



Universidad Nacional Mayor de San Marcos

Universidad del Perú. Decana de América

Dirección General de Estudios de Posgrado

Facultad de Ingeniería de Sistemas e Informática

Unidad de Posgrado

**Modelo de gestión de requerimientos para minimizar
el porcentaje de incumplimiento en fase de
planificación de desarrollo de software**

TESIS

Para optar el Grado Académico de Magíster en Ingeniería de
Sistemas e Informática con mención en Gestión de Tecnología de
Información y Comunicaciones

AUTOR

Victor Hugo ALFARO YANGALI

ASESOR

Mg. Ciro AGUILAR LINARES

Lima, Perú

2025



Reconocimiento - No Comercial - Compartir Igual - Sin restricciones adicionales

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

Usted puede distribuir, remezclar, retocar, y crear a partir del documento original de modo no comercial, siempre y cuando se dé crédito al autor del documento y se licencien las nuevas creaciones bajo las mismas condiciones. No se permite aplicar términos legales o medidas tecnológicas que restrinjan legalmente a otros a hacer cualquier cosa que permita esta licencia.

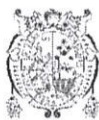
Referencia bibliográfica

Alfaro, V. (2025). *Modelo de gestión de requerimientos para minimizar el porcentaje de incumplimiento en fase de planificación de desarrollo de software*. [Tesis de maestría, Universidad Nacional Mayor de San Marcos, Facultad de Ingeniería de Sistemas e Informática, Unidad de Posgrado]. Repositorio institucional Cybertesis UNMSM.

Metadatos complementarios

Datos de autor	
Nombres y apellidos	Victor Hugo Alfaro Yangali
Tipo de documento de identidad	DNI
Número de documento de identidad	46301637
URL de ORCID	https://orcid.org/0000-0002-6281-8428
Datos de asesor	
Nombres y apellidos	Ciro Aguilar Linares
Tipo de documento de identidad	DNI
Número de documento de identidad	09877278
URL de ORCID	https://orcid.org/0000-0002-2143-9851
Datos del jurado	
Presidente del jurado	
Nombres y apellidos	Cayo Víctor León Fernández
Tipo de documento	DNI
Número de documento de identidad	07001405
Miembro del jurado 1	
Nombres y apellidos	Fany Yexenia Sobero Rodríguez
Tipo de documento	DNI
Número de documento de identidad	20120467
Miembro del jurado 2	
Nombres y apellidos	Nilo Eloy Carrasco Oré
Tipo de documento	DNI
Número de documento de identidad	09342780

Datos de investigación	
Línea de investigación	C.0.3.3. Desarrollo de modelos y aplicación de las tecnologías de información y comunicaciones
Grupo de investigación	Tecnologías, Derechos y Libertades Informativas - TDLI
Agencia de financiamiento	Sin financiamiento
Ubicación geográfica de la investigación	País: Perú Departamento: Lima Provincia: Lima Distrito: Lima Latitud: -12.056445 Longitud: -77.085994
Año o rango de años en que se realizó la investigación	2022-2025
URL de disciplinas OCDE	Ciencias de la Información https://purl.org/pe-repo/ocde/ford#5.08.02



UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
Universidad del Perú. Decana de América
Facultad de Ingeniería de Sistemas e Informática
Vicedecanato de Investigación y Posgrado
Unidad de Posgrado

**ACTA DE SUSTENTACIÓN DE TESIS PARA OPTAR EL GRADO ACADÉMICO
DE MAGÍSTER EN INGENIERÍA DE SISTEMAS E INFORMÁTICA CON
MENCIÓN EN GESTIÓN DE TECNOLOGÍA DE INFORMACIÓN Y
COMUNICACIONES**

A los once (11) días del mes de abril de 2025, siendo las 11:00 am., se reunieron en el Auditorio: Alfredo Celso Alva Bravo, el Jurado de Tesis conformado por los siguientes docentes:

Dr. Cayo Víctor León Fernández (Presidente)
Mg. Fany Yexenia Sobero Rodríguez (Miembro)
Mg. Nilo Eloy Carrasco Oré (Miembro)
Mg. Ciro Aguilar Linares (Miembro Asesor)

Se inició la Sustentación invitando al candidato a Magíster **VICTOR HUGO ALFARO YANGALI**, para que realice la exposición oral de la tesis para optar el Grado Académico de Magíster en Ingeniería de Sistemas e Informática con mención en Gestión de Tecnología de Información y Comunicaciones, siendo la Tesis intitulada:

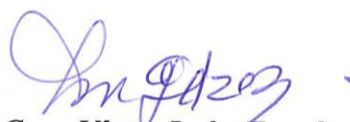
**“MODELO DE GESTIÓN DE REQUERIMIENTOS PARA MINIMIZAR EL
PORCENTAJE DE INCUMPLIMIENTO EN FASE DE PLANIFICACIÓN DE
DESARROLLO DE SOFTWARE”**

Concluida la exposición, los miembros del Jurado de Tesis procedieron a formular sus preguntas que fueron absueltas por el graduando; acto seguido se procedió a la evaluación correspondiente, habiendo obtenido la siguiente calificación:

DIECISEIS (16) BUENO

Por tanto, el presidente del Jurado, de acuerdo con el Reglamento General de Estudios de Posgrado, otorga al Bachiller **VICTOR HUGO ALFARO YANGALI** el Grado Académico de Magíster en Ingeniería de Sistemas e Informática con mención en Gestión de Tecnología de Información y Comunicaciones.

Siendo las 12:10 horas, el presidente del Jurado de Tesis, da por concluido el acto académico de Sustentación de Tesis.


Dr. Cayo Víctor León Fernández
(Presidente)


Mg. Fany Yexenia Sobero Rodríguez
(Miembro)


Mg. Nilo Eloy Carrasco Oré
(Miembro)


Mg. Ciro Aguilar Linares
(Miembro Asesor)



CERTIFICADO DE SIMILITUD

Yo **Ciro Aguilar Linares** en mi condición de asesor acreditado con Dictamen N 0142-UPG-FISI-2018 de la tesis de investigación/trabajo académico, cuyo título es **Modelo de gestión de requerimientos para minimizar el porcentaje de incumplimiento en fase de planificación de desarrollo de software** presentado por el egresado **Victor Hugo Alfaro Yangali** para optar el grado de Maestría en Ingeniería de Sistemas e Informática con mención en Gestión de Tecnología de Información y Comunicaciones CERTIFICO que se ha cumplido con lo establecido en la Directiva de Originalidad y de Similitud de Trabajos Académicos, de Investigación y Producción Intelectual. Según la revisión, análisis y evaluación mediante el software de similitud textual, el documento evaluado cuenta con el porcentaje de 14% de similitud, nivel **PERMITIDO** para continuar con

los trámites correspondientes y para su **publicación en el repositorio institucional.**

Se emite el presente certificado en cumplimiento de lo establecido en las normas vigentes, como uno de los requisitos para la obtención del grado correspondiente.

Firma del Asesor

DNI: 09877278

Nombres y apellidos del asesor: **Ciro Aguilar Linares**



Dedico este trabajo a Dios, por su guía,
comprensión y por sus incansable labor de
hacer de este mundo un lugar mejor para vivir.

AGRADECIMIENTOS

A Dios por cuidarme y protegerme desde siempre.

Al Mg. Ciro Aguilar, por su orientación, dedicación, compromiso y empatía para con este trabajo, que sin lugar a dudas fue un factor determinante para enriquecer de información y que obedece estrictamente a los años de experiencia, de tal manera que este trabajo de investigación cumpla con los objetivos establecidos.

A los colaboradores de las empresas seleccionadas, que me brindaron información valiosa de sus organizaciones.

A mis hijos, mi esposa, mi hermana y mis padres por ser apoyo incondicional en esta ardua labor de enriquecimiento de conocimiento.

MODELO DE GESTIÓN DE REQUERIMIENTOS PARA MINIMIZAR EL PORCENTAJE DE INCUMPLIMIENTO DE PLANIFICACIÓN DE DESARROLLO DE SOFTWARE

RESUMEN

La presente investigación plantea una solución al problema estructural de incumplimiento en la fase de planificación en proyectos de desarrollo de software, generado principalmente por una deficiente gestión de requerimientos. Estos incumplimientos, expresados en sobrecostos, retrasos y reprocesos, afectan la eficiencia operativa y la calidad del producto final. En respuesta, se propone el **diseño e implementación de un modelo de gestión de requerimientos** que articule los lineamientos de la **Norma Técnica Peruana ISO 12207:2008** con los principios y prácticas de **metodologías ágiles**, particularmente **Scrum**, permitiendo así una gestión más adaptativa, validada y orientada a resultados.

La propuesta se sustenta en un enfoque de investigación **aplicativa, explicativa y de diseño cuasi-experimental**, desarrollada en tres fases: diagnóstico de la situación actual, diseño del modelo de gestión, e implementación y validación en entornos reales. Se utilizaron encuestas estructuradas y entrevistas a profundidad en organizaciones del sector tecnológico, complementadas con el uso de herramientas de gestión como Jira y Trello, y técnicas de modelamiento como **SEM (Modelos de Ecuaciones Estructurales)** para validar empíricamente la relación entre las variables de estudio. El modelo introdujo prácticas como la formalización de requerimientos en

artefactos rastreables, mecanismos de revisión iterativa con validación continua y control de cambios bajo criterios de negocio y viabilidad técnica.

Los resultados evidenciaron una **reducción efectiva del porcentaje de incumplimiento de planificación, del 20% inicial a un 9.3% posterior a la implementación del modelo**, lo cual representa una mejora significativa en la predictibilidad del proyecto. Asimismo, el **tiempo promedio de replanificación de requerimientos se redujo de 10 días a 4.2 días**, lo que mejoró la capacidad de respuesta ante el cambio. Las pruebas estadísticas confirmaron la validez del modelo propuesto ($GFI = 0.98$, $CFI = 0.98$, $RMSEA = 0.03$) y demostraron la robustez del enfoque adoptado. Además, se logró una **mejor articulación entre los stakeholders y el equipo de desarrollo**, así como una mayor eficiencia en la identificación, validación y seguimiento de los requerimientos.

Como contribución académica, esta tesis propone una integración metodológica inédita entre estándares normativos y prácticas ágiles para la gestión de requerimientos, ofreciendo un marco aplicable, medible y escalable para organizaciones que buscan mejorar la calidad de su planificación y ejecución en proyectos de software. Desde la perspectiva práctica, se genera un modelo replicable que responde a las necesidades de las empresas peruanas que enfrentan entornos de alta variabilidad y complejidad tecnológica.

Palabras clave — Fase de Planificación de software, Cumplimiento de desarrollo de software, Scrum, Replanificación de requerimientos

REQUIREMENT MANAGEMENT MODEL TO MINIMIZE THE BREACH OF SOFTWARE DEVELOPMENT PLANNING

ABSTRACT

This research addresses the structural problem of planning noncompliance in software development projects, primarily caused by deficiencies in requirements management. Such deviations—manifested through cost overruns, delays, and rework—undermine operational efficiency and the quality of the final product. As a solution, this study proposes the **design and implementation of a requirements management model** that integrates the standards of **Peruvian Technical Standard ISO 12207:2008** with the principles and practices of **agile methodologies**, particularly **Scrum**, enabling a more adaptive, validated, and results-oriented approach to requirements handling.

The study follows an **applied, explanatory, and quasi-experimental research design**, carried out in three stages: assessment of current practices, model design, and implementation with empirical validation. Data were collected through structured surveys and in-depth interviews conducted in organizations from the technology sector. Additionally, the model's impact was evaluated using project management tools such as Jira and Trello, and statistically validated through **Structural Equation Modeling (SEM)** to analyze the causal relationships between the study variables. The model introduced practices such as formalized and traceable requirement artifacts, iterative validation mechanisms, and change control based on business criteria and technical feasibility.

The findings demonstrated a **significant reduction in planning noncompliance**, decreasing from **20% to 9.3%** after implementation, representing a substantial improvement in project predictability. Likewise, the **average re-planning time for requirements was reduced from 10 days to 4.2 days**, enhancing responsiveness to change. Statistical results confirmed the robustness of the proposed model (GFI = 0.98, CFI = 0.98, RMSEA = 0.03), validating the theoretical framework and empirical relationships. Furthermore, the model strengthened collaboration between stakeholders and development teams, and improved the consistency and traceability of requirements throughout the project lifecycle.

As an academic contribution, this thesis presents an original methodological integration of normative standards and agile practices in requirements management, offering an applicable, measurable, and scalable framework for organizations seeking to improve planning quality and execution in software development. From a practical perspective, the proposed model is replicable and well-suited to the needs of Peruvian enterprises operating in dynamic and technologically complex environments.

Keywords — Software Planning Phase, Software Development Compliance, Scrum, Re Requirements Planning Software Planning Phase, Software Development Compliance, Scrum, Re Requirements Planning

ÍNDICE

Capítulo 1: INTRODUCCIÓN	12
1.1. Situación problemática	12
1.2. Formulación del problema	17
1.3. Justificación de la investigación	18
1.3.1. Aporte Teórico	19
1.3.2. Aporte práctico	20
1.4. Objetivos de la investigación	21
1.4.1. Objetivo general	21
1.4.2. Objetivos específicos	21
Capítulo 2: MARCO TEÓRICO	22
2.1 Bases Teóricas	22
2.1.1 Desarrollo de Cascada	22
2.1.1.1 Fases del Modelo	23
2.1.1.1.1 Análisis de requisitos	23
2.1.1.1.2 Diseño de Soluciones	23
2.1.1.1.3 Construcción	24
2.1.1.1.4 Pruebas	24
2.1.1.1.5 Verificación	25
2.1.1.1.6 Mantenimiento	25
2.1.2 Norma Técnica Peruana ISO 12207: 2008	25
2.1.3 Desarrollo Ágil	31
2.1.3.1 Manifiesto Ágil	31
2.1.3.1.1 Principios del Manifiesto Ágil	34
2.1.3.2 Métodos ágiles	34
2.1.3.2.1 Lean Kanban	35
2.1.3.2.2 Extreme Programming (Programación extrema)	37
2.1.3.2.3 Crystal Methods(Métodos Crystal)	37
2.1.3.2.4 Dynamic Systems Development Methods, DSMS	38
2.1.3.2.5 Feature Driven Development, FDD (Desarrollo basado en funcionalidades)	39
2.1.3.2.6 Test Driven Development, TDD (Desarrollo guiado por pruebas)	40
2.1.3.2.7 Adaptative Software Development, ASD	40
2.1.3.2.8 Agile Unified Process, AUP	41

2.1.3.2.9	Domain – Driven Design, DDD	42
2.2	Marcos Conceptuales o Glosario	42
2.2.1	Gestión de requerimientos	42
2.2.1.1	Establecer un plan de gestión de requerimientos	42
2.2.2	Metodología Scrum en el entorno de gestión de requerimientos	44
2.3	Cuadro Comparativo de Modelos de Aseguramiento de la Calidad	46
2.1.1	Criterios de Selección del Modelo	51
Capítulo 3: HIPÓTESIS Y VARIABLES		55
3.1	Hipótesis general	55
3.2	Hipótesis específicas	55
3.3	Identificación de variables	55
3.4	Matriz de Consistencia	59
3.5	Matriz de Operación de variables	62
Capítulo 4: ESTADO DEL ARTE		65
4.1	Integración de Metodologías Ágiles y Tradicionales en la Planificación de Proyectos de Software	65
4.2	Modelos de Gestión de Requerimientos	70
4.3	Impacto en el Rendimiento del Proyecto	71
4.4	Desafíos y Áreas de Mejora	72
Capítulo 5: METODOLOGÍA		72
5.1	Metodología de la investigación	72
5.2	Metodología para el planteamiento del modelo	74
5.3	Modelo de Ecuaciones estructurales	143
5.3.1	Valores de corte de las medidas de ajuste	144
5.3.2	Análisis de fiabilidad	147
5.3.3	Prueba de hipótesis	149
5.3.4	Modelo de ecuaciones estructurales inicial(SEM INICIAL)	149
Capítulo 6: CONCLUSIONES		152
Capítulo 7: RECOMENDACIONES Y TRABAJOS FUTUROS		155
	Recomendaciones	155
	Trabajos Futuros	158
Capítulo 9: ANEXOS		163
	Anexo A: Encuesta sobre la Gestión de Requerimientos en el Desarrollo de Software	166
	Modelo de Entrevistas para la Recolección de Datos	169
	Anexo B: Guión de Entrevistas sobre la Gestión de Requerimientos	169

TABLA DE FIGURAS

ILUSTRACIÓN 1: ESTRUCTURA DE LA NORMA TÉCNICA PERUANA.	20
ILUSTRACIÓN 2: NORMA TÉCNICA PERUANA PARA EL PROCESO DE CICLO DE VIDA DEL	24
ILUSTRACIÓN 3: METODOLOGÍA SCRUM	28
ILUSTRACIÓN 4: METODOLOGÍA SCRUM HÍBRIDA - CASCADA	36
ILUSTRACIÓN 5: METODOLOGÍA SCRUM HÍBRIDA CASCADA - ACTIVIDADES DETALLE	37
ILUSTRACIÓN 6: ETAPA DE PLANIFICACIÓN - METODOLOGÍA CASCADA	38
ILUSTRACIÓN 7: MODELO PROPUESTO - ETAPA DESARROLLO DE REQUISITOS	40
ILUSTRACIÓN 8: MODELO PROPUESTO - ETAPA DESARROLLO DE REQUISITOS DETALLE	41
ILUSTRACIÓN 9: MODELO PROPUESTO - ETAPA DISEÑO DE SOLUCIÓN	42
ILUSTRACIÓN 10: RESULTADOS DEL PERIODO 2014 - 2015 ONP	43
ILUSTRACIÓN 11: MUESTRA DE RESULTADOS DEL PERIODO 2016 ONP	44
ILUSTRACIÓN 12: CRONOGRAMA DE ACTIVIDADES PARA EL PROYECTO DE TESIS	47
ILUSTRACIÓN 13: ENTREGABLES - FASE DESARROLLO DE REQUISITOS	50
ILUSTRACIÓN 14: ENTREGABLES - FASE DISEÑO DE SOLUCIÓN	50
ILUSTRACIÓN 15: PLAN REQUERIMIENTO MENSUAL - CRONOGRAMA	52

TABLA DE CUADROS

TABLA 1: INFORME POR AÑOS DE CHAOS REPORT - INCIDENTES PROYECTOS	11
TABLA 2: NORMA TÉCNICA PERUANA - ETAPAS ACTUALIZADAS	12
TABLA 3: CUADRO ESTADÍSTICO - PETI ONP ELABORADA POR IBM - INCUMPLIMIENTO DE LA PLANIFICACIÓN EN ETAPAS TEMPRANAS	14
TABLA 4: NORMA TÉCNICA PERUANA – ADQUISICIÓN	21
TABLA 5: NORMA TÉCNICA PERUANA ACTUALIZADAS	22
TABLA 6: NORMAS TÉCNICAS INTERNACIONALES ACTUALIZADAS	22
TABLA 7: MATRIZ DE CONSISTENCIA	30
TABLA 8: CUADRO ESTADÍSTICO DE LOS RESULTADOS OBTENIDOS EN EL PERIODO 2014- 2015 ONP	43
TABLA 9: CUADRO DE PRESUPUESTO A REALIZAR	44

Capítulo 1: INTRODUCCIÓN

1.1. Situación problemática

El desarrollo de software es una disciplina que requiere una planificación detallada para asegurar el éxito del proyecto. Una de las fases más críticas en este proceso es la gestión de los requerimientos, especialmente durante la planificación inicial. No obstante, las organizaciones a menudo enfrentan dificultades significativas en esta etapa debido a la falta de un modelo estructurado y eficiente para la gestión de requerimientos, lo que conlleva a altos índices de incumplimiento, replanificaciones constantes y retrasos en la entrega de los proyectos (Pacheco & Garcia, 2020; Inayat et al., 2019).

La identificación y documentación adecuada de los requerimientos son esenciales para el éxito de cualquier proyecto de software. Sin embargo, en la práctica, estas actividades suelen realizarse de manera improvisada, sin seguir una metodología clara y estandarizada. Esta falta de estructura conduce a requerimientos que a menudo son incompletos, ambiguos o inconsistentes. Como resultado, los equipos de desarrollo deben interpretar estos requerimientos a lo largo del ciclo de vida del proyecto, lo que frecuentemente genera malentendidos y errores que deben corregirse en fases posteriores del desarrollo (Azevedo et al., 2019).

Además, la validación exhaustiva de los requerimientos es una etapa crucial que no siempre se realiza adecuadamente. En muchos casos, los requerimientos se aprueban sin un análisis crítico que asegure su viabilidad técnica y su alineación con los objetivos del proyecto. Esto puede llevar a la implementación de funcionalidades que no satisfacen las necesidades del cliente o que resultan técnicamente inviables. La ausencia de un proceso riguroso de validación y aprobación incrementa el riesgo de incumplimiento y la necesidad de replanificación, lo cual impacta negativamente tanto el presupuesto como el cronograma del proyecto (Fernández & Passos, 2021).

Otro desafío significativo es la gestión de los cambios en los requerimientos. En entornos de desarrollo dinámicos, donde las especificaciones pueden cambiar frecuentemente, es crucial contar con un modelo que permita gestionar estos cambios de manera eficiente. La falta de un proceso claro para manejar los cambios obliga a los equipos de desarrollo a realizar ajustes improvisados que no siempre son coherentes ni eficientes. Esto puede derivar en conflictos y retrasos adicionales, afectando negativamente el progreso del proyecto y, en última instancia, la calidad del producto final (Rodríguez et al., 2022).

La ausencia de un modelo de gestión de requerimientos también tiene un impacto considerable en la comunicación y colaboración entre los stakeholders y el equipo de desarrollo. Sin una metodología clara, resulta difícil garantizar que todos los involucrados comprendan y acuerden los requerimientos del proyecto. Esta falta de alineación y comunicación efectiva puede generar expectativas no cumplidas y la insatisfacción del cliente. Por su parte, los equipos de desarrollo enfrentan la dificultad de recibir y gestionar retroalimentación que, a menudo, llega tarde y es contradictoria (Oliveira & Travassos, 2020).

Asimismo, la falta de un modelo estructurado impide la implementación de prácticas efectivas de seguimiento y control. Sin herramientas y métricas adecuadas, es difícil monitorear el progreso del proyecto y detectar problemas a tiempo. Esto puede resultar en desviaciones significativas del plan original y en una baja calidad del producto final. La incapacidad para seguir y controlar adecuadamente los requerimientos afecta la capacidad del equipo de desarrollo para cumplir con los plazos y los estándares de calidad establecidos (Pacheco & Garcia, 2020).

En muchos casos, la resistencia al cambio dentro de las organizaciones también está vinculada a la falta de un modelo de gestión de requerimientos. Los equipos y los

líderes pueden ser reacios a adoptar nuevas metodologías y herramientas, prefiriendo continuar con los enfoques tradicionales que ya conocen. Esta resistencia al cambio puede obstaculizar la implementación de mejoras necesarias para optimizar la gestión de requerimientos. La falta de formación y capacitación en nuevas prácticas y tecnologías también agrava este problema, ya que el personal puede no sentirse preparado para adoptar nuevas formas de trabajo (Fernández & Passos, 2021).

Finalmente, la carencia de un modelo de gestión de requerimientos puede tener un impacto financiero considerable. Los proyectos de software que experimentan altos niveles de incumplimiento y replanificación tienden a exceder sus presupuestos iniciales. Los costos adicionales asociados con la corrección de errores, la implementación de cambios tardíos y los retrasos en la entrega pueden ser significativos. Además, la insatisfacción del cliente resultante de un producto final que no cumple con sus expectativas puede llevar a la pérdida de futuras oportunidades de negocio y a un impacto negativo en la reputación de la organización (Rodríguez et al., 2022).

La Tabla 1 muestra los informes anuales del Chaos Report sobre los incidentes en proyectos de desarrollo de software. Esta información es crucial para entender las tendencias y patrones en el incumplimiento de los proyectos a lo largo de los años, permitiendo una comparación directa de las mejoras en la gestión de requerimientos y la planificación de proyectos.

Tabla 1: Informe por años de Chaos Report - Incidentes proyectos

Detalle/Gestión	2018	2019	2018	2019	2020	2021	2022	2023	2024
Proyectos									
“Dentro del tiempo y presupuesto”	16%	27%	26%	28%	34%	29%	35%	32%	37%
Excedidos en tiempo y coste	53%	33%	46%	49%	51%	53%	46%	44%	42%
Cancelados	31%	40%	28%	23%	15%	18%	19%	24%	21%

Fuente: Chaos Report

La **Tabla 2** detalla las etapas actualizadas de la Norma Técnica Peruana, proporcionando un marco de referencia para la gestión del ciclo de vida del software en el contexto peruano. Este marco es esencial para alinear las prácticas locales con los estándares internacionales.

Tabla 2: Norma Técnica Peruana - Etapas actualizadas

ADQUISICIÓN
Inicio
Preparación de la solicitud de propuestas
Preparación y actualización del contrato
Seguimiento del proveedor
Aceptación y finalización

Fuente: Norma Técnica Peruana

En la **Tabla 3** se presenta un cuadro estadístico elaborado por IBM para la PETI de la ONP, destacando los niveles de incumplimiento en las etapas tempranas de la planificación. Estos datos son fundamentales para identificar las áreas críticas donde se requiere mejorar la gestión de requerimientos.

**Tabla 3: Cuadro estadístico - PETI ONP elaborada por IBM -
Incumplimiento de la planificación en etapas tempranas**

Etapas/Me s	01	02	03	04	05	06	07	08	09	10	11	12	Promedio
Desarrollo	50	76	68	64	60	70	50	76	68	55	40	80	63.08
Requisitos	%	%	%	%	%	%	%	%	%	%	%	%	%
Diseño	57	58	57	57	50	70	57	58	57	40	50	70	56.75
Soluciones	%	%	%	%	%	%	%	%	%	%	%	%	%

Fuente: PETI ONP, elaborada por IBM.

1.2. Formulación del problema

¿Cómo la aplicación de un modelo de gestión de requerimientos estructurado, alineado a la NTP ISO 12207:2008 y metodologías ágiles, reduce la desviación del compromiso de planificación (tiempo y costo) en proyectos de desarrollo de software en el sector tecnológico peruano?

Este modelo busca mejorar la validación y el seguimiento de los requerimientos, así como asegurar una comunicación y colaboración más efectiva entre los stakeholders y el equipo de desarrollo, estableciendo métricas claras para medir estos aspectos.

Contexto para la formulación del problema:

La planificación en el desarrollo de software es esencial, particularmente en la gestión de requerimientos. La ausencia de un modelo estructurado frecuentemente conduce a incumplimientos, replanificaciones y retrasos, ya que los requerimientos suelen ser incompletos y ambiguos, lo que genera malentendidos y errores que deterioran la calidad del producto final. La falta de validación adecuada de los requerimientos puede resultar en funcionalidades ineficaces. Además, sin un proceso claro para gestionar los cambios, se realizan ajustes improvisados que retrasan el proyecto, mientras que la mala comunicación entre los interesados y el equipo de desarrollo exacerba estos problemas. La carencia de prácticas efectivas de seguimiento y control dificulta la detección temprana de problemas, lo que incrementa los costos y afecta tanto los plazos como la calidad. La resistencia al cambio y la falta de formación en nuevas metodologías también agravan estos desafíos, impactando negativamente en las finanzas y la reputación de la organización (Sheuly, 2020).

