



**UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL**

**SUBSISTEMA DE POSGRADO  
MAESTRÍA EN TRANSFORMACIÓN DIGITAL DE NEGOCIOS**

**TEMA:**

Estudio de Caso: Transformación Organizacional mediante la Integración de Metodologías Ágiles en una empresa de Software

**AUTOR:**

Narváez Narváez, Diego Andrés

**Componente práctico del examen complejo previo a la  
obtención del título de Magíster en Transformación Digital de  
Negocios**

**TUTOR**

Maruri Siguenza, Juan Carlos

**Guayaquil, Ecuador**

**14 de abril del 2026**



UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL

**SUBSISTEMA DE POSGRADO  
MAESTRÍA EN TRANSFORMACIÓN DIGITAL DE NEGOCIOS**

## **CERTIFICACIÓN**

Certificamos que el presente **componente práctico del examen complejo**, fue realizado en su totalidad por **Narvárez Narvárez, Diego Andrés**, como requerimiento para la obtención del título de **Magíster en Transformación Digital de Negocios**.

### **REVISOR**

f. \_\_\_\_\_

**Maruri Sigüenza, Juan Carlos**

### **DIRECTOR DE LA MAESTRÍA**

f. \_\_\_\_\_

**Carvache Franco, Orly Daniel**

**Guayaquil, a los 14 del mes de abril del año 2026**



UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL

**SUBSISTEMA DE POSGRADO  
MAESTRÍA EN TRANSFORMACIÓN DIGITAL DE NEGOCIOS**

## **DECLARACIÓN DE RESPONSABILIDAD**

Yo, **Narvárez Narvárez, Diego Andrés**

### **DECLARO QUE:**

El **componente práctico del examen complejo, Estudio de Caso: Transformación Organizacional mediante la Integración de Metodologías Ágiles en una empresa de Software**, previo a la obtención del título de **Magíster en Transformación Digital de Negocios**, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

**Guayaquil, a los 14 del mes de abril del año 2026**

### **EL AUTOR**

f. \_\_\_\_\_  
**Narvárez Narvárez, Diego Andrés**



UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL

**SUBSISTEMA DE POSGRADO  
MAESTRÍA EN TRANSFORMACIÓN DIGITAL DE NEGOCIOS**

## **AUTORIZACIÓN**

Yo, **Narvárez Narvárez, Diego Andrés**

Autorizo a la Universidad Católica de Santiago de Guayaquil a la **publicación** en la biblioteca de la institución el **componente práctico del examen complejo Estudio de Caso: Transformación Organizacional mediante la Integración de Metodologías Ágiles en una empresa de Software** cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.

**Guayaquil, a los 14 del mes de abril del año 2026**

**EL AUTOR**

f. \_\_\_\_\_  
**Narvárez Narvárez, Diego Andrés**



UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL

SUBSISTEMA DE POSGRADO  
MAESTRÍA EN TRANSFORMACIÓN DIGITAL DE NEGOCIOS

REPORTE DE SOFTWARE ANTIPLAGIO

**INFORME DE ANÁLISIS**  
magister

Diego.Narvaez

8%  
Textos sospechosos

< 1% Similitudes  
0% similitudes entre comillas  
< 1% entre las fuentes mencionadas

3% Idiomas no reconocidos

5% Textos potencialmente generados por la IA

Nombre del documento: Diego.Narvaez.docx  
ID del documento: 6dff54ffac2592912e49c3ffe64f4be4658726ddb  
Tamaño del documento original: 1,63 MB

Depositante: Juan Carlos Maruri Siguenza  
Fecha de depósito: 24/11/2025  
Tipo de carga: interface  
fecha de fin de análisis: 24/11/2025

Número de palabras: 20.663  
Número de caracteres: 142.755

Ubicación de las similitudes en el documento:



Fuentes de similitudes

Fuentes principales detectadas

Nº	Descripciones	Similitudes	Ubicaciones	Datos adicionales
1	localhost   Naturaleza jurídica del consorcio. http://localhost:8080/mlui/bitstream/3317/12032/3/1-UCSG-PRE-JUR-DER-297.pdf.txt 21 fuentes similares	1%		Palabras idénticas: 1% (252 palabras)
2	Tesis_Carrillo_González_v1.docx   Tesis_Carrillo_González_v1 #417040 Viene de de mi grupo 21 fuentes similares	1%		Palabras idénticas: 1% (235 palabras)
3	ENSAYO FINAL DELGADO QUEZADA SILVIA VALESKA.docx   ENSAYO FI... #0c5b7f Viene de de mi grupo 20 fuentes similares	< 1%		Palabras idénticas: < 1% (191 palabras)
4	Nathaly Freire Juan Vega_P73.docx   Nathaly Freire Juan Vega_P73 #f56993 Viene de de mi grupo 6 fuentes similares	< 1%		Palabras idénticas: < 1% (170 palabras)
5	repositorio.ucsg.edu.ec   Transformación digital para impulsar las ventas en las ... http://repositorio.ucsg.edu.ec/bitstream/3317/25013/1/UCSG-C323-24536.pdf 19 fuentes similares	< 1%		Palabras idénticas: < 1% (168 palabras)

Fuentes con similitudes fortuitas

Nº	Descripciones	Similitudes	Ubicaciones	Datos adicionales
1	Documento de otro usuario #388462 Viene de de otro grupo	< 1%		Palabras idénticas: < 1% (16 palabras)
2	www.iils.org https://www.iils.org/CDs2022/CD2022Spring/papers/CB055NZ.pdf	< 1%		Palabras idénticas: < 1% (23 palabras)
3	repositorio.ucsg.edu.ec   El impacto estratégico del marketing digital en la capta... http://repositorio.ucsg.edu.ec/bitstream/3317/23823/1/UCSG-C518-23363.pdf	< 1%		Palabras idénticas: < 1% (22 palabras)
4	doi.org   La transformación digital en la gestión de emprendimientos e innovació... https://doi.org/10.33970/teetes.v7.n2.2023.336	< 1%		Palabras idénticas: < 1% (20 palabras)
5	hdl.handle.net   La influencia de la Cultura Organizativa en el Rendimiento Empr... https://hdl.handle.net/11000/33773	< 1%		Palabras idénticas: < 1% (19 palabras)

TUTOR

f. \_\_\_\_\_

Maruri Siguenza Juan Carlos

## **AGRADECIMIENTO**

Expreso mi más sincero agradecimiento a mi alma mater, la Universidad Católica de Santiago de Guayaquil, por haberme acogido nuevamente en esta etapa académica, al igual que en mi formación como ingeniero. Este espacio me ha brindado las herramientas y el entorno necesarios para continuar creciendo tanto en lo profesional como en lo personal.

A mis profesores de la maestría, por su dedicación, guía y compromiso en cada clase, transmitiendo no solo conocimientos, sino también valores y experiencias que han enriquecido mi formación. Su acompañamiento ha sido fundamental para alcanzar este objetivo.

A mis compañeros de maestría, por compartir este camino de aprendizaje, por el apoyo mutuo, las ideas compartidas y las conversaciones que nos han permitido crecer juntos, tanto en el ámbito académico como en el personal.

A mi familia, por comprender y aceptar el tiempo que debí restarles para poder culminar esta meta, brindándome su paciencia, comprensión y aliento incondicional en cada etapa de este proceso.

**Narváez Narváez, Diego Andrés**

## **DEDICATORIA**

Dedico este trabajo a mis padres, por inculcarme valores, disciplina y el amor por el aprendizaje, siendo siempre mi ejemplo de esfuerzo, perseverancia y honestidad. Sus enseñanzas han sido la base sobre la que he construido cada paso de mi vida, y su apoyo incondicional ha sido un pilar fundamental para llegar hasta aquí.

A mis hijos, por ser la razón más grande para seguir superándome y por dar sentido a cada meta alcanzada. Su sonrisa y su cariño han sido mi motor en los momentos más desafiantes, recordándome que cada sacrificio vale la pena cuando se hace por amor.

A mi familia en general, por acompañarme en cada etapa de este camino, brindándome su apoyo sincero y, sobre todo, por comprender y aceptar el tiempo limitado que pude compartir con ellos durante este proceso. Gracias por su paciencia, por alentarme aun en mis ausencias y por confiar en que lograría culminar este objetivo.

Este logro es tan mío como suyo, porque detrás de cada página escrita y cada noche de estudio estuvo el respaldo, la comprensión y el amor de quienes siempre han creído en mí.

**Narvárez Narvárez, Diego Andrés**



UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL

**SUBSISTEMA DE POSGRADO**  
**MAESTRÍA EN TRANSFORMACIÓN DIGITAL DE NEGOCIOS**

**TRIBUNAL DE GRADUACIÓN**

f. \_\_\_\_\_

**(NOMBRES Y APELLIDOS)**

PRESIDENTE DEL TRIBUNAL DE GRADUACIÓN

f. \_\_\_\_\_

**(NOMBRES Y APELLIDOS)**

TUTOR

MIEMBRO TRIBUNAL DE GRADUACIÓN

f. \_\_\_\_\_

**(NOMBRES Y APELLIDOS)**

REVISOR

MIEMBRO DEL TRIBUNAL DE GRADUACIÓN

## Índice

<b>ÍNDICE</b>	<b>IX</b>
<b>ÍNDICE DE TABLAS</b>	<b>XI</b>
<b>ÍNDICE DE FIGURAS</b>	<b>XII</b>
<b>RESUMEN</b>	<b>XIII</b>
<b>ABSTRACT</b>	<b>XIV</b>
<b>INTRODUCCIÓN</b>	<b>1</b>
<b>PROBLEMÁTICA</b>	<b>1</b>
<b>OBJETIVOS</b>	<b>3</b>
<i>Objetivo General</i>	3
<i>Objetivos Específicos</i>	3
<b>CONTEXTO</b>	<b>4</b>
<b>PROCESOS DE DESARROLLO DE SOFTWARE</b>	<b>5</b>
<b>ORGANIGRAMA DEL ÁREA DE SISTEMAS</b>	<b>7</b>
<b>FLUJO DEL PROCESO ACTUAL DE DESARROLLO DE SOFTWARE</b>	<b>7</b>
<b>VISIÓN ESTRATÉGICA Y CULTURA ORGANIZACIONAL</b>	<b>9</b>
<b>DISPOSICIÓN Y EXPECTATIVAS DE TRANSFORMACIÓN</b>	<b>10</b>
<b>METODOLOGÍA</b>	<b>11</b>
<b>TÉCNICAS DE RECOLECCIÓN DE DATOS</b>	<b>12</b>
<i>Entrevistas semiestructuradas</i>	12
<i>Resumen de las entrevistas</i>	12
<i>Análisis temático</i>	13
<i>Mapeo de procesos</i>	13
<i>Evaluación de madurez</i>	14
<b>METODOLOGÍAS DE TRANSFORMACIÓN DIGITAL: KANBAN Y SCRUM EN RELACIÓN CON LAS PROBLEMÁTICAS</b>	<b>14</b>
<i>Kanban: Optimización del Flujo de Trabajo y Visibilidad</i>	15
<i>Scrum: Colaboración, Adaptabilidad y Entrega Incremental</i>	16
<i>Complementariedad entre Kanban y Scrum</i>	20
<i>Relación con CMMI: Un Enfoque Híbrido</i>	21
<b>PROPUESTA</b>	<b>22</b>
<b>IMPLEMENTACIÓN DE SCRUM</b>	<b>22</b>

<b>IMPLEMENTACIÓN DE KANBAN</b>	<b>28</b>
<b>INTEGRACIÓN CON CMMI</b>	<b>31</b>
<b>GESTIÓN DEL CONOCIMIENTO Y TRANSFORMACIÓN CULTURAL</b>	<b>32</b>
<b>PLAN DE CAPACITACIÓN</b>	<b>33</b>
<b>VALIDACIÓN</b>	<b>37</b>
<i>Validación Externa de la Propuesta con Expertos</i>	37
<i>Sistemas de indicadores para validar el resultado del modelo propuesto</i>	38
<i>Plan de Acción con Indicadores Integrados</i>	39
<i>Fase 1: Piloto 3-6 meses</i>	41
<i>Fase 2: Escalamiento (6-12 meses)</i>	41
<i>Fase 3: Consolidación 12+ meses</i>	44
<i>Monitoreo y Ajustes Continuos</i>	44
<b>CONCLUSIONES</b>	<b>46</b>
<b>RECOMENDACIONES</b>	<b>47</b>
<b>ANEXO 1</b>	<b>48</b>
<b>ANEXO 2</b>	<b>59</b>
<b>ANEXO 3</b>	<b>70</b>
<b>ANEXO 4</b>	<b>74</b>
<b>ANEXO 5</b>	<b>83</b>
<b>ANEXO 6</b>	<b>87</b>
<b>BIBLIOGRAFÍA</b>	<b>91</b>

## Índice de Tablas

Tabla 1 Roles y funciones dentro del Área de Sistemas	8
Tabla 2 Duración promedio actual por fase	11
Tabla 3 Tabla genérica de Ciclo Iterativo de Sprints	29
Tabla 4 Ejemplo Adaptado de que servicio <i>E-COMMERCE</i>	30
Tabla 5 Resumen de Herramientas Propuestas	39
Tabla 6 Indicadores Clave de Desempeño	43
Tabla 7 Fase 1: Piloto (3–6 meses)	45
Tabla 8 Fase 2: Escalamiento (6–12 meses)	46
Tabla 9 Fase 3: Consolidación 12+ meses	47
Tabla 10 Instrumentos de investigación	83

## Índice de Figuras

Figura 1 Organigrama del Área de Sistemas	22
Figura 2 Flujo del proceso actual del desarrollo de software	24
Figura 3 Modelo Cascada y sus problemas, pasando a Metodología Scrum	37
Figura 5 Ilustración de ejemplo adaptado	48
Figura 6 Flujo de Trabajo	50
Figura 7 MS-Teams	53
Figura 8 Moodle	54
Figura 9 Google Forms	96
Figura 10 Power BI	96

## Resumen

Este proyecto de investigación analiza la transformación organizacional de una empresa de desarrollo de software en Guayaquil que actualmente opera bajo el modelo en cascada, un enfoque lineal que genera retrasos, baja visibilidad del progreso, comunicación deficiente y alta incidencia de defectos. La calidad del producto y la satisfacción del cliente se impacta negativamente debido a las limitaciones. La propuesta del marco integral donde se combinan las metodologías Scrum y Kanban con estándares de calidad del modelo de madurez CMMI tiene como intención mejorar eficientemente la colaboración interdisciplinaria y fomentar la cultura de mejora continua. Desarrollar un modelo que eleve la madurez en el desarrollo de software y fortalezca la cultura organizacional es el objetivo. El estudio combina métodos cualitativos como entrevistas semiestructuradas y cuantitativos como encuestas tipo Likert, complementados con análisis temático y mapeo de procesos, para diagnosticar la situación actual y diseñar estrategias de intervención. La implementación se estructura en tres fases: un piloto para validar el marco, una etapa de escalamiento a otros equipos y una fase de consolidación para estandarizar buenas prácticas. Cada fase incluye capacitación, integración de herramientas colaborativas, ajustes metodológicos y seguimiento mediante KPIs. Se espera que esta transformación reduzca los tiempos de entrega entre un 20% y 30%, disminuya la tasa de defectos entre un 15% y 25%, y aumente la satisfacción del cliente y del equipo en un 30% a 40%, promoviendo una transición sostenible hacia un modelo ágil más competitivo.

**Palabras Clave:** Transformación organizacional, metodologías ágiles, mejora continua

## Abstract

This research project analyzes the organizational transformation of a software development company in Guayaquil that currently operates under the waterfall model, a linear approach that generates delays, low visibility of progress, poor communication, and a high incidence of defects. These constraints negatively affect product quality and customer satisfaction. To overcome these problems, a plan is suggested that combines the agile Scrum and Kanban methods with the maturity of the CMMI model to improve work, collaboration between different areas, and foster a culture of continuous improvement and innovation. The ultimate goal is to create a model that increases maturity in software development and strengthens group culture. The study combines qualitative methods such as semi-structured interviews and quantitative methods such as Likert surveys, complemented by thematic analysis and process mapping, to diagnose the current situation and design intervention strategies. There are three parts to the implementation process: a pilot to test the framework, a scaling phase to other teams, and a consolidation phase to make sure that best practices are followed. Training, adding collaborative tools, changing methods, and keeping an eye on things through KPIs are all parts of each phase. This change is expected to cut delivery time by 20% to 30%, cut the defect rate by 15% to 25%, and make customers and teams 30% to 40% happier, all of which will help the company move to a more competitive agile model in a way that lasts.

**Keywords:** *Organizational transformation, agile methodologies, continuous improvement*

## **Introducción**

Este estudio de caso aborda la transformación organizacional de una empresa tecnológica dedicada al desarrollo de software para distintos sectores a nivel nacional e internacional. La empresa enfrenta desafíos comunes en la gestión de proyectos y la cultura organizacional.

Entre las principales problemáticas identificadas se encuentran la falta de visibilidad del progreso de los proyectos, lo que ocasiona retrasos en las entregas y dificultades para alinear las expectativas de los clientes. También, se veía una negación al cambio por parte de los grupos, con modos estrictos y una comunicación mala entre partes, lo cual afectaba malo en la calidad del producto final y en la completitud del cliente.

La idea trae un plan completo que mezcla métodos rápidos como Scrum y Kanban con las reglas de calidad CMMI (Capability Maturity Model Integration), sumando un programa para cambiar la cultura, un sistema para manejar el conocimiento y un diseño extenso de enseñanza. Diseñado para ser replicable en contextos similares, el marco ofrece una solución práctica, adaptable y detallada para optimizar procesos, mejorar la calidad del software y fomentar una cultura colaborativa en un entorno competitivo. (Anderson, 2010)

## **Problemática**

El departamento de sistemas de la empresa, objeto del presente trabajo, opera bajo un modelo tradicional de gestión de proyectos, basado en el modelo en cascada, con procesos lineales, requisitos definidos rígidamente al inicio, comunicación centralizada y limitada entre departamentos, y entregas únicas al finalizar los proyectos. Dado el modo de operación actual del departamento de sistemas, se generan desafíos operativos y culturales significativos. Esta metodología ha provocado una serie de ineficiencias que afectan directamente la competitividad, la calidad del software y la capacidad de innovación de la empresa.

La falta de visibilidad en el progreso de los proyectos es una problemática central, ya que no se cuenta con herramientas de monitoreo en tiempo real. Esto

impide el rastreo efectivo de las tareas y el avance general, generando cuellos de botella recurrentes y retrasos en las entregas. Esta situación dificulta la alineación de expectativas con los clientes y la toma de decisiones informadas, impactando negativamente en la eficiencia operativa.

Asimismo, la ausencia de flujos de trabajo estandarizados conduce a inconsistencias en la calidad del software y a una alta tasa de defectos. Los desaciertos que suelen ser hallados en fases tardías incluso luego de la entrega crean la necesidad de retrabajos de alto costo en el que se involucran procesos tales como reubicación de personal e interrupción de nuevos desarrollos. Esto aumenta los gastos, los tiempos de desarrollo y, adicionalmente, resulta en una confianza del cliente desgastada y detrimento moral del equipo técnico.

Al final la cultura organizacional jerárquica se ve fragmentada con silos organizacionales entre departamentos como los de desarrollo, estudio y procedimientos, que delimita la contribución interdepartamental y la innovación. Esta inflexibilidad impide la comunicación fluida, frena el proceso de la toma de decisiones y minimiza la capacidad de respuesta rápida a las variaciones en los requerimientos del cliente, comprometiendo la adaptabilidad de la empresa en un entorno competitivo.

La puesta en marcha del modelo en cascada ha generado una serie de inacciones en el proceso de desarrollo de software. Una de las principales es la ausencia de visibilidad en el avance de los proyectos, por la falta de herramientas de monitoreo al instante. Esto impide el rastreo efectivo de las tareas y el avance general, generando cuellos de botella recurrentes y retrasos en las entregas. Esta situación dificulta la alineación de expectativas con los clientes y la toma de decisiones informadas, impactando negativamente en la eficiencia operativa.

Asimismo, la ausencia de flujos de trabajo estandarizados conduce a inconsistencias en la calidad del software y a una alta tasa de defectos.

Se identificó que la percepción general es de una gestión de proyectos que no permite la adaptación rápida a los cambios, lo que se traduce en una constante lucha por cumplir con los plazos y mantener la calidad. Los equipos

expresaron la necesidad de mayor transparencia en el flujo de trabajo y de procesos más flexibles que faciliten la colaboración y la resolución temprana de problemas. Estas entrevistas se encuentran disponibles en el marco de este estudio en la sección respectiva de Anexo 1 y Anexo 2.

La rigidez anteriormente descrita obstaculiza la comunicación fluida, ralentiza la toma de decisiones y reduce la capacidad de respuesta hacia los cambios de los requisitos del cliente lo que compromete la adaptabilidad de la empresa en un entorno competitivo.

## **Objetivos**

### **Objetivo General**

Desarrollar un marco de trabajo integral que mediante la integración de metodologías ágiles contribuya a la mejora de la madurez en el desarrollo de software, estableciendo bases sólidas para la calidad organizacional en el área de sistemas.

### **Objetivos Específicos**

Diagnosticar la situación actual del área de desarrollo de software, respecto a la cultura organizacional y el nivel de madurez de los procesos de desarrollo de software, identificando fortalezas y áreas de mejora en el desempeño del equipo.

Desarrollar estrategias que integren metodologías ágiles con CMMI a fin de optimizar el desempeño en el desarrollo de software, la colaboración interdepartamental y prácticas de mejora continua.

