



ARTÍCULO ORIGINAL
INFORMÁTICA EMPRESARIAL

Innovación estratégica de proyectos de software en Desoft: un enfoque integrado con Odo
Strategic innovation in software project management at Desoft: an integrated approach with Odo

Dany Jesús Valiente Prieto¹ <https://orcid.org/0000-0002-1249-6133>
Jeffrey Blanco González^{2*} <https://orcid.org/0000-0001-7020-025X>
Edistio Yoel Verdecia Martínez¹ <https://orcid.org/0000-0003-2911-723X>
Dianelys Tejeda Villazón² <https://orcid.org/0000-0003-3689-1127>

¹Empresa de Aplicaciones Informáticas, *Desoft. La Habana, Cuba*

²*Universidad Tecnológica de La Habana José Antonio Echeverría. La Habana, Cuba*

Correo para la correspondencia: jblanco@ind.cujae.edu.cu

RESUMEN:

La presente investigación aborda la optimización de la gestión de proyectos de desarrollo de software en Desoft, integrando Odo como marco de trabajo principal. Centrándose en la naturaleza intelectual y colaborativa del desarrollo de software. El estudio propone una metodología que combina herramientas de gestión ágil, tradicionales y estrategias operativas efectivas. A través de la implementación y utilización de Odo en el desarrollo, se logra mejorar significativamente la coordinación, eficiencia y calidad en la entrega de proyectos. La metodología desarrollada proporciona un conjunto coherente de principios, elementos del ciclo de vida, roles involucrados, artefactos y herramientas tecnológicas, facilitando una gestión de proyectos más adaptativa y alineada con los objetivos estratégicos de Desoft. Los resultados del estudio demuestran cómo este enfoque conduce a una mejora notable en la eficacia de la gestión de proyectos, ofreciendo así lecciones valiosas y aplicables a contextos similares en la industria del software.

Palabras claves: desarrollo de software, gestión de proyectos, metodologías, Odo, dirección estratégica.

ABSTRACT:

This research addresses the optimization of software development project management at Desoft, integrating Odo as the main framework. Focusing on the intellectual and collaborative nature of software development, the study proposes a methodology that combines agile, traditional management tools, and effective operational strategies. Through the implementation and use of Odo in development, it is possible to significantly improve coordination, efficiency, and quality in project delivery. The methodology developed provides a coherent set of principles, lifecycle elements, involved roles, artifacts, and technological tools, facilitating more adaptive project

management aligned with Desoft's strategic objectives. The results of the study demonstrate how this approach leads to a significant improvement in project management effectiveness, thus offering valuable and applicable lessons to similar contexts in the software industry.

Keywords: software development, project management, methodologies, Odoos ERP, strategic management.

1

Introducción

En la dinámica actual de la industria del software, caracterizada por su rápida evolución y la creciente demanda de soluciones digitales eficientes, la gestión eficaz de proyectos de desarrollo de software se ha vuelto crucial [1]. Este campo, constantemente impulsado por innovaciones tecnológicas, requiere una adaptación ágil a las cambiantes necesidades del mercado. El Proceso de desarrollo de software (PDSW), siendo una actividad que combina intensamente la intelectualidad y la creatividad, se apoya en un capital humano altamente especializado en diversas tecnologías y herramientas. Esta complejidad inherente subraya la necesidad de una coordinación meticulosa y una gestión efectiva de proyectos para garantizar entregas de calidad y a tiempo [2,3]. Ante este escenario, las metodologías tradicionales de gestión de proyectos, como el modelo en cascada, han mostrado limitaciones, particularmente en contextos de requisitos cambiantes o poco claros [4,5,6]. Aquí emerge la relevancia de las metodologías ágiles, que ofrecen flexibilidad y una respuesta rápida a la volatilidad del entorno tecnológico y comercial [1,2,4,7,8,9,10,11]. Sin embargo, la implementación de estas metodologías no siempre es directa ni adecuada para todas las organizaciones o proyectos, es esencial hallar una metodología que, sin comprometer la calidad y la entrega oportuna, permita optimizar los costos de producción [12,13]. Esto contribuye a la rentabilidad en el competitivo dominio del software. Esta diversificación metodológica refleja una evolución y adaptación continuas en la gestión de proyectos de software, en respuesta a los desafíos y oportunidades emergentes en la industria del software [14,15,16,17].

En respuesta a los desafíos persistentes en la gestión de proyectos, han emergido marcos de trabajo como Odoos, que proporcionan un conjunto de herramientas y prácticas para gestionar de manera efectiva el desarrollo y la implementación de aplicaciones empresariales. Odoos, con su enfoque modular y escalable, facilita la gestión de proyectos de desarrollo de software, aunque también introduce complejidades que requieren una cuidadosa planificación y gestión [18]. Sin embargo, existe una brecha notable en la adaptación y aplicación efectiva de estas metodologías en contextos organizacionales concretos, particularmente en empresas tecnológicas como Desoft. A nivel nacional, Desoft se distingue por su capacidad para innovar en el desarrollo de software, resaltando la necesidad de adaptar las metodologías de gestión a las características únicas de cada organización y proyecto. Aunque Desoft ha implementado una variedad de tecnologías y enfoques en el desarrollo de aplicaciones reconocidas, tales como: ENERGUX, FASTOS, DM PLUS, Zoom LC Odoos y otras, enfrenta el reto de carecer de una metodología de gestión de proyectos propia que se ajuste plenamente a las exigencias particulares de sus productos y las expectativas de sus clientes.

Los retos identificados en Desoft abarcan desde la alta fluctuación del personal y la falta de procedimientos sólidos para la selección y capacitación de personal clave, hasta problemas en la planificación y gestión de proyectos del PDSW. Estas dificultades se reflejan en la imprecisa planificación y estimación de costos, la baja reutilización de código y la adaptación insuficiente de soluciones maduras, así como en una comunicación deficiente entre los equipos de desarrollo. El mantenimiento de aplicaciones y la gestión del conocimiento a lo largo del ciclo de desarrollo presentan deficiencias significativas, lo que resulta en pérdidas económicas y opiniones negativas de los *stakeholders* sobre la calidad de las soluciones desarrolladas.

Esta situación resalta la necesidad de abordar de manera integral los desafíos asociados a las metodologías de gestión de proyectos de desarrollo de software, particularmente en el uso de

Odoos como marco de trabajo. La adaptación y aplicación efectiva de metodologías adecuadas en contextos específicos como el de Desoft es un área poco explorada en la literatura existente. Por lo tanto, surge la necesidad de optimizar el proceso de gestión de proyectos de desarrollo de software en Desoft, considerando las peculiaridades de la empresa y el marco de trabajo de Odoos. El objetivo de este estudio es desarrollar una metodología para la gestión de proyectos de desarrollo de software en Desoft, empleando Odoos como marco de trabajo. Esta metodología busca abordar las deficiencias específicas de Desoft, mejorando la eficiencia, calidad y satisfacción del cliente. Dicha investigación se propone como una contribución significativa al cuerpo de conocimiento existente, ofreciendo perspectivas y recomendaciones prácticas para otras organizaciones que enfrentan desafíos similares. La vinculación con estándares reconocidos como PMBOK, Agile y Scrum y la comprensión de cómo estas prácticas se entrelazan con las metodologías específicas para el desarrollo de software, permitirá una visión integrada que promueva una gestión más eficiente y efectiva [19,20,21,22]. Se espera que los hallazgos del estudio no solo beneficien a Desoft, sino que también proporcionen una base para futuras investigaciones y mejoras prácticas en la gestión de proyectos de desarrollo de software, alineando eficazmente los procesos de gestión con los objetivos estratégicos y las demandas tanto a nivel organizacional como de la industria.