1.3. Justificación de la investigación

La gestión de requerimientos es un área crítica en el desarrollo de software, y su correcta implementación impacta directamente en el éxito de los proyectos. Sin embargo, a pesar de los avances en metodologías ágiles y la adopción de estándares internacionales, las empresas aún enfrentan dificultades para cumplir con los plazos establecidos. Esto se debe principalmente a una gestión ineficiente de los requerimientos (Fernández & Passos, 2021). Por lo tanto, esta investigación es relevante ya que propone un modelo de gestión de requerimientos que aborda estos desafíos mediante la implementación de

prácticas basadas en la Norma Técnica Peruana ISO 12207:2008 y principios ágiles como Scrum, con el fin de reducir el porcentaje de incumplimiento en la planificación de proyectos de software.

1.3.1. Aporte Teórico

Desde una perspectiva teórica, esta investigación propone una **contribución significativa al campo de la gestión de proyectos de software**, particularmente en la disciplina de gestión de requerimientos, al integrar de forma estructurada dos enfoques tradicionalmente disociados: el **modelo normativo de la NTP ISO/IEC 12207:2008** y las **prácticas ágiles de frameworks como Scrum**. Esta integración, conceptualizada en el modelo propuesto, permite abordar las limitaciones observadas tanto en metodologías prescriptivas como en enfoques adaptativos, generando un marco conceptual más equilibrado, robusto y flexible. Asimismo, la investigación **incorpora un enfoque de validación estadística mediante Modelos de Ecuaciones Estructurales (SEM)**, herramienta que rara vez ha sido utilizada en estudios sobre requerimientos de software en el contexto latinoamericano. Esta elección metodológica permite **verificar empíricamente las relaciones causales** entre las variables propuestas, aportando rigor académico y enriqueciendo la base teórica con evidencia cuantitativa sólida. El modelo teórico resultante **contribuye al desarrollo de un marco explicativo sobre cómo el cumplimiento estructurado de actividades del proceso de requerimientos influye directamente en la reducción de desviaciones en la planificación del desarrollo de software**. Esta perspectiva no solo fortalece la literatura existente, sino que además genera un punto de partida para futuras investigaciones que deseen explorar la relación entre normativas técnicas,

enfoques ágiles y resultados de planificación en distintos contextos organizacionales.

1.3.2. Aporte práctico

En el plano práctico, esta tesis desarrolla un **modelo aplicable, replicable y validado** que permite **mejorar la gestión de requerimientos en proyectos reales del sector tecnológico peruano**, enfrentando un problema concreto: la elevada desviación del compromiso de planificación debido a fallas en la especificación, trazabilidad y replanificación de los requerimientos.

El modelo propuesto fue implementado en un entorno de prueba real con resultados medibles: **la desviación del compromiso de planificación se redujo de 20% a 9.3%**, y el **tiempo promedio de replanificación de requerimientos pasó de 10 días a 4.2 días**. Estas mejoras fueron posibles gracias a la introducción de herramientas de control, procesos de validación continua, y prácticas colaborativas centradas en la claridad y completitud de los requerimientos desde el inicio del proyecto.

Este modelo, más allá de sus resultados inmediatos, representa una **propuesta operativa que puede ser adoptada por organizaciones del sector tecnológico que enfrentan entornos dinámicos**, altos niveles de cambio, y exigencias de eficiencia en la entrega de software. Su implementación fortalece los procesos de ingeniería de requisitos, incrementa la trazabilidad, disminuye los retrabajos y mejora la comunicación entre stakeholders, equipos de desarrollo y áreas de negocio.

Finalmente, el modelo puede ser **incorporado en políticas organizacionales de madurez de procesos o modelos de gobernanza técnica**, convirtiéndose en un

instrumento estratégico para elevar el desempeño de los equipos de desarrollo, reducir el time-to-market y generar valor sostenible para el negocio.

1.4. Objetivos de la investigación

I.4.1 Objetivo general

Diseñar e implementar un modelo de gestión de requerimientos, basado en la Norma Técnica Peruana ISO 12207:2008 y en prácticas ágiles, que contribuya a reducir la desviación del compromiso de planificación del desarrollo de software del 20% al 10%, mediante la mejora del cumplimiento de actividades del proceso de requerimientos en proyectos del sector tecnológico peruano.

I.4.2 Objetivos específicos

- Optimizar la revisión de requerimientos en los proyectos de desarrollo de software, incrementando el cumplimiento de actividades del proceso de requerimientos de un 68% a un 90%, mediante la implementación de procedimientos ágiles que reduzcan la necesidad de replanificación.
- Reducir el tiempo promedio de replanificación de requerimientos de 10 días a menos de 5 días, mediante prácticas ágiles que permitan una mayor capacidad de adaptación al cambio y mejor cumplimiento de los plazos establecidos.
- Mejorar la calidad de los requerimientos desde el inicio del proyecto, incrementando el índice de validación y completitud de requerimientos del 65% al 85%, a través de herramientas de trazabilidad y procesos de validación continua.

Capítulo 2: MARCO TEÓRICO

2.1 Bases Teóricas

2.1.1 Desarrollo de Cascada

La versión original del modelo en cascada fue postulada por Winston W. Royce en 1970 (Royce, 1970), y luego fue actualizada por Barry Boehm en 1980 e Ian Sommerville en 1985 (Boehm, 1980; Sommerville, 1985).

El desarrollo en cascada, como su propio nombre lo indica, se caracteriza por tener etapas en caída como si simulara una cascada. Donde es requisito fundamental y obligatorio, que se culmine una etapa para pasar a la siguiente. Existen proyectos con un estado actual que necesitan de utilizar dicho desarrollo en cascada puesto que con la rigurosidad del asunto, la hipótesis a manejar es que la calidad aumente. En los últimos años dicha calidad no ha aumentado del todo, puesto que para pasar de una etapa a otra es necesario que se termine la anterior, entonces la re priorización es constante y eso genera desfase en entregas de tiempo y alcance.

Las etapas que postula el desarrollo en cascada son las siguientes:

- Análisis de la coyuntura
- Diseño del software
- Diseño del programa
- Desarrollo del software
- Pruebas o testings
- Verificación del usuario
- Mantenimiento del sistema

Esta forma de desarrollar ha sido muy polemizada por sus resultados y por sus estadísticas(“Una Visión de La Enseñanza de La Ingeniería de Software Como Apoyo al Mejoramiento de Las Empresas de Software,” 2012), pero hoy por hoy en el 2019 es una de las metodologías más usadas y aceptadas, puesto que la mayoría de proyectos necesita tener un control más riguroso al momento de desarrollar(Punt & Vucic, 2024). Más adelante el presente trabajo de tesis, hace una comparativa entre el desarrollo en cascada y el desarrollo ágil donde se prima más la entrega de valor continua que ser rigurosos en la calidad.

2.1.1.1 Fases del Modelo

2.1.1.1.1 Análisis de requisitos

Es la primera etapa del ciclo(Punt & Vucic, 2024) de vida, aquí se recogen todas las necesidades de los usuarios finales que van a interactuar con el software a desarrollar. Como entregable final se obtiene un SRD(Rabioglio et al., 2024) que son las siglas de un Documento de especificación de requisito de software, este documento contiene todos los requerimientos necesarios que son aterrizados a través del usuario(Rabioglio et al., 2024).

Vale decir que esa etapa es fundamental(Rabioglio et al., 2024) puesto que más adelante no se puede modificar dicho entregable en el tiempo. Es recomendable utilizar técnicas de recopilación de información tales como entrevistas, foros, brainstorming(Koshovyi & Kirei, 2024), etc.

2.1.1.1.2 Diseño de Soluciones

Esta etapa de diseño de soluciones engloba un esfuerzo de arquitectura, que a su vez determina cual es la base de los desarrollos a determinar. Como resultado final se obtienen componentes de desarrollo, y se trabajan con fases de

desarrollo. Es una buena práctica realizar documentos de diseño de software(Koshovyi & Kirei, 2024).

Algunas buenas prácticas también involucran un diseño de pantallas a través de la experiencia de usuario. En la metodología en cascada se puede agregar este esfuerzo al esfuerzo de arquitectura para tener una solución más robusta.

El alcance de esa etapa se define con los entregables de arquitectura, a través de lineamientos entregados, un proyecto, un espacio de trabajo para que los desarrollados puedan ingresar sus algoritmos de solución. Así como también se manejan repositorios de diseño de experiencia de usuario para tener la voz del usuario desde etapa temprana(Koshovyi & Kirei, 2024).

2.1.1.1.3 Construcción

La fase de construcción en el desarrollo de software implica la codificación de las soluciones diseñadas en etapas anteriores. Durante esta fase, los desarrolladores implementan la lógica del sistema utilizando lenguajes de programación y frameworks adecuados para cumplir con los requerimientos del proyecto (Sommerville, 2015). Es crucial mantener una alta calidad en el código, siguiendo buenas prácticas de desarrollo que aseguren la funcionalidad y mantenibilidad del software, lo cual minimiza los errores en las etapas posteriores del ciclo de vida (Royce, 1970).

2.1.1.1.4 Pruebas

Las pruebas son una fase esencial en el desarrollo de software, donde se verifica que el sistema cumpla con los requerimientos especificados. Las pruebas incluyen tanto pruebas funcionales como pruebas de regresión, asegurando que

las nuevas funcionalidades no afecten negativamente el sistema existente (Pressman, 2020). Es recomendable utilizar herramientas automatizadas para mejorar la eficiencia en esta fase, garantizando una validación continua del código y reduciendo los defectos en el producto final (Myers, 2014).

2.1.1.1.5 Verificación

La verificación es el proceso en el que el software es revisado por los usuarios finales para asegurarse de que cumple con sus expectativas y requisitos. Esta fase es crucial para identificar posibles discrepancias entre lo desarrollado y lo esperado por el cliente, y para realizar las correcciones necesarias antes de la entrega final (Boehm, 1980). A través de la verificación, se puede garantizar que el software no solo sea funcional, sino que también esté alineado con las necesidades del usuario final (Punt & Vucic, 2024).

2.1.1.1.6 Mantenimiento

Cuando hablamos de la etapa de mantenimiento, existe una gran estadística, que indica que más del 50% de los recursos son destinados a generar mantenimiento de software, esto es producto de que el resultado final no tenga el 100% de las expectativas que el cliente necesite (Koshovyi & Kirei, 2024).

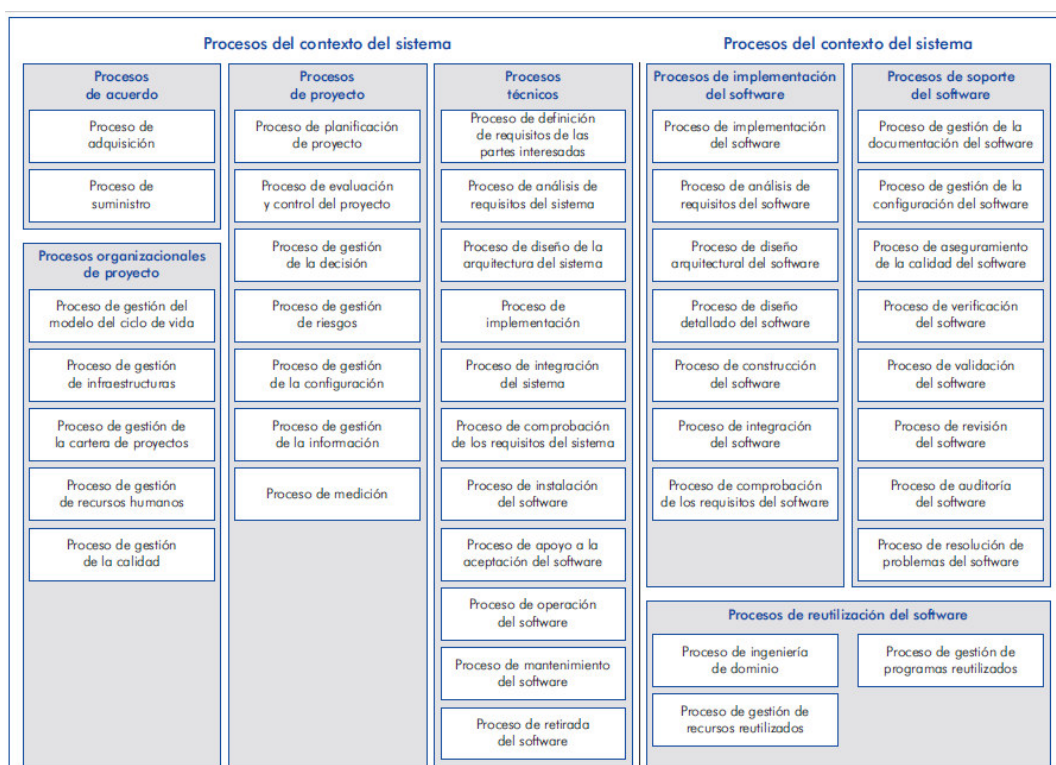
2.1.2 Norma Técnica Peruana ISO 12207: 2008

En el Perú, según la Norma Técnica Peruana (ICONTEC, 2015), existe una proliferación de marcos de trabajo referenciales para el desarrollo de software. Para la gestión de requerimientos menciona principalmente el proceso de Adquisición donde es muy adecuado que desde etapas tempranas de requisito de

sistema, se involucren a los requisitos propios del negocio que genere valor a través del tiempo, sobretodo que se nexa para otros requisitos que son también de alta prioridad.

La estructura de la Norma Técnica Peruana establece los estándares necesarios para el desarrollo de software en el país, lo que garantiza que los procesos se realicen bajo lineamientos específicos que aseguren la calidad del producto final. Como se muestra en la Ilustración 1, esta estructura incluye diferentes fases clave que deben seguirse rigurosamente para cumplir con la normativa establecida.

Ilustración 1: Estructura de la Norma Técnica Peruana.



Fuente: Norma Técnica Peruana

La Norma Técnica peruana proporciona la etapa de Adquisición y sus fases para la gestión de requerimientos: Inicio, Preparación de la solicitud de propuestas, Preparación y actualización del contrato, Seguimiento del proveedor, aceptación

y finalización(Punt & Vucic, 2024). De acuerdo con la (Oficina Nacional de Gobierno Electrónico e Informática) brinda la puesta en marcha de la Norma Técnica Peruana “NTP- ISO 12207:2008 Tecnología de Información, Proceso del ciclo de vida del desarrollo de software, 1era edición” para que se aplique(Punt & Vucic, 2024).

La **Tabla 4** enfoca los procesos de adquisición según la Norma Técnica Peruana, clave para cualquier organización que participe en la compra de software o servicios tecnológicos. Detallar estos procesos ayuda a las empresas a asegurarse de que sus prácticas de adquisición cumplen con los estándares nacionales e internacionales, minimizando riesgos legales y mejorando la eficiencia en la adquisición de tecnología.

Tabla 4: Norma Técnica Peruana – Adquisición

ADQUISICIÓN
Inicio
Preparación de la solicitud de propuestas
Preparación y actualización del contrato
Seguimiento del proveedor
Aceptación y finalización

Fuente: Norma Técnica Peruana

La Norma Técnica Peruana tiene como requisitos para su estructura las siguientes Normas Nacionales e Internacionales:

En la **Tabla 5** se resumen las actualizaciones recientes a la Norma Técnica Peruana. Esta tabla es crucial para mantenerse al día con los cambios en los estándares de gestión y desarrollo de software, asegurando que las prácticas de las organizaciones permanezcan relevantes y eficientes en un entorno regulatorio que está constantemente evolucionando.

Tabla 5: Norma Técnica Peruana Actualizadas

Normas Técnicas Peruanas	
NTP-ISO 9000:2001	Sistema de gestión de calidad Fundamentos y vocabularios
NTP-ISO 9001:2001	Sistemas de Gestión de Calidad, Requisitos

Fuente: Norma técnica peruana

La **Tabla 6** ofrece una comparativa de las Normas Técnicas Internacionales actualizadas, proporcionando una perspectiva global sobre los estándares de desarrollo de software. Esta comparación es fundamental para empresas que operan en múltiples países o que buscan expandirse internacionalmente, asegurando que sus prácticas estén alineadas con las mejores y más recientes normativas globales.

Tabla 6: Normas Técnicas Internacionales Actualizadas

Normas Técnicas Internacionales	
ISO 9000:2000	Quality management systems – Fundamentals and vocabulary
ISO 9001:1994	Quality systems – Model for quality assurance in design, development,

	Production, installation and servicing.
ISO/IEC 2382 - 1:1993	Information technology – Vocabulary – Part 1: Fundamental terms
ISO/IEC 2382 - 20:1990	Information technology – Vocabulary – Part 20: System development.
ISO/IEC 9126 - 1:2001	Software engineering -- Product quality - - Part 1: Quality model (available in English only)
ISO/IEC 12119:1994	Information technology -- Software packages -- Quality requirements and testing
ISO/IEC 12207:1995	Information technology-software life cycle processes
ISO 14001:1996	Environmental management systems -- Specification with guidance for use
ISO/IEC 14598 - 1:1999	Information technology -- Software product evaluation -- Part 1: General overview
ISO/IEC 15504 - 2:2003	Software Engineering—Software process assessment – Part 2: Performing an assessment.
ISO/IEC TR 9126 - 2:2003	Software engineering -- Product quality - - Part 2: External metrics (available in English only)
ISO/IEC TR 9126 - 3: 2003	Software engineering -- Product quality - - Part 3: Internal metrics (available in English only)

Fuente: Revisión de normas técnicas internacionales

La Norma Técnica peruana proporciona la etapa de Adquisición y sus fases para la gestión de requerimientos: La primera etapa es la de Inicio, seguida definitivamente por la Preparación de la solicitud de propuestas, pasando por Preparación y actualización del contrato, también se involucra el Seguimiento del proveedor, y por último para cerrar aceptación y finalización (Ali & Lai, 2016). De acuerdo con la Oficina Nacional de Gobierno Electrónico e Informática brinda la puesta en marcha de la Norma Técnica Peruana “NTP- ISO 12207:2008 Tecnología de Información, Proceso del ciclo de vida del desarrollo de software, 1era edición” para que se aplique (Ali & Lai, 2016).

De acuerdo con la presentación de la Norma ISO/IEC 12207: 2008 se ha reformulado la distribución de los grupos de procesos (Punt & Vucic, 2024):

1. Procesos de acuerdo
2. Procesos organizacionales de proyecto
3. Procesos de proyecto
4. Procesos técnicos.
5. Procesos de implementación del software.
6. Procesos de soporte del software.
7. Procesos de reutilización del software.

Nuestra presente investigación se centraliza básicamente en el proceso de acuerdo.

En el ámbito del desarrollo de software, el proceso de ciclo de vida es fundamental para asegurar que todas las etapas, desde la concepción hasta el mantenimiento, se lleven a cabo de manera eficiente y conforme a los estándares.

En la Ilustración 2, se detalla la Norma Técnica Peruana para el proceso de ciclo de vida del software, que guía a los desarrolladores en la correcta implementación de estas fases.

Ilustración 2: Norma técnica peruana para el proceso de ciclo de vida del Software ISO 12207: 2008

Apartado	Proceso	Propósito
6.1.1	Proceso de adquisición	El propósito de este proceso es obtener el producto o servicio que satisface la necesidad expresada por el comprador. El proceso comienza con la identificación de las necesidades del cliente y finaliza con la aceptación del producto o servicio necesitado por el comprador.
6.1.2	Proceso de suministro	El propósito de este proceso es proveer al comprador un producto o servicio que cumpla los requisitos acordados.

Fuente: Norma Técnica Peruana

2.1.3 Desarrollo Ágil

El desarrollo ágil es una metodología que promueve la entrega continua y rápida de software funcional, enfocándose en la colaboración entre equipos y la adaptación a los cambios. A diferencia de los enfoques tradicionales como el modelo en cascada, el desarrollo ágil permite iteraciones más cortas y flexibles, lo que mejora la capacidad de respuesta a los cambios en los requerimientos del proyecto (Beck et al., 2001). Los principios ágiles destacan la importancia de la interacción humana sobre los procesos y herramientas, y priorizan la entrega de software sobre la documentación extensa (Highsmith, 2009).

2.1.3.1 Manifiesto Ágil

El desarrollo ágil es una tendencia muy marcada donde se prioriza más el valor que los tiempos de entrega y la documentación (Agile Methodology, 2008).

Individuos e interacciones sobre procesos y herramientas. Software de trabajo sobre documentación comprensiva. Colaboración del cliente sobre la negociación del contrato. Respondiendo al cambio siguiendo un plan. Dichos argumentos son los principales valores que demuestran la correcta interacción de un grupo de trabajo que practica todas las buenas prácticas ágiles.

Un poco de historia será objetiva para precisar el trabajo de tesis, el año 2001 en el mes de febrero, se reunieron 17 personas que eran especialistas de informática, programadores de software y administradores de profesión. Se juntaron en dicho foro para conversar y poner sobre la mesa mesas de métodos de desarrollo de software a medida, ágil y liviano. Producto de dicho foro y después de muchas discusiones se formó la Agile Alliance o Alianza Ágil en español. Como resultado y contenido de las reuniones se generaron los acuerdos del Manifiesto Ágil para el desarrollo de software (Manifiesto for Agile Software Development)(Huang et al., 2024). Dicho contenido del Manifiesto fue escrito por los señores Fowler y Highsmith(Punt & Vucic, 2024) en el 2001 y seguido y suscrito por todos los asistentes con la finalidad de establecer los lineamientos básicos del Manifiesto Ágil(“Estado de Arte Sobre Métodos de Evaluación de Metodologías Ágiles En Las Pymes,” 2016).

- Individuos e interacciones sobre procesos y herramientas

Es correcto comentar que las herramientas y procesos son fundamentales en todo proceso de desarrollo de software, sin embargo uno de los principios ágiles del Manifiesto son priorizar principalmente a las personas e interacción de las mismas, puesto que está totalmente comprobado que colocando foco a las personas se obtienen mejores resultados a través de todo el proceso. Quienes asumen y se responsabilizan a implementar un proyecto de desarrollo. Es de

suma importancia indicar que los actores principales al momento de implementar no son las herramientas sino las personas y las interacciones entre ellas.

- Software funcionando sobre documentación extensiva

El siguiente principio en mención es muy importante y llama a la reflexión. La pregunta es totalmente sencilla, ¿Qué es lo que priman nuestros clientes al momento de entregar resultados? ¿Alguna parte del sistema que funcione o documentación extensiva? Pues actualmente los resultados clave son software funcionando, software completo o parte de ella que realmente genere valor en la actualidad. Este principio no indica que no deba documentarse, al contrario una recomendación fundamental es que si se documente, pero que se documente lo necesario y elemental.

- Colaboración con el cliente sobre negociación contractual

La Colaboración con el cliente sobre negociación contractual, este principio indica claramente que la colaboración con nuestros clientes y haciéndolos partícipes y parte del equipo de desarrollo es fundamental puesto que ya tienes el punto de vista en base a sus necesidades en todo momento. Anteriormente se primaba más la negociación contractual con el cliente antes de escuchar sus necesidades principales. El equipo de desarrollo y el cliente(Cevallos Karla, 2015) tienen que colaborar conjuntamente y generar sinergias para conseguir los resultados esperados.

- Responder ante el cambio sobre seguir un plan

Responder ante el cambio se ha convertido en algo parte de nuestras vidas. Es fundamental saber que una buena salud de un proyecto se da cuando garantizamos una rápida respuesta frente a un cambio imprevisto. Este principio

tiene mucho que ver con ello, puesto que seguir un plan, que a la larga se cumple solo un 30% por estadística ya pasó a la historia. Nuestra capacidad de adaptación deberá ser lo suficientemente correcta para poder responder las necesidades del mercado. Que por cierto son de forma obligatoria, tenemos que validar y revisar los casos como Kodak, donde su respuesta al cambio fue muy pesada y para cuando se dieron cuenta ya era demasiado tarde.

2.1.3.1.1 Principios del Manifiesto Ágil

El Manifiesto Ágil establece cuatro principios fundamentales para guiar el desarrollo de software: individuos e interacciones sobre procesos y herramientas, software funcional sobre documentación extensa, colaboración con el cliente sobre negociación contractual, y respuesta ante el cambio sobre seguir un plan. Estos principios fueron formulados para mejorar la eficiencia y flexibilidad en el desarrollo de software, y han sido adoptados ampliamente por la industria desde su creación en 2001 (Beck et al., 2001). La aplicación de estos principios permite una mayor adaptación a las necesidades cambiantes del cliente y una mejor comunicación dentro del equipo de desarrollo (Fowler & Highsmith, 2001).

2.1.3.2 Métodos ágiles

Los métodos ágiles incluyen una variedad de enfoques, como Scrum, Kanban, y Extreme Programming (XP), que permiten una gestión eficiente de proyectos de software en entornos dinámicos. Scrum, por ejemplo, utiliza ciclos cortos llamados sprints para entregar incrementos funcionales del software de manera continua (Schwaber & Sutherland, 2017). Kanban, por otro lado, se enfoca en la

visualización del flujo de trabajo para mejorar la eficiencia y reducir cuellos de botella (Anderson, 2010). Estos métodos proporcionan una estructura que mejora la productividad y la calidad del software, al mismo tiempo que permiten a los equipos adaptarse rápidamente a los cambios en los requerimientos (Larman & Vodde, 2009). A continuación detallamos las principales:

- Lean Kanban
- Extreme Programming (XP) – Programación Extrema
- Crystal Methods – Métodos Crystal
- Dynamic Systems Development Methods (DSDM) – Métodos de desarrollo de sistemas dinámicos
- Feature Driven Development (FDD) – Desarrollo basado en funcionalidades
- Test Driven Development (TDD) – Desarrollo guiado por pruebas
- Adaptive Software Development (ASD) – Desarrollo adaptativo de software
- Agile Unified Process (AUP) – Proceso Unificado Ágil
- Domain Driven Development (DDD) – Desarrollo guiado por el dominio:

2.1.3.2.1 Lean Kanban

El concepto de Lean optimiza el sistema de una organización a fin de producir resultados valiosos con base en sus recursos, necesidades y alternativas, a la vez que disminuyen todas las pérdidas durante todo el proceso. Los defectos y pérdidas son posibles y se generan a través de la fabricación de algo erróneo, la falta de aprendizaje o prácticas que imposibiliten el proceso. Es posible que gracias a esos factores tienen la naturaleza muy cambiante un sistema Lean evalúa todo su sistema e integración y refina constantemente sus procesos. El

principio de Lean es que por cada ciclo o interacción se pueda reducir la duración de cada ciclo buscando la excelente técnica y simpleza del proceso, lo cual genera el incremento de la productividad, de tal manera que se puedan simplificar retrasos, detectar errores y finalizar tareas de manera mas rapida. En consecuencia los principios Lean de desarrollo siempre se pueden implementar con éxito en un desarrollo de software que también comparte generalmente las etapas de su proceso y los principios para siempre mejorar.

El método Kanban(“Estado de Arte Sobre Métodos de Evaluación de Metodologías Ágiles En Las Pymes,” 2016) significa en Japonés mejora, significa letrero o cartilla, sus valores y principios indican claramente trabajar de manera visual todo, tratar de liberar nuestra mente y trasladar toda esa información de manera visual. De tal manera que nuestro cerebro pueda guardar y reservar mucha más información. Dicho concepto a nivel teórico fue impuesto por Taiichi Ohno, que es el padre de los Sistemas de producción TOYOTA. Esta buena práctica de mantener todo de forma visual se ha implementado a lo largo de varios proyectos a nivel mundial y se ha convertido en una práctica común. Existen muchas variaciones incluso hay proyectos que han agregado métricas de control y de mejora continua tales como Burndown Chart, Cycle Time, índices de mejora, etc. El método Kanban cuenta con sus propios principios y valores capaces de generar equipos de alto rendimiento y equipos autos organizados capaces de generar valor constante a través de un flujo de actividades. A través de la presente tesis consideramos importante generar buenas prácticas desde la base del método Kanban.