Proponer un plan de implementación de las metodologías ágiles con CMMI que incluya gestión del conocimiento y capacitación, para garantizar la continuidad y mejora de las prácticas ágiles en la organización.

Establecer indicadores clave de desempeño KPI que permitan evaluar la madurez organizacional del área de desarrollo de software y la efectividad de las metodologías ágiles implementadas.

## Contexto

Para el presente caso de estudio se considera una empresa tecnológica del sector de desarrollo de software, clasificada como PYME, ubicada en la ciudad de Guayaquil, Ecuador, en un entorno empresarial competitivo y dinámico. Con una presencia de más de 10 años en el mercado, se ha consolidado y mantenido como una empresa confiable para clientes de diversos sectores tanto a nivel nacional como internacional.

La empresa dispone de un equipo especializado de 15 empleados, 8 de los cuales pertenecen al equipo del área de sistemas, este equipo está conformado por desarrolladores, testers y analistas de sistemas quienes ofrecen soluciones tecnológicas innovadoras.

La actual estructura organizacional tiene una marcada segmentación entre los diferentes equipos de trabajo tal como un modelo tradicional, departamentos que operan de forma aislada, con poca o nula coordinación entre sí.

Por ejemplo, el equipo de desarrollo avanza en funcionalidades sin contar con retroalimentación constante del área de pruebas, mientras que la gestión de proyectos carece de visibilidad en tiempo real sobre el estado real del avance técnico. Las decisiones fluyen verticalmente y no existen espacios comunes de discusión o canales horizontales de comunicación. No se comparten conocimientos relevantes entre equipos se generan esfuerzos duplicados y demoras innecesarias en las entregas finales.

Los silos funcionales son, por tanto, uno de los principales desafíos que enfrenta la empresa y representan un obstáculo significativo para la adopción de metodologías ágiles, las cuales requieren un alto nivel de colaboración, transparencia y responsabilidad compartida entre todos los involucrados en el desarrollo de software.

La empresa se destaca por ofrecer servicios como la implementación de plataformas de e-commerce, desarrollo de aplicaciones móviles, software

personalizado y otras soluciones digitales que se adaptan a las necesidades específicas de sus clientes.

## **Procesos de Desarrollo de Software**

La empresa actualmente opera bajo un modelo tradicional de gestión de proyectos basado en el enfoque en cascada. Este modelo implica una secuencia lineal de fases, donde cada etapa debe completarse antes de iniciar la siguiente, sin retroalimentación continua ni iteraciones. A continuación, se describe textualmente el proceso actual de desarrollo de software, los involucrados y sus funciones.

En la Tabla 1, se presentan los roles y funciones dentro del Área de Sistemas, considerando las fases involucradas en el desarrollo de un Proyecto.

**Tabla 1 Roles y funciones dentro del Área de Sistemas**

<b>Fase del Proyecto</b>	<b>Responsable</b>	<b>Funciones</b>
<b>1. Inicio del Proyecto</b>	Gerente del área de sistemas	<ul style="list-style-type: none"> <li>- Reunirse con el cliente para recopilar requisitos iniciales.</li> <li>- Asignar recursos humanos y técnicos al proyecto.</li> <li>- Definir cronograma general del proyecto.</li> </ul>
<b>2. Análisis y Especificación de Requisitos</b>	Analista de sistemas	<ul style="list-style-type: none"> <li>- Documentar los requisitos funcionales y no funcionales.</li> <li>- Validar los requisitos con el cliente.</li> <li>- Entregar un documento formal al equipo de desarrollo.</li> </ul>
<b>3. Diseño del Sistema</b>	Arquitecto técnico o desarrollador senior	<ul style="list-style-type: none"> <li>- Crear diagramas UML para incluir definición y modelos de datos.</li> <li>- Definir la arquitectura técnica del sistema.</li> <li>- Comunicar el diseño al equipo de desarrollo.</li> </ul>
<b>4. Desarrollo</b>	Equipo de desarrollo (desarrolladores)	<ul style="list-style-type: none"> <li>- Programar funcionalidades según especificaciones.</li> <li>- Realizar pruebas unitarias básicas.</li> <li>- Enviar código al repositorio (actualmente GitHub).</li> </ul>
<b>5. Pruebas</b>	Equipo de testing (testers)	<ul style="list-style-type: none"> <li>- Ejecutar pruebas manuales y automatizadas.</li> <li>- Registrar defectos en documentos compartidos.</li> <li>- Validar correcciones del equipo de desarrollo.</li> </ul>
<b>6. Implementación / Despliegue</b>	Desarrollador senior o administrador de servidores	<ul style="list-style-type: none"> <li>- Migrar el sistema a producción.</li> <li>- Verificar correcto funcionamiento en entorno real.</li> <li>- Coordinar con el cliente para aceptación final.</li> </ul>
<b>7. Soporte y Mantenimiento</b>	Equipo técnico interno	<ul style="list-style-type: none"> <li>- Atender reportes de errores posteriores a la entrega.</li> <li>- Aplicar parches o actualizaciones menores.</li> <li>- Documentar cambios realizados.</li> </ul>

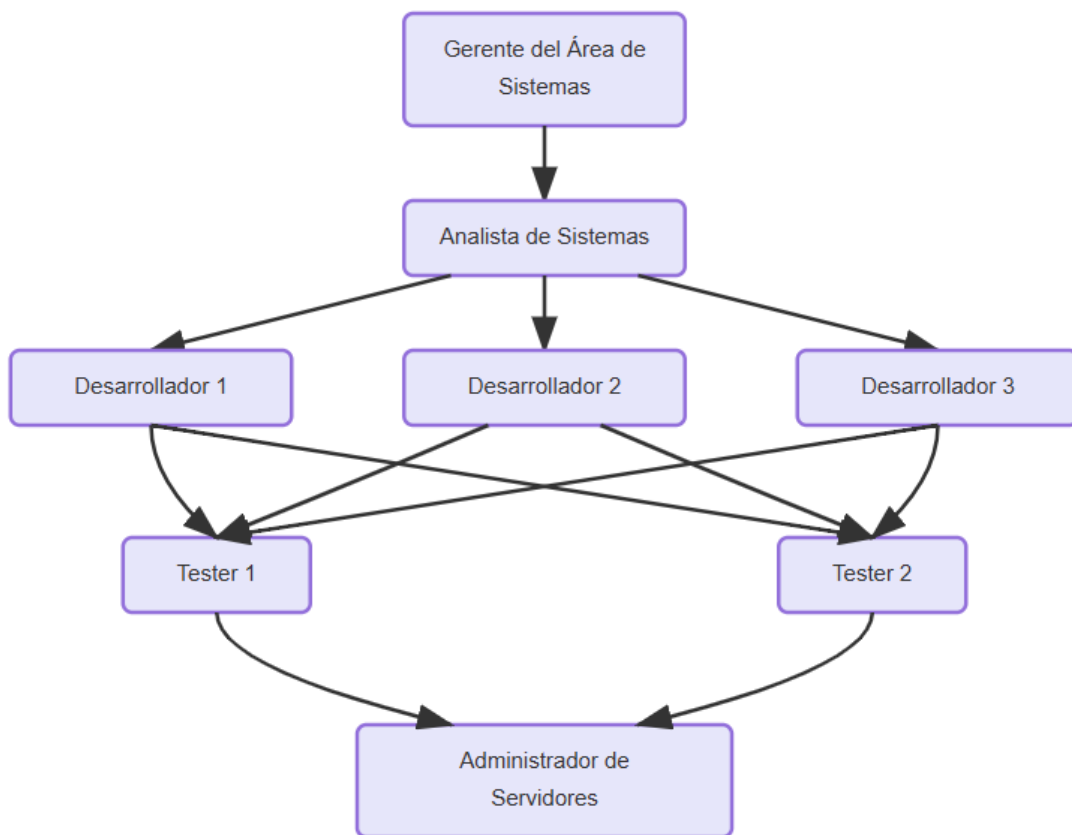
**Nota.** Funciones detalladas por rol en el Área de Sistemas.

## Organigrama del Área de Sistemas

En el Área de Sistemas se dispone de 8 personas.

La Figura 1 muestra la estructura jerárquica y las relaciones en la misma, cuya distribución interna se encuentra detallada con sus relaciones funcionales entre los distintos miembros del equipo.

**Figura 1 Organigrama del Área de Sistemas**

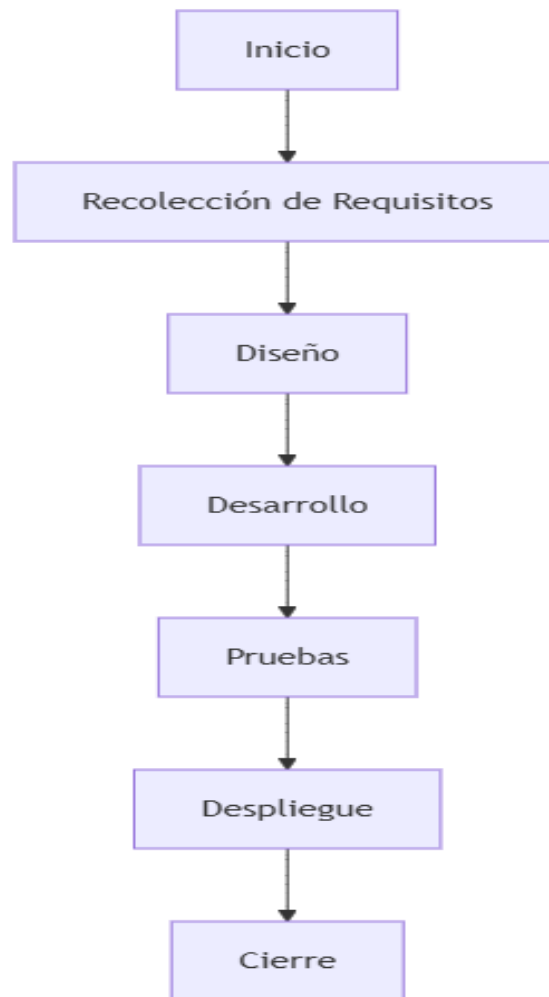


**Nota.** Organigrama del Área de Sistemas y sus relaciones

## Flujo del Proceso Actual de Desarrollo de Software

El Flujo del Proceso Actual de Desarrollo de Software se ilustra en la Figura 2, donde se visualizan las etapas y los puntos de interacción.

**Figura 2 Flujo del proceso actual del desarrollo de software**



**Nota.** Proceso actual del desarrollo de software

La información a continuación, en base a los proyectos la empresa especializada en comercio electrónico permite detallar la duración promedio actual por fases de un Proyecto.

La tabla 2 a continuación muestra una duración promedio por fase:

**Tabla 2 Duración promedio actual por fase**

<b>Fase</b>	<b>Ecommerce</b>	<b>ERP</b>
<b>Recolección de Requisitos</b>	2 semanas	3 semanas
<b>Diseño</b>	1 semana	2 semanas
<b>Desarrollo</b>	8 semanas	12 semanas
<b>Pruebas</b>	3 semanas	4 semanas
<b>Despliegue</b>	1 semana	1 semana
<b>Total estimado</b>	~15 semanas	~22 semanas

**Nota.** Tiempo promedio por fases

En el desarrollo de software se utilizan GitHub para control de versiones y Google Drive y correo electrónico para comunicación e intercambio de documentos. El testing se realiza manualmente, registrando defectos en hojas de Excel. El despliegue se efectúa mediante FTP y scripts personalizados. En este trayecto ha surgido la necesidad de emplear plataformas ágiles como Jira, Trello o Kanbanize, por lo que la coordinación se realiza de forma directa o por correo, sin un sistema estructurado de seguimiento de tareas, que es lo ideal.

## **Visión Estratégica y Cultura Organizacional**

La gerencia del área de desarrollo de software percibe que la empresa necesita modernizar sus procesos para mantenerse competitiva en un mercado dinámico. Según la entrevista realizada al gerente del área, las principales percepciones son las siguientes:

- Sobre la cultura organizacional: Existe cierta resistencia al cambio, especialmente en niveles operativos. Los equipos están acostumbrados a trabajar de manera individual y con poca colaboración cruzada.

- Sobre la adopción de nuevas metodologías: El gerente reconoce la necesidad de adoptar prácticas más ágiles, aunque desconoce en profundidad cuáles podrían ser las mejores opciones.
- Sobre la estructura actual: Considera que la organización está muy fragmentada, lo cual afecta la eficiencia y la capacidad de innovación.

### **Disposición y Expectativas de Transformación**

Observando el Anexo 1, se dispone de las preguntas y respuestas del personal del Área de Sistemas, cuya síntesis y hallazgos se detallan aquí:

El Gerente de Sistemas manifiesta apertura hacia una transformación cultural orientada a metodologías ágiles, siempre que exista apoyo externo y capacitación. Considera que este cambio podría mejorar la productividad, la calidad del software y la colaboración entre equipos, pero reconoce que la falta de experiencia y la resistencia a modificar prácticas arraigadas son obstáculos relevantes, los cuales plantea superar mediante formación constante y liderazgo activo.

El Analista de Sistemas describe un flujo de trabajo lineal, con escasa retroalimentación y participación limitada durante el desarrollo. La comunicación está limitada al correo electrónico, mientras que las herramientas no están centralizadas, y los estándares son problemáticos y, en este, se pierde información y la colaboración es imposible.

Del mismo modo, señala que con una metodología ágil trabajaría de manera más proactiva; por un lado, su participación sería promovida desde el principio, y por el otro, el resto del equipo también estaría mejor coordinado. Los desarrolladores de software operan según un modelo en cascada, lo que significa que las tareas se asignan de manera aislada: no hay interacción con los otros roles y entre ellos y no hay revisión general continua. La comunicación por correo es la más utilizada, mientras que el empleo de herramientas es incompleto y discontinua. A pesar de que demuestran una cierta resistencia a la transformación por miedo a ser sobrecargados con trabajo, estarían abiertos a implementar nuevas metodologías si les brindan capacitación y

acompañamiento, con la expectativa de mejorar la coordinación, la visibilidad del avance y disminuir el retrabajo.

Los probadores de software llevan a cabo pruebas manuales que documentan en hojas de cálculo y reportan los fallos a través del correo electrónico. No toman parte en etapas iniciales del desarrollo y solo al final reciben la información del producto lo que resulta en que la identificación de problemas demore. Este personal cree que su contribución podría aumentar si se utilizan herramientas que faciliten su integración desde el principio y con la aplicación de metodologías es pos de conseguir un diálogo fluido con los desarrolladores.

Al final del ciclo el Administrador de Servidores tiene un papel fundamental pero no así en la planificación, tampoco tiene visibilidad temprana de las modificaciones. Se evidencia entonces la falta de prácticas de DevOps, perspectiva que une operaciones de TI y con desarrollo del software. El Administrador en la entrevista indicó que fortalecería su rol si se le incluye desde el principio con procesos estandarizados.

En conjunto del análisis de las repuestas se puede inferir que en general el equipo presenta una disposición moderada al cambio, y esta se limita dada la inexperiencia en uso de dichas metodologías ágiles y también por la dependencia de procesos tradicionales. Los hallazgos muestran problemas comunes: comunicación fragmentada, integración tardía de roles clave y procesos lineales que generan retrasos y retrabajos. La expectativa general es que, con capacitación adecuada, liderazgo activo y adopción de prácticas ágiles y herramientas integradas, la organización podría mejorar la coordinación, la eficiencia y la calidad del producto entregado.

## **Metodología**

El estudio utiliza un enfoque combinando métodos cualitativos, orientados a explorar la cultura organizacional y las percepciones (Noteboom et al., 2021). Se trata de un diseño de estudio de caso de tipo explicativo, con elementos descriptivos y exploratorios, que busca comprender cómo las metodologías

ágiles y los estándares de calidad transforman los procesos y la cultura (Stouby Persson et al., 2022) .La unidad de análisis corresponde a equipos de desarrollo de software en una empresa del mismo rubro, con énfasis en los procesos y la dinámica cultural (Schwaber y Sutherland, 2020).

## **Técnicas de Recolección de Datos**

### ***Entrevistas semiestructuradas***

Las entrevistas se realizaron a personas con roles clave dentro del Área de Sistemas de la organización, con el objetivo de obtener una visión integral de los procesos actuales, la cultura organizacional y la percepción sobre posibles transformaciones metodológicas (Sommerville, 2016). Los participantes seleccionados han sido: Gerente de Sistemas, Analista de Sistemas, tres Desarrollador, dos testers y el administrador de servidores.

Entre el 15 y el 19 de abril de 2025, se llevaron a cabo una serie de entrevistas a través de videollamadas grabadas por Zoom con diversos profesionales de un equipo de sistemas. El proceso comenzó el 15 de abril con el Gerente de Sistemas, cuya entrevista duró 35 minutos. Dos días después, el 17 de abril, se entrevistaron a dos desarrolladores: el Desarrollador 1, en una sesión de 30 minutos, y el Desarrollador 2, en una de 28 minutos. El 18 de abril se desarrollaron tres entrevistas, la una fue al Analista de Sistemas, con una duración de 40 minutos, la del Desarrollador 3, duró 26 minutos, y la del Tester 1, se desarrolló en 25 minutos. Durante la última sesión del 19 de abril se cerraron las entrevistas del Tester 2, cuya sesión duró 22 minutos y la del Administrador de Servidores que se realizó en 25 minutos. Se utilizó Zoom como plataforma de comunicación virtual.

### ***Resumen de las entrevistas***

Un resumen ejecutivo de cada entrevista, incluyendo preguntas realizadas, respuestas clave y observaciones relevantes, se encuentra disponible en el Anexo 1 del presente documento, al igual de las consideraciones que se obtuvieron en las mismas en el Anexo 2.

## ***Análisis temático***

Dentro del análisis temático realizado, se identificaron varios patrones relevantes relacionados con la percepción de los involucrados sobre los procesos actuales, la cultura organizacional y la disposición hacia la transformación metodológica. Entre los patrones más significativos encontramos:

- Falta de conocimiento sobre metodologías ágiles: Tanto desarrolladores como testers manifestaron desconocimiento o comprensión limitada sobre Scrum y Kanban, lo cual representa una brecha de formación que afecta la adopción de nuevas prácticas.
- Cultura de trabajo individual: Los equipos operan de forma aislada, sin canales formales de comunicación entre desarrollo, pruebas y gestión, lo que refuerza la existencia de silos funcionales.
- Se ha notado resistencia al cambio: No una oposición negativa, sino temor a lo desconocido y a que aumente a carga de trabajo si no se conocen con seguridad los verdaderos beneficios de los nuevos métodos.
- Poca transparencia del flujo de trabajo: Se identificó que los técnicos y el gerente no cuentan con medios para monitorear el avance del trabajo en tiempo real, lo que genera retrasos, reprocesos y problemas para establecer prioridades.

Estos patrones se identificaron mediante la triangulación de datos obtenidos en entrevistas semiestructuradas, observaciones directas y análisis documental previo realizado meses atrás con una visión de implementación inminente de metodologías ágiles en la empresa con este histórico permitiendo construir una base para diseñar una propuesta de intervención ajustada a las necesidades reales de la organización.

## ***Mapeo de procesos***

El análisis comparativo se realizó contrastando los flujos de trabajo actuales de la empresa con los estándares recomendados por las metodologías ágiles, tales como Scrum y Kanban, y el modelo CMMI para el desarrollo de

software (Schwaber y Sutherland, 2020). En este sentido, se identificaron diferencias significativas en cuanto al origen y manejo de los flujos de trabajo:

- Origen de los flujos actuales: Los procesos siguen un enfoque lineal basado en cascada, donde cada fase comienza solo cuando la anterior termina. Este flujo se origina principalmente desde documentos de requisitos entregados por el analista y gestionados mediante correo electrónico, sin retroalimentación continua ni revisiones intermedias.
- Falta de ciclos iterativos: No hay entregas progresivas o flujos cíclicos, no se permite modificar el producto según las necesidades cambiantes del cliente.
- Visualización del flujo laboral: Actualmente no hay una representación visual del estado de las tareas en curso, lo que posibilita la justificación de la implementación de Kanban.
- Combinación entre roles: Los roles de la empresa Enel caso de estudio no están interconectados ya que el desarrollo progresa sin coordinación con la evaluación y la administración no tiene actualizaciones en tiempo real.

Este análisis permitió identificar áreas de mejora clave, tales como la necesidad de capacitación en metodologías ágiles, la adopción de herramientas de gestión visual y la creación de espacios de colaboración entre equipos.

### ***Evaluación de madurez***

Se empleó el marco CMMI para diagnosticar el nivel actual de madurez organizacional. Esta evaluación proporcionará una visión estructurada de las capacidades de la empresa en términos de gestión de proyectos, calidad del software y prácticas de desarrollo, sirviendo como base para establecer un plan de mejora incremental hacia niveles superiores de madurez (Sommerville, 2016).

### **Metodologías de Transformación Digital: Kanban y Scrum en Relación con las Problemáticas**

La transformación digital en esta empresa requiere la adopción de metodologías que permitan abordar los desafíos operativos y culturales

identificados. Entre las metodologías más efectivas para este propósito se encuentran Kanban y Scrum, ambas pertenecientes al marco ágil, que han demostrado ser herramientas poderosas para mejorar la eficiencia, la calidad del software y la colaboración interdepartamental. A continuación, se detalla cómo estas metodologías pueden resolver las problemáticas específicas que enfrenta la empresa.