Métodos

Este estudio presenta una metodología para la gestión de proyectos de desarrollo de software en Desoft, con una integración estratégica del marco de trabajo Odoos. La metodología propuesta se diseñó siguiendo las directrices del:

- Modelo de Calidad de Desarrollo de Aplicaciones Informáticas (MCDAI), establecido por el Centro de Calidad de Software de Cuba (CALISOFT) y la Oficina Nacional de Normalización (ONN) [23]
- Decreto 359 de 2019 y la Resolución 124 de 2019 del Ministerio de Comunicaciones de Cuba [24,25]

Estas normativas proporcionan un marco regulatorio que asegura la calidad y la estandarización en el desarrollo de software.

Se define una metodología de desarrollo de software como un compendio sistemático de técnicas y métodos que facilitan la gestión coherente y transparente de todas las fases del ciclo de vida de un proyecto de software, ofreciendo una guía detallada y una visión integral del proceso de desarrollo. Este conjunto de prácticas y recomendaciones se apoya en una diversidad de modelos procesales genéricos, delineando artefactos, roles y actividades específicas [1].

El MCDAI esboza una estructura de niveles de capacidad y madurez organizacional destinados a normalizar y mejorar continuamente los procesos de desarrollo, sustentados en las mejores prácticas a nivel internacional que son adaptados a la realidad cubana [23]. Consta de cuatro categorías fundamentales (Ver Figura 1) que clasifican los procesos base para proporcionar una perspectiva clara y objetiva en su definición y ejecución:

1. Gestión de la Organización: Conjunto de procesos que impactan globalmente en la entidad, llevados a cabo en niveles superiores o bajo la responsabilidad directa de la alta dirección.
2. Gestión de Proyecto: Colección de procesos enfocados en la estructuración y supervisión del trabajo proyectado.
3. Ingeniería: Incorporación de procesos técnicos esenciales durante el proceso de creación de aplicaciones informáticas.
4. Soporte: Serie de procesos de apoyo esenciales que complementan el desarrollo de aplicaciones informáticas.

1	Gestión de la Organización	Gestión de Proyectos	2
---	----------------------------	----------------------	---

Gestión de Procesos de la Organización (GPO) Gestión de Portafolios (GPf) Gestión de Conocimiento (GCn) Gestión de Adquisiciones (GA)	Planificación, Monitoreo y Control de Proyectos (PMCP)
3	Ingeniería
Ingeniería de Requisitos (IR) Desarrollo de Soluciones Técnica (DST) Pruebas de Software	Gestión de Riesgos (GR)
4	Soporte
	Medición y Análisis (MA) Aseguramiento de la Calidad (AC) Gestión de la Configuración (GCf)

Fig. 1. Categorías que agrupan los procesos base del MCDAl.
Fuente [23].

Cada una de estas categorías está diseñada para colaborar sinérgicamente, asegurando que la gestión y el desarrollo de software en Desoft se ejecuten con la mayor eficiencia y calidad posible.

Modelo General de Procesos

La metodología se estructura en torno a un modelo general de procesos, abarcando desde la concepción hasta la finalización del proyecto (Ver Figura 2). Se adoptó un enfoque híbrido que combina las características de un modelo en cascada para etapas bien definidas y la flexibilidad de métodos ágiles para iteraciones incrementales en las fases de conceptualización, modelación, construcción, y pruebas. Este modelo facilita la adaptabilidad a los cambios y la gestión eficiente de los recursos del proyecto.

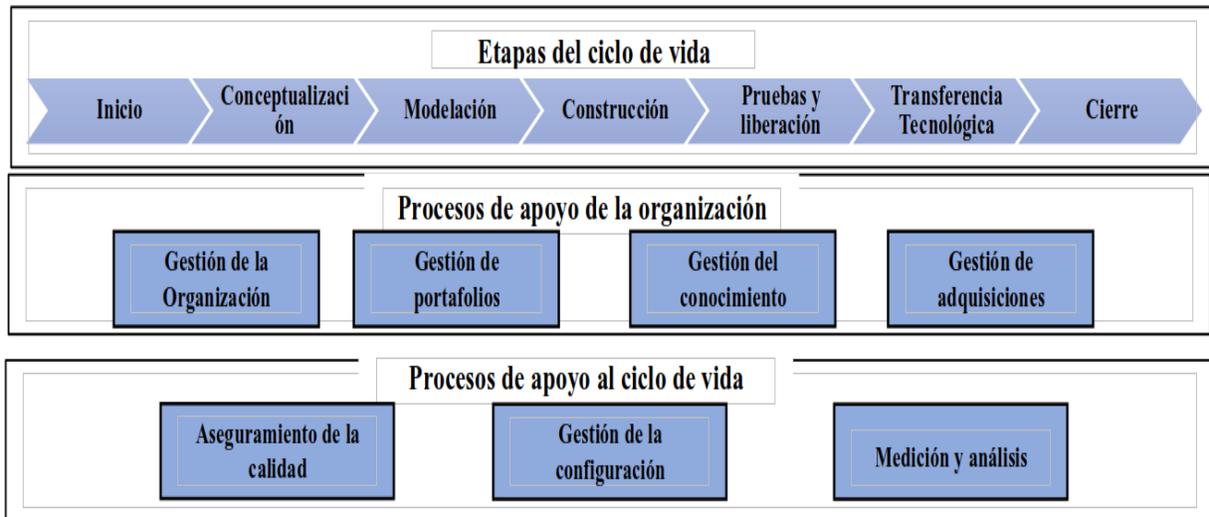


Fig. 2. Modelo de procesos general de gestión de proyectos de desarrollo de software.

Etapas del ciclo de vida

El ciclo de vida de desarrollo de software es un esquema que delinea el flujo secuencial y los criterios esenciales para la transición efectiva entre las distintas etapas del proceso de creación de software. Cada fase del ciclo de vida está diseñada para asegurar que todas las actividades y resultados necesarios se completen de manera sistemática antes de avanzar al siguiente conjunto de tareas. La estructuración clara de estas etapas y la definición precisa de los criterios de avance aseguran que el proceso de desarrollo de software sea manejable, predecible y de alta calidad, contribuyendo a la entrega exitosa del producto final.

Etapa 1: Inicio

Constituye la fase de planificación del proyecto, en la cual se efectúa un análisis exhaustivo de la organización cliente. Esta investigación preliminar es vital para adquirir una comprensión detallada del alcance proyectado, permitiendo así efectuar estimaciones precisas de tiempo, esfuerzo y costos asociados. En esta etapa se generan varios artefactos esenciales. El primero es el Acta de Constitución del Proyecto (ACP), un documento crucial firmado por Desoft y el cliente que establece la fecha de inicio y fin, presupuesto, esquema de ejecución y seguimiento del proyecto, así como los diferentes hitos y entregables por etapas. Para su elaboración, se utilizan herramientas como Odoos u Open Office. Se crea un Plan de Gestión de Proyectos (PGP), un documento interno que define el plan de recursos humanos del proyecto, el plan de involucrados, el plan de adquisiciones, el plan de iteraciones y el proceso de desarrollo. Se formula un Registro y Tratamiento de los Riesgos (RR), donde se identifican los riesgos asociados al proyecto, se planifican estrategias de mitigación y contingencia y se establece un proceso de monitoreo. Por último, se desarrolla un Cronograma General de Ejecución (CGE), firmado tanto por Desoft como por el cliente, que detalla las diferentes etapas del proyecto, incluyendo hitos, tareas, responsables y dependencias.