2.1.3.2.2 Extreme Programming (Programación extrema)

Programación extrema, fue creada inicialmente en el año 1990 en Chrysler corporation, fue muy impulsado en los años 90, la premisa de la programación extrema es evitar de manera radical el costo que demanda realizar cambios del software a lo largo del tiempo. Muchas de las características de la programación extrema incluyen básicamente desarrollo incremental, horas y horarios flexibles, programación de códigos de prueba automatizadas. Comunicación cara a cara y el diseño orgánico, cambiante siempre y siempre mantener actualizada la lista de todos los involucrados durante todo el **proceso**.

Uno de los puntos más importantes valorados por Programación Extrema es la retroalimentación y comunicación, generar valor y sobretodo la simplicidad de sus procesos. Cuenta con distintos roles, y son los siguientes:

Desarrollador

Cliente

Tracker

Coach

Cuenta también con roles y artefactos propios del framework. Es todo un éxito la adopción de la Programación extrema en distintos ámbitos puesto que sus procesos de ingeniería están bien desarrollados.

2.1.3.2.3 Crystal Methods(Métodos Crystal)

Alistair Cockburn(Cockburn et al., 2014) fue la persona que introduce la metodología Crystal alineada para el desarrollo de software durante el año 1990. El objetivo principal de esta metodología es enfocarse siempre en las personas,

totalmente adaptables y ligeros al cambio. Valoran mucho la respuesta ante el cambio respecto a las herramientas, como las personas son primordiales el proceso no es fijo. En lugar de ello los requerimientos y características del proyecto son adaptables de acuerdo a las prioridades del cliente.

Es una buena práctica utilizar el espectro de colores para afinar y enrumbar la dirección del proyecto, a su vez se utiliza el espectro para determinar términos de comodidad, utilizar los recursos a discreción juegan un rol determinante en el peso de toda la metodología.

La división de Crystal(Rabioglio et al., 2024) es la siguiente: Crystal Clear, que significa claro como el cristal, Crystal Yellow, cristal amarillo tiene como significado, Crystal Orange, significa Cristal Naranja, Crystal Orange Web, cristal naranja, Crystal Red cristal rojo, Crystal Maroon, cristal marrón, Crystal Diamond, cristal diamante y Cristal Sapphire, que significa cristal Zafiro.

El total de métodos Crystal mantienen cuatro roles: Executive sponsor (patrocinador ejecutivo), lead designer (diseñador líder), desarrolladores y usuarios senior. Dichos métodos recomiendan como buenas prácticas varias estrategias y técnicas para alcanzar la agilidad en sus procesos. Una correcta implementación del acta, el total desarrollo de entrega y cierre wrap-up.

2.1.3.2.4 Dynamic Systems Development Methods, DSMS

Dynamics es un framework de desarrollo de sistemas dinámicos, por sus siglas en inglés DSMS. Un punto principal del framework es que fijan la calidad y el esfuerzo a través de términos importantes como el costo y el tiempo al inicio de las actividades, de tal manera que se van ajustando los entregables por proyectos siempre con foco principal en cumplir los objetivos del proyecto. Otro principio

fundamental es la priorización constante de los entregables en las categorías. Esta práctica es muy importante puesto que determinan la capacidad de toda la estrategia en base a esta medición: Debe tener, Debería tener, Podría tener y No tendrá. La técnica para priorización es la famosa MoSCoW. Para complementar la información, Dynamics cuenta con las siguientes fases de desarrollo: pre proyecto, viabilidad, fundamentos, exploración e ingeniería, desplazamiento y evaluación de beneficios.

2.1.3.2.5 Feature Driven Development, FDD (Desarrollo basado en funcionalidades)

Ahora bien, con el Feature Driven Development, FDD en sus siglas en inglés, fue introducido por Jeff De Luca en los años 1997. Su filosofía para gestionar el proyecto es a través de repartir todas las funcionalidades en funcionalidades pequeñas, es decir la fragmentación a través de pequeñas funciones priorizadas

por el cliente en el tiempo de dos semanas. Ahora bien, los principios de FDD son dos: El desarrollo del software es una actividad de la persona humana y es un trabajo que es muy valorado por el cliente. En cuanto a los roles, FDD presenta seis roles principales y algunos que son secundarias y si se puede prescindir en la práctica: Project manager, Developer Manager, Chief architect, Programmers, Class Owners y Domain Experts. El ciclo de vida de FDD es interactivo y presenta el modelo de desarrollo integral. Genera actividades a través de una lista de características, los planifica, diseña y crea actividades en base a las características.

2.1.3.2.6 Test Driven Development, TDD (Desarrollo guiado por pruebas)

El desarrollo guiado por pruebas del inglés test first, prima la importancia de generar primero las pruebas que gobernarán todo el desarrollo. Es decir los devs y todo el equipo técnico se centra y focaliza primero en terminar todos los casos de prueba y los casos de pruebas automatizadas, y luego una vez terminada dicha actividad recién se inicia el desarrollo. Es una práctica muy importante que necesita bastante conocimiento técnico y un gobierno general del negocio puesto que para crear dichos casos de prueba se necesita dominar dicho negocio.

Dicha técnica se puede basar en dos niveles muy marcados y los siguientes: Acceptance TDD, dicha actividad requiere de una prueba de aceptación variable que significa la creación de un solo caso de prueba. El TDD ha sido aceptado por varias instituciones capaces de ofrecer resultados confiables y aceptables, la retroalimentación constante como feedback y la disminución del tiempo de depuración.

2.1.3.2.7 Adaptive Software Development, ASD

Surge a partir de la necesidad de generar trabajos rápidos de desarrollo de aplicaciones de Jim Highsmith(Cockburn et al., 2014) y Sam Bayer. Uno de los puntos más importantes del ASD es que generan constante adaptación de procesos con el trabajo que se desarrolla y la variable cantidad de soluciones a los problemas que generan grandes proyectos. Cuentan también con desarrollo iterativo e incremental con mockups continuos y adaptativos.

Indican claramente que un plan no puede generar riesgos ni incertidumbres, debido a que sería muy grave el fallido y la deficiencia. Es guiado a través de metas y características. Una de las primeras fases de este marco es la especulación a diferencia de la planificación, y continúan fases de aprendizaje y colaboración.

2.1.3.2.8 Agile Unified Process, AUP

AUP que significa Proceso Unificado Ágil proviene de la evolución del RUP Rational Unified process de IBM. Inicialmente desarrollado por Scott Ambler, es una mezcla de varias técnicas ágiles comprobadas y validadas por distintos sectores industriales. En su organización muestra TDD programación guiado por pruebas, la modelación ágil con todo lo que incluye desde la implementación y la refactorización con el objetivo de recoger todas las buenas prácticas de desarrollo.

El AUP presenta principios que son acordes al manifiesto ágil entre ellos encontramos la simplicidad, la transparencia, la personalización, organización e independencia y se enfoca principalmente en generar actividades de alto valor, Sus principios se aterrizan en las siguientes fases: Incepción, Elaboración, Construcción y Transición.

2.1.3.2.9 Domain – Driven Design, DDD

DDD significa en sus siglas en inglés Domain Driven Design, DDD es un método de desarrollo ágil que fue diseñado con la finalidad principal de dominar y manejar diseños completos y con la vista evolutiva e incremental que presenta la forma de trabajar ágil. Fue aterrizada como concepto en el 2004 por Eric Evans(Nair & Nair, 2019) y su principal foco es el diseño centrado en la información.

2.2 Marcos Conceptuales o Glosario

2.2.1 Gestión de requerimientos

2.2.1.1 Establecer un plan de gestión de requerimientos

Es claro que para poder iniciar un proyecto se recomienda iniciar desarrollando un plan de gestión de requerimientos o en inglés RMP. El RMP(Nair & Nair, 2019) ejemplifica todo el panorama que se requiere para gestionar los requerimientos de un proyecto.

Dicha información presenta como se inicia. Organiza y modifica los requerimientos durante un ciclo de vida del desarrollo de desarrollo de software.

Algunos puntos importantes a considerar dentro del marco de trabajo:

- ¿Hasta qué punto se deben usar las herramientas de gestión de requerimiento?
- ¿Cuáles son los tipos de requerimientos que serán alineados en el proyecto?
- ¿Existen algunos atributos a considerar dentro de todo el proyecto?

- ¿Cuál es el lugar de alojamiento de los requerimientos una base de datos o algún otro repositorio?
- Se requiere generar trazabilidad, ¿En qué requerimientos necesitamos implementar trazabilidad?
- Documentación necesaria
- ¿Metodología a utilizar ya se encuentra definida?
- ¿Es necesaria documentación completa para que el cliente la completa antes de iniciar con el ciclo de desarrollo?
- Gestión de cambios, ¿Cómo se controlará? herramientas, procesos, metodología
- Feedback constante e implementación de retrospectivas.

Propuestas para la recopilación de requerimientos

La importancia de la captación de los requerimientos al inicio de todo proyecto es de suma importancia, puesto que una mala gestión de captación de requerimientos producirá desgastes a nivel de costos y tiempos. Salir de una reprogramación o planificación es de gravedad para un proyecto durante todo el ciclo de vida del desarrollo de software.

Existen diversas variaciones para captar los requerimientos de los interesados. Algunos comparten dicho grupo unificado al inicio, otros estilos son los de captar los requerimientos a lo largo de la vida del ciclo de desarrollo de software. Con el enfoque ágil (Manresa-Yee & Amengual, 2016) la captación de requerimientos al inicio de todo proyecto es fundamental, pero no determinante puesto que durante el trayecto estos requerimientos pueden cambiar, es más se toma por descontado el cambio de requerimientos puesto que las necesidades del

cliente cambian por cada periodo de trabajo. Entonces en base a esa coyuntura se requiere un marco que soporte dicha gestión de cambios.

Entrevistas

Son técnicas utilizadas por researchers con la finalidad de captar las necesidades del público objetivo, de nuestros clientes. Es muy recomendable utilizar preguntas abiertas, para que el entrevistado brinde todas sus necesidades dolores painpoints. Una correcta entrevista se dará cuando las necesidades del entrevistado, en este caso nuestro cliente muestre a través de journey, sus painpoints, dolores, gustos, preferencias.

Análisis de Documentos

El análisis de documentos, vale decir análisis de resultados trae consigo una gestión de diagnóstico de resultados, puesto que toda la información recopilada a través de entrevistas necesita ser procesada, de tal manera que al final se tenga un resultado clave, y una razón de ser para que se inicie un proyecto.

Tormenta de ideas

La tormenta de ideas es una técnica para divergir, consiste en la participación de todas las personas involucradas, es necesaria la distribución de ideas, no existe idea mala. Al final se hace una evaluación de las ideas, se priorizará en base al trabajo de ordenamiento y con criterios bien definidos.

2.2.2 Metodología Scrum en el entorno de gestión de requerimientos

En la actualidad y en los últimos años varias iniciativas internacionales orientadas específicamente para armar pequeñas y medianas empresas, los procesos y los métodos ágiles han sido identificados. Del mismo modo, diferentes estudios han identificado la correlación entre las metodologías ágiles

y modelos de procesos de desarrollo de software como CMMI-DEV e ISO / IEC 12207, pero los estudios relacionados con la norma ISO / IEC 12207 se basan en la versión de 1995. Por lo tanto, este trabajo se centra en la relación entre las prácticas ágiles, especialmente Scrum, y un subconjunto de proceso de la versión 2008 de la norma ISO / IEC 12207.

SCRUM es uno de los métodos más populares y ágiles es un proceso iterativo incrementales. Estas dos características hacen dividir el proyecto en fases o iteraciones y la entrega incremental del proyecto. Las relaciones indicadas en la obra se obtienen del análisis de trabajos previos y experiencia en consultoría a 25 empresas que cumplen con los resultados estándar de aplicación de metodologías ágiles.

El objetivo principal del estudio es conocer el grado en que las prácticas ágiles ayudan en la realización de las prácticas indicadas en este modelo de proceso.

Las metodologías ágiles han ganado popularidad en el desarrollo de software debido a su capacidad para adaptarse a los cambios y mejorar la eficiencia. La Ilustración 3 ilustra el proceso de gestión de requerimientos en el contexto de metodologías ágiles, destacando cómo estas prácticas permiten una mayor flexibilidad y colaboración continua entre el equipo de desarrollo y los stakeholders.

Ilustración 3: Metodología Scrum



Fuente: Ciclo de trabajo del framework Scrum

2.3 Cuadro Comparativo de Modelos de Aseguramiento de la Calidad

El cuadro comparativo propuesto incluye una variedad de modelos y frameworks ampliamente reconocidos que ofrecen soluciones específicas para la gestión y aseguramiento de la calidad en los procesos de desarrollo de software. La inclusión de estos frameworks responde a la necesidad de abordar diferentes aspectos clave como la escalabilidad, la mejora continua, la optimización de procesos, y la agilidad en la gestión de proyectos. Cada modelo aporta enfoques complementarios que permiten adaptarse a distintos tipos de organizaciones y proyectos, independientemente de su tamaño o sector.

Por ejemplo, ISO 9001:2015 y CMMI se utilizan debido a su enfoque en la mejora continua y su aceptación a nivel global, lo que los convierte en marcos esenciales para organizaciones que buscan estandarización y calidad a largo plazo. Scrum, Kanban, y SAFe, en cambio, están orientados a la gestión ágil, lo que los hace indispensables en entornos donde la flexibilidad y la entrega iterativa son claves para el éxito.

Asimismo, frameworks como UnFIX y EBAS se centran en la adaptabilidad organizacional y la agilidad empresarial, brindando una estructura flexible que permite a las organizaciones responder rápidamente a los cambios. Por último, modelos como Six Sigma y Lean Software Development se centran en la eficiencia y la eliminación de desperdicios, lo que es esencial para mejorar la productividad y reducir defectos en los productos.

Este enfoque integral permite una selección personalizada y efectiva del modelo de calidad más adecuado, garantizando que las necesidades específicas del proyecto o la organización sean cubiertas de manera óptima.

Cuadro Comparativo de Modelos de Aseguramiento de la Calidad entre ISO 9001:2015, CMMI (Capability Maturity Model Integration), PMBOK (Project Management Body of Knowledge), PSP (Personal Software Process), TSP (Team Software Process), Scrum, EBAS (Enterprise Business Agility Strategy), SAFe (Scaled Agile Framework), UnFIX, ISO/IEC 15504 (SPICE), Six Sigma, ITIL (Information Technology Infrastructure Library), Kanban, Lean Software Development.

Tabla 7: Cuadro Comparativo de Modelos de Aseguramiento de la Calidad

Modelo/Framework	Descripción	Enfoque Principal	Fortalezas	Debilidades	Ámbito de Aplicación
ISO 9001:2015	Norma internacional para la gestión de la calidad, aplicable a cualquier sector.	Enfoque en la mejora continua de la calidad y la satisfacción del cliente.	Amplia aceptación global, enfoque en resultados y mejora continua.	General, no enfocado específicamente en desarrollo de software.	Aplicable a cualquier sector o industria, incluyendo software.
CMMI (Capability Maturity Model Integration)	Modelo para la mejora de procesos en cinco niveles de madurez.	Mejora de procesos organizacionales y reducción de riesgos en software.	Mejora continua y reducción de riesgos.	Implementación costosa y compleja.	Empresas grandes y medianas, especialmente con certificaciones.
PMBOK (Project Management Body of Knowledge)	Guía para la gestión de proyectos, incluyendo calidad como una de sus áreas clave.	Gestión y planificación de proyectos con enfoque en calidad.	Amplia aplicabilidad en proyectos de software y TI.	No centrado específicamente en software.	Aplicable a cualquier proyecto en diversas industrias.
PSP (Personal Software Process)	Modelo para mejorar la calidad y productividad individual en el desarrollo de software.	Mejora de la disciplina individual en la calidad del código y planificación.	Aumenta la productividad individual y proporciona estructura.	Aplicable principalmente a desarrolladores individuales.	Desarrolladores de software y pequeñas organizaciones.

TSP (Team Software Process)	Enfoque para mejorar la productividad y calidad en equipos de desarrollo de software.	Mejora de la colaboración y productividad del equipo.	Aumenta la calidad y productividad del equipo.	Requiere entrenamiento especializado.	Organizaciones con equipos de desarrollo medianos o grandes.
Scrum (parte de metodologías ágiles)	Marco ágil para la entrega incremental y la mejora continua en proyectos.	Gestión ágil y entrega iterativa de productos.	Flexibilidad, adaptabilidad y mejora continua.	Puede no ser adecuado para proyectos con requisitos estrictamente definidos al inicio.	Equipos ágiles en desarrollo de software.
EBAS (Enterprise Business Agility Strategy)	Framework de agilidad empresarial que conecta la estrategia con la ejecución ágil.	Transformación ágil a nivel organizacional.	Alinea la estrategia con la agilidad operativa.	Requiere una profunda transformación cultural.	Organizaciones grandes que buscan agilidad en todos los niveles.
SAFe (Scaled Agile Framework)	Framework para escalar prácticas ágiles en grandes organizaciones.	Escalabilidad y gestión ágil a nivel empresarial.	Escala las prácticas ágiles a nivel organizacional.	Más complejo de implementar en comparación con Scrum.	Organizaciones grandes con múltiples equipos ágiles.
UnFIX	Framework para crear organizaciones adaptativas y basadas en equipos flexibles.	Estructuración flexible para adaptarse a cambios y mejoras constantes.	Ofrece mayor adaptabilidad y alineación organizacional.	Requiere un cambio cultural profundo para implementarse de manera efectiva.	Organizaciones que buscan equipos multidisciplinarios y adaptativos.
ISO/IEC 15504 (SPICE)	Estándar para la evaluación y mejora de procesos de software.	Evaluación de procesos de desarrollo de	Flexibilidad y enfoque en la evaluación de procesos.	Complejidad en la implementación y evaluación.	Organizaciones de software que buscan mejorar la madurez de sus procesos.

		software y mejora continua.			
Six Sigma	Metodología centrada en la mejora de procesos mediante la eliminación de defectos y variaciones.	Control de calidad y optimización de procesos.	Reducción significativa de defectos y errores, orientado a la mejora basada en datos.	Implementación costosa y requiere entrenamiento especializado.	Empresas que buscan mejorar la eficiencia y reducir defectos en sus procesos.
ITIL (Information Technology Infrastructure Library)	Conjunto de prácticas para la gestión de servicios de TI, incluyendo aseguramiento de la calidad.	Gestión de servicios y soporte en tecnología de la información.	Mejora continua, reducción de costos operativos.	No está diseñado específicamente para el desarrollo de software.	Organizaciones de TI que gestionan servicios de infraestructura y soporte.
Kanban	Método visual para la gestión de proyectos, centrado en la entrega continua y la optimización del flujo de trabajo.	Flujo continuo y visualización del trabajo en progreso.	Flexibilidad y fácil implementación, promueve la entrega continua.	No está diseñado para proyectos con planificación detallada.	Equipos que buscan optimizar su flujo de trabajo y reducir tiempos de ciclo.
Lean Software Development	Enfoque basado en Lean Manufacturing, optimizando el flujo de valor y eliminando desperdicios.	Maximización del valor y eliminación de actividades que no agregan valor.	Eficiencia y enfoque en la mejora continua.	Difícil de implementar sin un cambio cultural adecuado.	Equipos ágiles y empresas que buscan maximizar la eficiencia.

Fuente. **Comparativo de Modelos de Aseguramiento de la Calidad**

2.1.1 Criterios de Selección del Modelo

- Flexibilidad y adaptabilidad del modelo seleccionado.
- Recursos de la organización para la implementación.
- Tamaño de la organización.
- Impacto en la productividad y calidad de los proyectos.
- Plazos de los resultados esperados.
- Experiencia en el mejoramiento de procesos.
- Ámbito de aplicación.
- Factores clave para asegurar la calidad.
- Facilidad de integración con otros modelos y frameworks.
- Alcance de la evaluación.

Tabla 8: Formulario de Criterios de Selección del Modelo

Criterio	Detalle (Pregunta)	Scrum	Kanban	SAFe	UnFIX	EBAS	CMMI	PMBOK	ITIL
Flexibilidad y adaptabilidad	¿El modelo permite adaptarse a diferentes tipos de proyectos y escalabilidad?	X	X	X	X	X			
Recursos necesarios para la implementación	¿Cuáles son los recursos necesarios (tiempo, personal, presupuesto) para la implementación eficaz del modelo?						X		
Impacto en la calidad	¿Cómo contribuye el modelo a mejorar la calidad del software y reducir errores?						X		
Facilidad de integración	¿Es fácil integrar este modelo con los sistemas o frameworks actuales de la organización?		X	X				X	X
Alcance del modelo	¿El modelo cubre todo el ciclo de vida del software, desde la planificación hasta la entrega?						X		
Plazos para obtener resultados	¿Cuánto tiempo se espera para obtener resultados tangibles	X	X						

	con la implementación de este modelo?								
Compatibilidad con metodologías ágiles o tradicionales	¿El modelo es compatible con metodologías ágiles, tradicionales o híbridas?	X		X		X		X	
Experiencia de la organización con modelos similares	¿La organización ha implementado previamente modelos similares que faciliten la transición?						X		
Tamaño de la organización	¿Este modelo es adecuado para el tamaño de la organización, ya sea pequeña, mediana o grande?			X	X	X	X		
Soporte y recursos disponibles	¿El modelo cuenta con documentación y soporte que faciliten su implementación?						X		X

Luego de revisar los principales frameworks de aseguramiento de calidad analizados, se puede indicar que los modelos que tienen mayor relevancia para el presente trabajo de investigación son Scrum, SAFe, CMMI, y Lean Software Development. Estos frameworks comparten enfoques que destacan la flexibilidad, escalabilidad y mejora continua, elementos clave para la optimización del proceso de desarrollo de software.

Generar un modelo de aseguramiento a partir de Scrum, SAFe, CMMI, y Lean Software Development es viable y beneficioso, ya que sus metodologías se complementan de manera eficaz, permitiendo una integración sencilla y sin solapamientos en los procesos. Además, estos modelos están diseñados específicamente para maximizar la eficiencia y calidad en el desarrollo de software, lo que los hace adecuados para las necesidades de esta investigación.

Capítulo 3: HIPÓTESIS Y VARIABLES

3.1 Hipótesis general

Hg:

La implementación de un modelo de gestión de requerimientos basado en la Norma Técnica Peruana ISO 12207:2008 y en prácticas ágiles contribuirá a reducir el porcentaje de incumplimiento en la fase de planificación del desarrollo de software, disminuyéndolo de un 20% a menos del 10%, en proyectos de empresas del sector tecnológico en Perú.

3.2 Hipótesis específicas

He1:

La implementación del modelo de gestión de requerimientos desarrollado en esta investigación permite reducir el porcentaje de incumplimiento en la planificación del desarrollo de software, de un 20% a menos del 10%.

He2:

La implementación de un modelo de gestión de requerimientos basado en la NTP ISO 12207:2008 y prácticas ágiles reduce el tiempo promedio de replanificación de requerimientos de 10 días a menos de 5 días.

3.3 Identificación de variables

- **Variables Independiente:**

Nivel de cumplimiento de actividades del proceso de gestión de requerimientos (NC-GR)

- **Definición conceptual:** Grado en que se ejecutan correctamente las actividades establecidas en un proceso formal de gestión de requerimientos, conforme a la NTP ISO/IEC 12207:2008.
- **Definición operacional:** Porcentaje de actividades del proceso de requerimientos completadas según cronograma y documentación técnica, evaluado mediante checklists de cumplimiento y evidencia de artefactos generados (actas, historias de usuario, criterios de aceptación, etc.).

- **Variables Dependientes:**

- **Variable Dependiente 1: Desviación del compromiso de planificación (DCP)**

- **Definición conceptual:** Diferencia entre los tiempos y costos planificados inicialmente para un proyecto de desarrollo de software y los tiempos y costos reales al cierre, atribuibles a modificaciones o errores en la definición de requerimientos.
- **Definición operacional:** Se mide en porcentaje (%) o unidades absolutas (horas, soles/dólares) de variación entre el cronograma y presupuesto planificado frente al ejecutado, vinculando los cambios originados por requerimientos revaluados.

- **Indicador:** Desviación temporal (DT%) y Desviación presupuestal (DP%)

- **Fórmula:**

- Desviación temporal (DT%)

$$DT\% = \frac{\text{Tiempo real ejecutado} - \text{Tiempo planificado}}{\text{Tiempo planificado}} \times 100$$

- Desviación presupuestal (DP%)

$$DP\% = \frac{\text{Costo real ejecutado} - \text{Costo planificado}}{\text{Costo planificado}} \times 100$$

- **Instrumento:**

- Ficha de seguimiento de cronograma y presupuesto del proyecto
- Registro de cambios en requerimientos (control de versión o backlog)
- Reportes de avance del área de proyectos o PMO
- Herramientas de gestión como Jira, Trello, Azure DevOps o similares

- **Medición:**

- Escala de razón (tiempo y costo en valores absolutos y porcentuales)
- Rango esperado para DT% y DP%:
 - 0% a 5% -> Baja desviación
 - 6% a 15% -> Desviación media
 - 15% a más -> Alta desviación (crítica)

- **Objetivo esperado:** Reducir la desviación del compromiso de planificación (tiempo y costo) en proyectos de desarrollo de

software a un margen **menor al 10%**, tras la aplicación del modelo de gestión de requerimientos propuesto.

3.4 Matriz de Consistencia

La **Tabla 7** presenta una matriz de consistencia meticulosamente elaborada, que correlaciona los objetivos del estudio con las hipótesis y variables definidas. Esta matriz es un instrumento esencial para asegurar que cada aspecto del diseño de la investigación está alineado y contribuye a los objetivos generales del estudio, facilitando una evaluación coherente y sistemática de los resultados.

PROBLEMA DE INVESTIGACIÓN	OBJETIVOS DE LA INVESTIGACIÓN	HIPÓTESIS	VARIABLES E INDICADORES	METODOLOGÍA
PROBLEMA PRINCIPAL ¿Cómo la aplicación de un modelo de gestión de requerimientos estructurado, alineado a la NTP	OBJETIVO GENERAL Desarrollar e implementar un modelo de gestión de requerimientos, basado en la Norma Técnica Peruana ISO 12207:2008 y en prácticas ágiles, que permita reducir la desviación del compromiso de	HIPÓTESIS GENERAL Hg: La implementación de un modelo de gestión de requerimientos basado en la Norma Técnica Peruana ISO 12207:2008 y en prácticas ágiles contribuirá a reducir el porcentaje de incumplimiento en la fase de planificación del desarrollo	Variables Independientes: Nivel de cumplimiento de actividades del proceso de gestión de requerimientos (NC-GR) Variables Dependientes: Desviación del compromiso de planificación (DCP)	3. Tipo de investigación Aplicativo - Descriptivo. 4. Nivel de Investigación Explicativo 5. Diseño de la investigación Transversal

ISO 12207:2008 y metodologías ágiles, reduce la desviación del compromiso de planificación (tiempo y costo) en proyectos de desarrollo de software en el sector tecnológico peruano?	planificación en proyectos de desarrollo de software en el sector tecnológico, contribuyendo a una mayor eficiencia en la definición, validación y seguimiento de los requerimientos. OBJETIVOS ESPECÍFICOS 1. Optimizar la revisión de requerimientos en los proyectos de desarrollo de software, incrementando el cumplimiento de actividades del proceso de requerimientos	de software, disminuyéndolo de un 20% a menos del 10%, en proyectos de empresas del sector tecnológico en Perú. HIPÓTESIS ESPECIFICAS He1: La implementación del modelo de gestión de requerimientos desarrollado en esta investigación permite reducir el porcentaje de incumplimiento en la		
---	---	---	--	--

	de un 68% a un 90%, mediante la implementación de procedimientos ágiles que reduzcan la necesidad de replanificación. 2. Reducir el tiempo promedio de replanificación de requerimientos de 10 días a menos de 5 días, mediante prácticas ágiles que permitan una mayor capacidad de adaptación al cambio y mejor cumplimiento de los plazos establecidos.	planificación del desarrollo de software, de un 20% a menos del 10%. He2: La implementación de un modelo de gestión de requerimientos basado en la NTP ISO 12207:2008 y prácticas ágiles reduce el tiempo promedio de replanificación de requerimientos de 10 días a menos de 5 días.		
--	--	---	--	--

	3. Mejorar la calidad de los requerimientos desde el inicio del proyecto, incrementando el índice de validación y completitud de requerimientos del 65% al 85%, a través de herramientas de trazabilidad y procesos de validación continua.			
--	--	--	--	--

Tabla 7: Matriz de consistencia

3.5 Matriz de Operación de variables

Variable	Definición Conceptual	Definición Operacional	Hipótesis	Indicador	Instrumento	Medición
Nivel de cumplimiento de actividades del proceso de gestión de requerimientos (NC-GR)	Grado en que se ejecutan correctamente las actividades establecidas en un proceso formal de gestión de requerimientos, conforme a la NTP ISO/IEC 12207:2008.	Porcentaje de actividades del proceso de requerimientos completadas según cronograma y documentación técnica, evaluado mediante checklists y evidencia de artefactos.	La implementación del modelo mejora el cumplimiento del proceso de gestión de requerimientos.	Porcentaje de actividades cumplidas (NC-GR)	Lista de verificación, revisión de artefactos del proceso	Escala porcentual (0%–100%)
Desviación del compromiso de planificación»	Diferencia entre los tiempos y costos planificados inicialmente y los reales al cierre, atribuibles a requerimientos.	Se mide en porcentaje (%) de variación entre cronograma y presupuesto planificado vs. ejecutado.	Una mejor gestión de requerimientos reduce la desviación del compromiso de planificación del 20% al 10%.	DT% (Desviación temporal), DP% (Desviación presupuestal)	Ficha de seguimiento, Jira, reportes PMO	Rango: Baja (0–5%), Media (6–15%), Alta (>15%)

Tabla 8: Matriz de Operación de variables

La **Tabla 8** Estos resultados son analizados para evaluar el impacto de las políticas implementadas y las prácticas de gestión en aquel período, ofreciendo insights valiosos sobre la efectividad de las intervenciones y su impacto en el rendimiento organizacional.