### ***Kanban: Optimización del Flujo de Trabajo y Visibilidad***

Kanban es una metodología visual que se centra en la gestión del flujo de trabajo y la mejora continua. Utiliza tableros Kanban, físicos o digitales, para representar el estado actual de las tareas, lo que permite a los equipos tener una visión clara de su progreso y detectar cuellos de botella en tiempo real. Esta característica es especialmente relevante para abordar las siguientes problemáticas:

- Falta de visibilidad en el flujo de trabajo: Uno de los principales problemas de la empresa es la ausencia de herramientas que permitan rastrear el progreso de los proyectos. Kanban resuelve esta limitación al proporcionar un tablero visual donde cada tarea tiene un estado claro (pendiente, en progreso, completada). Esto no solo mejora la transparencia, sino que también facilita la identificación de áreas problemáticas antes de que se conviertan en retrasos significativos.
- Procesos que no son estandarizados: La inconsistencia en flujos de trabajo quita calidad al software. Kanban al establecer límites de trabajo en progreso (WIP, por sus siglas en inglés) facilita a que los equipos no se sobrecarguen, al documentar y estandarizar pasos para avanzar una tarea se puede reducir la variabilidad en los procesos.
- Alta tasa de defectos: Los errores detectados con retraso generan retrabajos costosos. Kanban fomenta la entrega continua y la revisión constante de las tareas, lo que permite detectar problemas temprano en el proceso y corregirlos antes de que afecten entregables finales.

## ***Scrum: Colaboración, Adaptabilidad y Entrega Incremental***

Scrum es una metodología iterativa e incremental que divide el trabajo en ciclos cortos llamados sprints (generalmente de 2-4 semanas). Cada sprint culmina con una entrega funcional de software, lo que permite a los equipos adaptarse rápidamente a los cambios y entregar valor de manera continua. Esta metodología es ideal para abordar las siguientes problemáticas:

- **Cultura jerárquica y silos funcionales:** Las estructuras organizacionales tradicionales limitan la colaboración entre departamentos. Scrum promueve una cultura de equipo al requerir la participación activa de todos los miembros en reuniones clave como el Daily Scrum (reunión diaria), la planificación de sprints y las retrospectivas.
- **Cambios burocráticos y requisitos estrictos:** La compañía no puede adaptarse a las modificaciones de los pedidos de los clientes debido al modelo en cascada que emplea. Scrum trata este obstáculo al permitir que se hagan cambios durante el desarrollo. El equipo, al finalizar cada sprint, examina el producto con el cliente y ajusta el backlog según sea necesario para asegurar que el software esté a la altura de las expectativas del cliente en ese momento.
- **Entregas únicas al término de los proyectos:** La posibilidad de que el producto final no cumpla con los requerimientos del cliente se incrementa si la entrega es única y se realiza al término del proyecto. Scrum aborda este problema a través de entregas por etapas, lo que hace posible que el cliente verifique el avance del proyecto en fases iniciales y ajuste si fuera necesario.

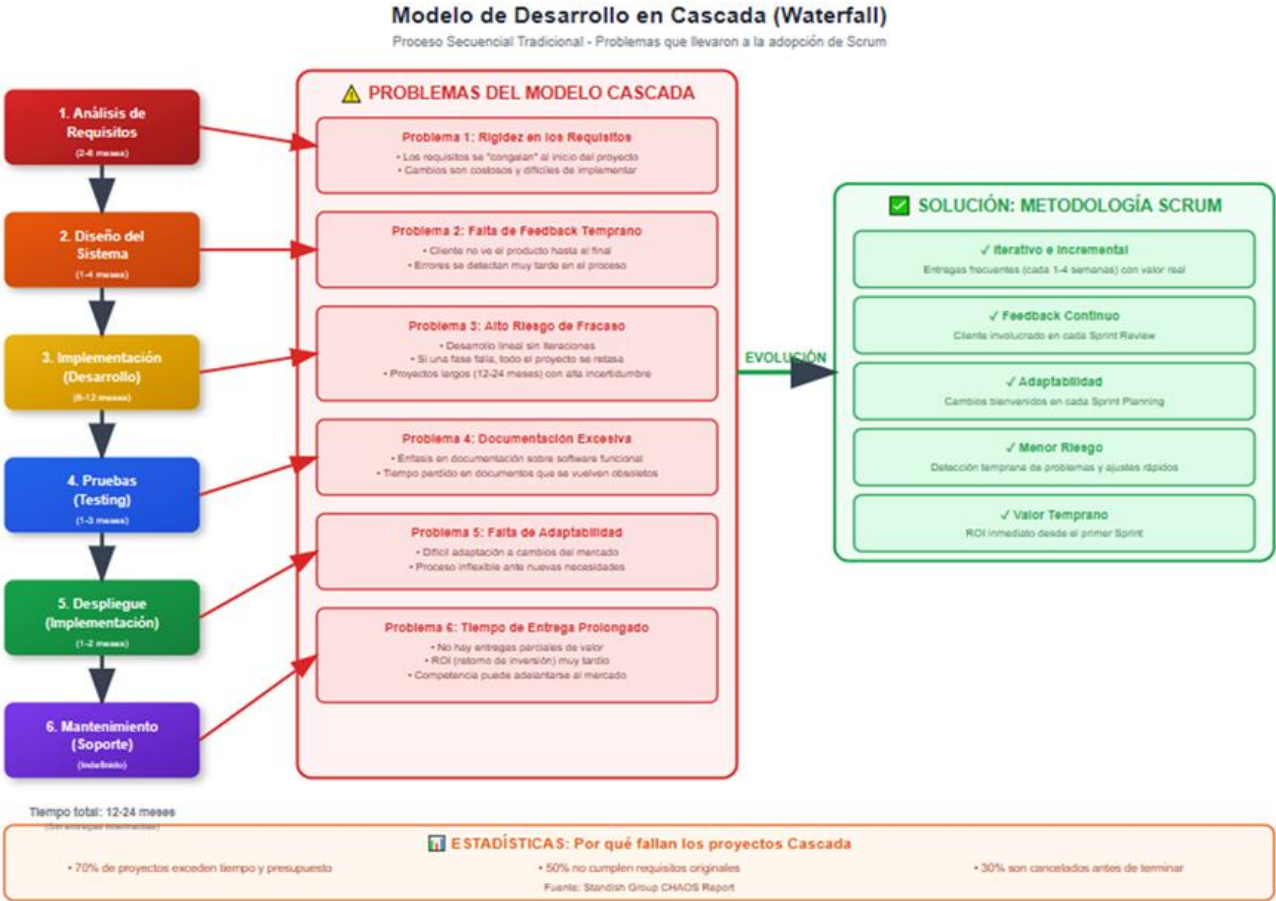
La Figura 3 describe el modelo de desarrollo en cascada o en inglés waterfall, cuyo enfoque tradicional y secuencial es actual en el Área de Sistemas para la gestión de proyectos de software. Se plantean las seis fases principales: análisis de requisitos, diseño del sistema, implementación, pruebas, despliegue y mantenimiento con un tiempo estimado de entre 12 a 24 meses en total. Cada etapa tiene que estar terminada antes de comenzar la siguiente, lo cual genera problemas en cuanto a la rigidez de los requisitos iniciales, dificultades para implementar cambios, escasa retroalimentación del cliente hasta el final,

documentación excesiva, poca adaptabilidad y un tiempo de entrega largo. De acuerdo con las estadísticas del mismo Departamento de Sistemas, el 70 % de los proyectos en cascada superan el tiempo y presupuesto, el 50 % no satisfacen los requerimientos iniciales y el 30 % son cancelados antes de ser completados.

Como solución, se plantea la metodología Scrum, que presenta un enfoque iterativo e incremental. Esta metodología integra entregas frecuentes, cada 1-4 semanas, con revisiones de sprints, feedback continuo del cliente, flexibilidad a los cambios mediante la planificación de sprints, detección temprana de problemas y un gran valor del tiempo ROI al entregar resultados rápidamente.

La Figura 3 dispone de un diagrama con una comparación visual entre el tradicional Modelo de Desarrollo en Cascada y la moderna Metodología Scrum, ilustrando claramente por qué las organizaciones han migrado hacia enfoques ágiles para el desarrollo de software.

Figura 3 Modelo Cascada y sus problemas, pasando a Metodología Scrum



**Nota.** Problemas del Modelo Cascada y paso a Scrum

El lado izquierdo del diagrama muestra las seis fases secuenciales del modelo Cascada, representadas como una cascada descendente donde cada etapa debe completarse antes de pasar a la siguiente:

- Análisis de Requisitos (rojo): La fase inicial donde se definen todos los requerimientos del sistema.
- Diseño del Sistema (naranja): Creación de la arquitectura y diseño detallado.
- Implementación (amarillo): Desarrollo y codificación del software.
- Pruebas (azul): Testing del sistema completo.
- Despliegue (verde): Implementación en el ambiente de producción.
- Mantenimiento (morado): Soporte y actualizaciones posteriores.

El diagrama identifica específicamente seis problemas principales que emergen de este enfoque secuencial:

- Rigidez en los requisitos
- Falta de retroalimentación temprana
- Alto riesgo de fracaso
- Documentación excesiva
- Falta de adaptabilidad
- Prolongados tiempos de entrega.

El lado derecho del diagrama presenta la Solución con Metodología Scrum, mostrando cómo este enfoque ágil resuelve sistemáticamente cada uno de los problemas identificados:

- Iterativo e Incremental: Reemplaza el flujo lineal con ciclos cortos que entregan valor constantemente
- Feedback Continuo: Incorpora la retroalimentación del cliente en cada iteración
- Adaptabilidad: Permite cambios y ajustes frecuentes según las necesidades del Sprint Planning
- Menor Riesgo: La detección temprana de problemas y los ajustes rápidos minimizan el riesgo de fracaso

- Valor Temprano: Genera retorno de inversión desde el primer Sprint completado

La parte inferior del diagrama representa el impacto estadístico y respalda esta transición con datos contundentes del Standish Group CHAOS Report, que revela que en el modelo Cascada:

- 70% de los proyectos exceden tiempo y presupuesto
- 50% no cumplen con los requisitos originales
- 30% son cancelados antes de terminar

Estas estadísticas demuestran empíricamente por qué la industria del software ha adoptado masivamente metodologías ágiles como Scrum.

El diagrama ilustra de manera clara y visual la evolución necesaria en el desarrollo de software: desde un modelo rígido, secuencial y propenso al fracaso, hacia un enfoque flexible, colaborativo y orientado al valor que caracteriza a Scrum. Esta transición no es solo una mejora técnica, sino una transformación fundamental en cómo las organizaciones abordan el desarrollo de productos digitales en un mundo empresarial cada vez más dinámico y competitivo.

La representación visual enfatiza que la adopción de Scrum no es simplemente una preferencia metodológica, sino una respuesta estratégica a las limitaciones inherentes y los riesgos documentados de la modelo Cascada tradicional.

### ***Complementariedad entre Kanban y Scrum***

Aunque Kanban y Scrum tienen enfoques diferentes, pueden complementarse para maximizar los beneficios en la transformación digital de la empresa:

- Visibilidad y flexibilidad combinadas: Mientras que Kanban proporciona una visión clara del flujo de trabajo, Scrum introduce estructura y cadencia a través de los sprints. Juntas, estas metodologías permiten a los equipos

mantenerse organizados mientras responden de manera flexible a los cambios.

- Mejora continua: Las metodologías promueven la retroalimentación constante y mejora permanente.
- Colaboración entre departamentos: Kanban y Scrum fomentan la cooperación entre unidades y grupos.

### ***Relación con CMMI: Un Enfoque Híbrido***

El estándar Capability Maturity Model Integration CMMI complementa las metodologías ágiles al proporcionar un marco estructurado para la gestión de procesos y la calidad del software. Al integrar Scrum y Kanban con CMMI, la empresa puede:

- Registrar y unificar los procedimientos: Kanban y Scrum son más flexibles, mientras que CMMI asegura que los procesos críticos se documenten de manera apropiada y satisfagan los estándares de calidad establecidos.
- Calcular y analizar el avance: CMMI establece indicadores específicos para analizar la madurez de una organización, lo que posibilita a la compañía calcular el efecto de poner en práctica Kanban y Scrum en cuanto a satisfacción del cliente, calidad y productividad.
- Disminuir los riesgos: La posibilidad de que el software falle disminuye al asegurar que los requisitos se manejen de forma eficaz y que las pruebas sean minuciosas, lo cual es posible gracias a la fusión de metodologías ágiles y CMMI.

La puesta en práctica Kanban y Scrum combinados con el estándar CMMI constituye una solución para tratar los problemas diagnosticados en el desarrollo de producto de la empresa. Se espera que la empresa esté en una mejor posición para competir en un mercado dinámico si implementa estas prácticas, afín de garantizar la satisfacción de sus clientes y además de sus colaboradores.

## Propuesta

La propuesta presenta un marco de trabajo integral que combina las metodologías ágiles Scrum y Kanban con el estándar de calidad CMMI, acompañado de un programa de transformación cultural, un sistema de gestión del conocimiento y un plan de capacitación. Este enfoque busca optimizar los procesos de desarrollo de software, mejorar la calidad del producto y fomentar una cultura organizacional colaborativa dentro de la organización (Anderson, 2010).

A continuación, se detalla cómo se implementarán estas metodologías y las herramientas específicas que se utilizarán:

### Implementación de Scrum

Adoptar un enfoque iterativo e incremental para el desarrollo de software, promoviendo entregas frecuentes y ajustes rápidos basados en retroalimentación.

- Herramientas:

Jira: Se utilizará Jira como la principal herramienta para gestionar los sprints, el backlog del producto y las tareas asignadas al equipo. Jira permitirá visualizar el progreso del equipo a través de tableros Kanban y diagramas de Gantt personalizados.

Diagramas de Gantt: Se integrarán diagramas de Gantt (a través de plugins en Jira o herramientas externas como Microsoft Project) para visualizar las dependencias entre tareas y asegurar la correcta planificación temporal.

- Prácticas clave:

Realización de reuniones diarias (Daily Stand-ups) para revisar el progreso del equipo.

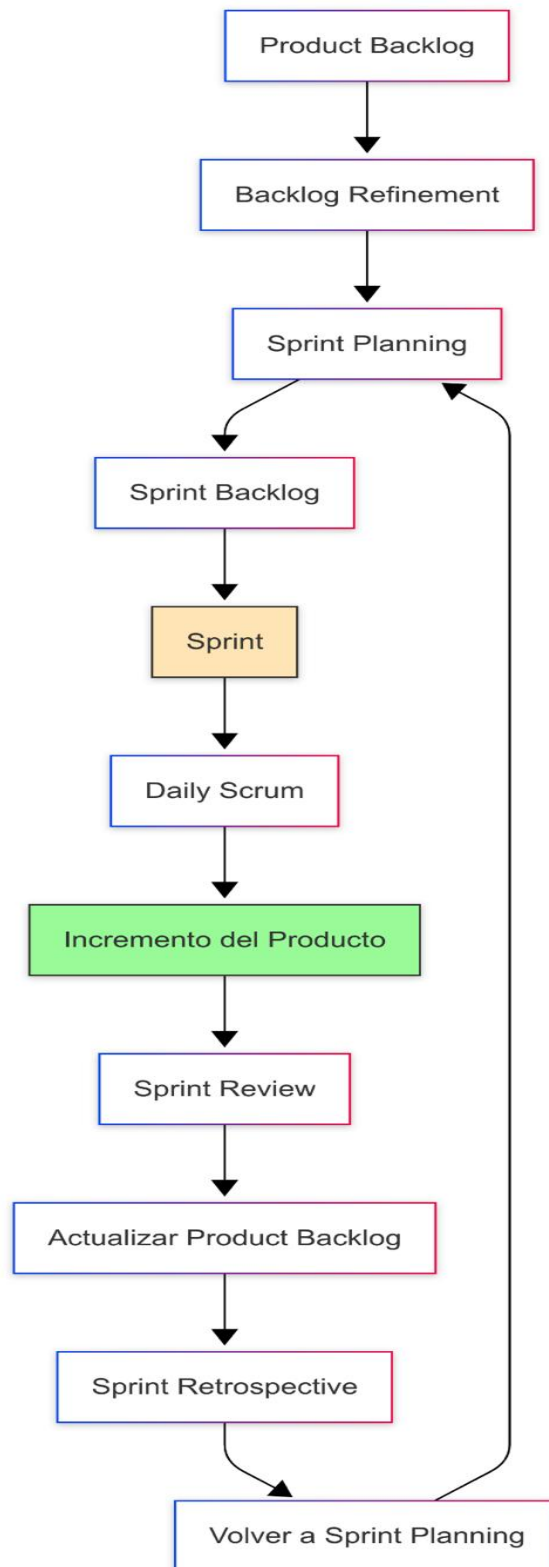
Planificación de sprints cada dos semanas o mensualmente, en función de cuán complicado sea el proyecto.

Al final de cada sprint, se realizan retrospectivas para determinar los aspectos que necesitan mejorarse y hacer ajustes en los procedimientos.

Scrum es un marco de trabajo que se basa en la iteración. El equipo, utilizando una lista priorizada de mejoras llamada product backlog, planifica cada sprint (que puede durar entre 1 y 4 semanas) eligiendo tareas para el sprint backlog y estableciendo un objetivo para el mismo. Se llevan a cabo daily scrums durante el sprint para coordinar los progresos y modificar el plan. Al finalizar, se obtiene un incremento de producto utilizable que se revisa en sprint review con los interesados para recibir retroalimentación, y el equipo reflexiona en la *sprint retrospective* para mejorar el siguiente ciclo.

Un modo de visualizar lo dicho, se encuentra en la Figura 4, ilustrando que Scrum es un marco ágil que organiza el trabajo en *sprints* cortos 1 a 4 semanas para entregar incrementos funcionales del producto. Parte de un *product backlog* priorizado, que se refina antes de cada ciclo. En el *sprint planning* se define el objetivo y plan de trabajo *sprint backlog*. Durante el *sprint*, el equipo trabaja de forma colaborativa, revisa avances en el *daily scrum*, presenta resultados en el sprint review y mejora procesos en la *retrospective*, repitiendo el ciclo con enfoque en transparencia, inspección y adaptación.

**Figura 4 Otro modo de visualizar Scrum**



**Nota.** Diagrama simplificado del proceso Scrum

Antes de presentar la Tabla 3 sobre el ciclo iterativo de Sprints, es importante contextualizar que este enfoque divide el desarrollo en etapas cortas y enfocadas, cada una con un objetivo claro Sprint Goal, un tiempo estimado y entregables específicos. Este esquema permite trabajar de forma incremental, validando resultados al final de cada Sprint y ajustando el plan según las necesidades del proyecto. De esta manera, se asegura un avance constante, controlado y adaptable, garantizando que cada iteración aporte valor tangible al producto final.

Se describen los títulos de cada columna de la Tabla 3.

- No. Sprint: Número secuencial del Sprint.
- Sprint Goal: Objetivo claro y medible para cada Sprint.
- Tiempo Estimado Promedio - Días: Duración estimada del Sprint generalmente entre 1 y 4 semanas.
- Entregables: Resultados tangibles que se deben entregar al final del Sprint.

**Tabla 3 Tabla genérica de Ciclo Iterativo de Sprints**

No. Sprint	Sprint Goal	Tiempo Estimado Promedio (Días)	Entregables
Sprint 1	Iniciar con la implementación del módulo principal	10	<ul style="list-style-type: none"><li>● Prototipo funcional listo</li><li>● Guía técnica inicial</li></ul>
Sprint 2	Conectar con sistemas externos	10	<ul style="list-style-type: none"><li>● API conectada</li><li>● Pruebas de integración completadas</li></ul>
Sprint 3	Crear funciones avanzadas	10	<ul style="list-style-type: none"><li>● Nuevas características clave</li><li>● Interfaz de usuario mejorada</li></ul>
Sprint 4	Mejorar el rendimiento	5	<ul style="list-style-type: none"><li>● Código optimizado</li><li>● Pruebas de carga y estabilidad</li></ul>
Sprint 5	Validar y ajustar	2	<ul style="list-style-type: none"><li>● Test con casos reales</li><li>● Documentación final actualizada</li></ul>

**Nota.** Ciclo Iterativo de Sprints

La Tabla 4 presenta un ejemplo adaptado con base en la Tabla 3 que trata el desarrollo de un servicio de e-commerce. El ciclo de trabajo se organiza en Sprints. Cada Sprint tiene un objetivo específico, un tiempo promedio de ejecución y entregables.

