



Universidad  
Nacional  
de Loja

# Universidad Nacional de Loja

Facultad de la Energía, las Industrias y los Recursos

Naturales no Renovables

Maestría en Ingeniería en Software

Desarrollo de un marco de trabajo para la gestión de riesgos de calidad en proyectos de desarrollo de software ágil, basado en la ISO 31000 y gestionados con Scrum

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

**AUTOR:**

Francisco Javier Alvarez Pineda

**DIRECTOR:**

Ing. Wilman Patricio Chamba Zaragocín, Mg. Sc.

Loja - Ecuador

2025

Educamos para **Transformar**

## **Certificación**

Loja, 20 de enero de 2025

Ing. Wilman Patricio Chamba Zaragocín, Mg. Sc.

**DIRECTOR DEL TRABAJO DE TITULACIÓN**

### **CERTIFICO:**

Que he revisado y orientado todo proceso de la elaboración del Trabajo de Titulación denominado: **Desarrollo de un marco de trabajo para la gestión de riesgos de calidad en proyectos de desarrollo de software ágil, basado en la ISO 31000 y gestionados con Scrum**, previo a la obtención del título de **Magíster en Ingeniería en Software**, de autoría del estudiante **Francisco Javier Alvarez Pineda**, con cédula de identidad Nro. **1103635643**, una vez que el trabajo cumple con todos los requisitos exigidos por la Universidad Nacional de Loja para el efecto, autorizo la presentación para la respectiva sustentación y defensa.

Ing. Wilman Patricio Chamba Zaragocín, Mg. Sc.

**DIRECTOR DEL TRABAJO DE TITULACIÓN**

## **Autoría**

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

**Firma:**

**Cédula de Identidad:** 1103635643

**Fecha:** 20/01/2025

**Correo electrónico:** fjalvarez@unl.edu.ec

**Teléfono:** 0997197893

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

Yo, **Francisco Javier Alvarez Pineda**, declaro ser autor del Trabajo de Titulación denominado: **Desarrollo de un marco de trabajo para la gestión de riesgos de calidad en proyectos de desarrollo de software ágil, basado en la ISO 31000 y gestionados con Scrum** como requisito para optar el título de **Magíster en Ingeniería en Software**, autorizo al sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos muestre la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Institucional.

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

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

Para constancia de esta autorización suscribo, en la ciudad de Loja, a los veinte días del mes de enero de dos mil veinticinco.

**Firma:**

**Autor:** Francisco Javier Alvarez Pineda

**Cédula de identidad:** 1103635643

**Dirección:** Cdla. Altos de Valle. Calle principal Franklin Hidalgo, calle secundaria Teresa Mora.

**Correo electrónico:** fjalvarez@unl.edu.ec

**Teléfono:** 0997197893

**DATOS COMPLEMENTARIOS:**

**Director del Trabajo de Titulación:** Ing. Wilman Patricio Chamba Zaragocín Mg. Sc.

## **Dedicatoria**

A mi esposa, mi madre, mi padre y a mi sobrina, quienes con su presencia física y espiritual constituyen el pilar más importante de mi vida.

*Francisco Javier Alvarez Pineda*

## **Agradecimiento**

A mi esposa, por su apoyo incondicional y sus palabras de motivación, que fueron fundamentales durante el desarrollo de mi Maestría. Al Ing. Wilman Chamba, director de este Trabajo de Titulación, quien, con su experiencia y guía, orientó de manera adecuada el desarrollo de este trabajo.