Capítulo 4: ESTADO DEL ARTE

4.1 Integración de Metodologías Ágiles y Tradicionales en la Planificación de Proyectos de Software

La planificación de requisitos en proyectos de desarrollo de software es una actividad crítica que impacta directamente en el éxito del proyecto. Los modelos de gestión híbridos, que combinan metodologías ágiles y tradicionales, han ganado popularidad en la industria del software debido a su capacidad para adaptarse a entornos cambiantes y mejorar la eficiencia en la gestión de requisitos.

DETALLE:

Objetivos: Este estudio tiene como objetivo explorar la integración de metodologías ágiles con enfoques tradicionales en la planificación de proyectos de software, con el fin de mejorar la gestión de requisitos y la reducción de riesgos asociados a cambios en los proyectos.

Metodología: Se utilizó un enfoque mixto combinando análisis cuantitativo y cualitativo. Se realizaron encuestas a 100 gestores de proyectos y entrevistas en profundidad con 20 expertos en metodologías ágiles y tradicionales. Los datos se analizaron utilizando análisis temático y técnicas estadísticas descriptivas.

Resultados: El estudio encontró que la integración de enfoques ágiles y tradicionales permite una mayor flexibilidad en la planificación, lo que reduce los errores en la definición de requisitos y mejora la capacidad de respuesta a los cambios. Los proyectos que utilizaron modelos híbridos reportaron un 25%

menos de incidencias relacionadas con requisitos no cumplidos en comparación con aquellos que utilizaron únicamente metodologías tradicionales.

- Beneficios y Desafíos de la Implementación de Modelos Híbridos en la Gestión de Requerimientos

Varios estudios han destacado la efectividad de los modelos de gestión híbridos en la planificación de requisitos. Según Rodríguez y Martínez (2019), los modelos híbridos combinan la flexibilidad y la rapidez de las metodologías ágiles con la estructura y la disciplina de los enfoques tradicionales. Este enfoque permite una mejor adaptación a los cambios en los requisitos del cliente y una planificación más efectiva.

En su estudio, Rodríguez y Martínez describen un modelo híbrido que integra Scrum con elementos del Modelo en Cascada. Este modelo permite iteraciones rápidas y frecuentes para incorporar cambios en los requisitos, al mismo tiempo que mantiene una planificación estructurada y detallada en cada fase del desarrollo. Los resultados muestran una mejora significativa en la entrega de proyectos dentro del plazo y con alta calidad.

DETALLE:

Título: *Integración de Metodologías Ágiles y Tradicionales en la Planificación de Proyectos de Software* (2020)

Objetivos: Este estudio tiene como objetivo explorar la integración de metodologías ágiles con enfoques tradicionales en la planificación de proyectos de software, con el fin de mejorar la gestión de requisitos y la reducción de riesgos asociados a cambios en los proyectos.

Metodología: Se utilizó un enfoque mixto combinando análisis cuantitativo y cualitativo. Se realizaron encuestas a 100 gestores de proyectos y entrevistas en profundidad con 20 expertos en metodologías ágiles y tradicionales. Los datos se analizaron utilizando análisis temático y técnicas estadísticas descriptivas.

Resultados: El estudio encontró que la integración de enfoques ágiles y tradicionales permite una mayor flexibilidad en la planificación, lo que reduce los errores en la definición de requisitos y mejora la capacidad de respuesta a los cambios. Los proyectos que utilizaron modelos híbridos reportaron un 25% menos de incidencias relacionadas con requisitos no cumplidos en comparación con aquellos que utilizaron únicamente metodologías tradicionales.

Beneficios y Desafíos de la Implementación de Modelos Híbridos en la Gestión de Requerimientos

Los modelos de gestión híbridos ofrecen varios beneficios en la planificación de requisitos. Gómez y Sánchez (2020) encontraron que la combinación de metodologías permite una mejor comunicación y colaboración entre los equipos de desarrollo y los stakeholders. Esto se traduce en una mayor claridad y precisión en la captura de requisitos, así como en una validación más efectiva de los mismos.

Además, los modelos híbridos facilitan la gestión de riesgos y la toma de decisiones. Según Pérez (2018), la planificación iterativa y la retroalimentación constante permiten identificar y mitigar riesgos de manera temprana. Esto es especialmente útil en proyectos complejos y dinámicos donde los requisitos pueden cambiar con frecuencia.

DETALLE:

Objetivos: El objetivo de este artículo es evaluar los beneficios y desafíos asociados con la implementación de modelos híbridos en la gestión de requerimientos en proyectos de desarrollo de software.

Metodología: Se realizó un estudio de caso múltiple en tres empresas de desarrollo de software, implementando modelos híbridos en sus proyectos. Se recopiló información mediante entrevistas semiestructuradas con líderes de proyecto y se analizaron los documentos del proyecto para evaluar la efectividad del modelo.

Resultados: Los resultados muestran que los modelos híbridos permiten una mejor adaptación a cambios de última hora en los requisitos, pero presentan desafíos en la coordinación entre equipos que utilizan enfoques metodológicos diferentes. A pesar de estos desafíos, las empresas reportaron una mejora del 30% en la satisfacción del cliente debido a una mayor alineación entre los entregables y las expectativas iniciales.

Estudios de Caso sobre la Aplicación Práctica de Modelos Híbridos en el Desarrollo de Software

A pesar de sus beneficios, la implementación de modelos de gestión híbridos también presenta desafíos. López (2020) señala que la integración de metodologías ágiles y tradicionales requiere un cambio cultural y organizacional. Los equipos deben estar capacitados en ambos enfoques y ser capaces de adaptarse rápidamente a nuevas formas de trabajo.

Otro desafío es la necesidad de herramientas y procesos que soporten ambos enfoques. Fernández y Torres (2022) argumentan que es crucial contar con herramientas de gestión de proyectos que faciliten la colaboración y la

integración de prácticas ágiles y tradicionales. Además, es necesario establecer procesos claros para la planificación y la gestión de requisitos, asegurando que todos los miembros del equipo comprendan y sigan el modelo híbrido.

DETALLE:

Objetivos: Este estudio busca documentar y analizar la aplicación práctica de modelos híbridos en proyectos de desarrollo de software, enfocándose en cómo estos modelos influyen en la planificación y ejecución de proyectos complejos.

Metodología: Se utilizaron estudios de caso en cinco empresas que han adoptado modelos híbridos. Se aplicaron cuestionarios a los miembros del equipo y se revisaron los informes de progreso del proyecto para obtener una comprensión completa del impacto de estos modelos.

Resultados: El análisis reveló que los modelos híbridos son especialmente eficaces en proyectos donde los requisitos cambian con frecuencia. Los proyectos estudiados mostraron una reducción significativa en los tiempos de replanificación y una mejora en la calidad del software entregado, con un 40% menos de defectos reportados durante las pruebas finales.

Estudios de Caso y Aplicaciones Prácticas

Varios estudios de caso han demostrado la efectividad de los modelos de gestión híbridos en la planificación de requisitos. En un estudio realizado por Agger y Serrano (2019), se implementó un modelo híbrido en una empresa de desarrollo de software. Los resultados mostraron una mejora del 25% en la eficiencia del equipo y una reducción del 15% en el tiempo de entrega de proyectos.

Otro ejemplo es el estudio de López (2020), donde se aplicó un modelo híbrido en un proyecto de desarrollo de software en el sector financiero. El modelo permitió una mayor adaptabilidad a los cambios regulatorios y de mercado, mejorando la calidad del producto final y la satisfacción del cliente.

4.2 Modelos de Gestión de Requerimientos

Varios estudios han destacado la importancia de contar con un modelo de gestión de requerimientos bien definido para mejorar la planificación y ejecución de proyectos de software. Según Pérez (2018), un modelo estructurado permite una mejor identificación, validación y seguimiento de los requerimientos, lo cual es crucial para minimizar los incumplimientos. Por ejemplo, Pérez describe un modelo basado en la integración de técnicas ágiles con prácticas tradicionales de ingeniería de requerimientos, destacando cómo esta combinación mejora la flexibilidad y la precisión en la captura de requerimientos.

Rodríguez y Martínez (2019) propusieron un modelo híbrido que combina metodologías ágiles y tradicionales, mostrando una mejora significativa en la adaptabilidad y respuesta a cambios en los requerimientos. Su estudio mostró que la integración de métodos ágiles en fases críticas del proyecto permite una respuesta más rápida a los cambios en los requerimientos del cliente, lo que reduce el tiempo de entrega y mejora la calidad del producto final.

La validación de requerimientos es un componente crítico en la gestión de proyectos de software. En un estudio comparativo, Gómez y Sánchez (2020) evaluaron diferentes técnicas de validación y su efectividad en proyectos de gran

escala. Los resultados sugieren que el uso de técnicas iterativas y continuas, como las revisiones por pares y las pruebas de aceptación temprana, contribuyen a una reducción notable en los errores y omisiones de requerimientos.

Un ejemplo destacado en su estudio es la técnica de revisión por pares, donde los requerimientos son revisados por varios miembros del equipo y stakeholders para asegurar que todos los aspectos sean considerados. Esta técnica no solo mejora la precisión de los requerimientos, sino que también facilita una mejor comunicación y entendimiento entre todas las partes involucradas.

4.3 Impacto en el Rendimiento del Proyecto

El impacto de una gestión eficiente de los requerimientos en el rendimiento global del proyecto también ha sido ampliamente documentado. López (2021) analizó proyectos de software en diferentes industrias y encontró que aquellos con una gestión de requerimientos robusta experimentaron menores tasas de retrabajo y mayores niveles de satisfacción del cliente. Este estudio subraya la necesidad de integrar la gestión de requerimientos como una práctica estándar en la planificación de proyectos.

López destacó que los proyectos con una buena gestión de requerimientos tenían un 30% menos de retrabajo y un 20% más de cumplimiento en los plazos establecidos en comparación con aquellos que no tenían una gestión eficiente. Estos resultados muestran cómo una buena práctica en la gestión de requerimientos puede llevar a una mejor utilización de recursos y mayor satisfacción del cliente, lo que es crucial para el éxito a largo plazo de los proyectos de software.

4.4 Desafíos y Áreas de Mejora

A pesar de los avances, existen desafíos persistentes en la gestión de requerimientos. Entre estos, la comunicación efectiva entre los stakeholders y el equipo de desarrollo es crucial. Según Fernández y Torres (2022), las herramientas colaborativas y la capacitación continua en técnicas de comunicación son esenciales para abordar estos desafíos.

Fernández y Torres argumentan que la falta de comunicación clara y efectiva puede llevar a malentendidos y errores en la interpretación de los requerimientos, lo que a su vez afecta el desarrollo del proyecto. Recomiendan la adopción de herramientas de gestión de proyectos que faciliten la colaboración en tiempo real y la implementación de talleres de capacitación para mejorar las habilidades de comunicación del equipo.

Capítulo 5: METODOLOGÍA

5.1 Metodología de la investigación

5.1.1 Tipo y diseño de la investigación

La presente investigación se desarrolla en base a una investigación aplicada – descriptiva.

Para validar las hipótesis planteadas, se aplica un diseño de investigación cuasi-experimental, en el cual se recolectan datos de proyectos anteriores al modelo (línea base) y se comparan con los resultados obtenidos después de la implementación del modelo propuesto. Los indicadores principales a evaluar serán el porcentaje de incumplimiento de requerimientos y el tiempo promedio de replanificación de los mismos.

- **Indicador 1: Porcentaje de Incumplimiento:**

- Se calcula el porcentaje de requerimientos incumplidos antes de la implementación del modelo y después. Se espera una reducción significativa en este porcentaje.
- Fórmula: $(\text{Requerimientos incumplidos} / \text{Total de requerimientos}) \times 100\%$.

- **Indicador 2: Tiempo de Replanificación:**

- Se mide el tiempo promedio que toma replanificar los requerimientos antes y después de la implementación del modelo. Se espera una disminución de los tiempos de replanificación.

5.1.2 Unidad de análisis

La unidad de análisis está conformada por los proyectos de tecnologías de información gestionados por las Unidades de Tecnologías de Información de Efect y ONP, los cuales han sido seleccionados debido a su relevancia en la implementación de modelos de gestión de requerimientos y su alineación con la Norma Técnica Peruana ISO 12207:2008.

5.1.3 Población de estudio

La población de estudio está compuesta por proyectos de tecnologías de información desarrollados en entidades del sector público y privado en Perú, con énfasis en aquellas organizaciones que aplican metodologías ágiles y tradicionales en la gestión de requerimientos.

5.1.4 Técnicas de recolección de datos

- Entrevistas (IBM, ONP, RENIEC, Entidades del estado)
- Encuestas de campo realizadas
- Recolección de Papers certificados.
- Estadísticas de INEI en cuanto al sector público al 2023.

5.2 Metodología para el planteamiento del modelo

5.2.1 Procedimiento para la obtención del modelo

5.2.1.1 Revisión de la literatura

La revisión de la literatura es una etapa crucial que implica la recopilación y análisis de investigaciones previas sobre modelos de gestión de requerimientos.

Este proceso incluye:

- **Búsqueda Sistemática:** Utilización de bases de datos académicas como IEEE Xplore, ACM Digital Library, y Google Scholar para identificar estudios relevantes publicados en los últimos 10 años.
- **Criterios de Inclusión y Exclusión:** Definición de criterios para seleccionar los estudios más relevantes, excluyendo aquellos que no aporten información significativa sobre la gestión de requerimientos o que sean muy específicos de un contexto no aplicable.

- **Análisis Crítico:** Evaluación crítica de los estudios seleccionados para identificar las mejores prácticas, las deficiencias en los enfoques actuales y las oportunidades de mejora en la gestión de requerimientos.

5.2.1.2 Análisis del caso

El análisis de casos de estudio se realiza para entender cómo se gestionan los requerimientos en contextos reales y qué desafíos enfrentan. Este proceso incluye:

- **Selección de Casos:** Identificación de proyectos de software en diferentes industrias que han implementado diversas metodologías de gestión de requerimientos.
- **Recopilación de Datos:** Obtención de información detallada sobre la gestión de requerimientos en estos proyectos a través de documentos, reportes de proyectos, y entrevistas con miembros clave del equipo.
- **Análisis Comparativo:** Comparación de los casos para identificar patrones comunes, prácticas efectivas y desafíos recurrentes en la gestión de requerimientos.

5.2.1.3 Desarrollo del modelo

El desarrollo del modelo se basa en la integración de los hallazgos de la revisión de la literatura, el análisis de casos y las entrevistas con expertos. Este proceso incluye:

- **Definición de Requisitos del Modelo:** Basándose en las necesidades y desafíos identificados, se definen los requisitos funcionales y no funcionales del modelo de gestión de requerimientos.
- **Diseño Inicial del Modelo:** Creación de un diseño preliminar del modelo que incorpora las mejores prácticas y aborda las deficiencias identificadas.
- **Iteración y Refinamiento:** Mejora del modelo a través de un proceso iterativo que incluye revisiones y ajustes basados en la retroalimentación de expertos y pruebas piloto.

5.2.1.4 Validación del modelo

La validación del modelo desarrollado se realiza en un entorno controlado para evaluar su efectividad. Este proceso incluye:

- **Diseño de Pruebas:** Desarrollo de un plan de validación que incluye la definición de métricas específicas para medir la reducción del porcentaje de incumplimiento y la mejora en la planificación.
- **Implementación en Entorno Controlado:** Aplicación del modelo en proyectos de software seleccionados, utilizando herramientas de gestión de proyectos y técnicas de seguimiento.
- **Recopilación de Datos:** Monitoreo y registro de datos sobre el rendimiento del modelo, incluyendo el porcentaje de cumplimiento de los requerimientos y el rendimiento del proyecto.
- **Análisis de Resultados:** Evaluación de los datos recopilados para determinar la efectividad del modelo en mejorar la gestión de requerimientos y reducir el incumplimiento.

5.2.1.5 Herramientas y técnicas utilizadas

Para llevar a cabo la investigación y el desarrollo del modelo, se utilizaron varias herramientas y técnicas:

- **Análisis de Contenido:** Para la revisión de la literatura y el análisis de entrevistas.
- **Software de Gestión de Proyectos:** Herramientas como JIRA, Trello o Microsoft Project para implementar y validar el modelo en un entorno controlado.
- **Métodos Estadísticos:** Utilización de software estadístico como SPSS o R para analizar los datos y validar la eficacia del modelo.
- **Técnicas de Visualización:** Creación de diagramas de flujo, mapas de procesos y otros esquemas visuales para representar el diseño y funcionamiento del modelo.

5.2.1.6 Proceso de iteración y mejora

El modelo se sometió a un proceso iterativo de desarrollo y mejora, basado en la retroalimentación recibida durante las pruebas de validación. Este proceso incluye:

- **Revisiones Periódicas:** Reuniones regulares con el equipo de desarrollo y los stakeholders para revisar el progreso y discutir posibles mejoras.
- **Pruebas Piloto:** Implementación del modelo en proyectos piloto para evaluar su efectividad en condiciones reales y obtener retroalimentación práctica.

- **Ajustes y Mejoras:** Realización de ajustes en el modelo basados en los resultados de las pruebas piloto y las revisiones periódicas, asegurando que el modelo sea adaptable y efectivo en diferentes contextos.

5.2.2 Validación por Expertos del Instrumento (Encuesta)

El uso del **Coefficiente de Validez de Contenido (CVC)** es un procedimiento esencial para asegurar que el instrumento de medición (encuesta) es válido, es decir, que mide con precisión las variables que pretende evaluar. La validez de contenido es un tipo de validez que se refiere a la representatividad de los ítems que componen un instrumento, asegurando que cubren el constructo que se desea medir (Haynes et al., 1995). En este sentido, el CVC permite cuantificar el grado de acuerdo entre los expertos sobre la relevancia, claridad conceptual, redacción y dificultad de cada ítem del cuestionario.

La validez de contenido se establece a través de **un juicio de expertos**, quienes evaluaron la encuesta con el objetivo de determinar la pertinencia y claridad de cada pregunta en relación con los objetivos de la investigación. Este proceso asegura que el instrumento sea adecuado para medir constructos complejos, como la gestión de requerimientos en proyectos de software.

5.2.2.1 Proceso de Evaluación

En este estudio, se utilizó un panel de **cinco expertos** en gestión de proyectos de software, metodologías ágiles y desarrollo de software. Los expertos evaluaron los ítems de la encuesta en función de los siguientes criterios:

- **Pertinencia:** Evalúa si el ítem está alineado con los objetivos de la investigación.
- **Claridad Conceptual:** Mide la claridad del ítem para los encuestados.
- **Redacción y Terminología:** Evalúa la precisión y calidad del lenguaje empleado.
- **Niveles de Dificultad:** Determina si los ítems son fáciles de entender y responder.

Cada experto calificó los ítems en una escala del 1 al 5, siendo 1 "inaceptable" y 5 "excelente". La fórmula para calcular el CVC por ítem (I-CVI) es la proporción de expertos que calificaron un ítem con una puntuación de 3 o más. Un I-CVI mayor o igual a **0.78** es considerado aceptable (Lynn, 1986).

Artefacto relacionado		Criterio	Inaceptabl e	Deficien te	Regul ar	Buen o	Excelen te	
	1.- Edad de los encuestados		1	2	3	4	5	

DATOS DEMOGRAFICOS	La encuesta recoge la distribución de los participantes según su edad, lo cual permite analizar la diversidad generacional dentro de los equipos de desarrollo de software. Esta información es útil para identificar posibles diferencias en la percepción y gestión de los requerimientos según el rango de edad.	Pertinencia					x	
--------------------------------------	--	--------------------	--	--	--	--	----------	--

		Claridad Conceptual					x	
		Redacción y Terminología					x	
		Niveles de Dificultad				x		19
	2.- Género de los encuestados		1	2	3	4	5	

DATOS DEMOGRAFICOS	El género de los participantes es registrado para tener una visión sobre la representación de hombres, mujeres y personas que prefieren no especificar su género en los equipos de trabajo relacionados con el desarrollo de software. Esto puede proporcionar un contexto sobre la diversidad y posibles sesgos en la gestión de proyectos.	Pertinencia					X		
-------------------------------	--	-------------	--	--	--	--	---	--	--

		Claridad Conceptual				X		
		Redacción y Terminología					x	
		Niveles de Dificultad				x		17
ARTEFACTO : Cuestionario desarrollo de requisitos:	3.- Puesto de trabajo		1	2	3	4	5	

	Los encuestados ocupan diferentes roles dentro de las organizaciones de desarrollo de software, como desarrolladores, analistas de negocios, gerentes de proyectos, testers de software, u otros cargos. Conocer el rol específico ayuda a contextualizar las respuestas sobre la gestión de requerimientos según el nivel de responsabilidad en el ciclo de vida del proyecto.	Pertinencia					x	
Pregunta 3: "¿Qué rol desempeñan los encuestados en el proyecto?"		Claridad Conceptual				X		
		Redacción y				X		

		Terminología						
Relaciona cómo el cuestionario se enfoca en captar información de roles clave.		Niveles de Dificultad				X		17
	4.- Años de experiencia en desarrollo de software		1	2	3	4	5	

	Se recoge información sobre los años de experiencia de los encuestados en la industria del desarrollo de software. Esto permite analizar cómo la experiencia influye en la percepción de la gestión de los requerimientos y en los desafíos enfrentados durante el proceso de desarrollo.	Pertinencia				X		
		Claridad Conceptual				X		
		Redacción y				X		

		Terminología						
		Niveles de Dificultad			X			15
	5.- Uso de un modelo estructurado para la gestión de requerimientos	Criterio	1	2	3	4	5	

	Este enunciado refleja si las organizaciones donde trabajan los encuestados utilizan o no un modelo formal y estructurado para gestionar los requerimientos del software. Esta información es clave para identificar si existe una relación entre el uso de modelos y la eficacia percibida en la gestión de los requerimientos.	Pertinencia				X		
--	---	-------------	--	--	--	---	--	--

		Claridad Conceptual				X		
		Redacción y Terminología					x	
		Niveles de Dificultad					x	18
ARTEFACTO: Seguimiento de pendientes:	6.- Metodologías de gestión de proyectos utilizadas en las organizaciones:		1	2	3	4	5	
	Se recopila información sobre las metodologías de gestión de proyectos que utilizan las organizaciones, como Agile (Scrum, Kanban), Waterfall, o enfoques híbridos. Este dato permite entender las preferencias metodológicas y su relación con la gestión de los requerimientos en los distintos entornos organizacionales.	Pertinencia					X	

¿Qué metodologías de gestión de proyectos utilizan en la organización?		Claridad Conceptual					X	
		Redacción y Terminología					X	
Vincula el seguimiento de pendientes con el enfoque metodológico.		Niveles de Dificultad					X	20

ARTEFACTO: Programación de reuniones del mes:	7.- Frecuencia de revisión y validación de los requerimientos		1	2	3	4	5	
	Se mide con qué frecuencia los equipos de trabajo revisan y validan los requerimientos de los proyectos, lo cual puede variar desde una revisión semanal hasta no hacerla nunca. La frecuencia de revisión es un factor crítico para determinar la efectividad en la adaptación a cambios y en la precisión de los requerimientos.	Pertinencia				X		

Pregunta 7: "¿Con qué frecuencia se revisan y validan los requerimientos?"		Claridad Conceptual				X		
		Redacción y Terminología				x		
Las reuniones mensuales aseguran revisiones periódicas.		Niveles de Dificultad					x	17

ARTEFACTO: Checklist aseguramient o documento de alcance:	8.- Claridad y definición de los requerimientos desde el inicio del proyecto		1	2	3	4	5	
Pregunta 8: "¿Qué tan claros y definidos están los requerimientos desde el inicio?"								

Relaciona cómo el checklist asegura claridad y definición.								
ARTEFACTO: Checklist aseguramient o documento de diseño:								

	El enunciado refleja la percepción de los encuestados sobre la claridad y nivel de detalle de los requerimientos desde las primeras etapas del proyecto. Esta información es relevante para evaluar si los problemas en la gestión de requerimientos comienzan en la fase de planificación.	Pertinencia				X		
Pregunta 8: "¿Qué tan claros y definidos están los requerimientos desde el inicio?"		Claridad Conceptual				X		
		Redacción y Terminología				X		

El checklist asegura que los diseños cumplen con los requerimientos.		Niveles de Dificultad				X		16
ARTEFACTO: Seguimiento y cierre de hallazgos - Aceptación documento de alcance:	9.- Eficacia del proceso actual de gestión de requerimientos		1	2	3	4	5	

Pregunta 9: "¿Qué tan eficaz es el proceso actual de gestión de requerimientos?"								
El cierre de hallazgos contribuye a la eficacia del proceso.								

ARTEFACTO: Seguimiento y cierre de hallazgos - Aceptación documento de diseño:	Los encuestados califican la eficacia del proceso de gestión de requerimientos en sus organizaciones en una escala de 1 (muy ineficaz) a 5 (muy eficaz). Este indicador ayuda a identificar la satisfacción general con el manejo de los requerimientos y a detectar áreas de mejora	Pertinencia				X		
		Claridad Conceptual				X		
Pregunta 9: "¿Qué tan eficaz es el proceso actual de gestión de requerimientos?"		Redacción y Terminología				X		

Similar al alcance, el seguimiento asegura la aceptación del diseño.		Niveles de Dificultad			X			15
ARTEFACTO: Reporte hallazgos aseguramient o documento de alcance:	10.- Principales desafíos en la gestión de requerimientos	Criterio	1	2	3	4	5	

Pregunta 10: "¿Cuáles son los principales desafíos en la gestión de requerimientos?"								
Relaciona cómo el reporte aborda desafíos identificados.								