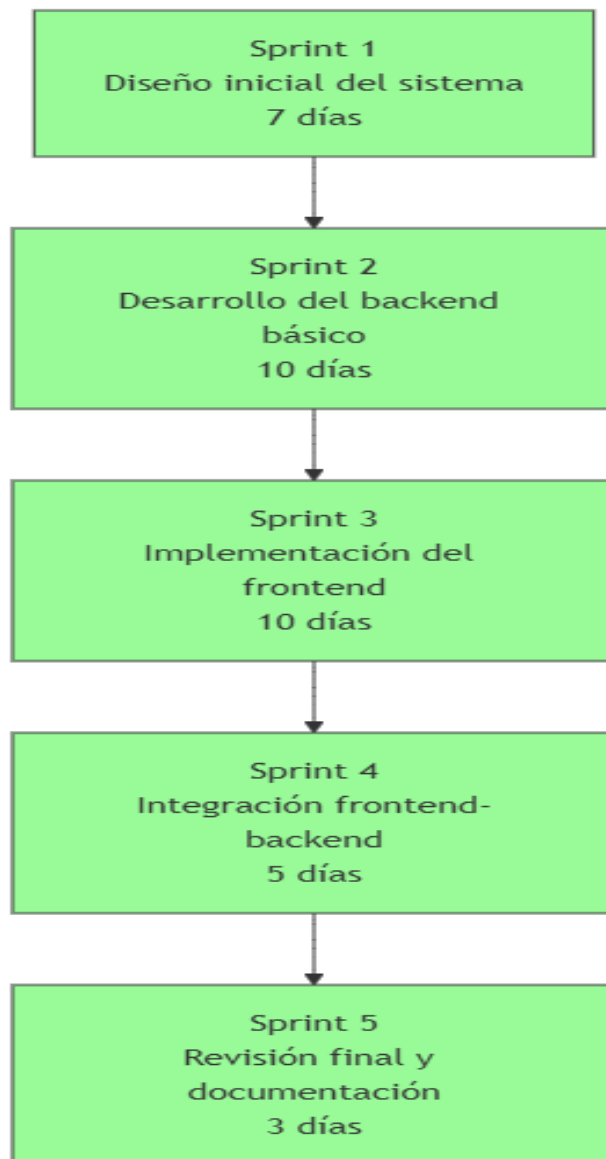
**Tabla 4 Ejemplo Adaptado de servicio E-COMMERCE**

No. Sprint	Sprint Goal	Tiempo Estimado Promedio (Días)	Entregables
Sprint 1	Diseño inicial del sistema.	7	<ul style="list-style-type: none"><li>- Maqueta del diseño de usuario (UI/UX).</li><li>- Especificaciones técnicas iniciales.</li></ul>
Sprint 2	Desarrollo del backend básico.	10	<ul style="list-style-type: none"><li>- API REST funcional.</li><li>- Base de datos configurada.</li></ul>
Sprint 3	Implementación del frontend.	10	<ul style="list-style-type: none"><li>- Interfaz de usuario completa.</li><li>- Pruebas unitarias del frontend.</li></ul>
Sprint 4	Integración frontend-backend.	5	<ul style="list-style-type: none"><li>- Comunicación bidireccional entre frontend y backend.</li><li>- Pruebas de integración.</li></ul>
Sprint 5	Revisión final y documentación.	3	<ul style="list-style-type: none"><li>- Resolución de bugs críticos.</li><li>- Documentación técnica detallada.</li></ul>

**Nota.** Adaptación del Servicio Ecommerce

En la Figura 6 se encuentra una ilustración de ejemplo adaptado, donde cada Sprint entrega componentes funcionales que van construyendo progresivamente el sistema completo hasta llegar a un producto terminado y documentado.

**Figura 5 Ilustración de ejemplo adaptado**



**Nota.** Ejemplo adaptado

Al momento de entrega del Producto Final, este ya cuenta con la aprobación del propio cliente, lo que hace que el Flujograma ya tenga que terminarse en su flujo como tal.

### **Implementación de Kanban**

Esto busca optimizar el flujo de trabajo mediante la visualización de tareas y la limitación del trabajo en progreso *WIP Working in Progress*, reduciendo cuellos de botella y mejorando la eficiencia operativa.

En la Figura 5, se considera Jira con metodología Kanban, dado que permite implementar prácticas clave que optimizan la gestión del trabajo y mejoran la productividad del equipo. Entre ellas, destaca la visualización del flujo de trabajo en tiempo real, lo que permite a todos los miembros conocer el estado de cada tarea y priorizar acciones de forma eficiente. Asimismo, se establece un límite de trabajo en progreso WIP, para asegurar que las tareas en curso se completen antes de iniciar nuevas, evitando la dispersión de esfuerzos. Finalmente, se realiza una revisión constante del tablero Kanban, identificando posibles bloqueos y aplicando ajustes que optimicen el flujo y la entrega de valor en cada etapa del proceso.

## Figura 6 Flujo de Trabajo

Proyectos / Cliente 01 E-Commerce

### Tablero NE

Resumen Cronograma **Tablero de kanban** Calendario Informes Lista Formularios Metas Todas las actividades Componentes Código Publicaciones More 4 +

Buscar tablero Epic Tipo Etiqueta Filtros rápidos Grupo: Consultas

BACKLOG 2	SELECCIONADO PARA DESARROLLO 3	EN CURSO 8	LISTO 3
<p>Vital para la Empresa</p> <p>8 dic 2025</p> <p>NE-1</p>	<p>4.1 Despliegue</p> <p>4. DESPLIEGUE Y MANTENIMIENTO</p> <p>NE-14</p>	<p>2. Diseño del Software</p> <p>2. DISEÑO DEL SOFTWARE</p> <p>NE-3</p>	<p>1. Ingeniería de Requerimientos</p> <p>1. INGENIERÍA DE REQUERIMIENTOS</p> <p>NE-2</p>
<p>Resumen Marcado</p> <p>1. INGENIERÍA DE REQUERIMIENTOS</p> <p>8 dic 2025</p> <p>NE-16</p>	<p>4. Despliegue y Mantenimiento</p> <p>4. DESPLIEGUE Y MANTENIMIENTO</p> <p>NE-5</p>	<p>1.3 Documentación</p> <p>1. INGENIERÍA DE REQUERIMIENTOS</p> <p>NE-8</p>	<p>1.2 Análisis</p> <p>1. INGENIERÍA DE REQUERIMIENTOS</p> <p>NE-7</p>
<p>+ Crear</p>	<p>4.2 Mantenimiento</p> <p>4. DESPLIEGUE Y MANTENIMIENTO</p> <p>NE-15</p>	<p>1.4 Gestión</p> <p>1. INGENIERÍA DE REQUERIMIENTOS</p> <p>NE-9</p>	<p>1.1 Identificación</p> <p>1. INGENIERÍA DE REQUERIMIENTOS</p> <p>NE-6</p>
		<p>2.1 Arquitectura de Software</p> <p>2. DISEÑO DEL SOFTWARE</p> <p>NE-10</p>	<p>Q Ver actividades más antiguas</p>
		<p>3.1 Desarrollo de Código</p> <p>3. IMPLEMENTACIÓN</p> <p>NE-12</p>	
		<p>2.2 Diseño de Componentes</p> <p>2. DISEÑO DEL SOFTWARE</p> <p>NE-11</p>	

Quickstart

**Nota.** Flujo de trabajo considerando Jira con metodología Kanban

- Prácticas clave:
  - Visualización del flujo de trabajo en tiempo real.
  - Establecimiento de límites de WIP para garantizar que las tareas se completen antes de iniciar nuevas actividades.
  - Revisión constante del tablero Kanban para identificar bloqueos y optimizar el proceso.

## **Integración con CMMI**

Hay que asegurar que los procesos de desarrollo cumplan con estándares de calidad reconocidos a nivel internacional, utilizando CMMI como marco estructurado para la mejora continua.

- Herramientas:

Jira + CMMI Plugins: Se integrarán plugins específicos en Jira para alinear los procesos ágiles con los requisitos de CMMI, como la documentación formal de requisitos y pruebas.

Documentación en Google Docs: Se utilizará Google Docs para almacenar y gestionar la documentación formal requerida por CMMI, incluyendo planes de proyecto, reportes de calidad y métricas históricas.

- Prácticas clave:

Auditorías internas periódicas para evaluar el nivel de madurez organizacional según CMMI.

Documentación detallada de los procesos, desde la ingeniería de requerimientos hasta el mantenimiento del software.

Uso de métricas clave para medir el impacto de la implementación de CMMI en la calidad del software.

## **Gestión del Conocimiento y Transformación Cultural**

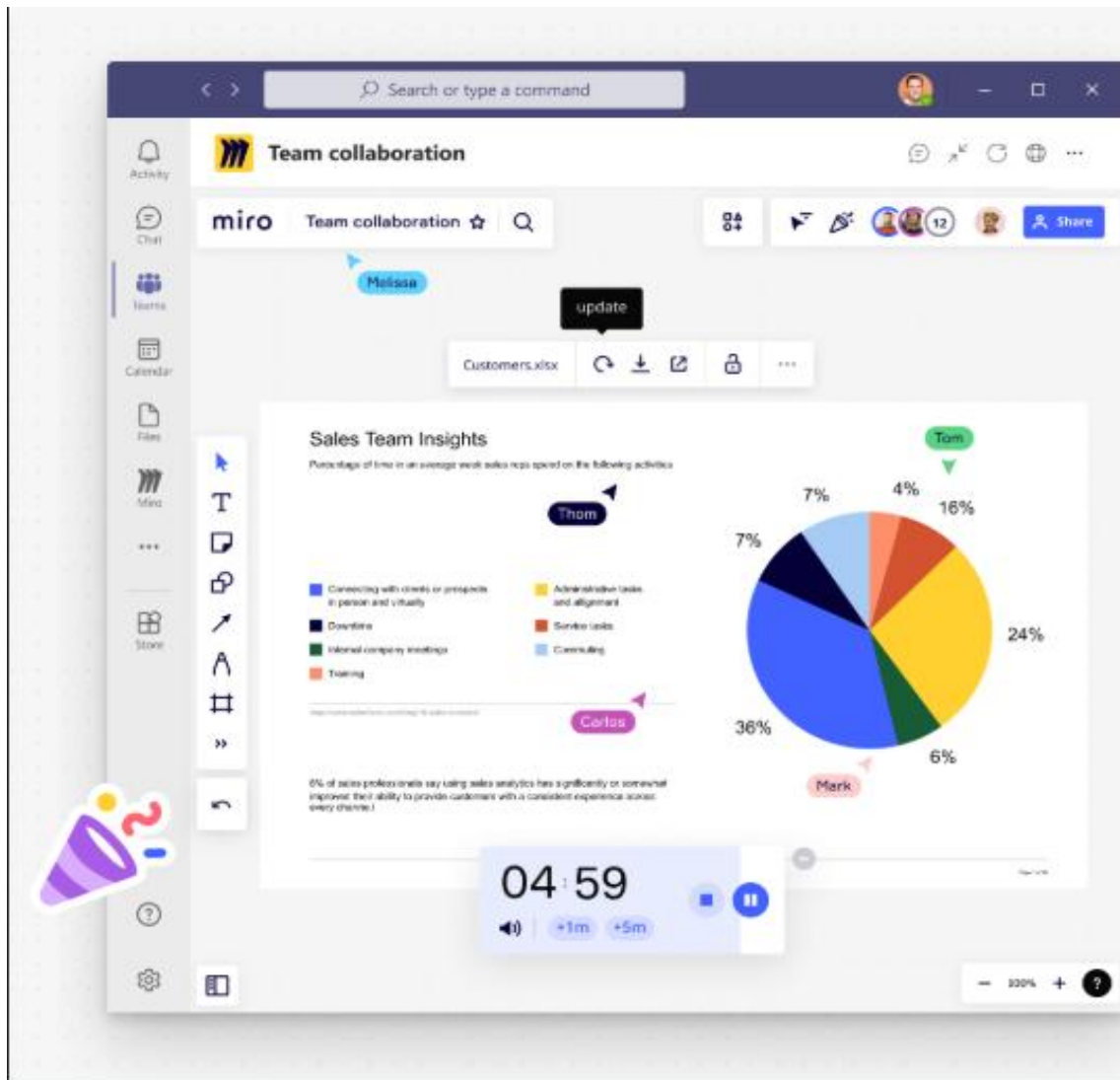
Fomentar una cultura de colaboración, innovación y mejora continua, asegurando que el conocimiento generado durante el proyecto sea capturado, almacenado y compartido de manera efectiva.

- **Herramienta Principal:**

Microsoft Teams: será la herramienta central para facilitar la comunicación interdepartamental, fomentar la colaboración en tiempo real y centralizar el conocimiento. A través de canales específicos para cada equipo y proyecto, se podrá compartir información, documentar lecciones aprendidas y organizar reuniones virtuales. Además, su integración con SharePoint permitirá crear bases de datos accesibles sobre mejores prácticas, soluciones a problemas comunes y documentación formal requerida por CMMI.

La Figura 7 muestra un ejemplo de captura de imagen con la herramienta Microsoft Teams.

**Figura 7 MS-Teams**



**Nota.** Trabajo en Microsoft Teams

## Plan de Capacitación

Hay que asegurar que el equipo tenga las habilidades necesarias para adoptar las nuevas metodologías y herramientas.

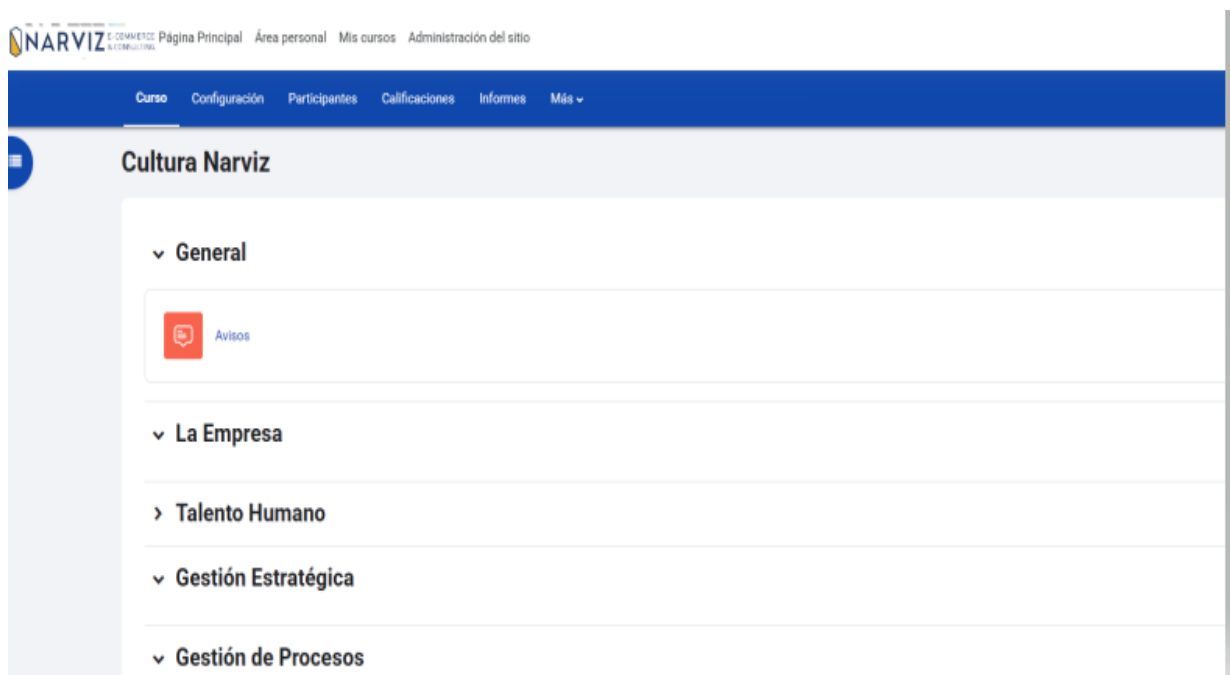
- Herramienta Principal:

Moodle en Cloud Privado: Moodle será la plataforma de capacitación principal, implementada en un entorno de cloud privado para garantizar seguridad y personalización. Esta herramienta permitirá

diseñar cursos específicos sobre Scrum, Kanban, CMMI y otras prácticas ágiles, así como evaluar el progreso del equipo mediante pruebas y encuestas. Su flexibilidad permitirá adaptar los contenidos a las necesidades del equipo y ofrecer capacitaciones continuas.

La Figura 8 muestra un ejemplo de captura de imagen con la herramienta Moodle en Cloud Privado.

**Figura 8 Moodle**



. **Nota.** Gestor de aprendizaje Moodle

- Prácticas clave:

Evaluación inicial del nivel de conocimiento del equipo mediante encuestas y pruebas diagnósticas.

Diseño de un plan de capacitación personalizado basado en las necesidades identificadas.

Monitoreo del avance mediante indicadores de rendimiento, como la implementación práctica de lo aprendido y la asistencia a cursos.

El marco de trabajo para la implementación exitosa de Scrum, Kanban y CMMI comprende las herramientas Microsoft Teams, Google Forms, Power BI, Moodle en Cloud Privado, y Jira. La propuesta comprende un programa de formación y transformación cultural con el fin de vencer la resistencia al cambio e impulsar una mentalidad de mejora constante. La compañía podrá optimizar sus procesos, elevar la calidad del software y continuar siendo competitiva si emplea estas metodologías y herramientas en conjunto.

La Tabla 5 que a continuación se visualiza, trata sobre las Herramientas Propuestas, que alimentan la validación tratada posteriormente:

**Tabla 5 Resumen de Herramientas Propuestas**

Área	Herramienta Principal	Propósito y Prácticas Clave
Aplicación de Scrum	Jira	- Administrar sprints, backlog y asignación de tareas.
		- Representar el avance mediante tableros Kanban y diagramas de Gantt.
		- Realizar reuniones diarias (Daily Stand-ups) y sesiones de retrospectiva.
Empleo de Kanban	Tablero Kanban en Jira	- Para que el flujo de trabajo se pueda ver, hace falta establecer tableros Kanban.
		- Para evitar sobrecargas, se deben fijar límites en el trabajo en progreso (WIP).
		- Analizar el tablero de manera continua para detectar bloqueos o cuellos de botella.
Integración con el CMMI	Jira y sus extensiones CMMI e IBM WEWM	- Modificar los procedimientos ágiles de acuerdo con las pautas de CMMI.
		- Registrar de forma organizada las métricas, los requisitos y las pruebas del pasado.
		- Llevar a cabo auditorías internas y analizar su incidencia o efecto sobre la calidad del software.
Gestión del Conocimiento	Microsoft Teams	- Fomentar el trabajo en equipo en tiempo real y la comunicación entre las áreas.
		- Crear canales temáticos para equipos y proyectos particulares.
		- Consolidar información a través de integraciones con Share Point.
Plan de Formación	Moodle en Nube Privado	- Desarrollar cursos individuales acerca de CMMI, Kanban y Scrum.
		- Evaluar de forma inicial el nivel de conocimiento del equipo.
		- Dar seguimiento al aprendizaje con indicadores de desempeño.
Monitoreo y Validación	Power BI	- Construir paneles de control que reflejen métricas clave (tiempo de entrega, defectos, etc.).

Área	Herramienta Principal	Propósito y Prácticas Clave
Encuestas de seguimiento	Google Forms	- Para realizar encuestas periódicas que midan la percepción del equipo sobre los cambios implementados.

**Nota.** Herramientas propuestas

## Validación

Esta parte tiene como objetivo medir el impacto de la implementación de Scrum, Kanban y CMMI en los procesos y resultados de la empresa.

La validación de la propuesta se realizó considerando las percepciones, expectativas y disposición al cambio de los involucrados en el área de sistemas de la empresa. Inicialmente, se llevaron a cabo entrevistas semiestructuradas para recopilar información sobre su conocimiento previo de las metodologías ágiles, sus expectativas personales frente a una posible transformación organizacional y su disposición para participar activamente en el proceso de cambio. Posteriormente, tras la presentación de la propuesta de transformación hacia metodologías ágiles y la explicación de cómo se los piensa involucrar en este proceso, se recibieron comentarios muy positivos que reflejan apertura y motivación frente a la iniciativa. Para complementar esta visión, y obtener un punto de vista distinto al de los actores directamente implicados en la transformación digital, se realizó además una entrevista a expertos.

### Validación Externa de la Propuesta con Expertos

Con el fin de garantizar la rigurosidad académica y la viabilidad práctica del modelo propuesto, se realizó una entrevista final con dos expertos en transformación digital y gestión ágil de proyectos (ver Anexo 3), diferentes a los participantes del diagnóstico inicial.

Los especialistas, con más de una década de experiencia en consultoría para la transformación ágil en firmas de software latinoamericanas, valoraron favorablemente la propuesta, resaltando su balance entre disciplina CMMI y flexibilidad ágil. Uno de ellos señaló: "Integrar CMMI con Scrum y Kanban es un

enfoque avanzado y poco común, pero muy adecuado para empresas que buscan certificación de calidad sin perder agilidad."

El otro experto manifiesta: "Totalmente de acuerdo. Lo que me parece innovador es el uso de herramientas como Jira para alinear los procesos ágiles con los artefactos de CMMI", es decir, está de acuerdo porque la base de todo desarrollo en progreso es definitivamente es la cultura y la práctica cotidiana de lo que se establece

Esta validación externa aporta una perspectiva crítica independiente, asegurando que la propuesta no solo responda a necesidades internas, sino que también esté alineada con buenas prácticas del sector y estándares internacionales.

### ***Sistemas de indicadores para validar el resultado del modelo propuesto***

El objetivo es establecer un sistema de validación basado en parámetros cuantitativos KPI y cualitativos que permita medir el impacto de la implementación de metodologías ágiles Scrum y Kanban y el marco CMMI en la madurez organizacional del área de desarrollo de software.

Se diseñó un sistema de validación que emplea una mezcla de indicadores cualitativos y cuantitativos para analizar, de manera rigurosa y sistemática, el efecto que la puesta en práctica de las metodologías Kanban y Scrum junto con CMMI tienen sobre la madurez del área dedicada al desarrollo de software. El sistema posibilita el seguimiento del progreso, la detección de áreas que necesitan mejorar y la confirmación de que las prácticas adoptadas están alineadas con los objetivos estratégicos de la empresa.

Los Indicadores Clave de Desempeño KPI seleccionados abarcan dimensiones críticas como la gestión de proyectos, la calidad del software, la colaboración interna, la madurez organizacional y el desarrollo del capital humano. Cada KPI ha sido definido con claridad, especificando su propósito, frecuencia de medición y responsables directos, lo que asegura su seguimiento oportuno y su integración en los ciclos de retroalimentación continua. La estructura facilita que la toma de decisiones se tomen a partir de pruebas y

permite comprobar si la incorporación de modelos de madurez y enfoques ágiles produce una transformación real sostenible.

En la Tabla 6 se presenta el detalle de los KPI propuestos, su definición, periodicidad y responsables, constituyendo así un marco de monitoreo integral para la evaluación del impacto organizacional.