Etapa 2: Conceptualización

En esta etapa se llevan a cabo las actividades relacionadas con el modelado del negocio (en caso de que sea necesario), identificación a través de los proveedores seleccionados de los requisitos funcionales y los no funcionales. Utilizando técnicas y métricas se determina la prioridad y complejidad y se validan los requisitos funcionales. Se establece la trazabilidad bidireccional de los requisitos y se gestionan los riesgos asociados a los mismos. En esta etapa se elabora la Descripción de Procesos de Negocio (DPN), un documento conjunto entre Desoft y el cliente que describe los procesos de negocio identificados, utilizando Open Office y herramientas de modelado CASE. También se genera la Especificación de Requisitos de Software (ERS), donde se documentan los requisitos funcionales y no funcionales del sistema. A esto se suma la creación de un Prototipo No Funcional (PNF), un artefacto opcional que simula el comportamiento o flujo funcional del sistema a desarrollar, elaborado con herramientas como Axure. Se compone un Glosario de Términos (GT), un documento interno opcional que recopila términos relevantes asociados al dominio del proyecto. Finalmente, se desarrolla un Modelo Conceptual (MC) utilizando una herramienta de modelado CASE, que representa gráficamente el diseño conceptual del sistema.

Etapa 3: Modelación

En esta etapa de Modelación se llevan a cabo las actividades relacionadas con el diseño de la solución: arquitectura de sistema, arquitectura de datos, diseño visual de la aplicación y diseños de casos de prueba. Se definen las relaciones entre los principales elementos estructurales de la aplicación, sus datos y diseño visual. Actualmente, se elabora el Documento de Arquitectura (DA), un componente integral que recopila toda la información relevante sobre las vistas arquitectónicas del sistema y los patrones arquitectónicos utilizados. Para su desarrollo, se emplean herramientas como Odoos u Open Office, y herramientas de modelado CASE. Paralelamente, se trabaja en el Modelo de Diseño (MD), un documento que registra en detalle el modelo de diseño de la propuesta de solución. Este modelo se crea utilizando paquetes como Open Office y herramientas CASE de modelado, para asegurar una representación precisa y funcional del diseño. Se elaboran las Pautas de Diseño de Interfaces (PDI), un documento que recoge las pautas de diseño más significativas para la solución propuesta. Este documento es crucial para garantizar que el diseño de la interfaz sea intuitivo y esté alineado con las necesidades del usuario. Finalmente, se genera un conjunto de Documentos de Casos de Pruebas por cada uno de los requisitos (DCP_Requisito) para cada requisito funcional identificado. Estos documentos son esenciales para reflejar el diseño de casos de prueba que se realizan, asegurando que todas las funcionalidades y requisitos del sistema se evalúen exhaustivamente, facilitando la documentación y seguimiento sistemático de las pruebas.

Etapa 4: Construcción

En esta etapa de construcción se implementan los requisitos funcionales identificados teniendo en cuenta los elementos especificados en las etapas de conceptualización y modelación. Se realizan pruebas unitarias y funcionales por parte del equipo de proyecto en aras de verificar que las funcionalidades implementadas estén en correspondencia con los requisitos identificados y detectar el mayor número de errores antes de entrar el sistema en pruebas para su liberación y posterior aceptación con el cliente. Como flujo de trabajo, antes de iniciar un nuevo desarrollo usando Odoo como marco de trabajo, se debe revisar los módulos a fines con los nuevos requerimientos y de existir usar los desarrollados y almacenados en el repositorio de código de Desoft, hechos en la Empresa, luego los disponibles en la tienda de Odoo libres de costos y por último los disponibles en las diferentes comunidades y repositorios de códigos nacional e internacionales. Solo después de descartados estos espacios es que se debe desarrollar un módulo completamente desde cero.

Durante la etapa de construcción se desarrollan tres componentes cruciales para garantizar el éxito y la usabilidad del software. El Código Fuente (CF), en el cual se traducen de forma meticulosa los requisitos funcionales y no funcionales definidos en una solución de software operativa y eficiente. Para facilitar este desarrollo, se utilizan IDEs (Entornos de Desarrollo Integrados, por sus siglas en inglés) especializados y GITLAB, una plataforma que apoya la colaboración y la gestión de versiones del código, permitiendo un trabajo coordinado y eficiente entre los desarrolladores. Paralelamente, se elabora el Manual de Usuario (MU), un documento esencial que describe y representa visualmente las funcionalidades del software. Este manual es una guía para los usuarios del sistema, asegurando que puedan comprender y utilizar eficientemente todas las características del software. Para su creación, se emplean herramientas como Help&Manual u Open Office, lo que garantiza que la documentación sea clara, completa y accesible para los usuarios finales. Se prepara un Manual de Instalación y Configuración (MIC). Este documento es vital para guiar a los administradores del sistema en los pasos necesarios para la correcta instalación y configuración del software, de acuerdo con los requisitos no funcionales de hospedaje planteados por el cliente. La claridad y precisión de este manual son fundamentales para garantizar una instalación sin problemas y una configuración adecuada del software en producción.

Estos elementos de construcción son esenciales para asegurar que el software no solo cumpla con las especificaciones técnicas, sino que también sea accesible y manejable para los usuarios finales, facilitando así una adopción exitosa del producto.

Etapa 5: Prueba y liberación

En esta etapa se llevan a cabo las actividades relacionadas con la revisión final (Liberación) del producto o parte de él, así como sus artefactos, dígame manuales de usuarios y manual de instalación y configuración. En esta revisión el probador verificará que la aplicación cumpla con los requisitos descritos y con los casos de prueba entregados por el jefe de proyecto, a su vez verificará que el manual de usuario describa correctamente todas y cada una de las funcionalidades implementadas. El probador verificará, junto con el jefe de proyecto, que el manual de instalación y configuración detalle cada uno de los pasos a seguir.

En esta fase del proyecto de Desoft, se llevan a cabo varios procesos y se generan múltiples documentos clave para asegurar la calidad y funcionalidad del software desarrollado. El jefe de proyecto emite una Solicitud de Pruebas (SP), un documento crucial que inicia formalmente el proceso de evaluación de calidad del software por parte del equipo de Calidad de Desoft. Esta solicitud, preparada con herramientas como Open Office, marca el comienzo de la fase de pruebas. El líder del grupo de pruebas desarrolla el Plan de Pruebas (PP), un documento detallado que define los tipos de pruebas que se realizarán en el software según la solicitud recibida. Este plan incluye la evaluación de funcionalidades, rendimiento, seguridad, usabilidad, entre otros aspectos, así como un cronograma detallado de todas las actividades de prueba. Una vez que se completa la evaluación del software, el grupo de pruebas emite un Acta de Evaluación de Calidad (AEC). Este documento es esencial para documentar los resultados de las pruebas y decidir si se cierra esta etapa del proyecto. En casos donde una prueba específica necesita ser abortada, se elabora un

Dictamen (DEP) por parte del probador, este documento justifica y detalla las razones para la interrupción de estas.