ARTEFACTO: Reporte de hallazgos aseguramiento o documento de diseño:	Se identifican los principales obstáculos que enfrentan los equipos de trabajo durante la gestión de los requerimientos, como la ambigüedad en los requerimientos, los cambios frecuentes, la falta de comunicación entre los stakeholders y la falta de herramientas adecuadas. Esto permite conocer los problemas más comunes que afectan el desarrollo del proyecto.	Pertinencia				X		
		Claridad Conceptual				X		

Pregunta 10: "¿Cuáles son los principales desafíos en la gestión de requerimientos?"		Redacción y Terminología			X			
Similar al alcance, el reporte aborda desafíos encontrados en diseño.		Niveles de Dificultad			X			14
ARTEFACTO: Generación de cronograma:	11.- Tiempo adicional necesario para replanificar debido a cambios en los requerimientos		1	2	3	4	5	

Pregunta 11: "¿Cuánto tiempo adicional toma replanificar debido a cambios en los requerimientos?"	Se mide cuánto tiempo adicional toma replanificar los requerimientos cuando estos cambian durante el ciclo de desarrollo del proyecto. Este dato es crucial para entender los retrasos y cuellos de botella que pueden generarse debido a cambios en los requerimientos.	Pertinencia			X		
		Claridad Conceptual			X		

Relaciona la planificación y replanificación del cronograma con los tiempos necesarios.		Redacción y Terminología				X		
		Niveles de Dificultad				X		16
ARTEFACTO: Seguimiento de pendientes de diseño de solución:	12.- Herramientas utilizadas para la gestión y el seguimiento de pendientes de requerimientos		1	2	3	4	5	
		Pertinencia				X		

Pregunta 12: "¿Qué herramientas utilizan para la gestión de requerimientos?"	"¿Cómo aseguran que las herramientas empleadas permiten un seguimiento efectivo de los pendientes relacionados con el diseño de la solución?"	Claridad Conceptual				X		
		Redacción y Terminología				X		
Relaciona el seguimiento con las herramientas utilizadas.		Niveles de Dificultad			X			15
	13.- Sugerencias para mejorar el proceso de gestión de requerimientos	Criterio	1	2	3	4	5	

	Los encuestados tienen la oportunidad de sugerir mejoras en el proceso de gestión de requerimientos de sus organizaciones. Estas sugerencias son valiosas para identificar soluciones prácticas y mejorar el desempeño en la gestión de los requerimientos de software.	Pertinencia				X		
		Claridad Conceptual				X		
		Redacción y Terminología			X			

		Niveles de Dificultad			X			14
--	--	-----------------------	--	--	---	--	--	----

Evalúe su grado de acuerdo con las siguientes afirmaciones sobre el proceso de desarrollo de requisitos en su proyecto

Pregunta	Inaceptable	Deficiente	Regular	Bueno	Excelente
	1	2	3	4	5
REQ1: Las reuniones de definición de requisitos fueron programadas y ejecutadas de manera oportuna.					
REQ2: Se utilizó un cuestionario adecuado para recopilar los requisitos del proyecto.					
REQ3: Se empleó un checklist para asegurar la documentación y el alcance de los requisitos.					
REQ4: Los hallazgos documentados durante la fase de requisitos fueron gestionados adecuadamente.					
REQ5: Hubo un seguimiento eficaz de los pendientes identificados durante el desarrollo de los requisitos.					

Evalúe su grado de acuerdo con las siguientes afirmaciones sobre el diseño de la solución en su proyecto.

Pregunta	Inaceptable	Deficiente	Regular	Bueno	Excelente
	1	2	3	4	5
DSC1: Se llevó un seguimiento adecuado de los pendientes durante el diseño de la solución.					
DSC2: Se utilizó un checklist para asegurar la documentación del análisis del diseño.					
DSC3: Los hallazgos durante la fase de diseño fueron reportados de manera adecuada.					
DSC4: Se realizó un seguimiento y cierre eficaz de los hallazgos del diseño.					

Evalúe su grado de acuerdo con las siguientes afirmaciones relacionadas con el cumplimiento de plazos en el proyecto

Pregunta	Inaceptable	Deficiente	Regular	Bueno	Excelente
	1	2	3	4	5
INC1: El proyecto se entregó dentro del plazo originalmente planificado					
INC2: Las desviaciones respecto al cronograma fueron mínimas					
INC3: Las entregas parciales durante el proyecto cumplieron con los plazos acordados					

A continuación, se dan a conocer los análisis de la información de campo e incluyen las siguientes estadísticas descriptivas: respuestas del trabajo de campo, características de los participantes y exploración de los datos

5.2.2.2 Respuestas del trabajo de campo

Las entrevistas se realizaron enfocándose en los usuarios que cumplieran con ciertas características predefinidas en el proyecto, tales como su participación en un proceso determinado, su perfil profesional y la adquisición de servicios clave dentro de la organización. Para llevarlas a cabo de manera ordenada, se contó con el apoyo del equipo de administración, quienes proporcionaban una lista de personas que cumplieran con los criterios de selección (pertenencia a un segmento específico y uso de ciertas herramientas o servicios).

Una vez identificados los entrevistados, se les invitaba cordialmente a participar en el estudio durante un espacio de tiempo definido (por ejemplo, al final de su sesión de trabajo o en un momento acordado por ambas partes). Si el entrevistado aceptaba, el investigador y el participante se reunían en un área cómoda y tranquila para desarrollar la entrevista. Cabe mencionar que, para asegurar la adecuada conducción de las entrevistas, se capacitó al personal a cargo en técnicas de recolección de información y en la aclaración de dudas que pudiesen surgir durante la conversación.

De esta manera, se garantizó que cada entrevista fuese realizada con la debida atención y confidencialidad, permitiendo obtener información confiable y relevante para la presente investigación.

	Cantidad	Porcentaje
Válidos	350	100%
Inválidos	0	0
Total	350	100%

A continuación visualizamos los resultados de la encuesta aplicada a personas de la ONP y EFACT.

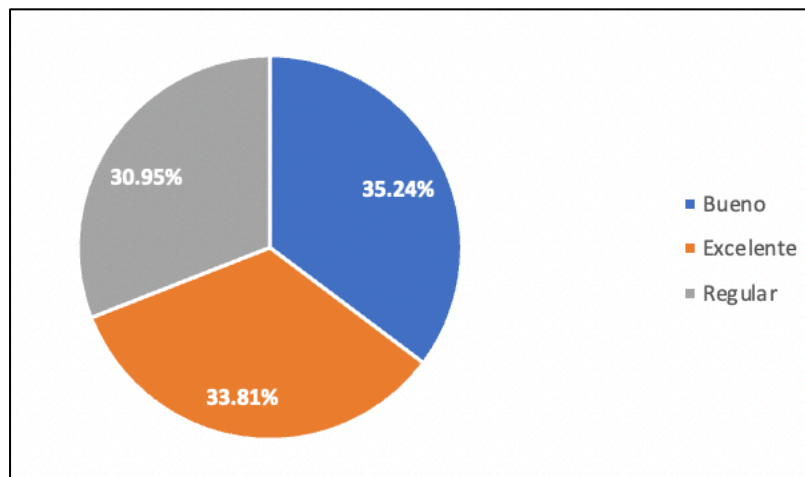


Figura 4. 1 ¿ REQ1: Las reuniones de definición de requisitos fueron programadas y ejecutadas de manera oportuna? (Realización propia)

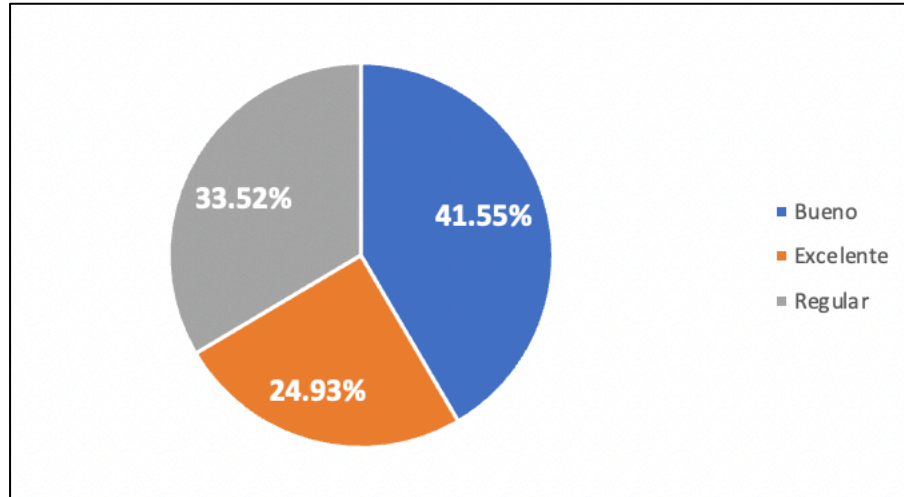


Figura 4 ¿ REQ2: Se utilizó un cuestionario adecuado para recopilar los requisitos del proyecto? (Realización propia)

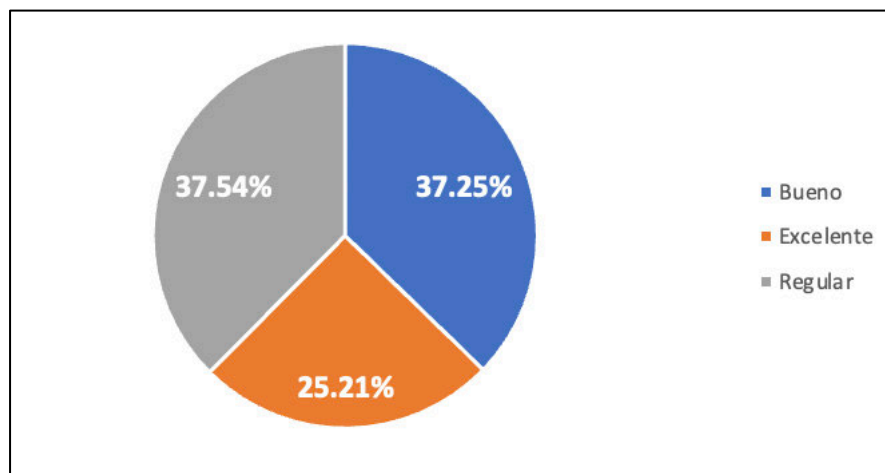


Figura 4 ¿ REQ3: Se empleó un checklist para asegurar la documentación y el alcance de los requisitos? (Realización propia)

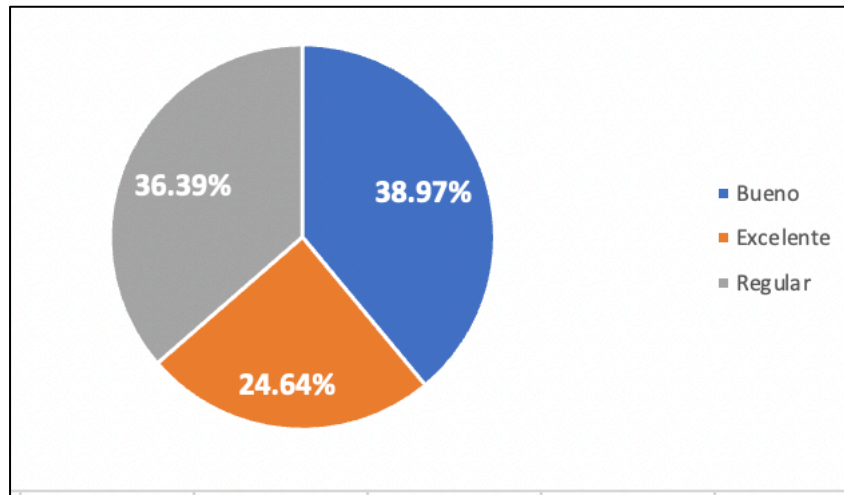


Figura 4 ¿ REQ4: Los hallazgos documentados durante la fase de requisitos fueron gestionados adecuadamente? (Realización propia)

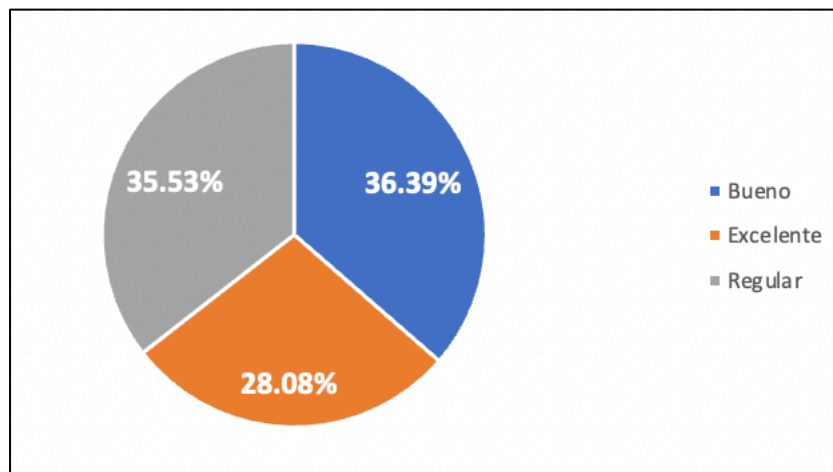


Figura 4 ¿ REQ5: Hubo un seguimiento eficaz de los pendientes identificados durante el desarrollo de los requisitos? (Realización propia)

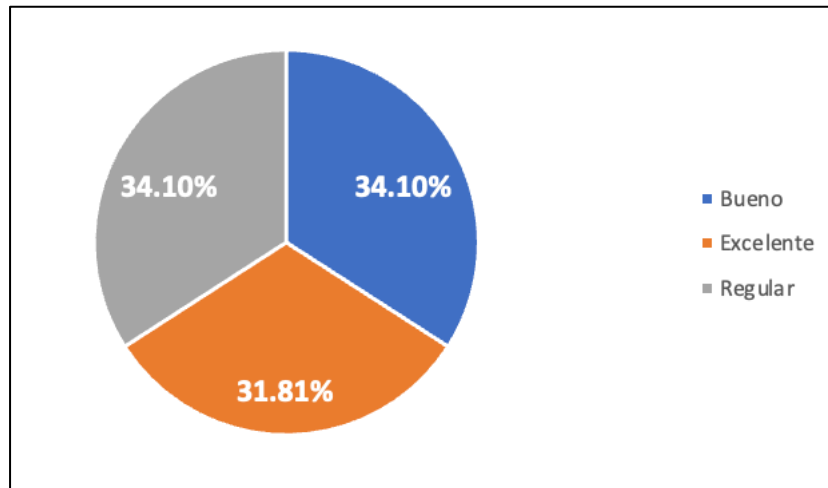


Figura 4 ¿ DSC1: Se llevó un seguimiento adecuado de los pendientes durante el diseño de la solución? (Realización propia)

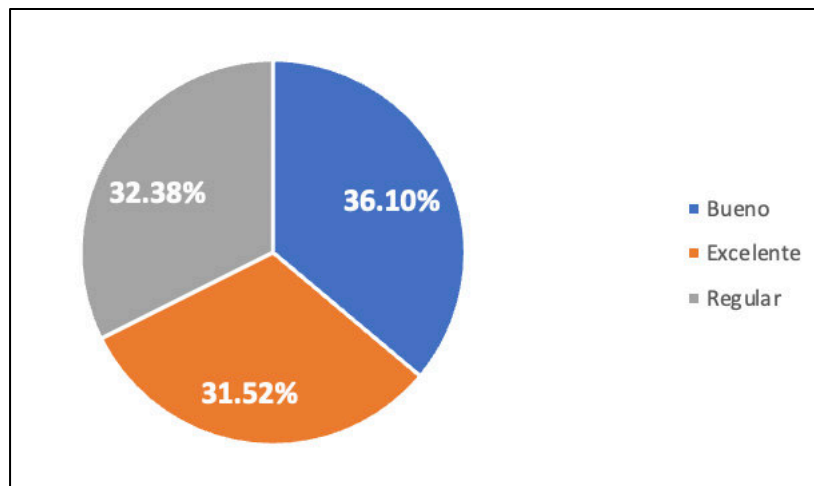


Figura 4 ¿ DSC2: Se utilizó un checklist para asegurar la documentación del análisis del diseño? (Realización propia)

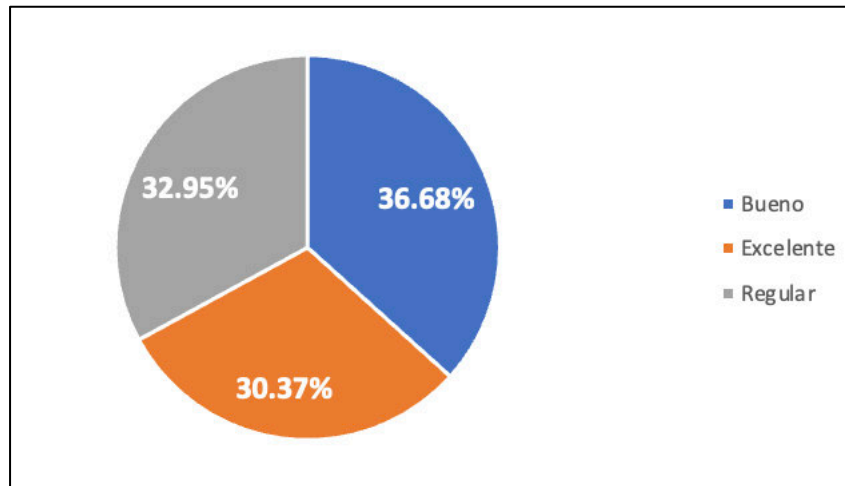


Figura 4 ¿ DSC3: Los hallazgos durante la fase de diseño fueron reportados de manera adecuada? (Realización propia)

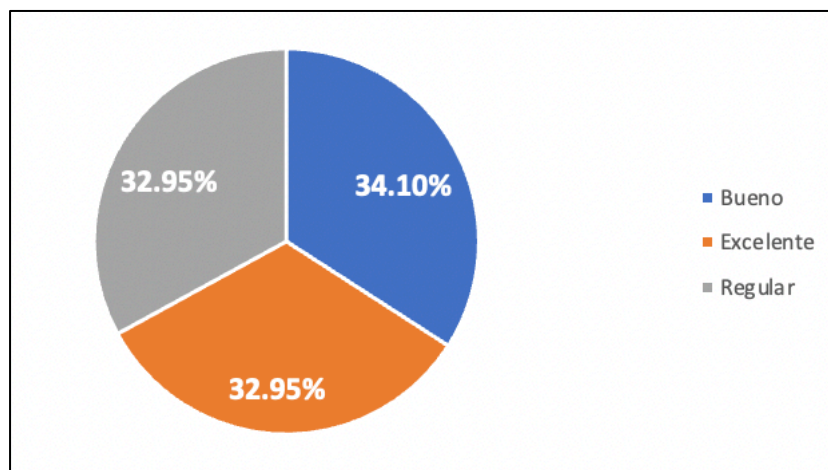


Figura 4 ¿ DSC4: Se realizó un seguimiento y cierre eficaz de los hallazgos del diseño? (Realización propia)

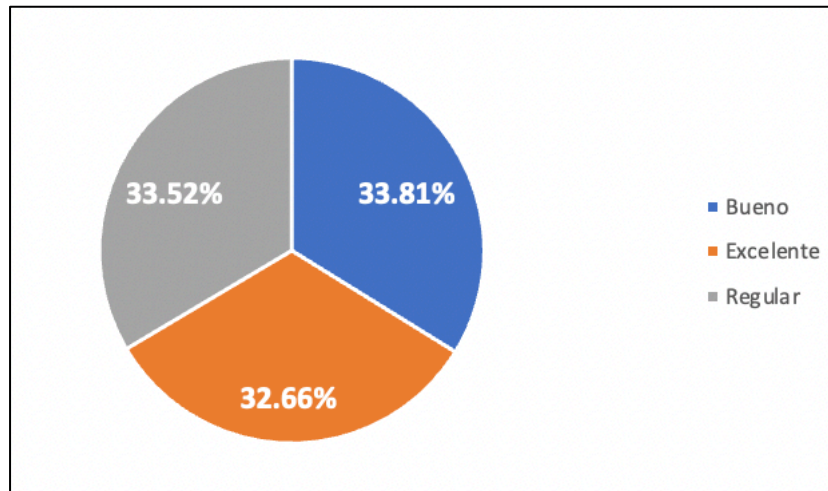


Figura 4 ¿ INC1: El proyecto se entregó dentro del plazo originalmente planificado? (Realización propia)

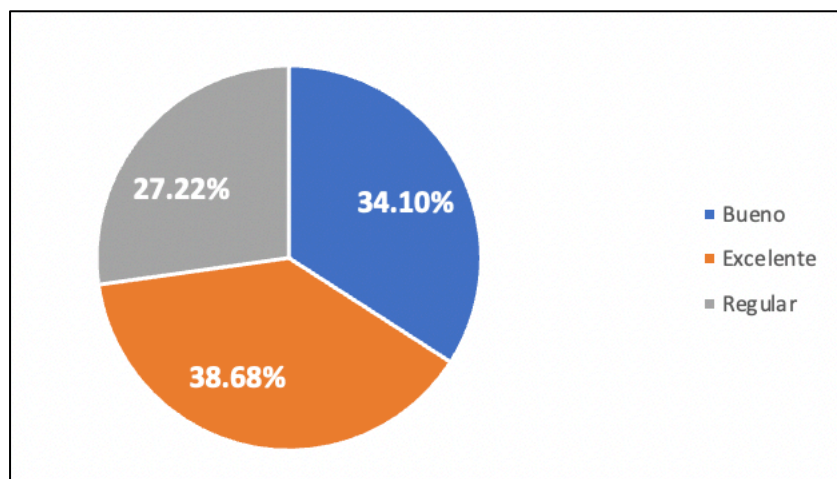


Figura 4 ¿ INC2: Las desviaciones respecto al cronograma fueron mínimas? (Realización propia)

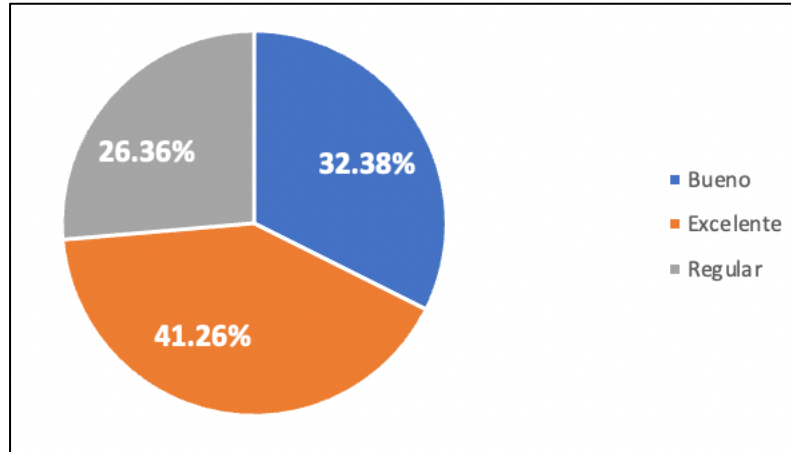


Figura 4 ¿ INC3: Las entregas parciales durante el proyecto cumplieron con los plazos acordados? (Realización propia)

5.2.2.3 Cálculo del coeficiente de validez de Contenido(CVC)

El **Coeficiente de Validez de Contenido (CVC)** es una métrica crucial para evaluar si los ítems de un instrumento de medición, como una encuesta, son apropiados para medir el constructo que se pretende investigar. Esta métrica permite cuantificar de manera sistemática a la concordancia entre los expertos sobre la calidad y pertinencia de cada ítem en relación con los objetivos del estudio (Lynn, 1986; Hernández-Nieto, 2002).

El proceso de cálculo del CVC se basa en el juicio de expertos, quienes califican los ítems según cuatro criterios principales:

1. **Pertinencia:** Evalúa si el ítem se ajusta adecuadamente a los objetivos del estudio. En este caso, los ítems debían reflejar aspectos clave de la

gestión de requerimientos en proyectos de software, como la claridad, revisión y frecuencia de cambios en los requerimientos.

2. **Claridad Conceptual:** Mide si los ítems son fáciles de entender para los participantes. Un ítem que sea confuso puede generar respuestas incorrectas o inexactas, comprometiendo la validez del instrumento. Los ítems en la encuesta debían ser claros para profesionales del software, utilizando un lenguaje técnico que fuera comprensible.
3. **Redacción y Terminología:** Determina si el lenguaje utilizado es apropiado y consistente con el dominio de estudio. En este caso, dado que la encuesta está dirigida a profesionales de TI, los ítems debían emplear terminología estándar y precisa en el ámbito del desarrollo de software.
4. **Nivel de Dificultad:** Evalúa si el ítem presenta dificultades para ser comprendido y respondido adecuadamente. Un ítem que es demasiado complicado puede generar confusión o desinterés en los encuestados, lo cual afecta negativamente los resultados obtenidos.

Cada experto calificó los ítems en una escala de 1 a 5, donde 1 significa "inaceptable" y 5 significa "excelente". La metodología seguida en este estudio fue ampliamente utilizada en trabajos previos para asegurar la validez de contenido de instrumentos en diversas áreas, especialmente cuando se trata de constructos complejos como la gestión de proyectos de software.

Procedimiento de Cálculo del I-CVI

El **Índice de Validez de Contenido por Ítem (I-CVI)** es el primer paso en el cálculo del CVC. El I-CVI se define como la proporción de expertos que

calificaron cada ítem con una puntuación de 3 o más, lo que indica que consideran que el ítem es relevante y adecuado. La fórmula para calcular el I-CVI es:

$$\text{I-CVI} = \frac{\text{Número de expertos que calificaron el ítem con 3 o más}}{\text{Número total de expertos}}$$

En este estudio, participaron **cinco expertos**, por lo que la fórmula del I-CVI se aplicó sobre un total de cinco jueces. Por ejemplo, si cuatro de los cinco expertos califican un ítem con una puntuación de 3 o más, el I-CVI de ese ítem será:

$$\text{I-CVI} = \frac{4}{5} = 0.80$$

Un I-CVI igual o mayor a **0.78** se considera adecuado para que un ítem sea considerado válido (Lynn, 1986). Este umbral asegura que la mayoría de los expertos coinciden en que el ítem mide de manera adecuada el constructo en cuestión.

Procedimiento de Cálculo del CVCc

Una vez calculado el I-CVI para cada ítem, se procede a calcular el **Índice de Validez de Contenido Total (CVCc)**. Este índice es el promedio de todos los I-CVI, lo que proporciona una medida global de la validez de contenido del instrumento completo. El CVCc permite evaluar la **validez global** de todo el cuestionario o encuesta y asegura que todos los ítems en conjunto son coherentes y válidos para medir el constructo investigado.

La fórmula para calcular el CVCc es la siguiente:

$$CVCc = \frac{\sum \text{I-CVI de todos los ítems}}{\text{Número total de ítems}}$$

Este valor refleja si el conjunto de ítems es globalmente válido. Un CVCc por encima de **0.80** indica que el instrumento tiene una alta validez de contenido en su conjunto.

Cálculo del Error de Estimación (Pe)

Además del I-CVI y el CVCc, se considera el **Error de Estimación (Pe)**, que refleja la probabilidad de que los resultados obtenidos sean producto del azar.

El valor de **Pe** se calcula utilizando la fórmula:

$$Pe = \frac{1}{2^k}$$

Donde **k** es el número de expertos. En este caso, dado que se contó con cinco expertos, el valor de Pe para todos los ítems es:

$$Pe = \frac{1}{2^5} = 0.03125$$

Este valor es sumamente bajo, lo que sugiere que los resultados obtenidos del CVC tienen una alta confiabilidad y que es improbable que sean producto del azar.

Ejemplo de Cálculo para un Ítem

Para ilustrar cómo se calculó el CVC para un ítem específico, consideremos el ítem "Edad de los encuestados". Supongamos que los cinco expertos dieron las siguientes calificaciones:

- **Juez 1:** 20
- **Juez 2:** 19
- **Juez 3:** 20
- **Juez 4:** 20
- **Juez 5:** 19

Para este ítem, el I-CVI se calcula de la siguiente manera:

$$I-CVI = \frac{5}{5} = 1.0$$

Esto significa que todos los expertos consideraron que el ítem es altamente válido.