### ***Plan de Acción con Indicadores Integrados***

Con el fin de asegurar que la ejecución del marco integrado, que une las metodologías ágiles Scrum y Kanban con los principios de madurez del modelo CMMI, sea eficaz, cuantificable y sostenible, se ha establecido un plan de acción organizado en etapas sucesivas. Este enfoque por etapas hace posible que el modelo se valide en un ambiente controlado, después se ajuste con base en la información empírica y las opiniones del equipo, y finalmente se escale de manera gradual y segura a toda la organización.

El plan se articula en tres fases clave: Piloto, Escalamiento y Consolidación, cada una con objetivos específicos, acciones concretas, responsables definidos y métricas asociadas directamente a los KPI establecidos previamente. Este modelo no solo hace más fácil la administración del cambio organizacional, sino que además posibilita una validación constante de cómo el modelo afecta la eficacia, calidad y madurez del grupo de desarrollo.

El plan también abarca sistemas de monitoreo y corrección en tiempo real, lo que asegura que cualquier obstáculo o desviación detectada sea abordada de manera proactiva. La inclusión de indicadores cualitativos y cuantitativos en cada etapa convierte este plan en un sistema dinámico de mejora incesante, que sigue los principios de la excelencia y del desarrollo ágil que fomenta CMMI.

**Tabla 6 Indicadores Clave de Desempeño**

Área	KPI	Definición	Frecuencia de Medición	Responsables
Gestión de Proyectos	Velocidad del Equipo	Número promedio de puntos de historia completados por sprint	Mensual	Scrum Master, Product Owner
	Tiempo Promedio de Entrega	Tiempo desde que una tarea entra en proceso hasta su finalización	Quincenal	Scrum Master
Calidad del Software	Tasa de Defectos por Línea de Código	Número de defectos encontrados por cada 1000 líneas de código	Mensual	QA Lead, Tester
	Porcentaje de Pruebas Automatizadas	Proporción de pruebas automatizadas sobre total de pruebas	Trimestral	QA Lead, DevOps
Colaboración y Comunicación	Satisfacción del Equipo (NPS Interno)	Índice de satisfacción obtenido mediante encuesta Likert mensual	Mensual	Coach Ágil
	Número de Bloqueos Resueltos por Sprint	Total de impedimentos resueltos durante el sprint	Semanal	Scrum Master
Madurez Organizacional	Nivel de Madurez Alcanzado	Evaluación periódica del nivel CMMI alcanzado	Semestral	Consultor CMMI
	Cumplimiento de Procesos	% de procesos documentados y aplicados según estándares CMMI	Mensual	Coordinador de Calidad
Capacitación	Participación en Capacitaciones	% de personal que asiste a sesiones de formación	Final de cada módulo	RRHH, Coach Ágil
	Aplicabilidad de Conocimientos Adquiridos	Autoevaluación de los participantes sobre aplicación práctica de lo aprendido	Post-capacitación	RRHH, Gerente de Proyecto

**Nota.** Indicadores de desempeño

A continuación, se detalla cada fase del Plan de Acción con Indicadores Integrados, cuya ampliación se dispone en el Anexo 5, junto con sus acciones, responsables, métricas y mecanismos de seguimiento.

### ***Fase 1: Piloto 3-6 meses***

Implementar el marco integrado (Scrum + Kanban + CMMI) en un equipo pequeño para validar su funcionamiento y ajustar antes del escalamiento.

La primera etapa del plan de implementación consiste en un proyecto piloto que se desarrollará durante un periodo de 3 a 6 meses con un equipo pequeño y multifuncional. La finalidad de esta etapa es comprobar la viabilidad del marco integrado, que abarca Scrum, Kanban y CMMI, en un ambiente real con una extensión limitada. Facilita la identificación de los primeros retos, la experimentación con técnicas y herramientas laborales y la creación de pruebas concretas sobre el impacto del modelo antes de que se difunda.

La Tabla 7 brinda información precisa acerca de las acciones fundamentales, los indicadores de seguimiento y los encargados designados para garantizar una ejecución que sea eficaz y medible.

### ***Fase 2: Escalamiento (6-12 meses)***

Extender el modelo a otros equipos, ajustando procesos según retroalimentación y métricas obtenidas en la fase piloto.

Tras la validación del modelo en la fase piloto, se inicia la etapa de escalamiento, que abarca un periodo de 6 a 12 meses. Durante esta fase, el marco integrado se extiende progresivamente a otros equipos de desarrollo, incorporando las lecciones aprendidas y los ajustes derivados del análisis de métricas y retroalimentación.

**Tabla 7 Fase 1: Piloto (3–6 meses)**

<b>Acciones Específicas</b>	<b>Métricas Asociadas</b>	<b>Responsables</b>
Dar inicio a talleres formativos	Nivel de asistencia a las capacitaciones	Scrum Master, Coach Ágil, Equipos multifuncionales
Configurar tableros Kanban y ciclos de Scrum, con revisiones semanales	Productividad o velocidad del equipo	Scrum Master, Equipos multifuncionales
Implementar prácticas de CMMI para registrar requisitos y ejecutar pruebas continuas	Porcentaje de defectos detectados	Coach Ágil, QA Lead
Registrar avances en Confluence, e incluir aprendizajes obtenidos	Grado de satisfacción del equipo (encuestas mensuales)	Equipos multifuncionales, Coach Ágil

**Nota.** Piloto fase 1

El enfoque se centra en la estandarización de procesos ágiles, la adaptación del modelo a diferentes contextos internos y el fortalecimiento de la capacidad organizacional mediante formación continua.

La Tabla 8 resume las acciones estratégicas, los indicadores clave de desempeño asociados y los responsables encargados de liderar esta transición a mayor escala.

**Tabla 8 Fase 2: Escalamiento (6–12 meses)**

<b>Acciones Específicas</b>	<b>Métricas Asociadas</b>	<b>Responsables</b>
Revisar resultados del piloto y ajustar herramientas/procesos	Tiempo promedio de entrega	2 Scrum Masters, Coach Ágil
Realizar nuevas rondas de capacitación adaptadas al contexto de nuevos equipos	Aplicabilidad de conocimientos adquiridos	RRHH, Coach Ágil
Establecer tableros Kanban y ciclos de Scrum estandarizados	Porcentaje de pruebas automatizadas	Scrum Masters, DevOps
Continuar evaluaciones de madurez bajo CMMI	Nivel de madurez alcanzado (CMMI)	Consultor CMMI, Coordinador de Calidad

**Nota.** Fase de Escalamiento

**Fase 3: Consolidación 12+ meses**

Estandarizar procesos y cultura ágil en toda la organización, asegurando sostenibilidad mediante auditorías periódicas.

La última etapa del plan, conocida como consolidación, tiene lugar desde el mes número 12 y se enfoca en la maduración y sostenibilidad del modelo que se ha implementado. En este momento, la atención va de la implementación activa hacia la normalización institucional y el seguimiento constante. Consolidar una cultura ágil, garantizar el cumplimiento de las normas CMMI y establecer procesos de mejora continua para facilitar la evolución autónoma de la entidad

es el propósito. Para ello, se implementan auditorías periódicas, se refuerzan las prácticas de colaboración y se mantienen los indicadores clave de rendimiento bajo un control constante.

La Tabla 9 presenta las acciones centrales de esta fase, junto con sus métricas asociadas y los responsables claves encargados de su supervisión y sostenibilidad a largo plazo.

**Tabla 9 Fase 3: Consolidación 12+ meses**

<b>Acciones Específicas</b>	<b>Métricas Asociadas</b>	<b>Responsables</b>
Estandarizar herramientas, procesos y roles en toda la organización	Cumplimiento de procesos documentados	Coordinador de Calidad, Dirección General
Realizar auditorías internas semestrales para evaluar cumplimiento CMMI	Evaluación de madurez CMMI	Consultor CMMI, Coordinador de Calidad
Reforzar la cultura de colaboración y mejora continua	Retroalimentación continua (retrospectivas)	Coach Ágil, Equipos multifuncionales

**Nota.** Fase de consolidación

### **Monitoreo y Ajustes Continuos**

Se aplican Mecanismos de Seguimiento que incluyen retrospectivas semanales para identificar mejoras inmediatas, encuestas mensuales de satisfacción del equipo y stakeholders, y tableros de control digitales (Jira, Google Docs) para la visualización en tiempo real de los KPIs. Además, se

realiza una revisión quincenal de métricas para garantizar decisiones basadas en datos y retroalimentación continua.

Estos mecanismos alimentan los Ajustes necesarios durante la implementación, como la reconfiguración de flujos de trabajo ante cuellos de botella, el rediseño de estrategias de capacitación si la aplicabilidad es baja, o la intervención temprana ante aumentos en la tasa de defectos o caídas en la velocidad del conjunto de individuos. Este ciclo de seguimiento y ajuste garantiza una implementación centrada en la mejora continua, dinámica y sostenible.

## Conclusiones

La implementación del marco integrado basado en Scrum, Kanban y el modelo de madurez CMMI se presenta como una solución viable y efectiva para superar las limitaciones del modelo en cascada vigente en la empresa. El diagnóstico evidenció problemas críticos: baja visibilidad del progreso, comunicación fragmentada, resistencia pasiva al cambio y ausencia de flujos de trabajo estandarizados. Estos factores generan retrasos, retrabajos y disminuyen la calidad del producto final.

Las entrevistas y los análisis confirmaron que el equipo tiene una actitud moderada hacia el cambio, siempre y cuando haya apoyo y formación práctica. La adopción de un nuevo modelo laboral tiene más posibilidades de éxito porque las expectativas expresadas por los colaboradores están alineadas con los beneficios de las metodologías ágiles.

Mediante una entrevista que se encuentra en el Anexo 3, confirmada el 10 de mayo del 2025, verificó la fortaleza e innovación del enfoque híbrido que fusionan Scrum, Kanban y CMMI. Los especialistas elogiaron la estructura de tres fases: piloto, escalamiento, consolidación, por su perspectiva gradual y realista. Destacaron la solidez plan de capacitación apoyado por Moodle, así como la consistencia en la vinculación entre las herramientas y la trazabilidad de los requisitos, como Jira. Se reconoció como efectiva la selección de herramientas para una implementación ágil y centrada en datos, por ejemplo Google Forms para realizar encuestas y Power BI para obtener métricas en tiempo real. Como sugerencias, propusieron establecer un comité trimestral de transformación, incluir “champions ágiles” en el equipo y añadir la automatización y CI/CD a la etapa de consolidación con el fin de fortalecer la viabilidad y sostenibilidad del planteamiento.

El plan sugerido, que se compone de etapas de piloto, escalamiento y consolidación, incluye estrategias para administrar el conocimiento, así como formación continua e instrumentos tecnológicos. Todo esto hace posible acortar los plazos de entrega, bajar la tasa de fallos y elevar la satisfacción equipo y del cliente. La combinación de prácticas ágiles con estándares CMMI aseguran un balance entre flexibilidad y control, impulsando una transformación organizacional sostenible.

## **Recomendaciones**

Implementar gradualmente el plan de adopción de metodologías ágiles, empezando por un proyecto piloto regulado que permita la verificación de herramientas, prácticas y métricas antes de escalar.

Crear un programa de capacitación continua que se centre en Kanban, Scrum y CMMI, priorizando la implementación práctica y directa en los proyectos actuales.

Para mejorar la comunicación, el seguimiento y la visibilidad de los procesos, emplear herramientas de colaboración centralizadas como Moodle, Jira y Microsoft Teams.

Fomentar una cultura de colaboración y mejora constante, motivando la retroalimentación continua, la participación temprana de todos los departamentos y la integración de áreas que normalmente están separadas.

Hacer un seguimiento constante de los indicadores clave de desempeño (KPI), ajustando las estrategias en función de los resultados y asegurando que se satisfagan los estándares de calidad.

Para garantizar que el modelo se mantenga y que la madurez organizacional siga progresando, es necesario llevar a cabo auditorías internas regularmente.

## Anexo 1

Resúmenes de Entrevistas Semiestructuradas:

Se entrevistó al Gerente del Área de Sistemas, el 15 de abril del año en curso, a través de una videollamada grabada por zoom con una duración de 35 minutos.

### Resumen Ejecutivo:

El gerente reconoció la necesidad de modernizar los procesos actuales de desarrollo de software, indicando que la empresa enfrenta desafíos relacionados con la falta de colaboración entre equipos, resistencia al cambio y baja madurez metodológica. Considera que la adopción de metodologías ágiles puede ser una solución viable, aunque identificó una brecha en conocimiento técnico y cultural en el equipo.

### Preguntas, respuestas y análisis:

1. **¿Cuáles son sus expectativas personales respecto a una posible transformación cultural en la empresa?** “Espero que mejore la productividad y la calidad del software, y que los equipos trabajen mejor juntos.”

*Esta respuesta refleja una visión estratégica orientada hacia la mejora operativa y la colaboración entre equipos como resultado directo de la transformación cultural.*

2. **¿Está dispuesto a liderar e impulsar una iniciativa de cambio hacia metodologías ágiles?** “Sí, siempre que haya apoyo externo y capacitación adecuada.”

*Esto indica que, aunque existe voluntad de liderar el cambio desde la dirección, también hay reconocimiento de la necesidad de acompañamiento técnico y formativo para asegurar el éxito de la iniciativa.*

3. **¿Qué herramientas actuales de gestión de proyectos considera más útiles y cuáles deberían mejorarse o reemplazarse?** “Actualmente

usamos hojas de cálculo y correos electrónicos como base de la gestión, lo que no es eficiente. Creo que deberíamos adoptar herramientas como Jira o Trello para tener más visibilidad y control del flujo de trabajo.”

*Esto revela que la infraestructura actual de gestión es limitada y que hay apertura a migrar hacia plataformas más robustas y colaborativas.*

4. **¿Qué tipo de capacitación considera prioritaria para que los equipos adopten con éxito metodologías ágiles?** “En primer lugar, capacitación fundamental en Kanban y Scrum, y luego talleres prácticos. Además, requerimos formación en herramientas digitales para administración y seguimiento de tareas.”

*La respuesta deja claro que la capacitación debe abarcar el marco metodológico y el uso de herramientas para la ejecución.*

5. **¿Cómo evaluaría el triunfo de la aplicación de metodologías ágiles en la compañía?** “Según mi criterio, seríamos exitosos si mejoráramos la calidad de los entregables, disminuyéramos el tiempo de entrega y aumentáramos el grado de satisfacción del cliente.”

*Esta respuesta establece indicadores clave que podrán utilizarse para evaluar el impacto del cambio.*

Se entrevistó al Analista de Sistemas, el 18 de abril del año en curso, a través de una videollamada grabada por zoom con una duración de 40 minutos.

### **Resumen Ejecutivo:**

El analista de sistemas destacó la dificultad de gestionar cambios en los requisitos una vez iniciado el proyecto, lo que a menudo genera retrabajos y frustración. Expresó la necesidad de una mayor interacción con los usuarios finales y los equipos de desarrollo para asegurar que los requisitos sean claros y se mantengan actualizados. Se mostró abierto a nuevas metodologías que permitan una mayor flexibilidad y colaboración.

## **Preguntas y respuestas:**

- 1. ¿Cuál considera que es el principal desafío en la gestión de requisitos del proyecto?** "El mayor problema es que, una vez definidos al inicio, es muy complicado cambiarlos. Cuando el cliente solicita algo nuevo, esto implica retrabajos y genera bastante frustración en el equipo."
- 2. ¿Cómo es la comunicación con el cliente durante el desarrollo?** "Después de la fase inicial casi no tenemos contacto directo. Todo se gestiona a través del gerente de proyecto, lo que a veces retrasa la clarificación de dudas."
- 3. ¿Qué dificultades observa en la colaboración entre los equipos internos?** "En ocasiones desarrollo y pruebas no comprenden del todo lo que se requiere. Hay una desconexión que termina afectando la calidad y el tiempo de entrega."
- 4. ¿Está familiarizado con metodologías ágiles como Scrum o Kanban?** "He leído algo sobre Scrum, y me parece interesante el concepto de iteraciones cortas con retroalimentación constante. Creo que ayudaría a evitar sorpresas al final del proyecto."
- 5. ¿Qué cambios le gustaría implementar para mejorar el flujo de trabajo?** "Me gustaría un proceso más flexible, que permita ajustar requisitos sobre la marcha y que desde el inicio todos tengamos la misma comprensión de lo que se necesita."

Se entrevistó al Desarrollador 1 -Desarrollador Senior-, el 17 de abril del año en curso, a través de una videollamada grabada por zoom con una duración de 30 minutos.

## **Resumen Ejecutivo:**

El desarrollador describió el flujo de trabajo actual como lineal y poco flexible, sin retroalimentación constante durante el ciclo de desarrollo. Manifestó interés en aprender sobre metodologías ágiles si estas son bien explicadas y

útiles para su rol diario. También destacó la importancia de una mejor comunicación con el área de pruebas.

### **Preguntas, respuestas y hallazgos:**

1. **¿Conoce usted qué son las metodologías ágiles y cómo pueden aplicarse en su trabajo?** “Estoy familiarizado con las palabras Kanban y Scrum, pero desconozco exactamente qué son y de qué manera podrían integrarse en mi día a día.”

**Hallazgo:** Esto evidencia que existe un entendimiento superficial de las metodologías ágiles, lo que revela la importancia de una formación práctica para subsanar esta brecha de conocimiento, tal como se estipula en los objetivos del estudio.

2. **¿Estaría dispuesto a participar en una capacitación sobre metodologías ágiles si le ofrecieran la oportunidad?** “Por supuesto, siempre que sea clara y pertinente a lo que hago todos los días.”

**Hallazgo:** La actitud abierta hacia la capacitación indica una oportunidad para implementar formación práctica que aborde las necesidades específicas del desarrollador.

3. **¿Mejoraría mucho la calidad final productiva del producto?** “Sería útil poder aclarar dudas durante el desarrollo, no solo al final cuando los cambios son más complicados.”

**Hallazgo:** Esta respuesta subraya la necesidad de un proceso con retroalimentación continua, un principio clave de las metodologías ágiles.

4. **¿Qué espera personalmente de un proceso de transformación metodológica?** “Espero que facilite la comunicación con otros equipos y nos permita adaptarnos mejor a los cambios.”

**Hallazgo:** La expectativa de mayor colaboración y flexibilidad refleja una percepción de las limitaciones actuales y un interés en enfoques más dinámicos.

Se entrevistó al Desarrollador 2 -Desarrollador de Backend-, el 17 de abril del año en curso, a través de una videollamada grabada por zoom con una duración de 28 minutos.

### **Resumen Ejecutivo:**

El desarrollador señaló que el trabajo se realiza de forma fragmentada y con poca coordinación entre integrantes del equipo. Indicó que la falta de una herramienta centralizada para gestionar tareas provoca retrasos y duplicación de esfuerzos. Expresó interés en aprender metodologías ágiles siempre que se implementen gradualmente y con acompañamiento.

### **Preguntas, respuestas y hallazgos:**

1. **¿Conoce usted qué son las metodologías ágiles y cómo pueden aplicarse en su trabajo?** “Sé que existen cosas como Scrum, pero nunca las he usado ni entiendo bien cómo podrían ayudarme en mi trabajo.”

**Hallazgo:** La falta de experiencia práctica con metodologías ágiles confirma la necesidad de un programa de capacitación que explique su aplicación de manera relevante para el rol del desarrollador.

2. **¿Estaría dispuesto a participar en una capacitación sobre metodologías ágiles si se ofreciera?** “Sí, si está bien estructurada y me muestra cómo aplicarlo en mi día a día.”

**Hallazgo:** La disposición a aprender sugiere que una capacitación enfocada en ejemplos concretos podría ser bien recibida.

3. **¿Mejoraría mucho la calidad final productiva del producto?** “Me ayudaría mucho poder trabajar más de cerca con el equipo de diseño desde el inicio para evitar errores al final.”

**Hallazgo:** Esta respuesta destaca la importancia de la colaboración temprana, un aspecto que las metodologías ágiles pueden fortalecer.

4. **¿Qué espera personalmente de un proceso de transformación metodológica?** “Quiero un proceso que nos dé más flexibilidad y nos permita reaccionar rápido a los cambios del cliente.”

**Hallazgo:** La expectativa de mayor adaptabilidad indica una alineación natural con los principios ágiles, lo que facilita la adopción de estas metodologías.

Se entrevistó al Desarrollador 3 -Desarrollador de Frontend-, el 18 de abril del año en curso, a través de una videollamada grabada por zoom con una duración de 26 minutos.

### **Resumen Ejecutivo:**

El desarrollador describió el flujo de trabajo como rígido y dependiente de un documento inicial que no siempre refleja cambios posteriores. Señaló que la comunicación con testers y analistas es limitada y que los ajustes se realizan tarde en el proceso. Mostró apertura a trabajar con metodologías ágiles si esto implica una mayor interacción entre los equipos y un seguimiento más claro de las tareas.

### **Preguntas, respuestas y hallazgos:**

1. **¿Conoce usted qué son las metodologías ágiles y cómo pueden aplicarse en su trabajo?** “He escuchado de Kanban y Scrum en algunos cursos, pero no los he aplicado ni sé cómo se adaptarían a nuestro flujo.”

**Hallazgo:** La falta de conocimiento práctico sobre metodologías ágiles refuerza la necesidad de un plan de capacitación enfocado en su implementación específica.

2. **¿Estaría dispuesto a participar en una capacitación sobre metodologías ágiles si se ofreciera?** “Por supuesto, siempre que sea relevante y me ayude a mejorar mi trabajo sin complicarlo.”