Cada vez que se completa una iteración de pruebas, se emite un Acta de Revisión de Calidad (ARC). Esta acta documenta el progreso y los hallazgos de cada ciclo de prueba, asegurando un registro detallado del proceso de evaluación. Se identifican y registran todas las no conformidades encontradas durante las pruebas en el Registro de No Conformidades (NC), ya sean errores funcionales, no funcionales o discrepancias con respecto a la ERS. Para la gestión y seguimiento de estas no conformidades, se utilizan herramientas como Odoo, Mantis o Redmine. Este proceso juega un papel vital en garantizar que el software desarrollado cumpla con los estándares de calidad y funcionalidad, alineándose con las expectativas del cliente y necesidades del usuario final.

Etapas 6: Transferencia Tecnológica

En esta etapa se realiza la instalación, configuración, prueba y puesta en marcha del software en el entorno real del cliente. Se ejecutan pruebas de aceptación y pruebas piloto de ser requeridas. También se realiza la capacitación y el acompañamiento a los usuarios finales para asegurar que adquieran los conocimientos necesarios en el manejo del software.

Durante esta etapa, se crea un Registro de Solicitudes de Cambio (RSC), un documento vital que lista todas las solicitudes de cambios en el software hechas por el cliente. Este registro, elaborado con Open Office, es crucial para asegurar que todas las modificaciones solicitadas por el cliente sean consideradas y, si es necesario, implementadas adecuadamente. Un elemento clave en esta etapa es el Acta de Entrega y Aceptación del Producto (AEAP). Este documento se firma por Desoft y el cliente, y marca la entrega oficial del software, el manual de usuario, y el manual de instalación y configuración. Con la firma de esta acta, se cierran formalmente las pruebas de aceptación y/o piloto, confirmando que el cliente ha revisado y está satisfecho.

Etapas 7: Cierre

En esta etapa se oficializa la finalización de todos los compromisos pactados con el cliente mediante la firma del acta de cierre del proyecto. Se analizan los logros y las deficiencias presentadas durante todo el desarrollo del proyecto, se archiva el expediente del proyecto y se liberan los recursos para su posterior reasignación. Al finalizar el proyecto, se redacta y firma el Acta de Cierre (AC) por ambas partes involucradas. Este documento es esencial ya que reconoce oficialmente el cierre del proyecto. La firma del Acta de Cierre confirma que todos los objetivos del proyecto se han alcanzado satisfactoriamente y que todas las entregas han sido aceptadas por el cliente. Para la preparación de esta acta, se emplea Open Office.

Estos documentos son fundamentales en las últimas etapas del proyecto, asegurando no solo una transferencia efectiva del software al cliente, sino también una conclusión clara y formal del proyecto, con todos los objetivos cumplidos y confirmados por las partes.

Todos los artefactos generados durante el ciclo de vida de desarrollo se mantienen en el repositorio documental de Desoft en el formato original y se exportan en formatos accesibles (.jpg, .pdf) para su revisión y uso. Estos artefactos que se generen y deban ser firmados pudieran utilizar firma digital y conservarse de esa forma en los debidos repositorios digitales.

Dado que Odoo se desarrolla en la empresa de forma distribuida entre todas estructuras que la conforman se establecen las bases de trabajo y conservación de este conocimiento en Desoft en el Manual de Arquitectura de Odoo donde se establecen las estructuras de los repositorios de códigos y documentación, nomenclatura de los módulos desarrollados, ramas de codificación para los entornos de desarrollo, preproducción y producción, arquitectura de Odoo (multicapas, composición de los módulos, estructura de los módulos y manifiesto), entre otros conceptos de vital importancia para proyectos que usan como marco de trabajo Odoo.

Roles involucrados en la metodología

En la metodología diseñada para el proyecto de Desoft, se ha establecido una estructura de roles clave, cada uno aportando de manera única al éxito del proyecto. El **jefe de la actividad de desarrollo** juega un papel crítico en supervisar y dirigir todas las actividades de desarrollo de

software, asegurando el cumplimiento de los objetivos y la adhesión a las mejores prácticas. Paralelamente, el **jefe de Proyecto** es el líder del equipo, es quien propone y mantiene. Es el máximo responsable del proyecto. Hace de enlace con los clientes. Se encarga de coordinar y de garantizar las condiciones necesarias para el desarrollo del trabajo [1], garantizando que cada fase se ejecute según lo planificado. El **Especialista de Gestión de Proyectos (PMO)** se centra en los aspectos de la gestión de proyectos, ofreciendo soporte clave y asegurando que las fases del proyecto fluyan sin contratiempos.

En el ámbito comercial, el **Especialista Comercial** se encarga de la correcta relación contractual entre las partes. El **Analista**, por su parte, es el responsable de analizar los procesos de negocios, especificar los requisitos y diseñar la interfaz de usuario [26].

El **Arquitecto** define las responsabilidades, operaciones, atributos y relaciones de una o varias clases y determina cómo ajustarlas al ambiente de desarrollo. Es responsable de diseñar la base de datos definiendo las tablas, los índices, las vistas, las restricciones, los procedimientos o parámetros de almacenamiento, así como todo aquello que resulte necesario para almacenar, recuperar y borrar los objetos persistentes [26]; mientras que el **Diseñador gráfico** es el responsable de definir la identidad visual de la aplicación informática que se genere, teniendo en cuenta los requisitos funcionales, así como los principios de usabilidad y accesibilidad acorde a la identidad visual que se le define al producto. El **desarrollador**, es quien produce el código del programa y escribe las pruebas unitarias, es decir implementa componentes, localiza y corrige errores o defectos; y el **probador** es el responsable de la planificación, diseño, implementación y evaluación de las pruebas, que incluye generar el plan y el modelo de pruebas, implementar los procedimientos, evaluar la envergadura y resultados de las mismas y su efectividad, así como generar un resumen de la evaluación [26]. El **Administrador de la calidad** es el responsable de revisar los documentos de ERS y MU, el plan de administración del proyecto, el plan de testeo, la fase de diseño arquitectónico, la fase de diseño detallado, políticas de control de cambios, control de errores y control de la configuración, así como revisar la documentación generada. El **Administrador de la Configuración** maneja aspectos críticos como el control de versiones y los cambios en el código. El **Especialista de soporte** colabora con el equipo de desarrollo del proyecto, interactuando esencialmente con el arquitecto y el desarrollador [26], garantizando las herramientas y tecnologías necesarias para los entornos de desarrollo y preproducción.

El **Implantador** tiene la responsabilidad de consumir o ajustar la documentación técnica como MU y MIC. Se encarga de planificar la transición del producto a la comunidad de usuarios. Para ser efectivo debe trabajar estrechamente vinculado con el jefe del proyecto [26]. El **Cliente**, aunque no es un miembro interno, es un actor clave, cuyas necesidades y retroalimentaciones son cruciales para guiar el desarrollo y la entrega final del software, así como el **Consultor o experto funcional** que tiene un conocimiento específico de un área en concreto, ayuda a resolver un problema puntual [1].

La combinación de estos roles forma un equipo cohesivo y eficiente, esencial para el desarrollo exitoso del proyecto en Desoft.