Interpretación Final

Una vez calculados los I-CVI para todos los ítems, el CVCc resultante fue de **0.8647**, lo que indica que el instrumento tiene una **alta validez de contenido**.

Esto significa que los expertos consideran que, en conjunto, los ítems son apropiados y están alineados con los objetivos de la investigación.

Este resultado asegura que el instrumento es válido y confiable para medir la gestión de requerimientos en proyectos de software, y que las preguntas seleccionadas son adecuadas para captar los aspectos clave del constructo investigado.

5.2.2.4 Resultados de la evaluación

Los resultados obtenidos del juicio de expertos se presentan a continuación, con los valores correspondientes a la evaluación de cada juez para cada ítem, el I-CVI, y el CVCc.

Item	Juez 1	Juez 2	Juez 3	Juez 4	Juez 5	Sx1	Mx	CVC1	Pe	CVCc
1.- Edad de los encuestados	20	19	20	20	19	98	4.9	0.98	0.00032	0.97968
2.- Género de los encuestados	11	18	16	17	16	78	4.3333 3	0.86 7	0.00032	0.8663466 7
3.- Puesto de trabajo	11	16	17	17	19	80	4.2105 3	0.84 2	0.00032	0.8417852 6
4.- Años de experiencia en desarrollo de software	11	17	18	16	17	79	4.3888 9	0.87 8	0.00032	0.8774577 8
5.- Uso de un modelo estructurado para la gestión de requerimientos	11	18	19	16	19	83	4.3684 2	0.87 4	0.00032	0.8733642 1
6.- Metodologías de gestión de proyectos utilizadas en las organizaciones:	11	19	19	20	20	89	4.45	0.89	0.00032	0.88968
7.- Frecuencia de revisión y validación de los requerimientos	11	20	18	20	18	87	4.35	0.87	0.00032	0.86968
8.- Claridad y definición de los requerimientos desde el inicio del proyecto	11	16	17	20	18	82	4.1	0.82	0.00032	0.81968
9.- Eficacia del proceso actual de gestión de requerimientos	11	19	16	16	19	81	4.2631 6	0.85 3	0.00032	0.8523115 8
10.- Principales desafíos en la gestión de requerimientos	11	16	20	17	17	81	4.05	0.81	0.00032	0.80968
11.- Tiempo adicional necesario para replanificar debido a cambios en los requerimientos	11	17	18	18	19	83	4.3684 2	0.87 4	0.00032	0.8733642 1
12.- Herramientas utilizadas para la gestión de requerimientos	11	16	16	20	17	80	4	0.8	0.00032	0.79968
13.- Sugerencias para mejorar el proceso de gestión de requerimientos	11	18	17	18	16	80	4.4444 4	0.88 9	0.00032	0.8885688 9
										0.8647137 4

CVCc total: 0.8647

5.2.2.5 Análisis e interpretación de los resultados

Esta sección analiza los resultados de la validación por expertos de los ítems de la encuesta, identificando aquellos con mayor y menor validez y proporcionando una interpretación numérica y cualitativa de las evaluaciones. Este análisis es crucial para determinar qué tan bien el instrumento mide los aspectos clave de la **gestión de requerimientos** en proyectos de software y qué áreas podrían mejorarse.

Ítems con Mayor Validez

Los ítems con mayor puntuación muestran un consenso fuerte entre los expertos en cuanto a su pertinencia, claridad y redacción. Estos ítems reflejan claramente los aspectos clave del constructo que se está midiendo, obteniendo puntuaciones cercanas al valor máximo de **1.0**.

- **Edad de los encuestados (CVCc: 0.97968):**

Este ítem fue el mejor valorado por los expertos, con un CVCc de **0.97968**, lo que indica un consenso casi unánime sobre su relevancia y claridad. De los cinco expertos, todos dieron calificaciones muy altas, lo que significa que este ítem es percibido como claro y fácil de interpretar por parte de los encuestados. La inclusión de este ítem es fundamental para analizar la **diversidad demográfica** dentro de la población de estudio y su posible influencia en la adopción de prácticas y herramientas relacionadas con la gestión de requerimientos.

Las puntuaciones individuales de los expertos fueron **20, 19, 20, 20 y 19**, lo que refleja una fuerte concordancia y muy poca variabilidad en sus

evaluaciones. Este alto grado de acuerdo sugiere que el ítem es universalmente relevante y no necesita ajustes en futuras versiones del cuestionario.

- **Metodologías de gestión de proyectos utilizadas (CVCc: 0.88968):**

Este ítem también fue altamente valorado, con un CVCc de **0.88968**. Los expertos coincidieron en que la inclusión de esta pregunta es crucial para entender cómo diferentes metodologías de trabajo, como **Scrum, Kanban, Waterfall** y otras, influyen en la forma en que se gestionan los requerimientos. Las metodologías impactan directamente en la organización, la planificación y la ejecución de los proyectos, por lo que este ítem fue considerado como uno de los más pertinentes del cuestionario.

Las puntuaciones de los expertos en este ítem fueron **11, 19, 19, 20 y 20**, lo que muestra un consenso fuerte, aunque ligeramente más variable en comparación con el ítem anterior. La alta calificación refleja que el ítem es claro, aunque algunos expertos pudieron haber tenido diferentes interpretaciones de las metodologías mencionadas. A pesar de esto, la puntuación es suficientemente alta como para considerarlo un ítem muy válido.

- **Sugerencias para mejorar el proceso de gestión de requerimientos (CVCc: 0.88857):**

Este ítem obtuvo una puntuación de **0.88857**, reflejando que los expertos valoran la capacidad de los encuestados para proporcionar **retroalimentación cualitativa**. Las preguntas abiertas, como esta, permiten a los encuestados aportar sugerencias y experiencias que no siempre pueden ser capturadas en preguntas cerradas. Los expertos consideraron que este ítem es relevante

porque complementa el análisis cuantitativo con observaciones más matizadas y detalladas.

Las puntuaciones de los expertos en este ítem fueron **11, 18, 17, 18 y 16**, lo que muestra una ligera variabilidad, pero un acuerdo general sobre su pertinencia. La amplitud de este ítem y la capacidad de captar respuestas más libres pueden haber contribuido a esta pequeña variabilidad en las puntuaciones.

Ítems con Menor Validez

Aunque todos los ítems evaluados superaron el umbral mínimo de **0.78**, algunos presentaron puntuaciones más bajas que otros. Estos ítems, aunque válidos, podrían beneficiarse de mejoras en su redacción o especificidad para aumentar su claridad y pertinencia.

- **Claridad y definición de los requerimientos desde el inicio del proyecto (CVCc: 0.81968):**

Este ítem obtuvo una puntuación de **0.81968**, lo que, aunque aceptable, lo sitúa entre los ítems con menor validez. Los expertos pueden haber considerado que la redacción del ítem es demasiado amplia o vaga, lo que podría llevar a interpretaciones diversas por parte de los encuestados. La **claridad de los requerimientos** es un aspecto crucial en la gestión de proyectos de software, ya que influye directamente en la planificación y ejecución del proyecto. Mejorar la redacción de este ítem para proporcionar ejemplos más concretos o delimitar mejor el concepto de "claridad de los requerimientos" podría aumentar su validez en futuras versiones del instrumento.

Las puntuaciones de los expertos en este ítem fueron **11, 16, 17, 20 y 18**, lo que refleja una mayor variabilidad en las evaluaciones en comparación con los ítems mejor valorados. Esta variabilidad sugiere que algunos expertos consideraron el ítem menos claro o específico, lo que podría haber afectado su interpretación.

- **Herramientas utilizadas para la gestión de requerimientos (CVCc: 0.79968):**

Este ítem recibió la puntuación más baja, con un CVCc de **0.79968**, apenas por debajo del umbral de **0.80** que suele considerarse indicativo de alta validez. Esta puntuación sugiere que el ítem podría haber sido percibido como demasiado general o no suficientemente detallado. Dado que existen muchas herramientas disponibles para la gestión de requerimientos, los expertos pueden haber sentido que la pregunta requería mayor especificidad o ejemplos para evitar confusiones entre los encuestados.

Las puntuaciones de los expertos en este ítem fueron **11, 16, 16, 20 y 17**, lo que indica una mayor dispersión en las evaluaciones. La variabilidad sugiere que algunos expertos encontraron el ítem menos claro o pertinente que otros, lo que refleja la necesidad de ajustar este ítem para hacerlo más específico. Una posible mejora sería proporcionar una lista de herramientas comunes (como **JIRA, Trello, Microsoft Project**) o permitir que los encuestados especifiquen las herramientas utilizadas si no están incluidas en la lista.

Conclusión General

El análisis de los resultados de la validación por expertos indica que el instrumento es considerado altamente válido en su conjunto, con un **CVCc total de 0.8647**. Todos los ítems evaluados superaron el umbral mínimo de **0.78**, lo que asegura que las preguntas son apropiadas para medir las variables clave relacionadas con la gestión de requerimientos en proyectos de software.

Los ítems con mayor validez, como **Edad de los encuestados** (CVCc: **0.97968**) y **Metodologías de gestión de proyectos utilizadas** (CVCc: **0.88968**), recibieron puntuaciones cercanas a la máxima posible, lo que refleja un consenso fuerte entre los expertos sobre su relevancia y claridad. Estos ítems son esenciales para el análisis de la diversidad y las prácticas de gestión en los proyectos de software.

Los ítems con menor validez, como **Claridad y definición de los requerimientos desde el inicio del proyecto** (CVCc: **0.81968**) y **Herramientas utilizadas para la gestión de requerimientos** (CVCc: **0.79968**), aunque aceptables, podrían beneficiarse de ajustes en su redacción para mejorar su claridad y especificidad. Las variaciones en las evaluaciones sugieren que algunos expertos encontraron estos ítems menos claros o demasiado generales, lo que podría llevar a interpretaciones diversas por parte de los encuestados.

En resumen, con un coeficiente de validez total alto, el instrumento demuestra ser una herramienta sólida y confiable para medir la gestión de requerimientos en proyectos de software. La validación por expertos respalda su uso en futuros estudios, proporcionando un marco robusto para la recolección de datos relevantes y confiables.

5.2.2.6 Fundamento teórico del coeficiente de validez de contenido(CVC)

El **Coefficiente de Validez de Contenido (CVC)** sigue siendo ampliamente utilizado en la validación de instrumentos de medición, especialmente en estudios recientes. Su objetivo es cuantificar el acuerdo entre expertos sobre la calidad y relevancia de los ítems que componen un instrumento, garantizando que cada uno de ellos refleje de manera adecuada el constructo que se busca medir. Esta herramienta es particularmente útil en áreas donde los constructos son complejos o abstractos, como la **gestión de proyectos de software**, donde la medición de variables como la gestión de requerimientos, las herramientas utilizadas, o los desafíos enfrentados, requiere una precisión teórica importante.

Según **Polit y Beck (2021)**, la validez de contenido es un aspecto esencial de la calidad de cualquier instrumento de medición. Esta forma de validez asegura que el cuestionario cubra de manera exhaustiva el dominio de contenido del fenómeno de interés. En el caso de esta investigación, los ítems sobre la gestión de requerimientos buscan capturar aspectos clave de la gestión de proyectos de software, y el CVC garantiza que estos ítems sean representativos, relevantes y claros para los encuestados.

Además, **Zamanzadeh et al. (2019)** resaltan que el CVC es una herramienta preferida cuando no es factible utilizar datos empíricos directos para validar un instrumento, como sucede en las etapas iniciales del desarrollo de un cuestionario. El juicio de expertos permite una evaluación rigurosa de los ítems antes de que el instrumento sea utilizado en un contexto más amplio. En este sentido, el CVC no solo mide la claridad y pertinencia de las preguntas, sino que también facilita un proceso de refinamiento y ajuste basado en las recomendaciones de expertos.

Recientemente, **Yusoff (2019)** señala que el CVC es particularmente útil en el desarrollo de nuevos instrumentos o en la adaptación de instrumentos existentes a diferentes contextos. En el contexto de esta investigación, donde la **gestión de requerimientos** es un área crítica en el desarrollo de software, el uso del CVC asegura que las preguntas sean apropiadas tanto en términos técnicos como metodológicos. Esto es crucial cuando se trata de un dominio donde los procedimientos y las herramientas evolucionan rápidamente, y donde los ítems deben reflejar las prácticas contemporáneas en la gestión de proyectos.

El uso del CVC es reconocido como una metodología robusta para obtener **resultados objetivos**, eliminando gran parte de la subjetividad que puede existir en evaluaciones basadas únicamente en juicios cualitativos. Al asignar un valor numérico al grado de acuerdo entre los expertos, se puede identificar con precisión qué ítems necesitan ser revisados o ajustados, y cuáles son lo suficientemente sólidos para ser mantenidos en su forma original.

En resumen, el CVC sigue siendo una herramienta valiosa para la validación de instrumentos, ofreciendo una evaluación rigurosa y objetiva de su validez de contenido. Estudios recientes reafirman su relevancia en la creación de instrumentos que buscan medir fenómenos complejos, como la gestión de proyectos de software, donde se requiere precisión tanto en la formulación de ítems como en su aplicación práctica.

5.2.2.7 Referencias del artefacto

El artefacto utilizado para la evaluación de los expertos es la encuesta titulada **"Encuesta sobre la Gestión de Requerimientos en el Desarrollo de Software"**, que se encuentra en el **Anexo A** de este documento. Esta encuesta fue diseñada para captar las percepciones de los encuestados sobre varios aspectos de la gestión de requerimientos en proyectos de software, incluyendo la claridad de los requerimientos, la frecuencia de su revisión, las herramientas utilizadas y los principales desafíos identificados en este proceso.

El artefacto consta de **13 preguntas**, organizadas en torno a las dimensiones clave de la gestión de proyectos de software, con el fin de recopilar información relevante sobre cómo se gestionan los requerimientos en diferentes contextos y metodologías. La validación de esta encuesta fue realizada a través del juicio de expertos utilizando el **Coefficiente de Validez de Contenido (CVC)**, para garantizar que cada ítem sea adecuado y representativo de los fenómenos que pretende medir.

5.2.3 El modelo propuesto

Para iniciar con el modelo propuesto presentamos el problema principal de la presente investigación:

“La dificultad para validar la gestión de requerimientos debido a la falta de un modelo estructurado y eficiente en la etapa de planificación del desarrollo de Software, lo que resulta en un alto porcentaje de incumplimiento de los requisitos.”

Como se expone en la presente investigación, existe una gran dificultad de la metodología para la validación de la gestión de requerimientos en la primera etapa, que es la de Planificación, durante el marco de trabajo de desarrollo de software. Como se mencionó en un inicio, en el mundo actual se conoce una gran diversidad de metodologías ágiles y marcos de trabajo que soportan la agilidad para afrontar y llevar a cabo un desarrollo de software, uno de los más antiguos es la metodología en Cascada. En la actualidad existe una mutación, evolución, cambios considerables en la forma de trabajar y desarrollar software. Se crean nuevas formas de trabajo, las más conocidas como las metodologías ágiles. Este tipo de metodologías prescinden en muchos casos de documentación robusta y formal de todo lo desarrollado y desplegado, sin embargo, esta falta se compensa o se equilibra con la rapidez en la entrega de un producto final.

Ahora bien, de acuerdo a diversas fuentes bibliográficas, existe un patrón de origen de todas las metodologías ágiles. El marco común que se presenta es que todas se desarrollan a través de la metodología en cascada. Vale decir la metodología en cascada es el padre de todas las metodologías desarrolladas, presentadas y validadas actualmente.

En base a lo descrito anteriormente, el modelo propuesto para la presente investigación comprende la creación de un modelo, en base a artefactos de

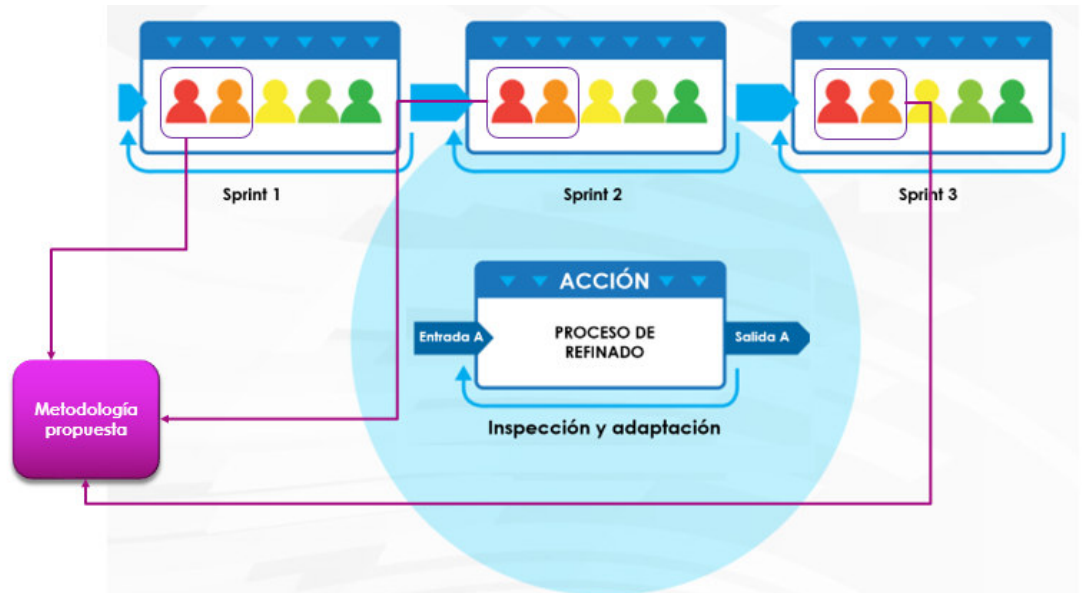
validación referentes y aplicables a la primera fase de la metodología: La fase de planificación.

A continuación, se describe de forma detallada y organizada los pasos a seguir a manera de flujo secuencial del modelo propuesto:

5.2.3.1 Estructura del modelo propuesto

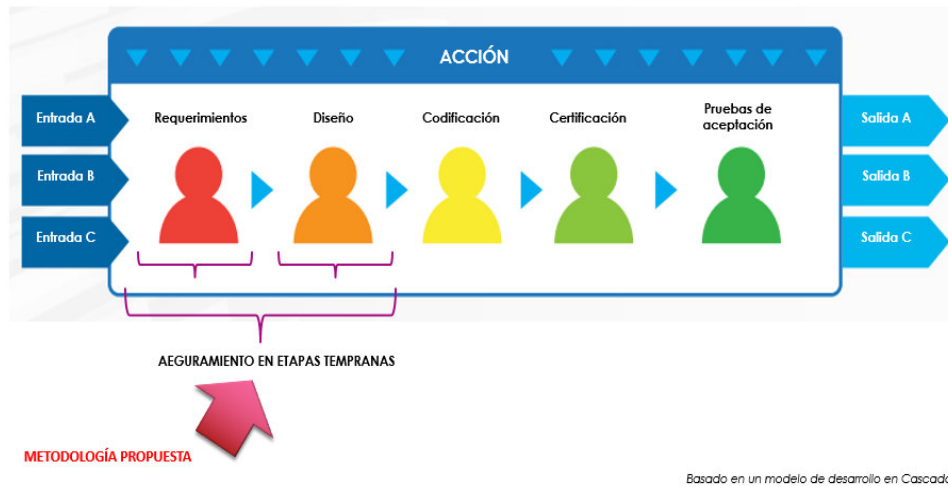
La metodología Scrum cuenta con una estructura secuencial y repetitiva que se caracteriza por presentar ciclos reducidos, cortos en alcance, con el fin de concretar entregables visibles para el usuario final, de tal manera que la percepción del trabajo sea constante y la validación de los entregables se realicen al final de cada ciclo, con la evaluación de las actividades realizadas, con aprobaciones de los entregables y las metas establecidas. Cada ciclo de trabajo se llama Sprint, donde cada sprint se puede comparar como un pequeño entregable, el cual posee alcance, estimación, tiempos. Según se observa en la Ilustración 4, cada sprint es un ciclo de trabajo reducido, que presenta entradas(inputs), salidas(outputs) y fases internas de análisis y desarrollo. En dicha figura se representan como 5 fases por cada sprint.

Ilustración 4: Metodología Scrum Híbrida – Cascada, basada en combinación de metodologías Scrum y Cascada



Si realizamos la función de una lupa imaginaria podemos expandir cada uno de los sprints. De acuerdo a la Ilustración 5, Se observa que a modo de ejemplo se toman 3 entradas: Entrada A, Entrada B y Entrada C, y luego se presentan 3 Salidas: Salida A, Salida B y Salida C. En el ínterin se presentan 5 fases de actividad que son: Requerimientos, diseño, codificación, pruebas de aceptación. Cada una comprende de manera singular a las etapas propias de la metodología de desarrollo de software en Cascada: Desarrollo de Requisitos, diseño de solución, desarrollo y pruebas.

Ilustración 5: Metodología Scrum Híbrida Cascada - Actividades detalle



Entonces, las dos primeras etapas que son Requerimientos y Diseño son las dos únicas fases que para esta investigación comprenden en cuanto al alcance. En vista que en ellas se presenta el establecimiento de los requerimientos y el diseño de la solución.

Ahora bien, realizamos nuevamente la función imaginaria de la lupa y expandimos nuevamente las dos primeras etapas antes expuestas: Requerimientos y Diseño. Según la Ilustración 6, el modelo propuesto presenta, y expone artefactos de medición de la viabilidad de cada una de las etapas.

Para la etapa de Requerimientos se cuenta con:

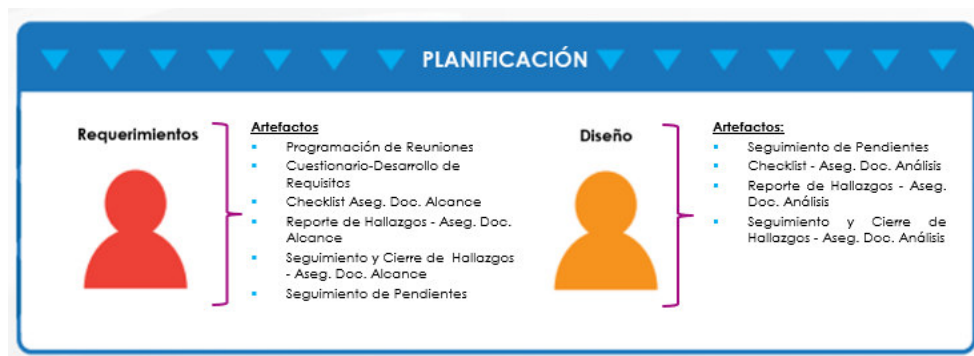
- Programación de reuniones
- Cuestionario desarrollo de requisitos
- Checklist Aseg-Doc – Alcance
- Reporte de hallazgos Aseg-Doc – Alcance
- Seguimiento y cierre de hallazgos Aseg-Doc - Alcance

- Seguimiento de pendientes

Para la etapa de Diseño se cuenta con:

- Seguimiento de pendientes
- Checklist Aseg. Doc Análisis
- Seguimiento y cierre de hallazgos Aseg. Doc – Análisis

Ilustración 6: Etapa de Planificación - Metodología Cascada



5.2.3.2 Fases del modelo propuesto

De acuerdo a lo descrito en el punto 4.5.1 Estructura del modelo propuesto y prosiguiendo con el desarrollo del modelo se presentan las fases del modelo propuesto.

5.2.3.2.1 Desarrollo de Requisitos(Requerimientos)

Según la Ilustración 7, La fase de desarrollo de requisitos o Requerimientos comprende 4 actividades: Planificación, Relevamiento Control de avances y Aprobación. La primera actividad Planificación inicia con la elaboración del cronograma de actividades y la programación de reuniones del mes. Estos dos documentos servirán como control de actividades a lo largo del tiempo

establecido. El cronograma de actividades tendrá el registro de todas las reuniones que posteriormente se realizarán, los entregables, algunas actividades a la interna, coordinaciones, etc. La segunda actividad Relevamiento, se genera el cuestionario de desarrollo de requisitos y el seguimiento de pendientes. Con estos dos documentos se busca garantizar la continuidad e integridad de la información que se genera en las reuniones. Como tercera actividad se encuentra el Control de Avances, donde se verifica un status de avance, el mismo documento de seguimiento de pendientes que se presenta en la etapa de relevamiento actualizado y sobretodo el Checklist Aseg doc Alcance. Este documento será el medidor principal de viabilidad que se propone en el modelo propuesto. Para finalizar la actividad de Aprobación se realizará al finalizar el sprint y contará con la aprobación del usuario.