**Hallazgo:** La disposición positiva sugiere que una capacitación bien diseñada podría tener un impacto significativo en el equipo.

3. **¿Mejoraría mucho la calidad final productiva del producto?** “Me gustaría tener revisiones más frecuentes para corregir problemas antes de que se acumulen al final.”

**Hallazgo:** Esta expectativa de retroalimentación continua está alineada con los principios ágiles, lo que indica un terreno fértil para su adopción.

4. **¿Qué espera personalmente de un proceso de transformación metodológica?** “Espero que nos ayude a coordinarnos mejor con otros equipos y a responder más rápido a los cambios.”

**Hallazgo:** La búsqueda de mejor coordinación y flexibilidad refleja una comprensión implícita de los beneficios de un enfoque ágil.

Se entrevistó al Tester 1-Tester Principal-, el 18 de abril del año en curso, a través de una videollamada grabada por zoom con una duración de 25 minutos.

### **Resumen Ejecutivo:**

El tester expresó preocupación por la baja calidad del software entregado, atribuyendo esto a la ausencia de pruebas tempranas y la falta de comunicación con el equipo de desarrollo. Indicó estar abierta a participar en una transformación metodológica si ello implica mayor participación desde el inicio del proyecto.

### **Preguntas, respuestas y hallazgos:**

1. **¿Ha tenido alguna vez contacto con metodologías ágiles como Scrum o Kanban?** “No, nunca he trabajado bajo ese enfoque.”

**Hallazgo:** Al igual que los desarrolladores, el tester muestra un desconocimiento práctico de las metodologías ágiles, lo que subraya la necesidad de incluir formación específica en el plan de transformación.

2. **¿Le gustaría conocer más sobre estas metodologías y participar en su implementación?** “Sí, si eso mejora la calidad del producto y nuestra forma de trabajar.”

**Hallazgo:** La actitud positiva hacia el aprendizaje y la implementación de metodologías ágiles indica una oportunidad para involucrar al tester en un proceso de cambio, siempre que se garantice relevancia práctica.

3. **¿Qué herramientas o procesos actuales dificultan su trabajo de pruebas?** “Usamos Excel para listas de verificación, pero no es eficiente. Nos falta una herramienta que facilite el seguimiento de defectos y la comunicación con los desarrolladores.”

**Hallazgo:** La ausencia de herramientas centralizadas para la gestión de pruebas resalta una limitación en el proceso actual que podría abordarse con metodologías ágiles y herramientas modernas.

4. **¿Qué cambios específicos espera de un nuevo enfoque metodológico?** “Espero poder participar desde el inicio del proyecto y tener reuniones regulares con los desarrolladores para evitar errores costosos al final.”

**Hallazgo:** La expectativa de mayor colaboración y participación temprana está alineada con los principios ágiles, lo que sugiere una buena receptividad a estos enfoques.

Se entrevistó al Tester 2-Tester de Soporte-, el 19 de abril del año en curso, a través de una videollamada grabada por zoom con una duración de 22 minutos.

### **Resumen Ejecutivo:**

El tester manifestó que uno de los principales problemas en el proceso actual es la detección tardía de errores debido a que las pruebas comienzan cuando el desarrollo ya está muy avanzado. Señaló que la comunicación con los desarrolladores es poco frecuente y generalmente reactiva. Expresó disposición a adoptar nuevas metodologías si se cuenta con herramientas de gestión de pruebas y capacitación adecuada.

## Preguntas, respuestas y hallazgos:

1. **¿Ha tenido alguna vez contacto con metodologías ágiles como Scrum o Kanban?** “He escuchado de Scrum y Kanban, pero nunca los he utilizado en mi trabajo.”

**Hallazgo:** La familiaridad superficial con metodologías ágiles refuerza la necesidad de un programa de capacitación que traduzca estos conceptos a la práctica diaria del tester.

2. **¿Le gustaría conocer más sobre estas metodologías y participar en su implementación?** “Sí, siempre que recibamos capacitación y que el proceso incluya herramientas que faciliten las pruebas y el seguimiento.”

**Hallazgo:** La disposición a aprender, condicionada a una capacitación adecuada y herramientas prácticas, indica un interés genuino en mejorar los procesos de prueba.

3. **¿Qué herramientas o procesos actuales dificultan su trabajo de pruebas?** “Hacemos pruebas manuales y usamos documentos compartidos, pero no tenemos un sistema unificado para rastrear errores, lo que genera confusión.”

**Hallazgo:** La falta de un sistema centralizado para la gestión de defectos destaca una oportunidad para introducir herramientas que complementen un enfoque ágil.

4. **¿Qué cambios específicos espera de un nuevo enfoque metodológico?** “Me gustaría estar involucrado desde las primeras etapas y tener una comunicación más fluida con el equipo de desarrollo para detectar problemas antes.”

**Hallazgo:** La expectativa de participación temprana y mejor comunicación refleja una alineación con los principios de colaboración y retroalimentación continua de las metodologías ágiles.

Se entrevistó al Administrador de Servidores, el 19 de abril del año en curso, a través de una videollamada grabada por zoom con una duración de 25 minutos.

### **Resumen Ejecutivo:**

El administrador de servidores describió un proceso de despliegue poco coordinado, donde recibe las aplicaciones al final del ciclo de desarrollo sin haber participado en las etapas previas. Esto genera problemas de compatibilidad y configuración que podrían evitarse con una mayor colaboración temprana. Mostró interés en metodologías que incluyan operaciones desde el inicio del proyecto.

### **Preguntas, respuestas y hallazgos:**

1. **¿Cómo describirías el impacto del proceso actual de despliegue en los servidores si se integrara con prácticas de Scrum o Kanban?**

“Actualmente, recibo el software al final, y surgen problemas de configuración porque no estoy involucrado antes. Con Scrum o Kanban, creo que podría participar en las iteraciones o tableros de trabajo para anticipar problemas.”

**Hallazgo:** El administrador considera que Scrum o Kanban pueden ayudar a una participación temprana más activa desde el comienzo, ya que la falta de integración temprana produce problemas evitables.

2. **¿Cómo es tu comunicación con el equipo de desarrollo y cómo piensas que Scrum o Kanban podrían mejorarla?** “Hasta el despliegue, la comunicación es casi inexistente. Con Kanban, podría ver el flujo de tareas para saber qué viene y prepararme mejor, y con Scrum, asistir a las reuniones diarias.”

**Hallazgo:** La escasa interacción con desarrollo resalta una oportunidad para que las prácticas de Scrum (reuniones diarias) o Kanban (visualización del flujo) mejoren la coordinación.

3. **¿Has tenido experiencia con metodologías como Scrum o Kanban en la gestión de infraestructura o despliegues?** “He oído que Scrum usa sprints y Kanban tableros, pero no los he aplicado. Creo que podrían ayudarme a alinear mejor los entornos con el desarrollo.”

**Hallazgo:** El conocimiento limitado sobre Scrum y Kanban sugiere la necesidad de capacitación específica para mostrar cómo estas metodologías pueden optimizar la gestión de servidores.

4. **¿Qué cambios esperas de adoptar Scrum o Kanban en tu trabajo como administrador de servidores?** “Espero estar en las reuniones de planificación de Scrum o seguir el tablero de Kanban para preparar los servidores desde el inicio y evitar sorpresas en producción.”

**Hallazgo:** La expectativa de involucrarse desde el principio y seguir un flujo de trabajo visible está alineada con los principios de Scrum (planificación iterativa) y Kanban (gestión visual), indicando buena receptividad.

## Anexo 2

### Consideraciones obtenidas de las respuestas en las entrevistas:

#### Gerente de

#### Sistemas

##### 1. Disposición del Equipo hacia el Cambio

*Apertura limitada:* Aunque el equipo no está completamente cerrado al cambio, existe un miedo generalizado a lo desconocido. Muchos temen que la adopción de nuevos métodos pueda traducirse en un aumento de su carga laboral.

*Resistencia pasiva:* No hay oposición abierta, pero sí una actitud de desinterés o poca participación en iniciativas de mejora.

##### 2. Principales Obstáculos para Implementar Nuevos Procesos

*Falta de conocimiento previo:* El equipo carece de experiencia en metodologías ágiles (como Scrum o Kanban) o en prácticas modernas como DevOps.

*Costumbre de trabajar de cierta manera:* Existe una alta dependencia de procesos tradicionales (modelo cascada), lo que dificulta la adopción de enfoques más dinámicos.

*Falta de capacitación:* No hay programas estructurados que preparen al equipo para adoptar nuevas herramientas o metodologías.

##### 3. Situación Actual de los Procesos

*Enfoque lineal y secuencial:* El flujo de trabajo sigue un modelo de cascada, con poca retroalimentación durante el desarrollo.

*Falta de integración temprana:* Los testers y el administrador de servidores participan principalmente al final del ciclo, lo que genera retrasos y problemas detectados tarde.

*Herramientas fragmentadas:* Se usan principalmente correo electrónico y GitHub, sin una herramienta centralizada de gestión de tareas ni comunicación estructurada.

#### **4. Expectativas de Transformación**

*Potencial de mejora:* Si se implementa con acompañamiento, capacitación y claridad, los nuevos métodos podrían mejorar la productividad y la calidad del producto.

*Necesidad de apoyo:* Los cambios deben venir acompañados de formación, tiempo de adaptación y liderazgo activo para superar la resistencia y generar confianza.

#### **Hallazgo Clave:**

El equipo muestra una disposición tímida al cambio, limitada por el miedo a la incertidumbre, la falta de conocimiento y la ausencia de un entorno de apoyo para adoptar nuevas metodologías.

#### **Recomendaciones Iniciales:**

Capacitación en metodologías ágiles (Scrum, Kanban) y herramientas colaborativas (Jira).

Pilotaje de nuevos procesos en proyectos pequeños para generar confianza y experiencia.

Mejora en la comunicación interna mediante reuniones diarias o sprints planificados.

Involucrar a todos los roles desde el inicio del desarrollo, especialmente a los testers y al administrador de servidores.

## **Analista de Sistemas**

### **1. Perspectiva del Equipo sobre el Cambio**

*Resistencia por falta de conocimiento:* Hay algo de escepticismo, pero también una cierta apertura al cambio. Para muchos, las herramientas nuevas son más un impedimento que un verdadero avance.

*Incertidumbre sobre beneficios:* Los integrantes del equipo no tienen claro cómo los nuevos procesos o herramientas pueden simplificar su trabajo, lo que causa una resistencia pasiva.

### **2. Flujo de Trabajo Presente**

*Modelo lineal y cascada:* El flujo de trabajo es secuencial y tiene escasa retroalimentación a lo largo del desarrollo. Todo inicia con un documento de requisitos que se codifica sin revisiones intermedias importantes.

*Ausencia de integración temprana:* La falta de participación activa del analista a lo largo del desarrollo restringe la habilidad de prever inconvenientes o modificar requisitos en tiempo real.

### **3. Utilización de Herramientas**

*Dependencia del email o correo electrónico:* La comunicación interna se lleva a cabo, sobre todo, por medio correo electrónico, lo que puede provocar demoras o pérdidas de información.

*Empleo parcial de GitHub:* Hay desarrolladores que emplean GitHub para controlar versiones, pero no existe una herramienta centralizada que facilite la administración eficaz de documentación compartida, pruebas o tareas.

*Ausencia de estandarización:* La falta de un estándar claro o una herramienta para la gestión de proyectos causa inconsistencias y obstaculiza la colaboración.

#### **4. Expectativas de Transformación**

*Posibilidad de rol más activo:* En un entorno más ágil, el rol del analista podría evolucionar a uno más colaborativo, participando desde el inicio del proyecto y facilitando la comprensión de requisitos en tiempo real.

*Necesidad de herramientas centralizadas:* Se espera que con una transformación metodológica se adopten herramientas que permitan una mejor coordinación, seguimiento y documentación compartida.

##### **Hallazgo Clave:**

El analista percibe un entorno de trabajo con baja disposición al cambio, debido a la falta de conocimiento previo, la costumbre de trabajar de forma lineal y la ausencia de herramientas centralizadas que faciliten la colaboración.

##### **Recomendaciones Iniciales:**

Adopción de herramientas centralizadas (Jira) para gestionar tareas, requisitos y seguimiento del desarrollo.

Capacitación en metodologías ágiles para que el equipo entienda cómo estos enfoques pueden facilitar el trabajo, no complicarlo.

Incorporación del analista en reuniones diarias o planificación de sprints, para que participe activamente en la evolución del producto.

Introducir revisiones intermedias para evitar el modelo totalmente lineal y permitir ajustes tempranos.

#### **Desarrolladores de Software**

##### **1. Descripción del Flujo de Trabajo Actual**

*Modelo lineal y cascada:* Los desarrolladores coinciden en que el flujo de trabajo es secuencial y basado en el modelo de cascada: se inicia con un documento de requisitos, se asignan tareas y se desarrolla de forma aislada. No hay revisión constante ni integración temprana.

*Falta de retroalimentación:* El proceso carece de mecanismos para ajustar el desarrollo en tiempo real. Los errores se detectan tarde, lo que incrementa el esfuerzo de corrección.

*Trabajo en silos:* Cada desarrollador avanza por su cuenta, con poca colaboración interna o con otros roles como testers o analistas.

## **2. Uso de Herramientas**

*Herramientas fragmentadas y poco estandarizadas:* No hay una herramienta única ni un estándar claro para gestionar tareas, seguimiento o comunicación.

*Dependencia del correo electrónico:* La comunicación interna, se basa principalmente en correos y chats, provocando incoherencias y la información se pierde.

*Uso limitado de GitHub:* GitHub se utiliza, por algunos, para control de versiones, pero, para gestionar proyectos, no se considera una herramienta principal.

*Ausencia de herramientas ágiles:* Herramientas tipo Jira, Trello, o cosas parecidas, no se emplean; Esto, de verdad limita la visibilidad del avance y la automatización del flujo de trabajo.

## **3. Percepción sobre la Disposición al Cambio**

*Resistencia basada en miedo y desconocimiento:* Los desarrolladores expresan cierta resistencia pasiva al cambio, motivada por el miedo a aumentar su carga laboral sin apoyo claro.

*Apertura condicionada:* Señalan que están dispuestos a probar nuevas formas de trabajo, siempre que haya capacitación, acompañamiento y claridad sobre los beneficios.

*Preocupación por la sobrecarga:* Muchos temen que la adopción de nuevas herramientas o metodologías implique más trabajo sin un beneficio inmediato o visible.

#### **4. Expectativas de Mejora**

*Mejora en la comunicación y coordinación:* Esperan que con una transformación metodológica se establezcan canales de comunicación más claros y frecuentes.

*Uso de herramientas centralizadas:* Ven con buenos ojos la posibilidad de usar una herramienta única que permita gestionar tareas, seguir el progreso y facilitar la colaboración.

*Reducción de retrabajos:* Confían en que con metodologías ágiles y mayor integración temprana, se puedan evitar errores detectados tarde y reducir retrabajos.

#### *Hallazgo Clave:*

Los desarrolladores perciben un entorno de trabajo con procesos anticuados, falta de herramientas adecuadas y resistencia al cambio motivada por el miedo a la sobrecarga y la falta de claridad sobre los beneficios reales de nuevas metodologías.

#### **Recomendaciones Iniciales:**

Introducir herramientas centralizadas como Jira o Trello para gestionar tareas, facilitar el seguimiento y mejorar la transparencia del flujo de trabajo.

Entrenamiento en técnicas y metodologías para trabajar con prácticas ágiles (Kanban, Scrum) así como para utilizar herramientas de administración de proyectos y CI/CD.

Implementación de pilotos en proyectos pequeños para experimentar nuevos métodos sin causar resistencia ni sobrecarga.

Promover la cooperación temprana entre analistas, testers y desarrolladores con el fin de identificar problemas antes y disminuir los retrabajos.

Involucrar a los desarrolladores en la planificación y el diseño de modificaciones para fomentar su participación y confianza.

## **Testers de Software**

### **1. Proceso de Pruebas Actual**

*Pruebas manuales que predominan:* Los testers llevan a cabo pruebas manuales, empleando listas de verificación en Excel y comunicando los errores a través del correo electrónico.

*Escasez de automatización:* No se emplean herramientas de pruebas automatizadas, lo cual restringe la eficiencia y la habilidad de hacer pruebas repetitivas con agilidad.

*Sin integración temprana:* Dado que las pruebas se realizan al final del ciclo de desarrollo, no es posible identificar problemas en fases iniciales, lo que incrementa el tiempo que toma corregirlos.

### **2. Acceso a Información del Desarrollo**

*Información escasa y con retraso:* Los testers no tienen acceso al producto hasta que este está terminado, lo cual les impide prever inconvenientes o participar en la definición de criterios de aceptación desde el comienzo.

*Ausencia de comunicación continua:* El equipo de pruebas y el de desarrollo no tienen un flujo informativo claro ni canales de comunicación bien organizados.

### **3. Colaboración con el Equipo de Desarrollo**

*Colaboración sumamente restringida:* La interacción con los desarrolladores es escasa durante el ciclo de desarrollo y se da sobre todo al final, cuando ya surgen errores.

*Decisiones realizadas sin su involucramiento:* A menudo se enteran de decisiones relevantes solo después de que se han implementado, lo que disminuye su capacidad para impactar la calidad del producto desde el principio.

#### **4. Percepción sobre el Cambio**

*Falta de interés en nuevas prácticas:* Muchos no ven beneficios claros en adoptar pruebas tempranas o automatizadas, lo que genera escepticismo.

*Poca comprensión del valor de la calidad integrada:* Existe una visión limitada del rol del tester como parte activa del proceso de desarrollo, más allá de la corrección final.

#### **5. Expectativas de Mejora**

*Mayor participación temprana:* Los testers esperan poder participar desde el inicio del desarrollo, para detectar problemas antes y colaborar en la definición de requisitos de calidad.

*Uso de herramientas especializadas:* Confían en que con una transformación metodológica se adopten herramientas dedicadas a la gestión de pruebas, lo que permitiría mayor trazabilidad y eficiencia.

*Mejora en la comunicación:* Esperan que con metodologías ágiles se establezcan canales de comunicación más claros y frecuentes con el equipo de desarrollo.

#### **Hallazgo Clave:**

Los testers perciben un entorno de trabajo con procesos de pruebas obsoletos, poca participación en el desarrollo y una colaboración muy limitada con el equipo de desarrollo, lo que afecta la calidad del producto y la eficiencia del proceso.

#### **Recomendaciones Iniciales:**

Incorporar a los testers desde el inicio del desarrollo, para que participen en la definición de requisitos y criterios de aceptación.

Introducir herramientas de gestión de pruebas (Jira con módulo de pruebas) para mejorar la trazabilidad y seguimiento de defectos.

Capacitación en pruebas automatizadas y en el uso de herramientas que permitan ejecutar pruebas repetitivas con mayor rapidez.

Adoptar metodologías ágiles que promuevan la integración continua y la participación activa de los testers en todo el ciclo de desarrollo.

Establecer canales de comunicación estructurados entre desarrollo y pruebas para facilitar la resolución de dudas en tiempo real y evitar errores detectados tarde.

## **Administrador de Servidores**

### **1. Colaboración con el Equipo de Desarrollo**

*Interacción limitada y tardía:* El administrador de servidores señala que su participación ocurre principalmente al final del ciclo de desarrollo, sin intervención temprana en la planificación o diseño.

*Decisiones tomadas sin su conocimiento:* Muchas veces se entera de cambios importantes después de haber sido implementados, lo que complica la integración y el despliegue.

### **2. Comunicación y Coordinación**

*Falta de canales estructurados:* No existe una comunicación constante ni canales definidos entre desarrollo y operaciones, lo que genera fricciones y retrasos.

*Falta de visibilidad del proceso:* El administrador no tiene acceso a información clara sobre el estado del desarrollo ni sobre los cambios que se están implementando.

### **3. Uso de Herramientas y Prácticas**

*Falta de integración y automatización:* No se utilizan prácticas de DevOps ni herramientas de integración continua/despliegue continuo (CI/CD), lo que limita la capacidad de desplegar cambios de forma rápida y segura.

*Herramientas obsoletas o ausentes:* No hay una cultura de trabajo conjunto entre desarrollo y operaciones, ni se usan herramientas que faciliten esa integración (como Jenkins, Ansible, Docker, Kubernetes, etc.).

#### **4. Percepción sobre el Cambio**

*Poca visibilidad del valor del cambio:* El administrador percibe resistencia en el equipo hacia nuevas formas de trabajo, en parte por la falta de comprensión de cómo estas pueden beneficiarlos.

*Miedo a la complejidad adicional:* Hay preocupación de que nuevas herramientas o procesos impliquen mayor carga de trabajo sin un beneficio claro.

#### **5. Expectativas de Mejora**

*Mayor involucramiento temprano:* El administrador espera poder participar desde el inicio del desarrollo, para anticipar problemas de infraestructura y garantizar entornos más estables.

*Adopción de prácticas de DevOps:* Confía en que con metodologías como DevOps y herramientas de automatización, se podrían construir entornos más predecibles y fiables desde el inicio.

*Mejora en la comunicación y coordinación:* Espera que con una transformación metodológica se establezcan flujos de trabajo más colaborativos y canales de comunicación más claros entre desarrollo y operaciones.

#### **Hallazgo Clave:**

El rol del administrador de servidores está desconectado del flujo de desarrollo actual, lo que genera retrasos, retrabajos y dificultades técnicas al momento del despliegue. Existe una oportunidad clara para introducir prácticas de DevOps y mejorar la colaboración temprana entre equipos.

### **Recomendaciones Iniciales:**

Incluir al administrador de servidores desde el inicio del desarrollo, para anticipar necesidades técnicas e infraestructura crítica.