Cada rol en el proyecto no solo aporta habilidades y conocimientos específicos, sino que también enriquece el proceso de desarrollo, asegurando que el producto final sea de alta calidad y cumpla con los objetivos y expectativas del cliente. Además, procesos de apoyo se llevan a cabo a lo largo de todo el ciclo de vida del proyecto, brindando una estructura y soporte adicionales que son esenciales para su éxito. Esta metodología integral y sus procesos de apoyo al ciclo de vida han demostrado ser efectivos en Desoft, asegurando que todos los aspectos del proyecto se manejen de manera eficiente y efectiva, desde su inicio hasta su cierre (ver Tabla 1).

Tabla 1. Artefactos, abreviaturas, herramientas y tecnologías de la metodología de los procesos de apoyo.

Procesos	Entregable o Artefactos Generados (Abreviatura)	Descripción	Roles	Herramientas y Tecnologías
Aseguramiento de la Calidad	Registro de Revisiones de Inconsistencias (RRI)	Documento elaborado al concluir cada etapa de desarrollo para identificar inconsistencias en los artefactos del expediente del proyecto.	Jefe de Proyectos y administrador de la calidad.	Open Office
	Lista de Chequeo de Revisiones Técnicas Formales (RTF)	Documento empleado para realizar revisiones técnicas detalladas del documento de Especificación de Requisitos.		
	Lista de Chequeo del	Documento usado para una revisión exhaustiva del producto desarrollado,		

	Producto (LCH_Prod)	asegurando su conformidad con los requisitos.		
Gestión de la Configuración	Acta de Solicitud de Cambio (ASC)	Documento emitido para formalizar las solicitudes de cambio en el proyecto, realizadas por el cliente.	Jefe de Proyectos, administrador de la configuración, administrador de la calidad, equipo de proyecto y comité de control de cambio.	Open Office
	Lista de Verificación para Auditorías de la Configuración (LVAC)	Documento preparado por el administrador de configuración para auditar la configuración del proyecto.		
	Reporte de Estado de la Configuración (REC)	Documento generado por el administrador de configuración para informar sobre el estado actual de la configuración del proyecto.		
Medición y Análisis	Guía de Indicadores para el Análisis y la Medición (GIAM)	Documento de soporte para el proceso de medición y análisis, configurado en la herramienta de gestión de proyectos.	Dirección de la empresa, administrador de la calidad, jefe de proyecto y equipo de proyecto.	Open Office, Odoo
	Indicadores del Proyecto (IP)	Documento que presenta los resultados del análisis de indicadores del proyecto, obtenidos manualmente o a través de la herramienta de gestión.		

En relación con la gestión específica del proyecto y la interacción con terceros, se establecen varios documentos clave:

1. Acta de Entrega y Aceptación (AE_A): Este documento se crea para cada artefacto entregado al cliente. Funciona como una confirmación formal de que el cliente ha recibido y aceptado cada artefacto.
2. Acta de Detención o Cancelación (AD_C): Se utiliza en situaciones donde el proyecto se detiene o cancela, ya sea por solicitud del cliente o por decisiones administrativas. Esta acta documenta oficialmente la pausa o finalización del proyecto.
3. Informe de Avance del Proyecto (IAP): Este informe se emite periódicamente para reflejar el estado actual del proyecto. Sirve como un registro detallado del monitoreo y control ejercido sobre el proyecto.
4. Minutas de Reunión (MR): Se redactan para documentar los acuerdos y decisiones tomadas tanto en reuniones con el cliente (MR_E) como en reuniones internas del proyecto (MR_I). Estas minutas son esenciales para mantener un registro claro de la comunicación y las decisiones a lo largo del proyecto.

Para una gestión documental eficiente, cada proyecto tiene su expediente organizado en una herramienta específica, como Gitlab o el gestor documental de Odoo. Esta organización facilita el acceso y la revisión de los documentos y asegura que toda la información relevante del proyecto esté centralizada y disponible para los interesados (Ver Tabla 2).

Tabla 2. Estructura del expediente de proyectos en el gestor documental.

Separador o carpetas	Artefactos	Etapa Ciclo de Desarrollo
	Nombre (Abreviatura)	
Planificación	Plan de gestión de proyectos (PGP)	Inicio y se actualiza durante la ejecución.
	Registro y tratamiento de los riesgos (RR)	Inicio y se actualiza durante la ejecución.
	Cronograma General de Ejecución	Inicio y se actualiza durante la

	(CGE)	ejecución.
Acuerdos con involucrados	Acta de constitución del proyecto (ACP)	Inicio
	Acta de entrega y aceptación (AE_A)	En todas las etapas.
	Acta de detención o cancelación (AD_C)	Cualquier etapa.
	Acta de cierre (AC)	Cierre
Reuniones	Informe de avance del proyecto (IAP)	En todas las etapas.
	Minuta de reunión (MR_I o MR_E)	En todas las etapas.

La Empresa, enfrentando el desafío de una alta rotación de personal y la imperiosa necesidad de preservar su capital intelectual, enfatiza la importancia de documentar exhaustivamente cada aspecto del proyecto. Esta documentación, abarcando tanto la gestión interna como la externa, es fundamental para asegurar la continuidad y la integridad del conocimiento dentro de la organización.

En este contexto, los Procesos de Apoyo de la Organización se han alineado estratégicamente con los delineados por el MCDAI. Esta sincronización busca fomentar una sinergia óptima con el modelo, consolidando los procesos que impactan directamente en toda la organización y que se ejecutan a un alto nivel, bajo la supervisión directa de la dirección. Esta integración incluye:

1. Actualización continua del plan estratégico de la organización, abarcando su misión, visión, valores, política, objetivos estratégicos y anuales, junto con el plan de acciones y recursos.
2. Monitoreo riguroso del cumplimiento del plan estratégico, adoptando medidas correctivas ante cualquier desviación.
3. Constante actualización de los modelos de negocio de la organización, ajustándose a la propuesta de valor para los clientes, segmento de mercado objetivo, estrategias de comunicación, sistemas de pago, recursos, costos, procesos, socios y proveedores.
4. Revisión y mantenimiento de los modelos de ciclo de vida para el desarrollo de productos de software, incluyendo sus etapas, procesos, resultados, interrelaciones y secuencias.
5. Definición clara de roles y responsabilidades, junto con la provisión de los recursos necesarios para su ejecución.

Para evaluar la eficacia de esta metodología, se emplearon técnicas de análisis estadístico, enfocadas en aspectos clave como:

- El rendimiento del proyecto, el cual se evaluó analizando en particular la gestión del alcance, del tiempo y de los recursos. Para ello, se realizó una revisión documental de los expedientes de proyectos de una muestra seleccionada. Los resultados mostraron que los indicadores de Índice de Cumplimiento del Alcance (ICA), Desviación del cronograma (DCr) e Índice de avance (IAv) fueron calificados como **buenos**. Los indicadores de Productividad (Pro) y Retención de talentos (RTa) fueron calificados como "regulares". Estos últimos dos indicadores se vieron afectados principalmente por la situación del entorno, no por la metodología en sí.
- La calidad del software se evaluó en función de las métricas asociadas a esta actividad en Desoft y se comparó con la media del mercado nacional, según estándares publicados por CALISOFT. Los resultados mostraron que el Índice de Calidad (ICa) fue evaluado como "bueno con mejoras".
- Las comunicaciones internas y externas, los riesgos, las partes interesadas, la organización y la planificación, así como la satisfacción del cliente, se evaluaron mediante encuestas a especialistas, clientes, jefes de proyectos y directivos vinculados a la actividad de desarrollo de software. Los resultados mostraron que todos los indicadores fueron calificados como: **buenos con mejoras**.