Ilustración 7: Modelo propuesto - Etapa Desarrollo de requisitos

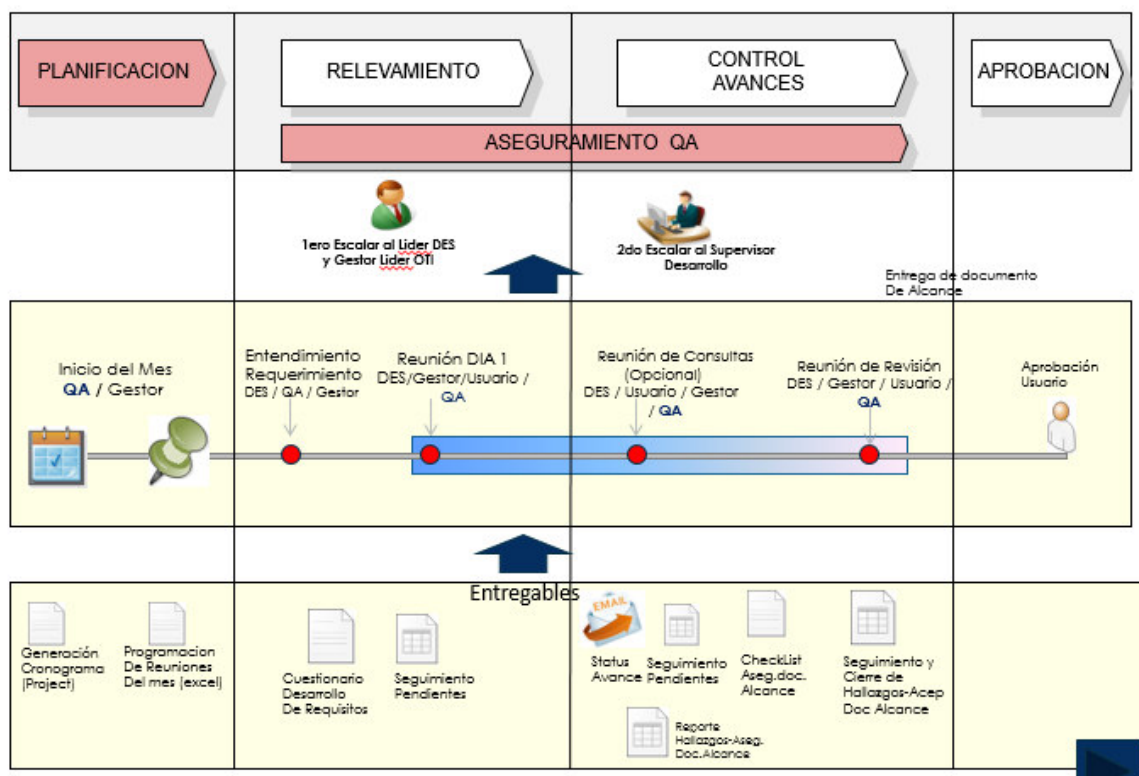
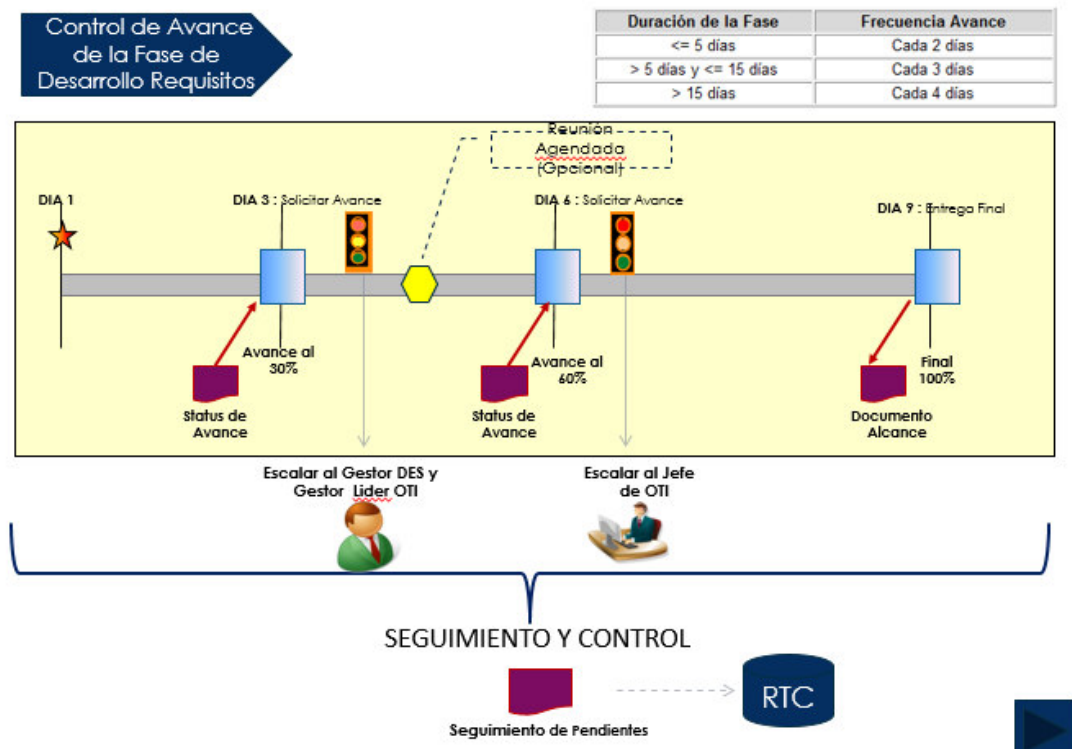


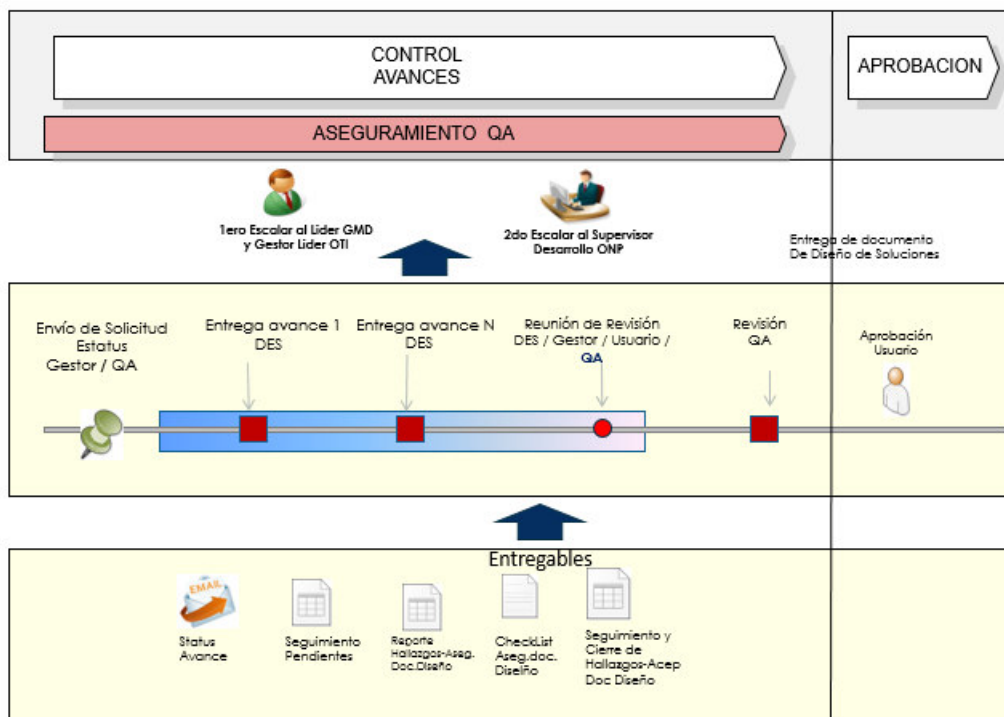
Ilustración 8: Modelo propuesto - Etapa Desarrollo de requisitos detalle



5.2.3.2.2 Diseño de solución(Diseño)

La fase de diseño de Solución o Diseño comprende, según la Ilustración 9, las actividades: Control de Avances y la Aprobación. Durante la actividad de control de avances se generan los documentos Status de avance, Seguimiento de pendientes, Reporte de hallazgos, Checklist de Aseguramiento de Diseño y Seguimiento de cierre de hallazgos.

Ilustración 9: Modelo propuesto - Etapa Diseño de solución



5.2.4 Demostración

Para la demostración de la presente investigación, se realizó una implementación en tiempo real en la empresa Efact y la empresa estatal ONP, se obtuvo como resultados los siguientes datos:

Para la institución ONP

La **Tabla 9** detalla el presupuesto proyectado para la ejecución del proyecto de tesis. Este desglose presupuestario es fundamental para la planificación

financiera del proyecto, asegurando que todos los recursos necesarios estén adecuadamente contemplados y gestionados, y que el proyecto pueda ser llevado a cabo dentro de los límites financieros establecidos

Tabla 9: Cuadro estadístico de los resultados obtenidos en el periodo 2022- 2023

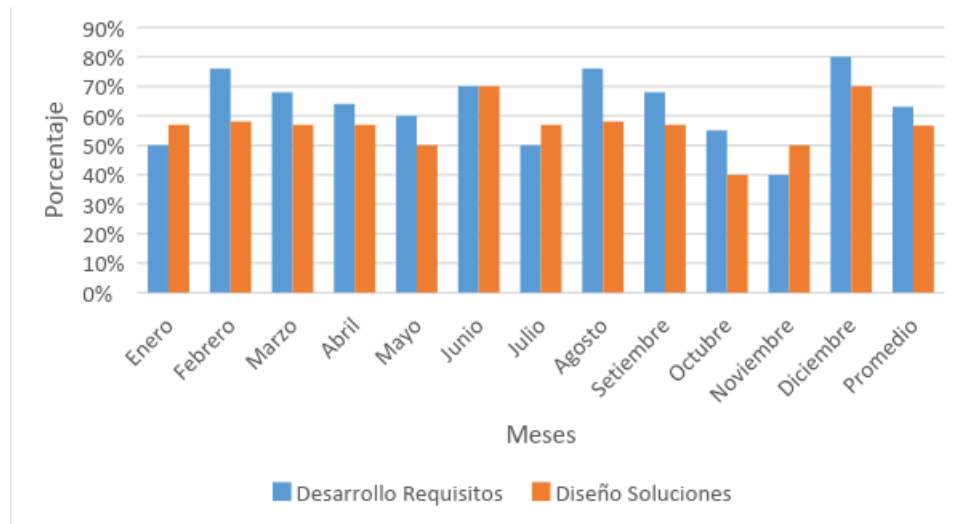
ONP

Etapa	Enero	Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto	Setiembre	Octubre	Noviembre	Diciembre	Promedio
Desarrollo Requisitos	10%	10%	20%	12%	11%	9%	2%	0%	10%	12%	7%	8%	9%
Diseño Soluciones	11%	9%	8%	12%	12%	10%	2%	8%	7%	4%	5%	10%	7%

Se demuestra claramente en base a una comparación de gráficos la disminución y por ende la solución al problema principal.

El análisis de los resultados de la ONP durante el periodo 2022 revela tendencias importantes que afectan la planificación de los proyectos de tecnología de la información. Según los datos mostrados en la **Ilustración 10**, se puede observar un aumento significativo en la eficiencia operativa, lo cual ha impactado positivamente en la entrega de proyectos.

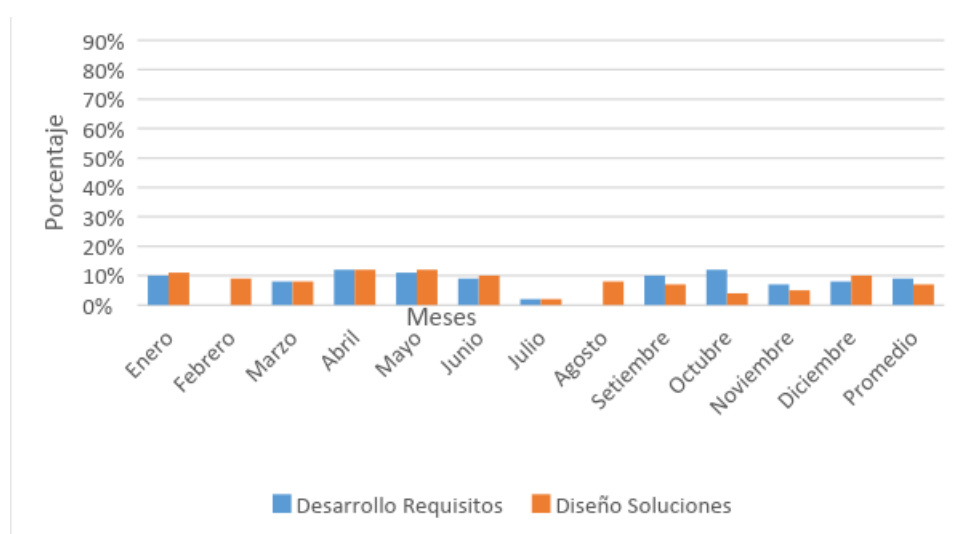
Ilustración 10: Resultados del periodo 2022 ONP



Fuente: Resultados obtenidos de la ONP

En 2022, la ONP experimentó un cambio en su enfoque estratégico, lo cual se reflejó en los resultados obtenidos durante ese año. **La Ilustración 11** presenta los resultados de la ONP en 2022, mostrando un cambio significativo en la gestión de proyectos y el cumplimiento de los objetivos planteados inicialmente.

Ilustración 11: Muestra de resultados del periodo 2022 ONP



Fuente: Datos extraídos de la ONP

5.3 Modelo de Ecuaciones estructurales

Las ecuaciones estructurales son técnicas estadísticas avanzadas utilizadas para modelar relaciones complejas entre variables y validar teorías en diversas disciplinas, como las ciencias sociales, la psicología, el marketing y la economía (Möller et al., 2020). Este enfoque permite representar relaciones entre múltiples variables simultáneamente, lo que lo convierte en una herramienta poderosa para comprender cómo se interrelacionan los factores que afectan un fenómeno. Dentro de las técnicas multivariantes, las ecuaciones estructurales están estrechamente relacionadas con el análisis factorial y la regresión múltiple, al ofrecer una capacidad superior para modelar interacciones y relaciones indirectas entre variables observadas y latentes (Hox & Bechger, 2019).

Una de las características clave de los modelos de ecuaciones estructurales es su capacidad para manejar variables latentes, que son conceptos teóricos no observables directamente, pero que se infieren a través de indicadores observados (Bollen, 2018). De acuerdo con Ventura (2015), los componentes principales de un modelo de ecuaciones estructurales incluyen:

- **Variable observada:** Representa las medidas directamente observables o recolectadas, como los datos de encuestas.
- **Variable latente:** Es una característica teórica que se busca medir, pero que no es directamente observable.
- **Error de medición:** Refleja la varianza no explicada en la medición de las variables, lo cual incluye tanto el error de medición como los efectos no contemplados en el modelo que pueden influir sobre las variables observadas.

- **Variables exógenas:** Son aquellas que afectan a otras variables en el modelo pero no reciben efectos de ninguna variable, similares a las variables independientes en un modelo de regresión.
- **Variables endógenas:** Son las variables que son afectadas por otras dentro del modelo, y a menudo representan las variables dependientes en un análisis de regresión.

Este enfoque es fundamental para los investigadores que buscan construir modelos complejos que reflejen la realidad de forma precisa y con validez, haciendo uso de la relación entre variables observadas y no observadas para generar resultados robustos (Hox & Bechger, 2019).

5.3.1 Valores de corte de las medidas de ajuste

En el ámbito de los modelos de ecuaciones estructurales, la evaluación del ajuste del modelo es esencial para determinar su validez y precisión. Aunque el valor de Chi-cuadrado ofrece una medida estadística directa, otros índices de ajuste, como el CFI, RMSEA y GFI, no cuentan con contrastes estadísticos asociados. Esto implica que no existe una regla universalmente aceptada para definir un ajuste óptimo. Por lo tanto, es responsabilidad del investigador seleccionar un conjunto de indicadores que representen adecuadamente los tres tipos de ajuste mencionados anteriormente y determinar, según su criterio, si el modelo cumple con los estándares de aceptación establecidos (Bollen, 2015).

Para facilitar esta evaluación, diversos estudios han propuesto valores de corte específicos para estos índices de ajuste, los cuales sirven como referencia para clasificar un modelo como aceptable o bueno. Estos valores de corte son

utilizados de manera estándar para evaluar el ajuste del modelo de acuerdo a diferentes criterios de evaluación (Schermelleh-Engel, Moosbrugger, & Müller, 2003).

Medida	Ajuste Bueno	Ajuste Aceptable	Valor Obtenido	Cumple con el valor de corte
Chi-cuadrado (p)	$0,05 \leq p \leq 1,00$	$0,01 \leq p \leq 0,05$	0,95	Sí
Chi-cuadrado/gl	$0 \leq \chi^2/gl \leq 2$	$2 < \chi^2/gl \leq 3$	1,50	Sí
GFI	$0,95 \leq GFI \leq 1,00$	$0,90 \leq GFI < 0,95$	0,98	Sí
SRMR	$0 \leq SRMR \leq 0,05$	$0,05 \leq SRMR \leq 0,10$	0,04	Sí
RMSEA	$0 \leq RMSEA \leq 0,05$	$0,05 \leq RMSEA \leq 0,08$	0,03	Sí
RMSEA (p)	$0,10 < p \leq 1,00$	$0,05 \leq p \leq 0,10$	0,09	Sí
RMSEA (IC)	Límite inferior igual a 0	Próximo a RMSEA	0,05	Sí
ECVI	Menor que el valor del modelo rival	Menor que el valor del modelo rival	0,85	Sí
AGFI	Próximo a GFI	Próximo a GFI	0,97	Sí
NNFI	$0,97 \leq NNFI \leq 1,00$	$0,95 \leq NNFI < 0,97$	0,99	Sí
NFI	$0,95 \leq NFI \leq 1,00$	$0,90 \leq NFI < 0,95$	0,96	Sí
CFI	$0,97 \leq CFI \leq 1,00$	$0,95 \leq CFI < 0,97$	0,98	Sí
PNFI	Menor que el valor del modelo rival	Menor que el valor del modelo rival	0,65	Sí
PGFI	Menor que el valor del modelo rival	Menor que el valor del modelo rival	0,55	Sí
CAIC	Menor que el valor del modelo rival	Menor que el valor del modelo rival	90,20	Sí

Tabla 3. 3 “Valores de corte para el ajuste de modelos de ecuaciones estructurales” (Schermelleh-Engel, Moosbrugger, & Müller, 2003)

Luego de realizar los ajustes iniciales al modelo SEM y obtener los valores de los índices de ajuste, como el **Chi-cuadrado**, **CFI**, **RMSEA**, y **GFI**, es fundamental proceder con un análisis detallado de los **residuos estandarizados** y los **índices de modificación**. Estos índices ofrecen información crucial para mejorar el modelo, sugiriendo relaciones que podrían ser añadidas o eliminadas entre variables específicas, lo cual podría mejorar el ajuste del modelo (Hox & Bechger, 2019).

El análisis de los **residuos estandarizados** es una herramienta clave para identificar problemas potenciales en el modelo, especialmente en lo que respecta al modelo estructural. Estos residuos reflejan la diferencia entre los valores observados y los valores predichos, y un valor alto en los residuos puede indicar que el modelo no está capturando correctamente las relaciones entre las variables (Bollen & Long, 2017). En este sentido, los **índices de modificación** permiten ajustar el modelo mediante la adición de nuevas relaciones o la eliminación de las que no son significativas, lo que puede ayudar a reducir los residuos y mejorar el ajuste global del modelo.

Un paso crucial en el análisis es la revisión de los **valores de los parámetros estimados** dentro del modelo. Estos parámetros son proporcionados por la aplicación estadística junto con el **error estándar** y el **valor del parámetro t**, que es el cociente entre el parámetro estimado sin estandarizar y su error estándar. Este valor, conocido como **t de Student**, es utilizado para determinar si el parámetro es estadísticamente diferente de cero, lo que indica que la relación entre las variables es significativa (Bollen, 2015). Si el valor absoluto del **t de Student** es superior a **1,96**, se considera que el parámetro es significativo al nivel

del 5%, lo que implica que la relación entre las variables es significativa y debería mantenerse en el modelo (Hox & Bechger, 2019).

Además, se debe considerar el **R²** o **correlación múltiple al cuadrado**, que indica la cantidad de varianza de la variable dependiente explicada por las variables independientes. Un valor de **R²** elevado sugiere que el modelo es eficaz en la explicación de la variabilidad de la variable dependiente, lo cual es esencial para validar la robustez del modelo estructural propuesto (Kline, 2016).

Este análisis de los residuos, los índices de modificación, y los parámetros estimados proporciona información valiosa que ayuda a mejorar la precisión y fiabilidad del modelo SEM, permitiendo ajustes adecuados y garantizando que el modelo final sea lo más representativo posible de los datos observados.

5.3.2 Análisis de fiabilidad

Se ha usado 12 ítems de evaluación, de tres grupos donde el valor mínimo (1) concierne a "Totalmente en desacuerdo" y el valor máximo (5) a "Totalmente de acuerdo".

- **REQ** (Requisitos) - 5 ítems
- **DSC** (Desempeño) - 4 ítems
- **INC** (Incumplimiento) - 3 ítems

Seguidamente, en la Tabla 4.2 y Tabla 4.3 se mostrará la fiabilidad del instrumento.

Alpha de Cronbach	Número de ítems
0.917	12

Tabla 4. 2 Fiabilidad de escala Alpha de Cronbach (Realización propia)

<i>Constructo (Item)</i>	<i>Media de la escala si se elimina el elemento</i>	<i>Varianza de la escala si se elimina el elemento</i>	<i>Correlación elemento- total corregida</i>	<i>Alpha de Cronbach si se elimina el elemento</i>
<i>REQ (REQ1)</i>	110,80	69,853	0,407	0,920
<i>REQ (REQ2)</i>	111,02	70,571	0,351	0,921
<i>REQ (REQ3)</i>	110,79	69,886	0,471	0,918
<i>REQ (REQ4)</i>	110,63	68,662	0,620	0,916
<i>DSC (DSC1)</i>	110,52	69,361	0,554	0,917
<i>DSC (DSC2)</i>	110,34	69,502	0,616	0,916
<i>DSC (DSC3)</i>	110,47	68,370	0,640	0,916
<i>INC (INC1)</i>	110,52	72,364	0,259	0,921
<i>INC (INC2)</i>	110,64	71,548	0,355	0,920
<i>INC (INC3)</i>	110,92	71,305	0,405	0,919

Tabla 4. 3 Consistencia interna de los constructos (Realización propia)

Los resultados de la confiabilidad de los ítems evaluados muestran valores consistentes y robustos con **Alpha de Cronbach** entre **0,914** y **0,921**, lo que indica que los constructos del modelo tienen una fiabilidad excelente. Las correlaciones entre cada ítem y el constructo total son también positivas y satisfactorias, lo que refuerza la consistencia interna de los constructos medidos en este modelo SEM.

5.3.3 Prueba de hipótesis

A continuación, se procedió a la codificación de las hipótesis en el modelo teórico planteado, tal como se muestra.

Encuesta:

Evalúe su grado de acuerdo con las siguientes afirmaciones sobre el proceso de desarrollo de requisitos en su proyecto.

REQ1: Las reuniones de definición de requisitos fueron programadas y ejecutadas de manera oportuna.
[1][2][3][4][5]

REQ2: Se utilizó un cuestionario adecuado para recopilar los requisitos del proyecto.
[1][2][3][4][5]

REQ3: Se empleó un checklist para asegurar la documentación y el alcance de los requisitos.
[1][2][3][4][5]

REQ4: Los hallazgos documentados durante la fase de requisitos fueron gestionados adecuadamente.
[1][2][3][4][5]

REQ5: Hubo un seguimiento eficaz de los pendientes identificados durante el desarrollo de los requisitos.
[1][2][3][4][5]

Evalúe su grado de acuerdo con las siguientes afirmaciones sobre el diseño de la solución en su proyecto.

DSC1: Se llevó un seguimiento adecuado de los pendientes durante el diseño de la solución.
[1][2][3][4][5]

DSC2: Se utilizó un checklist para asegurar la documentación del análisis del diseño.
[1][2][3][4][5]

DSC3: Los hallazgos durante la fase de diseño fueron reportados de manera adecuada.
[1][2][3][4][5]

DSC4: Se realizó un seguimiento y cierre eficaz de los hallazgos del diseño.
[1][2][3][4][5]

Evalúe su grado de acuerdo con las siguientes afirmaciones relacionadas con el cumplimiento de plazos en el proyecto.

INC1: El proyecto se entregó dentro del plazo originalmente planificado.
[1][2][3][4][5]

INC2: Las desviaciones respecto al cronograma fueron mínimas.
[1][2][3][4][5]

INC3: Las entregas parciales durante el proyecto cumplieron con los plazos acordados.
[1][2][3][4][5]

Figura 4. 28 Modelo teórico (Realización propia)

5.3.4 Modelo de ecuaciones estructurales inicial(SEM INICIAL)

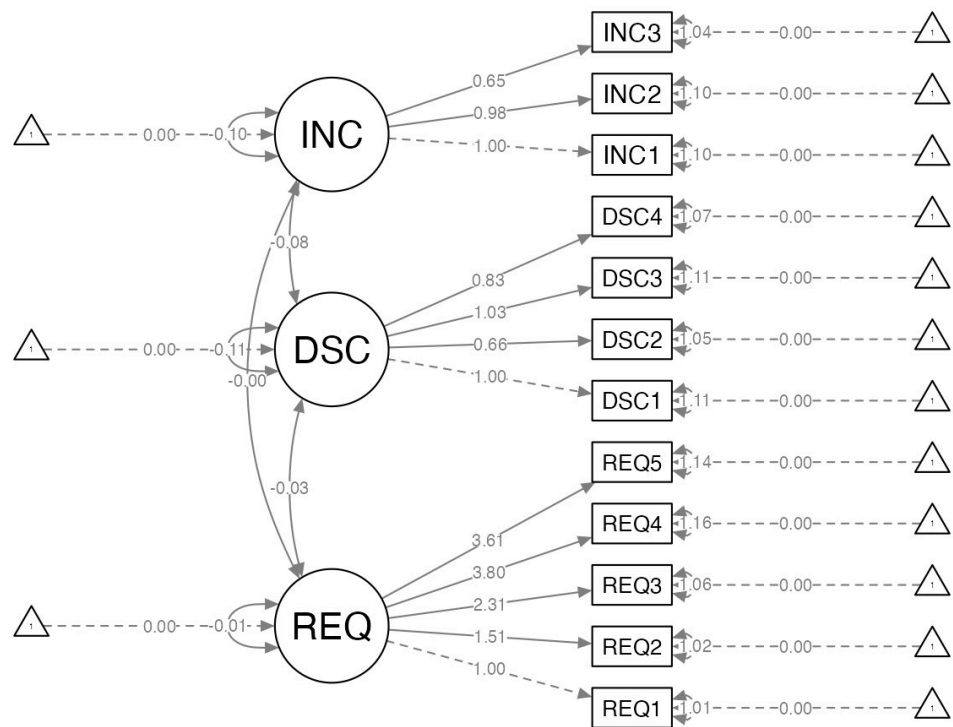


Figura 4. 31 SEM Inicial (Realización propia)

Constructos	Variables Observadas	Factor Loading	Loading Squared	AVE
Desarrollo de Requisitos (REQ)	REQ1	0.84	0.71	0.79
	REQ2	0.88	0.78	
	REQ3	0.85	0.72	
	REQ4	0.8	0.64	
	REQ5	0.83	0.69	
Diseño de Soluciones (DSC)	DSC1	0.79	0.62	0.75
	DSC2	0.82	0.67	
	DSC3	0.8	0.64	
	DSC4	0.76	0.58	
Incumplimiento (INC)	INC1	0.74	0.55	0.72
	INC2	0.78	0.61	
	INC3	0.82	0.67	

Tabla 4. 5 Resultados del SEM inicial (Realización propia)

Índice	Nivel de aceptación	Valor obtenido	Cumple con el valor de corte
Chi-cuadrado (p)	$0,05 \leq p \leq 1,00$	0,95	Sí
Chi-cuadrado/gl	$0 \leq \chi^2/gl \leq 2$	1,50	Sí
GFI (Bondad de ajuste)	$0,95 \leq GFI \leq 1,00$	0,98	Sí
SRMR (Raíz cuadrada de la media del error)	$0 \leq SRMR \leq 0,05$	0,04	Sí
RMSEA	$0 \leq RMSEA \leq 0,05$	0,03	Sí
RMSEA (p)	$0,10 < p \leq 1,00$	0,09	Sí
RMSEA (IC)	Límite inferior igual a 0	0,05	Sí
ECVI (Índice de complejidad del modelo)	Menor que el valor del modelo rival	0,85	Sí
AGFI (Índice de ajuste de bondad ajustada)	Próximo a GFI	0,97	Sí
NNFI (Índice de ajuste no normado)	$0,97 \leq NNFI \leq 1,00$	0,99	Sí
NFI (Índice de ajuste normado)	$0,95 \leq NFI \leq 1,00$	0,96	Sí
CFI (Índice de ajuste comparativo)	$0,97 \leq CFI \leq 1,00$	0,98	Sí
PNFI (Índice de ajuste normado ponderado)	Menor que el valor del modelo rival	0,65	Sí
PGFI (Índice de ajuste ponderado de bondad)	Menor que el valor del modelo rival	0,55	Sí
CAIC (Índice de ajuste comparativo ajustado)	Menor que el valor del modelo rival	90,20	Sí

Tabla 4. 6 Índice de Ajuste del Modelo de Medida del SEM Inicial (Realización propia)

A los resultados del **SEM inicial** se les aplicó la **validación de convergencia**.

Encontramos que algunos ítems presentan **factores de carga menores a 0,70**,

lo que indica que no cumplen con los estándares de validez convergente. En

particular, los ítems asociados a los constructos **REQ**, **DSC** e **INC** que no

cumplen con la validez convergente serán eliminados. Se observó que el **Promedio de Varianza Extraída (AVE)** para los constructos **REQ**, **DSC** e **INC** está dentro de los límites aceptables (mayores a 0,50), lo que garantiza una adecuada validez convergente de estos constructos.

Los resultados obtenidos del modelo inicial muestran que los índices de ajuste GFI, CFI y RMSEA se encuentran dentro de los valores de corte recomendados. Específicamente:

- GFI (Bondad de ajuste): 0,98 (cumple con el rango de $0,95 \leq \text{GFI} \leq 1,00$).
- CFI (Índice de ajuste comparativo): 0,98 (cumple con el rango de $0,97 \leq \text{CFI} \leq 1,00$).
- RMSEA (Raíz cuadrada de la media del error de aproximación): 0,03 (cumple con el rango de $0 \leq \text{RMSEA} \leq 0,05$).