Introducir prácticas de DevOps que promuevan la colaboración entre desarrollo y operaciones, y permitan despliegues más ágiles y confiables.

Capacitación en herramientas de CI/CD como Jenkins, Docker, Ansible o GitHub Actions, para automatizar procesos y mejorar la calidad del despliegue.

Establecer canales de comunicación estructurados entre desarrollo, pruebas y operaciones para facilitar la resolución de dudas y decisiones técnicas en tiempo real.

Documentar y estandarizar los procesos de despliegue y configuración, para reducir la dependencia de conocimiento tácito y facilitar la continuidad.

### Anexo 3

Se realizó el 10 de mayo del año en curso a través de videollamada por zoom, duración 55 minutos, a expertos para la validación externa de la propuesta.

#### Participantes:

- **Experto A:** Msc. Jorge Vera – Desarrollo de Sistemas de Información y Tecnología Digital Educativa

Licenciado en Sistemas de Información, Diplomado en Educación Universitaria, Magister en Educación Informática, Doctorante en Educación en la línea de Educación Digital, Asistente en Departamento Metodológico y Carrera Docente, Desarrollo de Sistemas en la Unidad Educativa San José La Salle de Guayaquil – Ecuador, Dirección de Sistemas de la Facultad de Filosofía de la Universidad de Guayaquil, Asistente en Gestión del Conocimiento de la Facultad de Filosofía de la Universidad de Guayaquil, Desarrollo de Sistema de Investigación Científica de la DIPA (Dirección de Investigación y Proyectos Académicos) de la Universidad de Guayaquil, Asistente del Departamento Metodológico y Carrera Docente de la Universidad Bolivariana del Ecuador

- **Experto B:** Msc. Carlos Puig – Gerente de Transformación Digital

Magíster en Mejoramiento de Procesos, con más de 14 años de experiencia liderando iniciativas de transformación digital, automatización y mejora continua. Especialista en metodologías ágiles (Scrum), Lean y Six Sigma, ha dirigido equipos multidisciplinarios en proyectos de innovación, arquitectura tecnológica y adopción de modelos operativos modernos orientados a incrementar la eficiencia organizacional y fomentar la cultura de mejora continua.

*Pregunta 1: ¿Cuál es su impresión general sobre la propuesta de integrar Scrum, Kanban y CMMI en una empresa de desarrollo de software como la descrita?*

*Msc. Jorge Vera:*

La propuesta es muy sólida. Lo que más me llama la atención es que no cae en el extremo común de “ágil o CMMI”, sino que busca un equilibrio inteligente. Integrar CMMI con Scrum y Kanban es un enfoque avanzado y poco común, pero muy adecuado para empresas que buscan certificación de calidad sin perder agilidad. Muchas organizaciones ven a CMMI como burocrático, pero aquí se usa como un marco de soporte, no como una camisa de fuerza. La forma en que se vincula la trazabilidad de requisitos con las historias de usuario de Scrum, y la gestión visual de Kanban, es coherente y práctica.

*Msc. Carlos Puig:*

Totalmente de acuerdo. Lo que me parece innovador es el uso de herramientas como Jira para alinear los procesos ágiles con los artefactos de CMMI. Conseguir que un tablero Kanban se ajuste a los estándares de documentación CMMI no es sencillo. Sin embargo, al implementar una estructura definida con campos obligatorios, historiales de cambios y trazabilidad automática, se logra. Incluir auditorías internas desde la etapa piloto también demuestra un enfoque maduro, que es no solo operativo, sino también de cumplimiento.

*Pregunta 2: ¿Qué opinión tiene sobre el diseño del plan de implementación en tres etapas: piloto, escalamiento y consolidación?*

*Msc. Carlos Puig:*

Es un enfoque metodológicamente correcto. Empezar con un piloto controlado es clave para mitigar riesgos. Me gusta que no se intenta cambiar todo de golpe, sino que se validan herramientas, métricas y dinámicas de equipo en un entorno pequeño. El uso de KPIs claros (velocidad del equipo, tasa de defectos, satisfacción) desde el inicio es un acierto. Lo que más valoro es que el plan no solo mide resultados técnicos, sino también culturales, como la disposición al cambio y la colaboración.

*Msc. Jorge Vera:*

Sí, y el paso de escalamiento es realista. Muchas propuestas fracasan porque pasan del piloto a la implementación total sin ajustes. Aquí se incluye una fase de retroalimentación, capacitación adaptada y estandarización de procesos,

lo cual es esencial. Además, se debe proyectar alcanzar un Nivel 2 de CMMI en un piloto y avanzar hacia el Nivel 3 como consolidación.

*Pregunta 3: ¿Qué opinan sobre la selección de herramientas: Jira, Moodle, Power BI, Microsoft Teams y Google Forms?*

*Msc. Jorge Vera:*

Excelente elección. Jira es ideal para gestionar tanto Scrum como Kanban, y con plugins puede soportar CMMI. Moodle en cloud privado es una decisión acertada: permite personalizar la capacitación, evaluar el progreso y mantener la seguridad de la información. Lo que más me gusta es que no se limita a “capacitar”, sino que crea un sistema de gestión del conocimiento con Teams y Google Docs. Eso es clave para evitar que el conocimiento quede en una sola persona.

*Msc. Carlos Puig:*

Power BI es un valor agregado que muchas propuestas omiten. Tener dashboards en tiempo real con métricas de desempeño (tiempo de entrega, calidad, satisfacción) permite tomar decisiones basadas en datos, no en percepciones. Y Google Forms para evaluar la capacitación y el clima organizacional es simple, pero efectivo. No siempre hace falta una herramienta compleja cuando una solución ligera cumple el objetivo.

*Pregunta 4: El plan de capacitación y transformación cultural es un componente central. ¿Lo consideran viable y suficiente?*

*Msc. Carlos Puig:*

Sí, y es uno de los puntos fuertes. Muchas transformaciones fracasan por subestimar el cambio cultural. Aquí no solo se capacita en herramientas, sino que se fomenta una mentalidad de mejora continua con retrospectivas, sesiones de innovación y participación temprana de testers y administradores. El hecho de que el Scrum Master y el coach ágil sean responsables del piloto también es clave: necesitas líderes que guíen el cambio, no solo técnicos que apliquen procesos.

*Msc. Jorge Vera:*

Coincido. El plan de capacitación es progresivo, práctico y medible. No es un taller único, sino un proceso continuo con evaluación de desempeño. Además, el uso de Moodle permite escalarlo sin perder calidad. Lo único que añadiría sería una métrica de “retención de conocimiento” tras los cursos, para asegurar que lo aprendido se aplica en la práctica.

*Pregunta 5: ¿Qué recomendación final harían para fortalecer esta propuesta?*

*Msc. Jorge Vera:*

Mantener el enfoque híbrido, pero con énfasis en la sostenibilidad. Sugiero incorporar un comité de transformación con representantes de cada área, que revise los avances cada trimestre. También sería útil definir un “champion ágil” en cada equipo, alguien que impulse la adopción desde dentro.

*Msc. Carlos Puig:*

Yo recomendaría que, durante la fase de consolidación, se comience a incluir CI/CD (integración continua) y automatización de pruebas. En la fase piloto no es urgente, aunque si la compañía quiere crecer con calidad, necesitará herramientas de DevOps. La propuesta, además, está sólidamente fundamentada, es realizable y ofrece un enfoque humano que no es usual en los proyectos de transformación digital.

## **Anexo 4**

### **Modelo de la Transformación Organizacional para una empresa de software mediante la Integración de Metodologías Ágiles**

El Modelo (cuya ampliación se dispone en el Anexo 5) trata lo siguiente en etapas:

#### ***Ingeniería de Requerimientos***

El modelo combina Scrum y Kanban con CMMI para un inicio sólido. Los requisitos se gestionan visualmente con historias de usuario o tableros Kanban, mientras CMMI asegura definiciones claras y trazabilidad. La colaboración continua con el cliente en ágiles se complementa con validaciones formales de CMMI, equilibrando flexibilidad y control.

En cuanto a las prácticas fundamentales, se establecerá una rutina de revisión cada quince días donde se analizarán las métricas clave con el objetivo de detectar oportunidades de mejora. Paralelamente, se realizarán ajustes constantes en los procesos operativos, tomando como base tanto los datos recopilados como la retroalimentación proporcionada por los miembros del equipo.

## Figura 9 Google Forms

The screenshot shows a Google Form titled "Solicitud de trabajo" (Job Application). The form is set against a light blue background. At the top, there is a header with the title and a description: "En caso de emergencia, llámanos al (54) 000-0000 o envíanos un correo electrónico a [go\\_reply@example.com](mailto:go_reply@example.com)". Below this, there are several sections:

- Información personal**: Descripción (opcional)
- Nombre \***: Texto de respuesta breve
- Dirección de correo electrónico \***: Texto de respuesta breve
- Describe el problema**: Descripción (opcional)
- Resumen \***: Texto de respuesta breve

On the right side, there is a vertical toolbar with icons for adding, deleting, and moving questions. At the top of the form, there are tabs for "Preguntas", "Respuestas", and "Configuración".

**Nota.** Google Forms para encuestas

## Figura 10 Power BI

The screenshot displays the Microsoft Power BI Desktop interface. The window title is "Untitled - Power BI Desktop". The interface is divided into several sections:

- Home**: A sidebar on the left with "Home" and "Open" options.
- Select a data source or start with a blank report**: A row of six buttons: "Blank report", "OneLake data hub", "Excel workbook", "SQL Server", "Learn with sample data", and "Get data from other sources".
- Recommended**: A section with four report cards: "Intro—What is Power BI?", "DQ-CSV-Adam", "csv-deployment-pipeline", and "Game Revenue".
- Recent**: A table at the bottom showing recent reports.

Name	Location	Opened
sales_performance_report.pbix	C:\Users\user\Documents\sales_performance_report.pbix	16 hours ago
DQ-CSV-Adam	Power BI & Learning\Business\adams	6 days ago
csv-deployment-pipeline	Power BI & Learning\Business\adams	6 days ago

**Nota.** Herramienta para analítica de datos

**Diseño del Software**

Se integra la adaptabilidad del diseño emergente ágil con la estructura de CMMI. Ágiles permite ajustes rápidos con retroalimentación, mientras CMMI exige una arquitectura documentada y alineada con estándares. Esto asegura un diseño robusto pero flexible ante cambios.

### ***Implementación***

La rapidez de sprints ágiles con integración continua se combina con el control de calidad de CMMI. Las pruebas unitarias y de aceptación se integran al desarrollo, mientras CMMI garantiza cumplimiento y documentación, logrando entregas frecuentes y confiables.

### ***Despliegue y Mantenimiento***

Se aplican entregas incrementales y mejora continua de las metodologías ágiles, junto con procesos formales de CMMI para un despliegue seguro y mantenimiento documentado. Así, el software se mantiene funcional y relevante en el tiempo.

### **Programa de Transformación Cultural**

La transformación cultural es un aspecto crucial para el éxito de la implementación de metodologías ágiles y CMMI. Un programa de transformación cultural bien diseñado puede ayudar a superar la resistencia al cambio y fomentar una cultura de colaboración, innovación y mejora continua (cuya ampliación conceptual se dispone en el Anexo 5), por lo que se considera: Cultura de colaboración, sistema de gestión del conocimiento y un plan de capacitación.

### ***Cultura de colaboración***

Las metodologías ágiles promueven el trabajo en equipo mediante reuniones diarias, revisiones de sprint y retrospectivas, fortaleciendo la comunicación y la cooperación. CMMI complementa este enfoque asegurando que todas las partes interesadas trabajen alineadas con los objetivos del proyecto, consolidando un entorno colaborativo y orientado a resultados.

### ***Sistema de gestión del conocimiento***

En ágiles, el conocimiento se captura de forma continua mediante retrospectivas y lecciones aprendidas, se almacena en herramientas como Jira, y se comparte en reuniones y capacitaciones. CMMI aporta un enfoque sistemático y documentado, garantizando que la información se gestione de forma estructurada y eficiente para su reutilización.

### ***Plan de capacitación***

Las metodologías ágiles requieren formación continua e incremental en metodologías, herramientas y técnicas, apoyada en talleres y sesiones prácticas. CMMI añade un esquema formal y documentado, asegurando que el equipo adquiera y mantenga las competencias necesarias, evaluando su impacto mediante modelos como Phillips, Brinkerhoff, Kaufman o Kirkpatrick.

### **Instrumentos de Investigación**

La investigación continua en la empresa enfocará la mejora permanente, así como al inicio del presente trabajo, basado en entrevistas.

### ***Enfoque Mixto***

Con el fin de obtener una comprensión integral y rigurosa de los factores que influyen en la adopción de metodologías ágiles dentro de la organización, se opta por un diseño de investigación que combina enfoques cualitativos y cuantitativos. Este enfoque mixto permite no solo capturar las percepciones subjetivas y dinámicas culturales del entorno organizacional, sino también medir de forma objetiva actitudes, niveles de satisfacción y disposición hacia el cambio metodológico. Al integrar múltiples fuentes de datos, se fortalece la validez de los hallazgos y se enriquece el análisis, facilitando recomendaciones más fundamentadas y contextualmente relevantes. Por ello se considera:

### **Cualitativo:**

Entrevistas: Para analizar la cultura organizacional y las percepciones sobre la adopción de metodologías ágiles (Schwaber y Sutherland, 2020).

### **Cuantitativo:**

Encuestas Likert: Para medir la disposición hacia prácticas ágiles y la satisfacción del equipo (Anderson, 2010).

### ***Modo de Uso de Instrumentos de Investigación***

Para la recolección de datos, se emplean dos instrumentos complementarios, seleccionados estratégicamente según el tipo de información requerida y los objetivos de cada enfoque metodológico. Con las entrevistas semiestructuradas se realizaron con actores clave de la organización, con el fin de examinar en detalle los obstáculos culturales vinculados a la implementación de metodologías ágiles, como se indica en la Tabla 6. Estas entrevistas se realizaron de acuerdo con un protocolo que contenía preguntas abiertas y sus transcripciones fueron analizadas con el software NVivo, que ayudó a clasificar y examinar temáticamente los datos cualitativos.

Por otro lado, las encuestas con escala Likert se utilizan para medir de manera cuantitativa la percepción de efectividad y el nivel de satisfacción del personal en relación con los métodos ágiles, así como su actitud hacia estos. Estas encuestas, elaboradas en Google Forms, emplearán escalas numéricas de 1 a 5 (donde 1 significa "Completamente en desacuerdo" y 5 "Completamente de acuerdo"), lo cual permitirá cuantificar las actitudes y realizar un análisis estadístico posterior.

La recopilación de datos es tanto amplia como profunda gracias a la combinación de estas herramientas, lo que se apoya en el enfoque mixto del estudio.

**Tabla 10 Instrumentos de investigación**

<b>Instrumento</b>	<b>Propósito</b>	<b>Herramientas</b>
Entrevistas semiestructuradas	Identificar barreras culturales	Guía de preguntas + NVivo
Encuestas Likert	Medir disposición hacia prácticas ágiles	Google Forms + Escalas 1-5

**Nota.** Instrumentación de la investigación

### **Ingeniería de Requerimientos**

La Ingeniería de Requerimientos es la fase inicial del ciclo de vida del desarrollo de software y es crucial para el éxito del proyecto. En esta fase, se identifican, analizan, documentan y gestionan los requisitos del software.

#### **Identificación de Requerimientos:**

**Metodologías Ágiles (Scrum y Kanban):** En Scrum, los requisitos se capturan en forma de historias de usuario, que son descripciones breves y simples de una funcionalidad desde la perspectiva del usuario final. En Kanban, los requisitos se visualizan en un tablero Kanban, lo que permite una gestión visual del flujo de trabajo.

**CMMI:** CMMI proporciona un marco para la gestión de requisitos, asegurando que los requisitos sean claramente definidos, documentados y gestionados a lo largo del ciclo de vida del proyecto.

#### **Análisis de Requerimientos:**

**Metodologías Ágiles:** Las metodologías ágiles fomentan la colaboración continua con el cliente y las partes interesadas para refinar y priorizar los

requisitos. Esto se logra a través de reuniones de refinamiento del Product Backlog y revisiones de sprint.

**CMMI:** CMMI enfatiza la importancia de analizar los requisitos para asegurar que sean completos, consistentes y alcanzables. Esto incluye la validación y verificación de los requisitos.

#### **Documentación de Requerimientos:**

**Metodologías Ágiles:** Aunque las metodologías ágiles favorecen la documentación ligera, se utilizan herramientas como Google Docs y Jira para documentar y gestionar los requisitos.

**CMMI:** CMMI requiere una documentación formal y detallada de los requisitos, lo que asegura que todas las partes interesadas tengan una comprensión clara y compartida de lo que se debe desarrollar.

#### **Gestión de Requerimientos:**

**Metodologías Ágiles:** La gestión de requisitos en metodologías ágiles es dinámica y adaptativa. Los requisitos pueden cambiar y evolucionar a lo largo del proyecto, y se gestionan a través de herramientas como Jira y su Tablero Kanban.

**CMMI:** CMMI proporciona un marco para la gestión de cambios en los requisitos, asegurando que los cambios sean evaluados, aprobados y documentados de manera formal.

#### **Diseño del Software**

El diseño del software es la fase en la que se define la arquitectura y los componentes del software.

#### **Arquitectura del Software:**

**Metodologías Ágiles:** En metodologías ágiles, el diseño del software se realiza de manera iterativa e incremental. Se utilizan técnicas como el diseño emergente, donde la arquitectura evoluciona a lo largo del proyecto.

**CMMI:** CMMI enfatiza la importancia de un diseño bien definido y documentado. Esto incluye la definición de la arquitectura del software, los componentes y las interfaces.

### **Diseño de Componentes:**

**Metodologías Ágiles:** Los componentes del software se diseñan y desarrollan en iteraciones cortas, lo que permite una retroalimentación rápida y ajustes continuos.

**CMMI:** CMMI requiere un diseño detallado y documentado de los componentes del software, asegurando que cumplan con los requisitos y estándares de calidad.

### **Implementación**

La implementación es la fase en la que se desarrolla el código del software.

### **Desarrollo del Código:**

**Metodologías Ágiles:** El desarrollo del código se realiza en iteraciones cortas (sprints), con entregas frecuentes y continuas. Se utilizan prácticas como la integración continua y el desarrollo guiado por pruebas (TDD).

**CMMI:** CMMI enfatiza la importancia de seguir estándares de codificación y prácticas de desarrollo de software, asegurando que el código sea de alta calidad y mantenible.

### **Pruebas del Software:**

**Metodologías Ágiles:** Las pruebas se realizan de manera continua e integrada en el proceso de desarrollo. Se utilizan técnicas como pruebas unitarias, pruebas de integración y pruebas de aceptación.

**CMMI:** requiere un enfoque sistemático y documentado para las pruebas del software, asegurando que el software cumpla con los requisitos y estándares de calidad.

## **Despliegue y Mantenimiento**

El despliegue y mantenimiento son las fases finales del ciclo de vida del desarrollo de software.

### **Despliegue del Software:**

**Metodologías Ágiles:** El despliegue del software se realiza de manera continua e incremental, con entregas frecuentes y continuas. Se utilizan prácticas como la entrega y el despliegue continuos.

**CMMI:** CMMI enfatiza la importancia de un proceso de despliegue bien definido y documentado, asegurando que el software se despliegue de manera segura y eficiente.

### **Mantenimiento del Software:**

**Metodologías Ágiles:** El mantenimiento del software se realiza de manera continua e incremental, con entregas frecuentes y continuas. Se utilizan prácticas como la mejora continua y la retroalimentación continua.

**CMMI:** CMMI requiere un enfoque sistemático y documentado para el mantenimiento del software, asegurando que el software se mantenga y evolucione de manera segura y eficiente.

## Anexo 5

### Programa de Transformación Cultural

La transformación cultural es un aspecto crucial para el éxito de la implementación de metodologías ágiles y CMMI. Un programa de transformación cultural bien diseñado puede ayudar a superar la resistencia al cambio y fomentar una cultura de colaboración, innovación y mejora continua.

#### **Cultura de Colaboración:**

**Metodologías Ágiles:** Las metodologías ágiles fomentan una cultura de colaboración y trabajo en equipo. Se utilizan técnicas como las reuniones diarias (daily stand-ups), las revisiones de sprint y las retrospectivas para fomentar la colaboración y la comunicación.

**CMMI:** CMMI enfatiza la importancia de una cultura de colaboración y trabajo en equipo, asegurando que las partes interesadas trabajen en busca de lograr los objetivos del proyecto.

#### **Cultura de Innovación:**

**Metodologías Ágiles:** Las metodologías ágiles fomentan una cultura de innovación y mejora continua. Se utilizan técnicas como las retrospectivas y las sesiones de innovación para fomentar la innovación y la mejora continua].

**CMMI:** CMMI enfatiza la importancia de una cultura de innovación y mejora continua, asegurando que el software se desarrolle y mantenga de manera innovadora y eficiente.

#### **Cultura de Calidad:**

**Metodologías Ágiles:** Las metodologías ágiles fomentan una cultura de calidad y excelencia. Se utilizan técnicas como las pruebas y la integración continuas para asegurar la calidad del software.

**CMMI:** CMMI enfatiza la importancia de una cultura de calidad y excelencia, asegurando que el software cumpla con los requisitos y estándares de calidad.

### **Sistema de Gestión del Conocimiento**

Un sistema de gestión del conocimiento bien diseñado puede ayudar a capturar, almacenar y compartir el conocimiento y las mejores prácticas, asegurando que el conocimiento se utilice de manera efectiva y eficiente.

