectiva o revisión de métricas, se identifican acciones de mejora que se convierten en ítems de este backlog. Algunos ejemplos:

- Mejorar la definición de criterios de aceptación.
- Ajustar el tiempo de las reuniones diarias.
- Probar nuevas herramientas de integración continua.
- Actualizar plantillas de documentación.

Este backlog asegura que las mejoras detectadas se prioricen y ejecuten, evitando que queden en el olvido. De esta manera, la metodología evoluciona junto con el equipo.

5.4.1.5 Experimentación controlada con nuevas prácticas ágiles

Para mantener la metodología actualizada y adaptada a las nuevas necesidades, se fomenta la experimentación controlada. Esto implica:

- Seleccionar una nueva práctica ágil (por ejemplo, sesiones de refinamiento técnico o Mob Programming).
- Definir una hipótesis (por ejemplo: “Mejorará la calidad del backlog técnico”).
- Aplicarla durante un sprint y medir el impacto.
- Analizar los resultados en la retrospectiva y decidir si se adopta o descarta.

Este enfoque promueve la innovación continua dentro de la metodología, manteniéndola ágil no solo en el desarrollo de software, sino también en su propia evolución.

6. Conclusiones

I. En cumplimiento al primer objetivo específico del proyecto, se optó por la integración de SCRUM, Extreme Programming XP y Lean Software Development, las cuales proporcionan grandes prácticas ágiles, componentes y principios ágiles para el desarrollo de proyectos de software, como la entrega de valor continuo, la adaptabilidad y mejora continua, la gestión proactiva de riesgos, colaboración, comunicación, entre otros.

II. El cumplimiento del segundo objetivo permitió diseñar el proceso ágil gracias a la integración de los elementos seleccionados de cada una de las metodologías investigadas, logrando una sinergia en cada uno de sus componentes que permita obtener una estructura organizativa más clara, un alta en la calidad del código a desarrollar, mejora en la eficiencia y reducción de tiempos de entregas para el proyecto.

III. Dando cumplimiento al tercer objetico específico se encuentra que al integrar diferentes metodologías ágiles es posible optimizar los resultados incorporando el rol del Risk Manager, y su principal responsabilidad es la identificación y mitigación de riesgos durante el desarrollo del software y así mismo implementar estrategias de mitigación y evaluación de la efectividad de las acciones preventivas de riesgos.

IV. Por último, se asignaron métricas y parámetros como el tiempo de ciclo, cantidad de información recolectada, entre otras, que permitirán la identificación de aspectos como el rendimiento operativo, calidad técnica y evolución que presenta el equipo durante la implementación de la metodología desarrollada.

7. Recomendaciones

- Si bien el objetivo de este proyecto es el desarrollo de una metodología ágil para desarrollo software para la empresa Inchcape Digital Delivery Center, se espera tener un acercamiento con la compañía en un futuro para llevar a cabo la implementación de esta metodología en el desarrollo de un sistema de información que permita a la empresa la identificación de tecnologías desactualizadas y así tomar las debidas acciones para mitigar posibles problemas que genere el tener sistemas desactualizados.
- En adición, principalmente esta metodología se encuentra desarrollada para la implementación en proyecto de desarrollo de sistema de información, sin embargo, es imperativo mencionar que, gracias a las estrategias de mejora continua y adaptaciones establecidas para la metodología, se permite la implementación de la metodología en diversos proyectos de software en diferentes empresas, debido a su estructura, principios y desarrollo ágiles.

Referencias

(PMI)., P. M. (2021). *A Guide to the Project Management Body of Knowledge (PMBOK Guide). .* Project Management Institute.

Anderson, D. J. (2010). *Kanban: Successful Evolutionary Change for Your Technology Business.* Blue Hole Press.

Atlassian. (n.d.). *Metodología Lean*. Retrieved from Atlassian: https://www.atlassian.com/es/agile/project-management/lean-methodology

Beck, K. (2000). *Extreme Programming Explained: Embrace Change.* Addison-Wesley.

Boehm, B. (1988). A Spiral Model of Software Development and Enhancement. ACM SIGSOFT Software Engineering Notes.

Bryson, J. M. (2018). *Strategic Planning for Public and Nonprofit Organizations: A Guide to Strengthening and Sustaining Organizational Achievement.* John Wiley & Sons.

Cohn, M. (2004). *User Stories Applied: For Agile Software Development.* Addison-Wesley Professional.

Communications, I. E. (2024, Mayo 9). *Inchcape*. Retrieved from https://www.inchcapeea.com/2024/05/09/inchcapes-digital-delivery-model-is-reshaping-the-auto-industry/

Conforto, E. C. (2016). Agile project management and stage-gate model—A hybrid framework for technology-based companies. *Journal of Engineering and Technology Management*, 40, 1-14.

Digital.ai. (2020). *Digital.ai*. Retrieved from Digital.ai anuncia la 14.ª edición anual State of Agile Informes: https://digital.ai/es/press-releases/14th-annual-state-agile-report-shows-60-respondents-have-increased-speed-market-and/

Dingsøyr, T. N. (2012). A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software*, 85(6), 1213-1221.

Highsmith, J. (2002). *Agile Software Development Ecosystems.* Addison-Wesley.

Hosseini, S. (2023). *Xcrum: A Synergistic Approach Integrating Extreme Programming with Scrum.* Retrieved from Cornell University: https://arxiv.org/abs/2310.03248

Inchcape. (2024, Mayo 3). *Inchcape*. Retrieved from https://www.inchcape.com/americas-en/we-celebrate-3-years-of-digital-excellence-at-inchcape/

Kerzner, H. (2019). *Using the Project Management Maturity Model: Strategic Planning for Project Management.* Wiley.

Kremic, N. (n.d.). *DELTAMATRIX*. Retrieved from Why are Agile Teams 25% More Productive?: https://www.deltamatrix.com/why-are-agile-teams-25-more-productive/

Laudon, K. C. (2016). *Management Information Systems: Managing the Digital Firm.* Pearson.

Leffingwell, D. (2011). *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise.* Addison-Wesley Professional.

Mary Poppendieck, T. P. (2013). *Lean Software Development - An Agile Toolkit.* Crawfordsville, Indiana: Pearson Education Corporate Sales Division.

Moore, S. (2022, abril 13). *Gartner.* Retrieved from https://www.gartner.es/es/articulos/las-7-principales-tendencias-en-ciberseguridad-para-2022

Poppendieck, M. P. (2003). *Lean Software Development: An Agile Toolkit.* Addison-Wesley.

Project Management Institute, I. (2017). *GUÍA PRÁCTICA DE ÁGIL.* Pennsylvania: Project Management Institute, Inc.

Qureshi, M. R. (2012). Empirical Evaluation of the Proposed eXScrum Model: Results of a Case Study. *International Journal of Computer Science and Issues*.

Schwaber, J. S. (2020). *The 2020 Scrum Guide*. Retrieved from SCRUM GUIDES: https://scrumguides.org/

Schwaber, K. &. (2020). Retrieved from The Scrum Guide: scrum.org

SCRUM@SCALE. (n.d.). *Spotify: Un estudio de caso Scrum@Scale*. Retrieved from SCRUM@SCALE: https://www.scrumatscale.com/es/spotify-a-scrumscale-case-study

Sotomayor, S. G. (2024). *iebs*. Retrieved from Metodologías ágiles: ¿Qué son y cuáles son más utilizadas?: https://www.iebschool.com/blog/que-son-metodologias-agiles-agile-scrum/

Stair, R. &. (2015). *Principles of Information Systems.* Cengage Learning.