*Francisco Javier Alvarez Pineda*

## Índice de contenidos

<b>Portada .....</b>	<b>i</b>
<b>Certificación.....</b>	<b>ii</b>
<b>Autoría.....</b>	<b>iii</b>
<b>Carta de Autorización.....</b>	<b>iv</b>
<b>Dedicatoria .....</b>	<b>v</b>
<b>Agradecimiento.....</b>	<b>vi</b>
<b>Índice de contenidos .....</b>	<b>vii</b>
Índice de tablas .....	xi
Índice de figuras .....	xii
Índice de anexos .....	xiii
<b>1. Título .....</b>	<b>1</b>
<b>2. Resumen.....</b>	<b>2</b>
Abstract .....	3
<b>3. Introducción .....</b>	<b>4</b>
<b>4. Marco teórico.....</b>	<b>6</b>
4.1. Antecedentes.....	6
4.2. Bases teóricas. ....	6
4.2.1. Calidad de software.....	6
4.2.2. Calidad en USO. ....	7
4.2.3. Riesgos en desarrollo de software.....	7
4.2.4. Gestión de riesgos. ....	7
4.2.5. Marco de gestión de riesgos.....	8
4.2.6. Estándar ISO 31000. ....	8
4.2.7. Estándar ISO/IEC 25022.....	9
4.2.8. Scrum. ....	9
4.2.9. Design Science Research (DSR).....	10

4.2.10. Las herramientas de colaboración participativa.....	11
<b>5. Metodología .....</b>	<b>12</b>
5.1. Área de estudio.....	12
5.2. Procedimiento.....	12
5.2.1. Objetivo 1: Diseñar el marco de trabajo basado en la Revisión Bibliográfica y en el diagnóstico de principales riesgos que afectan la calidad de software en proyectos que usan Scrum.....	12
5.2.1.1. Revisión Bibliográfica (RB), aplicando los lineamientos principales de Barbara Kitchenham.....	13
5.2.1.1.1. Determinar los principales riesgos de calidad en proyectos Scrum.....	14
5.2.1.1.2. Documentar los riesgos de calidad encontrados.....	15
5.2.1.2. Caracterizar los riesgos en el desarrollo de software, mediante la norma ISO 31000.....	16
5.2.1.2.1. Seleccionar los puntos relevantes de la normativa aplicable al TT.....	16
5.2.1.2.2. Documentación de caracterización de los riesgos en base a la norma...	17
5.2.1.3. Proponer marco de trabajo.....	20
5.2.2. Objetivo 2: Evaluar el marco de trabajo propuesto utilizando la norma ISO/IEC 25022.....	20
5.2.2.1. Definición del modelo de calidad en Uso basado en la ISO/IEC 25022. .	21
5.2.2.1.1. Seleccionar los puntos relevantes de la normativa aplicable al TT.....	21
5.2.2.1.2. Definir la matriz de modelo de calidad en uso basado en la ISO/IEC 25022.....	22
5.2.2.2. Evaluación del marco de trabajo.....	22
5.2.2.3. Actualización de modelo final del marco de trabajo.....	22
<b>6. Resultados.....</b>	<b>23</b>
6.1. Objetivo 1: Diseñar el marco de trabajo basado en la Revisión Bibliográfica y en el diagnóstico de principales riesgos que afectan la calidad de software en proyectos que usan Scrum.....	23

6.1.1.	Revisión Bibliográfica (RB), aplicando los lineamientos principales de Barbara Kitchenham. ....	23
6.1.1.1.	Determinar los principales riesgos de calidad en proyectos Scrum.....	24
6.1.1.2.	Documentar los riesgos de calidad encontrados. ....	26
6.1.2.	Caracterizar los riesgos en el desarrollo de software, mediante la norma ISO 31000.....	29
6.1.2.1.	Seleccionar los puntos relevantes de la normativa aplicable al TT. ....	29
6.1.2.2.	Documentación de caracterización de los riesgos en base a la norma.....	31
6.1.2.3.	Análisis y evaluación de riesgos. ....	32
6.1.2.4.	Tratamiento de riesgos de calidad.....	36
6.1.3.	Proponer el marco de trabajo. ....	44
6.1.3.1.	Definición de la estructura del marco de trabajo. ....	44
6.1.3.2.	Diseño y desarrollo del artefacto. ....	44
6.1.3.3.	Documentación de riesgos en el marco de trabajo.....	45
6.2.	Objetivo 2: Evaluar el marco de trabajo propuesto utilizando la norma ISO/IEC 25022. ....	54
6.2.1.	Definición del modelo de calidad en Uso basado en la ISO/IEC 25022. ...	54
6.2.1.1.	Seleccionar los puntos relevantes de la normativa aplicable al TT. ....	54
6.2.1.2.	Definir la matriz de modelo de calidad en uso basado en la ISO/IEC 25022. ....	55
6.2.2.	Evaluación del marco de trabajo.....	58
6.2.2.1.	Aplicación de la matriz de calidad en Uso.....	58
6.2.2.2.	Medición e interpretación de la calidad en Uso.....	60
6.2.3.	Actualización de modelo final del marco de trabajo.....	62
<b>7.</b>	<b>Discusión .....</b>	<b>65</b>
	Objetivo 1: Diseñar el marco de trabajo basado en la Revisión Bibliográfica y en el diagnóstico de principales riesgos que afectan la calidad de software en proyectos que usan Scrum. ....	65