#### **Captura del Conocimiento:**

**Metodologías Ágiles:** Las metodologías ágiles fomentan la captura del conocimiento de manera continua e incremental. Se utilizan técnicas como las retrospectivas y las sesiones de lecciones aprendidas para capturar el conocimiento.

**CMMI:** CMMI enfatiza la importancia de capturar el conocimiento de manera sistemática y documentada, asegurando que el conocimiento se capture de manera efectiva y eficiente.

#### **Almacenamiento del Conocimiento:**

**Metodologías Ágiles:** Las metodologías ágiles fomentan el almacenamiento del conocimiento de manera accesible y utilizable. Se utilizan herramientas como Google Docs y Jira para almacenar y gestionar el conocimiento.

**CMMI:** CMMI enfatiza la importancia de almacenar el conocimiento de manera sistemática y documentada, asegurando que el conocimiento se almacene de manera efectiva y eficiente.

#### **Compartir el Conocimiento:**

**Metodologías Ágiles:** Las metodologías ágiles fomentan el compartir el conocimiento de manera continua e incremental. Se utilizan técnicas como las reuniones de equipo y las sesiones de capacitación para compartir el conocimiento.

**CMMI:** CMMI enfatiza la importancia de compartir el conocimiento de manera sistemática y documentada, asegurando que el conocimiento se comparta de manera efectiva y eficiente.

### **Plan de Capacitación**

Un plan de capacitación bien diseñado puede ayudar a asegurar que el equipo tenga las habilidades y conocimientos necesarios para implementar y mantener las metodologías ágiles y CMMI de manera efectiva y eficiente. Pudiendo aplicarse modelos como el Phillips (ROI), Brinkerhoff (Método de las Historias de Éxito, Kaufman, Kirkpatrick, entre otros.

### **Capacitación en Metodologías Ágiles:**

**Metodologías Ágiles:** Las metodologías ágiles requieren una capacitación continua e incremental. Se utilizan técnicas como los talleres y las sesiones de capacitación para asegurar que el equipo tenga las habilidades y conocimientos necesarios.

**CMMI:** enfatiza la importancia de una capacitación sistemática y documentada, asegurando que el equipo tenga las habilidades y conocimientos necesarios para implementar y mantener las metodologías ágiles y CMMI.

### **Capacitación en CMMI:**

**Metodologías Ágiles:** Las metodologías ágiles requieren una capacitación continua e incremental en CMMI. Se utilizan técnicas como los talleres y las sesiones de capacitación para asegurar que el equipo tenga las habilidades y conocimientos necesarios.

**CMMI:** CMMI enfatiza la importancia de una capacitación sistemática y documentada en CMMI, asegurando que el equipo tenga las habilidades y conocimientos necesarios para implementar y mantener CMMI.

### **Capacitación en Herramientas y Técnicas:**

**Metodologías Ágiles:** Las metodologías ágiles requieren una capacitación continua e incremental en herramientas y técnicas. Se utilizan

técnicas como los talleres y las sesiones de capacitación para asegurar que el equipo tenga las habilidades y conocimientos necesarios.

**CMMI:** CMMI enfatiza la importancia de una capacitación sistemática y documentada en herramientas y técnicas, asegurando que el equipo tenga las habilidades y conocimientos necesarios para implementar y mantener las herramientas y técnicas.

## **Anexo 6**

### **Fase 1: Piloto (3-6 meses)**

**Objetivo:** Para validar el funcionamiento del marco integrado (Scrum + Kanban + CMMI) y ajustarlo antes de expandirlo, lo implementaremos en un equipo pequeño.

#### **Acciones Concretas:**

Comenzar con talleres de capacitación.

Establecer tableros Kanban y ciclos de Scrum, con revisiones cada semana.

Usar CMMI con el propósito de documentar requerimientos y llevar a cabo pruebas de manera constante.

Registrar avances en Google Docs, incluyendo lecciones aprendidas.

#### **Métricas Vinculadas:**

Rapidez del equipo

Porcentaje de defectos

Satisfacción del personal

Asistencia a capacitaciones

#### **Responsables:**

Maestro de Scrum

Entrenador Ágil

Grupos de trabajo multifuncionales

## **Fase 2: Escalamiento (6-12 meses)**

**Objetivo:** Ampliar el modelo a otros equipos, modificando los procesos conforme a la retroalimentación y las métricas conseguidas durante la fase piloto.

### **Acciones Específicas:**

Revisar resultados del piloto y ajustar herramientas/procesos.

Realizar nuevas rondas de capacitación adaptadas al contexto de nuevos equipos.

Establecer tableros Kanban y ciclos de Scrum estandarizados.

Continuar evaluaciones de madurez bajo CMMI.

### **Métricas Asociadas:**

Tiempo promedio de entrega

Porcentaje de pruebas automatizadas

Nivel de madurez alcanzado (CMMI)

Aplicabilidad de conocimientos adquiridos

### **Responsables:**

2 Scrum Masters

Coach Ágil

Consultor CMMI

RRHH

## **Fase 3: Consolidación (12+ meses)**

**Objetivo:** Estandarizar procesos y cultura ágil en toda la organización, asegurando sostenibilidad mediante auditorías periódicas.

### **Acciones Específicas:**

Estandarizar herramientas, procesos y roles en toda la organización.

Realizar auditorías internas semestrales para evaluar cumplimiento CMMI.

Reforzar la cultura de colaboración y mejora continua.

Monitorear KPIs clave y hacer ajustes proactivos.

**Métricas Asociadas:**

Cumplimiento de procesos documentados

Número de bloqueos resueltos

Retroalimentación continua (retrospectivas)

Evaluación de madurez CMMI

**Responsables:**

Consultor CMMI

Dirección General

Coordinador de Calidad

Monitoreo y Ajustes Continuos

Mecanismos de Seguimiento:

**Retrospectivas semanales:** Para identificar áreas de mejora inmediata.

**Encuestas mensuales de satisfacción del equipo y stakeholders.**

**Tableros de control digitales (Jira, Google Docs):** Visualización en tiempo real de KPIs.

**Revisión quincenal de métricas:** Ajustes basados en datos y retroalimentación del equipo.

Ajustes:

- Reconfiguración de flujos de trabajo si se detectan cuellos de botella.
- Rediseño de estrategias de capacitación si la aplicabilidad es baja.
- Intervención temprana ante aumento de defectos o disminución de velocidad.

## Bibliografía

- Ali, A. (2024). *Driving Digital Advancement in SMEs: Harnessing AI for Enhanced Innovation and Operational Excellence*. researchgate.
- Anderson, D. (2010). *Successful Evolutionary Change for Your Technology Business*. Blue Hole Press.  
[https://books.google.com.ec/books/about/Kanban.html?hl=es&id=RJ0VUkfUWZkC&redir\\_esc=y](https://books.google.com.ec/books/about/Kanban.html?hl=es&id=RJ0VUkfUWZkC&redir_esc=y)
- Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Highsmith, J., . . . Thomas, D. (2001). *agilemanifesto.org*.  
<https://agilemanifesto.org/iso/es/manifesto.html>
- Bhatia, S., Yadav, H., & Shawez, M. (2025). Impact of Digital Transformation on the Traditional Business Model. *Gyan Management*, 1-13.
- Bhayyalal, U., & Kothari, D. (2022). *Research Methodology Techniques and Trends*. CRC Press.
- Cohn, M. (2010). *Succeeding with Agile: Software development using Scrum*. Addison-Wesley.  
[https://books.google.com.ec/books/about/Succeeding\\_with\\_Agile.html?id=ldT6AgAAQBAJ&redir\\_esc=y](https://books.google.com.ec/books/about/Succeeding_with_Agile.html?id=ldT6AgAAQBAJ&redir_esc=y)
- DataReportal. (2025). *Digital 2025: Ecuador*.  
<https://datareportal.com/reports/digital-2025-ecuador>
- De León, D. (2024). Transformación digital como factor de cambio de la matriz productiva de Ecuador. *Economía y Desarrollo*, 168(2).
- Espinosa Vélez, M., & Armijos Buitrón, V. (2022). La transformación digital y su incidencia en el e-commerce en Ecuador. *Memorias de la Décima Segunda Conferencia Iberoamericana de Complejidad, Informática y Cibernética (CICIC 2022)*, 169-174.
- Funeraria Jardin De Los Recuerdos. (n.d.). [Videos publicados en su página de Facebook]. Piñas, El Oro.

[https://www.facebook.com/funerariajardindelosrecuerdos/videos?locale=es\\_LA](https://www.facebook.com/funerariajardindelosrecuerdos/videos?locale=es_LA)

GAD Piñas. (2023). *Compilado de actualización del plan de desarrollo y ordenamiento territorial*. <https://pinas.gob.ec/servicios/componente-territorial/>

García Paredes, R., & Merchán Jacome, V. (2022). La conducta del consumidor en la era digital y su incidencia en la búsqueda de información. *RECIAMUC*, 6(1), 379-388.  
[https://doi.org/https://doi.org/10.26820/reciamuc/6.\(1\).enero.2022.379-388](https://doi.org/https://doi.org/10.26820/reciamuc/6.(1).enero.2022.379-388)

Guido Trujillo, V., Desmond Mejía, A., & Lino Rodríguez, A. (2023). Los asistentes virtuales y la calidad de servicio al cliente. *TECHNO REVIEW. International Technology, Science and Society Review/Revista Internacional de Tecnología, Ciencia y Sociedad*, 13(4), 1-10.  
<https://doi.org/https://doi.org/10.37467/revtechno.v13.4816>

Hadi, M., Martel, C., Huayta, F., Rojas, C., & Arias, J. (2023). *Metodología de la investigación guía para proyecto de tesis*. Instituto Universitario de Innovación Ciencia y Tecnología INUDI.

Hernández, S., & Mendoza, T. (2018). *Metodología de la investigación: las rutas cuantitativa, cualitativa y mixta*. McGraw-Hill.

INCIPY. (2025, abril 10). *Madurez AI & Data Driven de las empresas*. Digital disruption:  
[https://incipy.typeform.com/to/GIMJfeFq?utm\\_medium=email&\\_hsenc=p2ANqtz--dBKyy56pS69inkcl8CzFN3JumGENDpdCkNEI45O2D7es5RXHNvo0Ohkb1vxXo5czZJzxBJUM3iluyeb1Mx2dHSIbzb&\\_hsmi=352183246&utm\\_content=352183246&utm\\_source=hs\\_email](https://incipy.typeform.com/to/GIMJfeFq?utm_medium=email&_hsenc=p2ANqtz--dBKyy56pS69inkcl8CzFN3JumGENDpdCkNEI45O2D7es5RXHNvo0Ohkb1vxXo5czZJzxBJUM3iluyeb1Mx2dHSIbzb&_hsmi=352183246&utm_content=352183246&utm_source=hs_email)

INEC. (2022). *Censo Ecuador 2022*. Quito-Ecuador.

Jain, A. (2025). Enhancing e-commerce accessibility with AI-powered chatbots: Integrating accessibility tools and navigation assistants. *Journal of Information Systems Engineering and Management*, 10(15s), 797-809.

- Kottler, P. (1996). Leading Change Why Transformation Efforts Fail. *Harvard Business Review Press*, 11. <https://irp-cdn.multiscreensite.com/6e5efd05/files/uploaded/Leading%20Change.pdf>
- Mendoza, C., Castro, D., Valderrama, G., & Acosta, J. (2024). Optimizing business performance: Marketing strategies for small and medium businesses using artificial intelligence tools. *Migration Letters*, 21(S1), 193-201.
- META. (2025, Junio 8). *Plataforma de WhatsApp Business*. Conversation-Based Pricing: <https://developers.facebook.com/docs/whatsapp/pricing>
- Meta. (2025, Mayo 18). *Precios basados en conversaciones*. Developers Facebook: [https://developers.facebook.com/docs/whatsapp/pricing/?locale=es\\_ES](https://developers.facebook.com/docs/whatsapp/pricing/?locale=es_ES)
- Ministerio de Telecomunicaciones y Sociedad de la Información. (2022). *Agenda de transformación digital del Ecuador 2022-2025*.
- Netzley, M., & Speculand, R. (2022). Digital Maturity Index Report 2022 closing the digital Knowing-Doing Gap. *Bridges*, 2-21.
- Nonaka, I., & Takeuchi, H. (1995). *The Knowledge-creating Company: How Japanese Companies Create the Dynamics of Innovation*. Oxford University Press. [https://books.google.com.ec/books/about/The\\_Knowledge\\_creating\\_Company.html?hl=de&id=B-qxrPaU1-MC&redir\\_esc=y](https://books.google.com.ec/books/about/The_Knowledge_creating_Company.html?hl=de&id=B-qxrPaU1-MC&redir_esc=y)
- Noteboom, C., Ofori, M., Sutrave, K., & El-Gayar, O. (2021). Agile Project Management: A Systematic Literature Review of Adoption Drivers and Critical Success Factors. *Proceedings of the 54th Hawaii International Conference on System Sciences*. <https://scholarspace.manoa.hawaii.edu/server/api/core/bitstreams/37cf9dea-7864-4a93-ab68-b52e44488e53/content>
- OpenAI. (2025, 6 8). *Precios de API*. <https://openai.com/es-419/api/pricing/>
- Osterwalder, A., & Pigneur, Y. (2010). *Business Model Generation*.

- Plekhanov, D., Franke, H., & Netland, T. (2023). Digital transformation: A review and research agenda. *European Management Journal*, 41(6), 821-844.  
<https://doi.org/https://doi.org/10.1016/j.emj.2022.09.007>
- Prosci. (2023). *The Prosci ADKAR Model for Organizational Change Success*.
- Rana, N., Begum, N., Nishat, M., & Mishra, A. (2025). Customer experiences with service robots in hotels: a review and research agenda. *Journal of Hospitality Marketing & Management*, 34.  
<https://doi.org/https://doi.org/10.1080/19368623.2024.2403640>
- Rieth, M., Onnasch, L., & Hagemann, V. (2024). Adaptable automation for a more human-centered work design? Effects on human perception and behavior. *International Journal of Human-Computer Studies*, 186.  
<https://doi.org/https://doi.org/10.1016/j.ijhcs.2024.103246>
- Salazar Mariscal, E., Cajas Rodríguez, M., Angulo Vélez, D., & Ruedas Palacios, F. (2025). Digitalización de las Pymes en Ecuador: Desafíos, oportunidades y casos de éxito. 6(1), 513-524.
- Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide*.  
<https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf>
- Sendpulse. (n.d.). *SendPulse: Automatiza tu marketing*.  
<https://sendpulse.com/latam>
- Septiani, N., Bitsy, A., & Jayanagara, O. (2024). Logistics Business Model Strategies in Facing Changes in Big Data and Blockchain Technology: A Business Model Canvas Approach. *Blockchain Frontier Technology*, 3(2), 126-131.
- SIMERTPI. (2023). *Sistema Municipal de Estacionamiento Rotativo Tarifario de Piñas*. <https://pinas.gob.ec/servicios/componente-de-atencion-a-la-ciudadania/unidad-de-transporte-terrestre-transito-y-seguridad-vial/>
- Sommerville, I. (2016). *Software engineering*. Pearson.  
<https://dn790001.ca.archive.org/0/items/bme-vik-konyvek/Software%20Engineering%20-%20Ian%20Sommerville.pdf>

- Spence, M. (2024). AI'S PROMISE FOR THE GLOBAL ECONOMY. 61(3).
- Statista. (2025). *Most popular social networks worldwide as of February 2025, by number of monthly active users*.  
<https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/>
- Stouby Persson, J., Bruun, A., Kristín Larusd, M., & Axel Nielse, P. (2022, Diciembre). Agile software development and UX design: A case study of integration by mutual adjustment. *Information and Software Technology*, 152. <https://doi.org/https://doi.org/10.1016/j.infsof.2022.107059>
- Tabim, V., Franco, C., & Hoerde, J. (2024). Digital transformation in e-commerce logistics: a case study on the digital maturity of the last-mile area. *Brazilian Journal of Operations and Production Management*, 21(1). <https://doi.org/https://doi.org/10.14488/BJOPM.1641.2024>
- Tuboalabo, A., Buinwi, U., Gbemisola, C., Johnson, E., & Adama, J. (2025). Circular economy integration in traditional business models:. 6(6), 1105-1123.



## **DECLARACIÓN Y AUTORIZACIÓN**

Yo, **Diego Andrés Narváez Narváez** con C.C. # 0926762212 autor del **componente práctico del examen complejo, Estudio de Caso: Transformación Organizacional mediante la Integración de Metodologías Ágiles en una empresa de Software** previo a la obtención del título de **Magíster en Transformación Digital de Negocios** en la Universidad Católica de Santiago de Guayaquil.

1.- Declaro tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de titulación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la SENESCYT a tener una copia del referido trabajo de titulación, con el propósito de generar un repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Guayaquil, **14 de abril de 2026**

f. \_\_\_\_\_

**Diego Andrés Narváez Narváez**

C.C. # 0926762212



## REPOSITORIO NACIONAL EN CIENCIA Y TECNOLOGÍA

### FICHA DE REGISTRO DE TESIS/TRABAJO DE TITULACIÓN

<b>TEMA Y SUBTEMA:</b>	<b>Estudio de Caso: Transformación Organizacional mediante la Integración de Metodologías Ágiles en una empresa de Software</b>		
<b>AUTOR(ES)</b>	Diego Andrés Narváez Narváez		
<b>REVISOR(ES)/TUTOR(ES)</b>	Juan Carlos Maruri Siguenza		
<b>INSTITUCIÓN:</b>	Universidad Católica de Santiago de Guayaquil		
<b>FACULTAD:</b>	Subsistema de Posgrado		
<b>CARRERA:</b>	Maestría en Transformación Digital de Negocios		
<b>TÍTULO OBTENIDO:</b>	Magíster en Transformación Digital de Negocios		
<b>FECHA DE PUBLICACIÓN:</b>	<b>14 de abril del 2026</b>	<b>No. DE PÁGINAS:</b>	95
<b>ÁREAS TEMÁTICAS:</b>	Metodología de desarrollo, Sistemas, Transformación Organizacional		
<b>PALABRAS CLAVES/ KEYWORDS:</b>	Transformación Organizacional, Metodologías Ágiles, Mejora Continua, Organizational Transformation, Agile Methodologies, Continuous Improvement		

#### RESUMEN:

Este proyecto de investigación analiza la transformación organizacional de una empresa de desarrollo de software en Guayaquil que actualmente opera bajo el modelo en cascada, un enfoque lineal que genera retrasos, baja visibilidad del progreso, comunicación deficiente y alta incidencia de defectos. La calidad del producto y la satisfacción del cliente se impacta negativamente debido a las limitaciones. La propuesta del marco integral donde se combinan las metodologías Scrum y Kanban con estándares de calidad del modelo de madurez CMMI tiene como intención mejorar eficientemente la colaboración interdisciplinaria y fomentar la cultura de mejora continua. Desarrollar un modelo que eleve la madurez en el desarrollo de software y fortalezca la cultura organizacional es el objetivo. El estudio combina métodos cualitativos como entrevistas semiestructuradas y cuantitativos como encuestas tipo Likert, complementados con análisis temático y mapeo de procesos, para diagnosticar la situación actual y diseñar estrategias de intervención. La implementación se estructura en tres fases: un piloto para validar el marco, una etapa de escalamiento a otros equipos y una fase de consolidación para estandarizar buenas prácticas. Cada fase incluye capacitación, integración de herramientas colaborativas, ajustes metodológicos y seguimiento mediante KPIs. Se espera que esta transformación reduzca los tiempos de entrega entre un 20% y 30%, disminuya la tasa de defectos entre un 15% y 25%, y aumente la satisfacción del cliente y del equipo en un 30% a 40%, promoviendo una transición sostenible hacia un modelo ágil más competitivo.

#### ABSTRACT

This research project analyzes the organizational transformation of a software development company in Guayaquil that currently operates under the waterfall model, a linear approach that generates delays, low visibility of progress, poor communication, and a high incidence of defects. These constraints negatively affect product quality and customer satisfaction. To overcome these problems, a plan is suggested that combines the agile Scrum and Kanban methods with the maturity of the CMMI model to improve work, collaboration between different areas, and foster a culture of continuous improvement and innovation. The ultimate goal is to create a model that increases maturity in software development and strengthens group culture. The study



combines qualitative methods such as semi-structured interviews and quantitative methods such as Likert surveys, complemented by thematic analysis and process mapping, to diagnose the current situation and design intervention strategies. There are three parts to the implementation process: a pilot to test the framework, a scaling phase to other teams, and a consolidation phase to make sure that best practices are followed. Training, adding collaborative tools, changing methods, and keeping an eye on things through KPIs are all parts of each phase. This change is expected to cut delivery time by 20% to 30%, cut the defect rate by 15% to 25%, and make customers and teams 30% to 40% happier, all of which will help the company move to a more competitive agile model in a way that lasts.

<b>ADJUNTO PDF:</b>	<input checked="" type="checkbox"/> SI	<input type="checkbox"/> NO
<b>CONTACTO CON AUTOR/ES:</b>	<b>Teléfono:</b> 0996531131	E-mail: diego_n2@hotmail.com
<b>CONTACTO CON LA INSTITUCIÓN:</b>	Nombre: (Apellidos, Nombres completos)	
	Teléfono: +593-4-(registrar teléfonos)	
	E-mail: (registrar los emails)	
<b>SECCIÓN PARA USO DE BIBLIOTECA</b>		
<b>Nº. DE REGISTRO (en base a datos):</b>		
<b>Nº. DE CLASIFICACIÓN:</b>		
<b>DIRECCIÓN URL (tesis en la web):</b>		