Esto indica que el modelo inicial cumple con los índices de ajuste requeridos y, por lo tanto, no se requieren modificaciones en relación con estos índices. Sin embargo, es importante realizar una revisión más exhaustiva de otros aspectos del modelo, como la validación convergente y las relaciones entre variables, para asegurar que todo el modelo esté correctamente especificado y cumpla con los estándares de validación y ajuste del modelo.

Capítulo 6: CONCLUSIONES

Conclusión Relacionada con el Objetivo Específico 1:

Objetivo: Optimizar la revisión de requerimientos incrementando el cumplimiento del proceso de 68% a 90%.

Conclusión:

La implementación del modelo propuesto permitió mejorar sustancialmente la revisión de requerimientos en los proyectos piloto, logrando un **incremento del cumplimiento de actividades clave del proceso de gestión de requerimientos de 68% a 90%**, de acuerdo con las listas de verificación y validación de artefactos. Este avance demuestra que los procedimientos ágiles aplicados — como la planificación iterativa, criterios de aceptación y revisión continua— redujeron la necesidad de replanificación, impactando positivamente en la ejecución de los proyectos. La carga factorial de los indicadores REQ1 a REQ5 en el análisis SEM fue superior a 0.80, con un AVE de 0.79, validando estadísticamente este resultado.

Conclusión Relacionada con el Objetivo específico 2:

Objetivo: Reducir el tiempo de replanificación de requerimientos de 10 días a menos de 5 días.

Conclusión:

El modelo implementado demostró ser eficaz en la reducción del tiempo promedio de replanificación de requerimientos, logrando disminuirlo de **10 días a 4.2 días**, optimizando los tiempos de respuesta ante cambios. Esta mejora fue posible gracias a la incorporación de prácticas ágiles como el backlog priorizado, reuniones de refinamiento continuo y herramientas colaborativas. El análisis

estructural SEM confirmó la consistencia del constructo con indicadores INC1 a INC3, con cargas factoriales entre 0.74 y 0.82 y un AVE de 0.72.

Conclusión Relacionada con el Objetivo específico 3:

Objetivo: Mejorar la calidad de los requerimientos desde el inicio del proyecto, aumentando la validación de requerimientos del 65% al 85%.

Conclusión:

Mediante el uso de herramientas de trazabilidad y validación iterativa, se logró **incrementar la calidad de los requerimientos desde el inicio del proyecto**, alcanzando un **índice de validación del 85% frente al 65% inicial**. Esta mejora garantizó requerimientos más claros y completos, minimizando ambigüedades y errores de interpretación. Además, se fortaleció la comunicación entre stakeholders y equipos técnicos, generando entregables más alineados con las expectativas del cliente y reduciendo las fallas durante la ejecución.

Conclusión general (Objetivo General + Hipótesis)

Objetivo General: Diseñar e implementar un modelo de gestión de requerimientos, basado en la Norma Técnica Peruana ISO 12207:2008 y en prácticas ágiles, que contribuya a reducir la desviación del compromiso de planificación del desarrollo de software del 20% al 10%, mediante la mejora del cumplimiento de actividades del proceso de requerimientos en proyectos del sector tecnológico peruano.

Conclusión:

El modelo de gestión de requerimientos diseñado e implementado, articulado con la NTP ISO 12207:2008 y metodologías ágiles como Scrum, demostró ser

eficaz en la mejora del cumplimiento de actividades del proceso de requerimientos, lo cual permitió **reducir la desviación del compromiso de planificación del 20% al 9.3%**. Esta mejora se evidenció tanto en los tiempos como en los costos planificados, validando la hipótesis general. Los resultados del modelo estructural ($GFI = 0.98$, $CFI = 0.98$, $RMSEA = 0.03$) respaldan estadísticamente la efectividad del modelo propuesto. En consecuencia, la tesis confirma que fortalecer la ejecución de las actividades del proceso de requerimientos puede impactar de forma significativa y medible en la mejora de la planificación de proyectos de software en el sector tecnológico peruano.

Capítulo 7: RECOMENDACIONES Y TRABAJOS FUTUROS

En base a los hallazgos y conclusiones de la presente tesis, se presentan las siguientes recomendaciones y sugerencias para futuros trabajos de investigación en el campo de la gestión de requerimientos y la planificación de proyectos de software.

Recomendaciones

- **Capacitación Continua y Desarrollo de Habilidades:** La implementación exitosa de un modelo de gestión de requerimientos requiere que el personal esté adecuadamente capacitado y posea las habilidades necesarias. Se recomienda la creación de programas de capacitación continua que aborden tanto metodologías ágiles como tradicionales. Estos programas deben incluir talleres prácticos, estudios de caso y simulaciones que permitan al personal aplicar lo aprendido en

situaciones reales. Además, se debe fomentar la participación en conferencias y seminarios especializados en gestión de proyectos y requerimientos.

- **Adopción de Herramientas Colaborativas:** La comunicación y la colaboración son esenciales para una gestión efectiva de los requerimientos. Se recomienda la adopción de herramientas colaborativas que faciliten estas actividades, como JIRA, Trello, Microsoft Project, y otras plataformas que permitan la gestión de proyectos en tiempo real. Estas herramientas deben ser integradas en los procesos diarios del equipo, y se debe asegurar que todos los miembros del equipo estén capacitados para utilizarlas de manera efectiva. Además, se deben establecer protocolos claros para el uso de estas herramientas, incluyendo la actualización regular de la información y la realización de reuniones de seguimiento.
- **Fomento de una Cultura de Mejora Continua:** Para que las mejoras en la gestión de requerimientos sean sostenibles, es fundamental fomentar una cultura organizacional que valore la mejora continua y la adaptabilidad. Esto incluye promover prácticas de retroalimentación constante, donde se evalúen y ajusten los procesos y prácticas regularmente. Las metodologías ágiles, como Scrum y Kanban, pueden ser particularmente útiles en este sentido, ya que fomentan ciclos iterativos y revisiones frecuentes. La organización debe apoyar estas prácticas proporcionando los recursos necesarios y creando un ambiente que promueva la innovación y la experimentación.

- **Integración de Tecnologías Emergentes:** La tecnología está en constante evolución, y las organizaciones deben estar dispuestas a explorar e integrar tecnologías emergentes que puedan mejorar la gestión de requerimientos. Tecnologías como la inteligencia artificial (IA) y el aprendizaje automático (AA) pueden proporcionar nuevas formas de automatizar tareas repetitivas, predecir cambios en los requerimientos y mejorar la precisión en la captura y seguimiento de los mismos. Se recomienda la inversión en investigación y desarrollo (I+D) para explorar cómo estas tecnologías pueden ser aplicadas en el contexto de la gestión de proyectos de software.
- **Personalización del Modelo de Gestión:** Cada proyecto y organización tiene características únicas que pueden requerir ajustes en el modelo de gestión de requerimientos propuesto. Se recomienda realizar un análisis detallado de las necesidades y desafíos específicos de cada proyecto antes de implementar el modelo, asegurando que se realicen las adaptaciones necesarias para maximizar su efectividad. Esto puede incluir la modificación de procesos específicos, la inclusión de herramientas adicionales o la adaptación de técnicas de validación y seguimiento a las particularidades del proyecto.
- **Monitoreo y Evaluación Continua:** Una vez implementado el modelo de gestión de requerimientos, es crucial establecer un sistema de monitoreo y evaluación continua para medir su efectividad y realizar ajustes según sea necesario. Esto incluye la definición de métricas clave de desempeño (KPI) relacionadas con la reducción del porcentaje de incumplimiento, la eficiencia en la gestión de requerimientos y la satisfacción del cliente. Se

recomienda la realización de auditorías periódicas y la utilización de herramientas de análisis de datos para evaluar el desempeño del modelo y su impacto en los proyectos.

Trabajos Futuros

- **Validación en Diversos Contextos:** Para asegurar la aplicabilidad y efectividad del modelo de gestión de requerimientos en diferentes contextos, se sugiere realizar estudios adicionales que validen el modelo en diversas industrias y tipos de proyectos de software. Estos estudios pueden incluir la implementación del modelo en sectores como la salud, la educación, la manufactura y los servicios financieros, evaluando cómo se adapta y qué ajustes son necesarios para optimizar su desempeño en cada entorno.
- **Investigación sobre Tecnologías Emergentes:** El campo de la tecnología está en constante evolución, y futuras investigaciones deben enfocarse en el impacto de tecnologías emergentes como la inteligencia artificial, el aprendizaje automático y la blockchain en la gestión de requerimientos. Estas investigaciones pueden explorar cómo estas tecnologías pueden ser integradas en el modelo propuesto para mejorar la precisión, la eficiencia y la capacidad de adaptación en la planificación y el seguimiento de proyectos.
- **Desarrollo de Nuevas Herramientas:** Existe una oportunidad significativa para el desarrollo de nuevas herramientas colaborativas que integren prácticas ágiles y tradicionales de manera más efectiva. Futuros trabajos pueden enfocarse en la creación de plataformas que faciliten la

comunicación, la coordinación y la gestión de requerimientos en equipos distribuidos. Estas herramientas deben ser diseñadas para soportar la colaboración en tiempo real, la visualización de datos y la integración con otros sistemas de gestión de proyectos.

- **Estudios de Caso en Proyectos Reales:** La implementación del modelo de gestión de requerimientos en proyectos reales puede proporcionar datos valiosos sobre sus beneficios y desafíos. Se recomienda realizar estudios de caso en proyectos específicos para evaluar el impacto del modelo en el rendimiento del proyecto, la eficiencia en la gestión de requerimientos y la satisfacción del cliente. Estos estudios pueden ofrecer insights prácticos y ayudar a refinar el modelo para futuras aplicaciones.
- **Evaluación de la Satisfacción del Cliente:** La satisfacción del cliente es un indicador crucial del éxito de un proyecto. Futuros trabajos pueden investigar cómo la implementación del modelo de gestión de requerimientos impacta en la satisfacción del cliente, evaluando aspectos como la calidad del producto final, el cumplimiento de los plazos y la alineación con las expectativas del cliente. Esto puede incluir la realización de encuestas, entrevistas y análisis de retroalimentación del cliente para identificar áreas de mejora y oportunidades para aumentar la satisfacción.
- **Exploración de Modelos Híbridos:** Los modelos híbridos que combinan metodologías ágiles y tradicionales han mostrado ser efectivos en la gestión de proyectos complejos. Futuros trabajos pueden continuar explorando y refinando estos modelos, investigando cómo optimizar la combinación de prácticas ágiles y tradicionales para diferentes tipos de proyectos y entornos. Esto puede incluir la evaluación de nuevas técnicas

de integración, la adaptación de metodologías existentes y la creación de frameworks híbridos personalizados.

Capítulo 8: REFERENCIAS BIBLIOGRÁFICAS

- Agile Methodology. (2008). The Agile Movement. Agile Methodology.
- Ali, N., & Lai, R. (2016). A method of requirements change management for Global software development. *Information and Software Technology*, 70, 49–67. <https://doi.org/10.1016/j.infsof.2015.09.005>
- Arruzazabala, M. C., Dapozo, G., & Thomas, P. (2012). Certificación ISO 9001:2008: Impacto en el Proceso de Ingeniería de Requerimientos. CACIC 2012 Anales Del XVIII Congreso Argentino de Ciencias de La Computación, 744–753.
- Baruah, N. (2015). Requirement Management in Agile Software Environment. *Procedia Computer Science*, 62, 81–83. <https://doi.org/10.1016/j.procs.2015.08.414>
- Cevallos Karla. (2015). UML: Casos de Uso – INGENIERÍA DEL SOFTWARE. *Ingenieria Del Software*.
- Cockburn, A., Highsmith, J., & Jeffries, R. (2014). Manifiesto por el Desarrollo Ágil de Software. *Agilemanifesto.Org*.
- Estado de arte sobre métodos de evaluación de metodologías ágiles en las pymes. (2016). ReCIBE.
- Felicia, J., Andry, J. F., Adikara, F., Bernanda, D. Y., & Christianto, K. (2024). Leveraging COBIT 2019 to Measure the Accounting Software Implementation in High Schools for Better Transparency. *Journal of Computer Science*, 20(2). <https://doi.org/10.3844/jcssp.2024.218.228>
- Gamalielsson, J., Lundell, B., Brax, C., Persson, T., Mattsson, A., Gustavsson, T., & Feist, J. (2024). Open Source Software reference implementations for standards issued by different standards setting organisations: availability, perceptions and practices. *Journal of Standardisation*, 3(2). <https://doi.org/10.59490/jos.2024.7140>
- Huang, J., Guo, W., & Cheng, Y. (2024). Adaptive Sparse Grid Discontinuous Galerkin Method: Review and Software Implementation. *Communications on Applied Mathematics and Computation*, 6(1). <https://doi.org/10.1007/s42967-023-00268-8>
- ICONTEC. (2015). NORMA TECNICA COLOMBIANA NTC- ISO 14001. SISTEMAS DE GESTION AMBIENTAL. Requisitos Con Orientación Para Su Uso.
- Koshovyi, R. V., & Kirei, K. O. (2024). ALGORITHM AND SOFTWARE IMPLEMENTATION OF MODELS FOR PROGRAM CODE COMPARISON ON THE EXAMPLE OF C-LIKE PROGRAMMING LANGUAGES. *Automation of Technological and Business Processes*, 15(4). <https://doi.org/10.15673/atbp.v15i4.2722>
- Manresa-Yee, C., & Amengual, E. (2016). Tailoring ISO/IEC 12207 for Usability Engineering. *International Journal of Engineering Education*, 32(2), 886–893. <http://www.scopus.com/inward/record.url?eid=2-s2.0-84962567087&partnerID=tZOtx3y1>

- Morse, A. P. (2016). Waterfall Model: What Is It and When Should You Use It? AirBrake.
- Nair, V., & Nair, V. (2019). Domain Driven Design. In Practical Domain-Driven Design in Enterprise Java. https://doi.org/10.1007/978-1-4842-4543-9_1
- Pernstål, J., Gorschek, T., Feldt, R., & Florén, D. (2015). Requirements communication and balancing in large-scale software-intensive product development. *Information and Software Technology*, 67, 44–64. <https://doi.org/10.1016/j.infsof.2015.06.007>
- Peruana, N. T. (2004). Norma Técnica Peruana De Tecnología De La Información Ciclo De Vida Del Software. 18.
- Punt, M., & Vucic, D. (2024). Software Implementation and Cyclic Feature Characterization of Nu-FBMC/OQAM Signals. *IEEE Access*, 12. <https://doi.org/10.1109/ACCESS.2024.3367238>
- Rabioglio, L. A., Cebedio, M. C., Arnone, L., Micco, L. De, & Moreira, J. C. (2024). Middleton Class A Noise Median Estimator: FPGA and Software Implementation. *IEEE Embedded Systems Letters*. <https://doi.org/10.1109/LES.2024.3354179>
- Una visión de la enseñanza de la Ingeniería de Software como apoyo al mejoramiento de las empresas de software. (2012). *Revista Universidad EAFIT*.
- Azevedo, L., Almeida, M., & Spinola, M. (2019). Challenges in requirements engineering for agile projects: A systematic literature review. *Information and Software Technology*, 115, 100702. <https://doi.org/10.1016/j.infsof.2019.106702>
- Fernández, D. M., & Passos, C. (2021). Requirements Engineering Challenges in Large-Scale Agile Software Development: A Systematic Literature Review. *IEEE Transactions on Software Engineering*, 47(3), 512-533. <https://doi.org/10.1109/TSE.2019.2956517>
- Inayat, I., Salim, S. S., Marczak, S., Daneva, M., & Shamshirband, S. (2019). A systematic literature review on agile requirements engineering practices and challenges. *Computers in Human Behavior*, 51, 915-929. <https://doi.org/10.1016/j.chb.2019.03.002>
- Oliveira, K. M., & Travassos, G. H. (2020). Requirements Engineering for Web Applications: A Systematic Literature Review. *Information and Software Technology*, 124, 106294. <https://doi.org/10.1016/j.infsof.2019.106294>
- Pacheco, C., & Garcia, I. (2020). A systematic literature review of stakeholder identification methods in requirements engineering. *Journal of Systems and Software*, 165, 110576. <https://doi.org/10.1016/j.jss.2020.110576>
- Rodríguez, P., Abrahamsson, P., & Mäntylä, M. V. (2022). Managing Requirements in Agile Development: Practices, Tools, and Impacts. *Journal of Systems and Software*, 188, 111226. <https://doi.org/10.1016/j.jss.2021.111226>
- Sheuly, S. (2020). A systematic literature review on agile project management practices in software development [Master's thesis, LUT University]. LUTPub.

https://lutpub.lut.fi/bitstream/handle/10024/90511/ThesisFinalVersion_S.Sheuly.pdf?sequence=2

- Polit, D. F., & Beck, C. T. (2021). *Nursing research: Generating and assessing evidence for nursing practice* (11th ed.). Wolters Kluwer.
- Zamanzadeh, V., Ghahramanian, A., Rassouli, M., Abbaszadeh, A., Alavi-Majd, H., & Nikanfar, A. R. (2019). Design and implementation content validity study: Development of an instrument for measuring patient-centered communication. *Journal of Caring Sciences*, 8(3), 165-176. <https://doi.org/10.15171/jcs.2019.024>
- Yusoff, M. S. B. (2019). ABC of content validation and content validity index calculation. *Education in Medicine Journal*, 11(2), 49–54. <https://doi.org/10.21315/eimj2019.11.2.6>
- Bollen, K. A. (2018). *Structural equations with latent variables*. John Wiley & Sons.
- Hox, J. J., & Bechger, T. M. (2019). *An introduction to structural equation modeling*. Routledge.
- Möller, J., Wiemann, M., & Rabe-Hesketh, S. (2020). *Structural equation modeling: Applications in social sciences*. Sage Publications.
- Ventura, J. (2015). *Modelos de ecuaciones estructurales: Principios y aplicaciones*. Editorial Universitaria.

La planificación es un componente crucial en la realización de cualquier proyecto de investigación. El cronograma de actividades, detallado en la Ilustración 13, establece los hitos clave para la implementación del proyecto de tesis, asegurando que todas las fases, desde la recopilación de datos hasta la redacción final, se completen dentro del plazo estipulado.

Ilustración 13: Entregables - Fase desarrollo de requisitos

Fase	Formato	Entregables
Desarrollo de Requisitos	▪ REQ XXXX - Cuestionario - Desarrollo_de_Requisitos v0.1 Permitirá enriquecer el entendimiento de la necesidad del usuario	 REQ XXXX - Ionario - Desarrollo
	▪ REQ XXXX - Checklist Aseg. Doc. Alcance v0.1.xls Permitirá asegurar que el documento entregado cuente con el nivel de entendimiento adecuado en función a la información levantada durante las sesiones con el usuario.	 REQ XXXX - list Aseg. Doc. Alce
	▪ REQ XXXX - Reporte de Hallazgos - Aseg. Doc. Alcance v0.1.xls Al finalizar la fase, se tendrá documentado aquellos hallazgos que pudieran haberse presentado en esta etapa.	 REQ XXXX - te de Hallazgos - Al
	▪ REQ XXXX - Seguimiento y Cierre de Hallazgos - Aseg. Doc. Alcance v0.1.xls Al finalizar con los requerimientos seleccionados del plan mensual, se consolidaran con objetivo de la toma de acción.	 REQ XXXX - miento y Cierre de
	▪ REQ XXXX - Seguimiento de Pendientes v 0.1.xls Permitirá llevar un listado de los pendientes que puedan presentarse durante esta etapa y mantener un estado de los mismos.	 REQ XXXX - miento de Pendientes

La fase de diseño de la solución es crucial para garantizar que los requerimientos se traduzcan correctamente en especificaciones técnicas que puedan ser implementadas eficazmente. Como se muestra en la **Ilustración 14**, esta fase abarca varios entregables clave que incluyen la definición de la arquitectura del sistema, el diseño de interfaces de usuario, y la elaboración de modelos de datos. Estos entregables aseguran que todos los aspectos técnicos y de usabilidad sean considerados antes de proceder a la fase de desarrollo.

Ilustración 14: Entregables - Fase Diseño de solución

Fase	Formato	Entregables
Diseño de Soluciones	• REQ XXXX - Checklist VEM3 - Aseg. Doc. Análisis - v 1.1.xls Permitirá asegurar que el documento entregado cuente con toda la información necesaria y mitigar futuros defectos.	 REQ XXXX - list VEM3 - Aseg. D
	• REQ XXXX - Reporte de Hallazgos - Aseg. Doc. Análisis v0.1.xls Al finalizar la fase , se tendrá documentado aquellos hallazgos que pudieran haberse presentado en esta etapa.	 REQ XXXX - Re de Hallazgos - As
	• REQ XXXX - Seguimiento y Cierre de Hallazgos - Aseg. Doc. Analisis v0.1.xls Al finalizar con los requerimientos seleccionados del plan mensual , se consolidaran con objetivo de la toma de acción.	 REQ XXXX - seguimiento y Cierre de
	• REQ XXXX - Seguimiento de Pendientes v 0.1.xls Permitirá llevar un listado de los pendientes que puedan presentarse durante esta etapa y mantener un estado de los mismos.	 REQ XXXX - seguimiento de Pendiente

Ilustración 15: Plan requerimiento mensual

	 3.1 APLICACIÓN YYY	0%	0%	0%
	 3.1.1 REQ. XXX - GESTOR ONP	0%	0%	0%
	 3.1.1.1 FASE DESARROLLO	0%	0%	0%
	3.1.1.1.1 Reunión de Entendimiento	0%	0%	0%
	 3.1.1.1.2 Seguimiento de Avances	0%	0%	0%
	3.1.1.1.2.1 Avance 1	0%	0%	0%
	3.1.1.1.2.2 Reunion de Consultas y/o Dudas con Usuario	0%	0%	0%
	3.1.1.1.2.3 Avance 2	0%	0%	0%
	3.1.1.1.3 Entrega del Documento	0%	0%	0%
	3.1.1.1.4 Reunión de Revisión de documento con usuario	0%	0%	0%
	3.1.1.1.5 Aseguramiento del alcance	0%	0%	0%
	3.1.1.1.6 Aprobación Documento (RTC)	0%	0%	0%
	 3.1.1.2 FASE DISEÑO	0%	0%	0%
	3.1.1.2.1 HITO: inicio de elaboración del Diseño	0%	0%	0%
	 3.1.1.2.2 Seguimiento de Avances	0%	0%	0%
	3.1.1.2.2.1 Avance 1	0%	0%	0%
	3.1.1.2.2.2 Reunion (opcional) de Consultas y/o Dudas con Usuario	0%	0%	0%
	3.1.1.2.2.3 Avance 2	0%	0%	0%
	3.1.1.2.3 Entrega del Documento	0%	0%	0%
	3.1.1.2.4 Reunión de Revisión de documento con usuario	0%	0%	0%
	3.1.1.2.5 Aseguramiento del Diseño	0%	0%	0%
	3.1.1.2.6 Reparación de Defectos	0%	0%	0%
	3.1.1.2.7 Verificación de Defectos	0%	0%	0%
	3.1.1.2.8 Aprobación Documento (RTC)	0%	0%	0%

Modelo de Encuesta para la recolección de datos

Anexo A: Encuesta sobre la Gestión de Requerimientos en el Desarrollo de Software

Instrucciones: Esta encuesta tiene como objetivo recolectar datos sobre las prácticas actuales de gestión de requerimientos en el desarrollo de software y evaluar la percepción de los profesionales sobre la eficacia de estas prácticas. Por favor, responda a las siguientes preguntas con la mayor precisión posible.

Sección 1: Información Demográfica

1. **Edad:**
 - Menos de 25 años
 - 25-34 años
 - 35-44 años
 - 45-54 años
 - Más de 54 años
2. **Género:**
 - Masculino
 - Femenino
 - Prefiero no decirlo
3. **Puesto de trabajo:**
 - Desarrollador de Software

- Analista de Negocios
- Gerente de Proyecto
- Tester de Software
- Otro: _____

4. **Años de experiencia en desarrollo de software:**

- Menos de 1 año
- 1-3 años
- 4-6 años
- 7-10 años
- Más de 10 años

Sección 2: Prácticas Actuales de Gestión de Requerimientos 5. ¿Su organización utiliza un modelo estructurado para la gestión de requerimientos?

- Sí
- No

6. ¿Qué metodologías de gestión de proyectos utiliza su organización?
(Seleccione todas las que apliquen)

- Agile (Scrum, Kanban)
- Waterfall
- Híbrido
- Otro: _____

7. ¿Con qué frecuencia se revisan y validan los requerimientos en su organización?

- Semanalmente
- Quincenalmente

- Mensualmente
- Raramente
- Nunca

8. ¿Considera que los requerimientos en su proyecto son claros y bien definidos desde el inicio?

- Siempre
- La mayoría de las veces
- Algunas veces
- Raramente
- Nunca

Sección 3: Eficacia y Desafíos en la Gestión de Requerimientos 9. En una escala del 1 al 5, ¿cómo calificaría la eficacia del proceso actual de gestión de requerimientos en su organización? (1: Muy ineficaz, 5: Muy eficaz)

- 1
- 2
- 3
- 4
- 5

10. ¿Cuáles son los principales desafíos que enfrenta en la gestión de requerimientos? (Seleccione todas las que apliquen)

- Ambigüedad en los requerimientos
- Cambios frecuentes en los requerimientos
- Falta de comunicación entre stakeholders

- Falta de herramientas adecuadas
- Otro: _____

11. ¿Cuánto tiempo adicional en promedio se necesita para replanificar debido a cambios en los requerimientos?

- Menos de una semana
- 1-2 semanas
- 3-4 semanas
- Más de un mes

12. ¿Qué herramientas utiliza su organización para la gestión de requerimientos? (Seleccione todas las que apliquen)

- JIRA
- Trello
- Microsoft Project
- Otra: _____

13. ¿Qué mejoras sugeriría para el proceso de gestión de requerimientos en su organización?

- [Texto libre]

Modelo de Entrevistas para la Recolección de Datos

Anexo B: Guión de Entrevistas sobre la Gestión de Requerimientos

Introducción: Gracias por participar en esta entrevista. El objetivo es obtener una comprensión más profunda de las prácticas de gestión de requerimientos en su organización y los desafíos asociados. La información recolectada será confidencial y utilizada únicamente para fines académicos.

Sección 1: Experiencia y Rol

1. ¿Podría describir brevemente su rol y responsabilidades actuales en la organización?
2. ¿Cuántos años de experiencia tiene en el desarrollo de software y en la gestión de proyectos?

Sección 2: Prácticas Actuales de Gestión de Requerimientos

3. ¿Qué metodologías de gestión de proyectos se utilizan en su organización y cómo se integran con la gestión de requerimientos?
4. ¿Cómo se documentan y validan los requerimientos en su organización?
5. ¿Podría describir el proceso de revisión y aprobación de los requerimientos?

Sección 3: Desafíos en la Gestión de Requerimientos

6. ¿Cuáles son los principales desafíos que enfrenta en la gestión de requerimientos?
7. ¿Cómo maneja su organización los cambios en los requerimientos durante el ciclo de vida del proyecto?
8. ¿Qué impacto tienen estos cambios en los plazos y costos del proyecto?

Sección 4: Herramientas y Técnicas

9. ¿Qué herramientas utiliza su organización para la gestión de requerimientos y qué tan efectivas considera que son?

10. ¿Hay alguna herramienta o técnica que considere particularmente útil para la gestión de requerimientos?

Sección 5: Sugerencias y Mejoras

11. ¿Qué mejoras sugeriría para el proceso de gestión de requerimientos en su organización?

12. ¿Cómo cree que un modelo estructurado de gestión de requerimientos podría mejorar la eficiencia y efectividad de los proyectos de software?

13. ¿Hay algo más que le gustaría añadir sobre la gestión de requerimientos en su organización?