Los Indicadores Clave de Rendimiento (KPIs), extraídos a través de Odoos y otras herramientas analíticas, proporcionaron una visión valiosa sobre la eficiencia en el uso del tiempo, presupuesto y

cumplimiento de los objetivos del proyecto. Estos resultados y análisis forman parte integral del proceso de mejora continua y son esenciales para la optimización futura de la metodología.

Durante el ciclo de vida de los proyectos seleccionados, se realizaron revisiones periódicas y ajustes basados en los comentarios de las partes interesadas y en los resultados del análisis estadístico. Esto permitió una mejora continua del proceso y la adaptación a las necesidades cambiantes del proyecto y del cliente. La experiencia adquirida durante la implementación de esta metodología proporcionó valiosas lecciones sobre la gestión efectiva de proyectos de software. Se identificaron mejores prácticas, especialmente en la integración de metodologías ágiles dentro de un marco estructurado, y cómo esto puede mejorar la flexibilidad y la capacidad de respuesta a los cambios.

La retroalimentación obtenida de los usuarios finales y del equipo de desarrollo fue crucial para refinar la metodología. Esta investigación contribuye al campo de la ingeniería de software con un enfoque práctico y adaptable para la gestión de proyectos, especialmente en contextos donde se requiere equilibrar la estructura y la agilidad. La metodología propuesta fue validada en el contexto de DESOFT La Habana demostrando su efectividad en la mejora de la coordinación, eficiencia y entrega de proyectos de software.

La integración exitosa de Odoos como marco de trabajo y la adopción de un modelo híbrido de gestión de proyectos resalta su aplicabilidad en entornos de desarrollo de software similares. Los hallazgos y metodologías desarrolladas en este estudio tienen implicaciones significativas para futuras investigaciones, especialmente en la exploración de modelos híbridos de gestión de proyectos y la personalización de marcos de trabajo como Odoos para diferentes contextos organizacionales y para su extensión en el resto de las dependencias de Desoft en Cuba que realizan procesos similares.

Resultados

La evaluación de la metodología propuesta para la optimización de la gestión de proyectos en Desoft, utilizando Odoos como marco de trabajo, se llevó a cabo mediante un enfoque práctico y participativo. Se realizó una serie de sesiones de trabajo con un grupo diverso de especialistas de Desoft La Habana, incluyendo el subdirector de desarrollo, ocho jefes de proyectos, siete analistas, tres desarrolladores, dos diseñadores, dos arquitectos y cuatro probadores. La metodología fue socializada y evaluada a través de encuestas implementadas en el módulo de encuestas de Odoos. Los resultados obtenidos proporcionan una visión valiosa sobre la percepción de los involucrados en el proceso de desarrollo de software respecto a la nueva metodología.

La Figura 3 ilustra la distribución de las respuestas obtenidas en la encuesta. La gráfica muestra cómo los especialistas valoraron diferentes aspectos de la metodología, reflejando una tendencia general hacia valoraciones positivas.

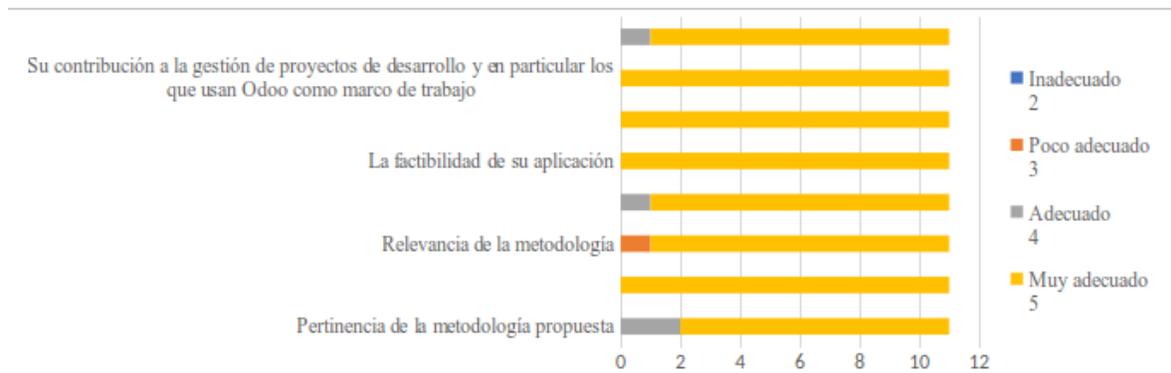


Fig. 3. Percepción de los involucrados en el proceso de desarrollo de software respecto a la nueva metodología.

Tabla 3. Evaluación cuantitativa general por cada aspecto.

No	Aspectos consultados	Puntuación promedio
1	Pertinencia de la metodología propuesta.	4.82
2	Correspondencia entre los principios, el alcance y las diferentes etapas de la metodología.	5.00
3	Relevancia de la metodología.	4.82
4	Lógica y coherencia entre las etapas y actividades a desarrollar.	4.91
5	La factibilidad de su aplicación.	5.00
6	Agilidad o simplificación para las diferentes etapas de la gestión del proceso de desarrollo.	5.00
7	Contribución a la gestión de proyectos de desarrollo que usan Odoos como marco de trabajo.	5.00
8	Posibilidad de sistematizar y generalizar la metodología en Desoft.	4.91
	Promedio	4.93

La Tabla 3 reafirma que la metodología propuesta fue evaluada de forma positiva por el grupo de expertos en gestión de proyectos de software. El promedio general de las calificaciones fue de 4.93 puntos sobre 5.00. Los aspectos mejor valorados fueron la correspondencia entre los principios, el alcance y las diferentes etapas (5.00 puntos), la factibilidad de su aplicación (5.00 puntos), la agilidad o simplificación para las diferentes etapas de la gestión del proceso de desarrollo (5.00 puntos) y su contribución a la gestión de proyectos de desarrollo que usan Odoos como marco de trabajo (5.00 puntos). Estos resultados indicaron que la metodología fue adecuada para el contexto de Desoft y que fue factible de aplicar en la práctica. Los resultados revelaron una alineación efectiva de la metodología propuesta con las necesidades y expectativas reales del entorno de desarrollo de Desoft. La adecuación entre los principios, el alcance y las diferentes etapas demostró que la metodología estaba bien estructurada y coherente. La viabilidad de su aplicación mostró que la metodología era realista y podía ser implementada con éxito en Desoft. La flexibilidad para las distintas etapas de la gestión del proceso de desarrollo evidenció que la metodología permitía adaptaciones según las necesidades particulares de cada proyecto. Finalmente, la contribución a la gestión de proyectos de desarrollo que utilizan Odoos como marco de trabajo señaló que la metodología era idónea para el contexto específico de desarrollo en Desoft. Los aspectos con menor calificación fueron la pertinencia (4.82 puntos) y la relevancia (4.82 puntos) de la metodología. Estos resultados indican que los expertos consideran que la metodología podría mejorarse en estos aspectos. Aunque la pertinencia y la relevancia de la metodología fueron bien valoradas, es importante centrar esfuerzos adicionales en asegurar una comprensión más profunda y una aplicación más efectiva de estos aspectos en proyectos futuros. Los resultados de la encuesta son fundamentales para la validación de la metodología propuesta y su implementación en Desoft. La retroalimentación positiva de los expertos no solo valida el enfoque adoptado, sino que también proporciona una base sólida para la mejora continua. Además, estos hallazgos son relevantes para el diseño de estrategias futuras de capacitación y desarrollo, asegurando que la metodología no solo sea adoptada, sino también optimizada y adaptada a las necesidades cambiantes de Desoft y del entorno del desarrollo de software.

Discusión

La implementación de la metodología híbrida en Desoft ha demostrado ser una solución efectiva, combinando las fortalezas de los enfoques tradicionales y ágiles. Mientras las metodologías tradicionales proporcionan una base sólida de documentación y estructura, clave para la gestión del conocimiento, las metodologías ágiles ofrecen la flexibilidad necesaria para responder a las cambiantes demandas del mercado. Esta dualidad ha permitido a Desoft no solo adaptarse a los retos actuales, sino también fomentar la innovación y el crecimiento en un entorno empresarial dinámico. La adopción de esta metodología híbrida responde a un enfoque estratégico, asegurando la alineación con los objetivos de la empresa y potenciando la soberanía tecnológica y la sostenibilidad económica.

La metodología implementada en Desoft se fundamenta en cinco pilares claves como la adaptabilidad, la excelencia en calidad, el trabajo colaborativo, un enfoque estratégico, y la promoción de la soberanía tecnológica. Estos principios son cruciales para orientar la implementación, asegurando que la metodología no solo sea pertinente y aplicable, sino también eficaz en el contexto específico de Desoft. La flexibilidad es un aspecto particularmente vital, posibilitando la personalización de la metodología para satisfacer las necesidades únicas de cada proyecto y adaptarse a las tecnologías emergentes, con un énfasis especial en la integración de Odoos como un marco de trabajo. Además, esta metodología está estrechamente alineada con la estrategia empresarial de Desoft, especialmente en respuesta a cambios en el modelo de negocio, como la transición hacia el Software como Servicio (SaaS) o el Todo como Servicio (XaaS). Esto implica una integración efectiva de la investigación, el desarrollo y la innovación, no solo para la creación de productos y servicios, sino también para su efectiva comercialización y distribución.

La implementación de la metodología se llevó a cabo en cuatro fases, comenzando con la definición y categorización de roles, seguida de la configuración de herramientas y capacitación, y culminando con la integración de la metodología en todos los aspectos de la gestión de proyectos. Cada fase ha contribuido significativamente a la mejora de los procesos de gestión de proyectos en Desoft, aumentando la eficiencia, la calidad y la satisfacción del cliente (Ver Fig. 4).

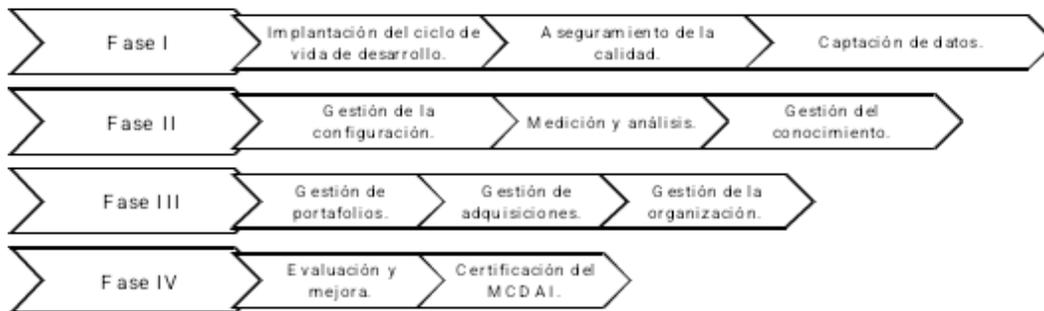


Fig. 4. Fases propuestas para la implementación de la metodología.

Durante la **Fase I** del proceso de implementación, se estableció el modelo de ciclo de vida del proyecto en Desoft. Antes de su implementación, se definieron y clasificaron los distintos roles organizacionales involucrados en el desarrollo de software. Además, se procedió al montaje y la configuración de las herramientas esenciales como Gitlab, Mantis, Axure, Odoos y Manual&Help entre otras. También se impartió formación a todos los participantes sobre la metodología y el uso de estas herramientas. Con el objetivo de asegurar la calidad, se adoptó un enfoque proactivo mediante la creación de listas de chequeo para las revisiones técnicas formales y para la evaluación del producto, utilizando el módulo de encuestas de Odoos. Además, se puso en marcha el módulo de gestión de proyectos de Odoos, iniciando la recopilación de datos.

En la **Fase II**, se inició la gestión de la configuración a través de Odoos, estableciendo listas de verificación para las auditorías de configuración. Se avanzó en la captación de datos y en la

elaboración de indicadores y tableros para la gestión de proyectos. En paralelo se comenzó a integrar el proceso de gestión del conocimiento dentro de la estructura de gestión de proyectos de software.

La **Fase III** se centró en la implantación del proceso de gestión de portafolios y en la evaluación integral de la gestión de proyectos de desarrollo. Se integraron los procesos de gestión de adquisiciones y de gestión organizacional en la dinámica de los proyectos de desarrollo de software. También, se solicitó a las entidades correspondientes la evaluación para la certificación del MCDAl.

Finalmente, en la **Fase IV**, se llevó a cabo una evaluación a nivel de dirección para determinar la efectividad de la metodología aplicada. Dado los resultados, se definieron mejoras y se estableció un cronograma para su implementación. Como corolario de este proceso, se aspira la certificación del MCDAl en Desoft La Habana a nivel básico, consolidando así las bases para una gestión de proyectos más estructurada y eficiente.

La integración de Odoos ha presentado desafíos únicos, principalmente relacionados con la personalización y adaptación del software a las necesidades específicas de Desoft. Sin embargo, estas dificultades han proporcionado oportunidades significativas para mejorar la gestión de proyectos y la toma de decisiones estratégicas. La experiencia adquirida ha sido invaluable para refinar la metodología y adaptarla a las necesidades cambiantes de Desoft y del sector del software en general.

Si bien los resultados han sido positivos, es importante reconocer las limitaciones de la investigación. La metodología se ha implementado en un contexto específico, y sería beneficioso explorar su aplicabilidad en diferentes tipos de proyectos de software y en otras organizaciones. Futuras investigaciones podrían centrarse en adaptar y probar la metodología en una variedad de entornos de desarrollo de software, así como en estudiar su impacto a largo plazo en la eficiencia y la innovación.

La discusión sobre la metodología implementada en Desoft subraya su relevancia y eficacia para afrontar los retos actuales en la gestión de proyectos de desarrollo de software. Esta metodología no solo responde a las necesidades inmediatas de la empresa, sino que también establece un camino hacia una mayor adaptabilidad, calidad y eficiencia en el futuro. La experiencia de Desoft proporciona buenas lecciones para otras organizaciones en la industria del software, destacando la importancia de una metodología que equilibre estructura y flexibilidad, y que esté alineada con la visión estratégica y los objetivos a largo plazo de la empresa.

Conclusiones

1. La investigación confirmó que la incorporación de la gestión de la calidad y la promoción del trabajo en equipo han mejorado significativamente la eficiencia y la entrega de proyectos. Estos elementos son cruciales para alcanzar un alto estándar en el producto final y han resultado en una mejor satisfacción del cliente y en la recepción positiva del equipo interno.
2. La metodología implementada en Desoft ha evidenciado una alta adaptabilidad y coherencia con las necesidades del entorno de desarrollo de software. La personalización, que integra prácticas ágiles dentro de un marco estructurado, se alinea efectivamente con los objetivos estratégicos de la empresa, permitiendo una gestión de proyectos más flexible y responsive a las tendencias tecnológicas actuales, particularmente con la adopción de Odoos como marco de trabajo.
3. La adopción de la metodología ha reforzado la soberanía tecnológica y contribuido al crecimiento económico en el contexto de Desoft, en concordancia con los movimientos estratégicos hacia modelos de negocio innovadores como SaaS y XaaS. Este enfoque

estratégico asegura que Desoft no solo se mantiene competitivo, sino que también avanza en la vanguardia tecnológica y económica del país.

4. La metodología, validada por la valoración positiva de los especialistas, se presenta como un modelo replicable y escalable. Las fases de implementación han demostrado ser efectivas para la mejora continua y la certificación de procesos, sugiriendo un camino claro para la generalización de estas prácticas en Desoft y otras organizaciones similares. Los hallazgos proporcionan un marco sólido para futuras investigaciones y desarrollos prácticos en la gestión de proyectos de software.

Referencias

1. MAIDA, E.G.P., JULIÁN. "Metodologías de desarrollo de software". 2015, Universidad Católica Argentina: Argentina", en línea en <https://repositorio.uca.edu.ar/handle/123456789/522>.
2. SÁENZ-BLANCO, F.G.-S., Frederick & Ramos-Rivera, Julián C, Establishment of agile teams for software development: review of literature. *Dimensión Empresarial*, 2017. 16(2): p. 39-54.
3. Vila Grau, J.L., AGILE PROJECT MANAGEMENT ACCORDING TO THE PRINCE2 AND PMBOK MODELS. 2021.
4. MOLINA MONTERO, B., H. VITE CEVALLOS, AND J. DÁVILA CUESTA, Agile methodologies against traditional methods in the software development process. *Espirales*, 2018.
5. VELÁSQUEZ RESTREPO, S.M., ET AL., A comparative review about traditional and modern software development methodologies. *Revista CINTEX*, ISSN: 2422-2208, 2019. 24(2): p. 13-23.
6. PRESSMAN, R.S., *Software Engineering: A Practitioner's Approach* (7a. edición ed.). New York: McGraw-Hill Education. 2013.
7. SANDRA MILENA VELÁSQUEZ RESTREPO, J.D.V.-M., Marta Ester Gómez-Adasme, Ana Alexandra Pino – Martínez, Erika Julieta Restrepo-Zapata & Sebastián Londoño-Marín., Una revisión comparativa de la literatura acerca de metodologías tradicionales y modernas de desarrollo de software. *Revista CINTEX*, 2019. 24(2): p. 13-23.
8. Pieterse, B.A., Kofi;Manu, Patrick;Mohandes, Saeed Reza;Cheung, Clara;Yunusa-Kaltungo, Akilu Less. *Examining critical project management skills for successful delivery of major maintenance projects: insights from the United Kingdom energy sector*. 2022 [cited Engineering, construction, and architectural management; Available from: <https://www.emerald.com/insight/content/doi/10.1108/ECAM-07-2022-0658/full/html>
9. DINGSOEYR, T., Falessi , D., & Power, K. Agile Development at Scale: The Next Frontier. *IEEE Software*, 36(2), 30-38, 2019.
10. JOHANNA PATRICIA ZUMBA GAMBOA, C.A.L.A., Evolución de las Metodologías y Modelos utilizados en el Desarrollo de Software. *INNOVA Research Journal - Revista mensual de la UIDE extensión Guayaquil*, 2018, ISSN: 2477-9024 Vol. 3(No. 10): p. 20-33.
11. Játiva, L.M.E.J.X., Evolución de las Metodologías de Desarrollo de la Ingeniería de Software en el Proceso la Ingeniería de Sistemas Software y Determinación de una metodología adaptable orientada a una organización pequeña. *Repositorio de la Universidad de las Fuerzas Armadas ESPE*, 2008.
12. González González, F., S.L. Calero Castañeda, and D.F. Loaiza Buitrago, Comparison of waterfall and agile methodologies for increasing productivity in software development. *Santiago de Cali University*, 2019.
13. Pardini, B., *Entrevista VP Product & Technology*. Cali., 2018.
14. Dybá, T. and T. Dingsoyr, Empirical studies of agile software development: A systematic review. *Information and software technology*, 2008. 50(9-10): p. 833-859.
15. Florentine, S. 7 agile certifications to take your career to the next level[En línea] Disponible en: <http://www.cio.com>, 2017.
16. Manjul G, Joey F., Weidong X Relationships between IT department culture and agile software development practices: An empirical investigation. *International Journal of Information Management* 44, 2019

17. Lisdania de la Caridad Delgado Olivera, L.M.D.A., Modelos de Desarrollo de Software. Revista Cubana de Ciencias Informáticas, 2020. Vol. 15, No. 1, enero-marzo, 2021: p. Pág. 37-51, ISSN 2227-1899.
18. Pons, P.S., DESARROLLO DE UNA APLICACIÓN PARA ODOO ERP 2020, Universitat Politècnica de València: Valencia.
19. Hirsch, M., Moving from a Plan Driven Culture to Agile Development. ZQhIke Engineering AG Wiesenstrasse 10a CH-8952 Schlieren, 2018.
20. S. Vélez, J.A.Z.y.A.H., Gestión de Proyectos: origen, instituciones, metodologías, estándares y certificaciones. Entre Ciencia e Ingeniería, 2018. vol. 12, no. 24: p. 68 - 76.
21. Reyes, J.M.C., Análisis e implementación del Sistema de Planificación de Recursos Empresariales ERP Odoos empleando una metodología ágil de desarrollo de software, mediante el caso de estudio de Control Escolar del Centro de Enseñanzas del LANIA. 2020, Instituto Tecnológico Superior de Martínez de la Torre.
22. Luis Miguel Egas, J.X.J., Evolución de las Metodologías de Desarrollo de la Ingeniería de Software en el Proceso la Ingeniería de Sistemas Software y Determinación de una metodología adaptable orientada a una organización pequeña. 2019.
23. CALISOFT, MODELO DE LA CALIDAD PARA EL DESARROLLO DE APLICACIONES INFORMÁTICAS (MCDAI)- GUÍA GENERAL, 2022.
24. Ministerio de Comunicaciones (MINCOM), Decreto 359 sobre el desarrollo de la industria cubana de programas y aplicaciones informáticas, 2019.
25. Ministerio de Comunicaciones (MINCOM), Resolución 124 que reglamenta la producción de los programas y aplicaciones informáticas y la evaluación de su calidad, 2019.
26. López Trujillo, Y.A.A., Margarita, ROLES EN EL PROCESO DE DESARROLLO DE SOFTWARE PARA LAS EMPRESAS CUBANAS. Instituto Superior Politécnico José Antonio Echeverría, 2006. vol. XXVII, núm. 1: p. pp. 31-35.