Objetivo 2: Evaluar el marco de trabajo propuesto utilizando la norma ISO/IEC 25022. ....	66
<b>8. Conclusiones .....</b>	<b>69</b>
<b>9. Recomendaciones .....</b>	<b>71</b>
<b>10. Bibliografía .....</b>	<b>72</b>
<b>11. Anexos .....</b>	<b>75</b>

## Índice de tablas

<b>Tabla 1.</b> Criterios de inclusión, exclusión y pregunta de calidad de software. ....	23
<b>Tabla 2.</b> Riesgos de calidad de software. ....	24
<b>Tabla 3.</b> Riesgos de calidad de software afinados y agrupados por categoría. ....	26
<b>Tabla 4.</b> Categorías de riesgos de calidad. ....	28
<b>Tabla 5.</b> Factores comunes de proyectos Scrum. ....	31
<b>Tabla 6.</b> Datos tabulados de la encuesta. ....	33
<b>Tabla 7.</b> Riesgos críticos que impactan en la calidad del producto. ....	34
<b>Tabla 8.</b> Riesgo de calidad propuesto por el tesista. ....	35
<b>Tabla 9.</b> Tabla de equivalencia de riesgos. ....	35
<b>Tabla 10.</b> Caracterización de riesgos de calidad. ....	38
<b>Tabla 11.</b> Marco de trabajo de riesgos de calidad. ....	46
<b>Tabla 12.</b> Características de calidad seleccionadas para evaluar el artefacto “Norma ISO/IEC 25022”. ....	54
<b>Tabla 13.</b> Preguntas por subcaracterística para evaluar el artefacto. ....	55

## Índice de figuras

<b>Figura 1.</b> Procedimiento de ejecución del objetivo 1.....	13
<b>Figura 2.</b> Fase de ejecución en herramienta Parsifal.....	14
<b>Figura 3.</b> Riesgos de calidad detectados en la RB.....	15
<b>Figura 4.</b> Riesgos y categorías de calidad depurados.....	16
<b>Figura 5.</b> Actividades ejecutadas de la norma ISO 31000. ....	17
<b>Figura 6.</b> Riesgos de calidad priorizados. ....	19
<b>Figura 7.</b> Procedimiento de ejecución del objetivo 2.....	21
<b>Figura 8.</b> Características de la normativa ISO/IEC 25022.....	21
<b>Figura 9.</b> Principios, marco de referencia, proceso.....	30
<b>Figura 10.</b> Relación entre categorías de riesgos y contexto de Scrum.....	33
<b>Figura 11.</b> Análisis de probabilidad e impacto.....	34
<b>Figura 12.</b> Estructura del marco de trabajo propuesto. ....	45
<b>Figura 13.</b> Ponderación de características para evaluar la calidad en uso.....	57
<b>Figura 14.</b> Ponderación de características para evaluar la calidad en uso R01.....	58
<b>Figura 15.</b> Ponderación de características para evaluar la calidad en uso R02.....	59
<b>Figura 16.</b> Ponderación de características para evaluar la calidad en uso R03.....	59
<b>Figura 17.</b> Ponderación de características para evaluar la calidad en uso R04.....	60
<b>Figura 18.</b> Ponderación de características para evaluar la calidad en uso R05.....	60
<b>Figura 19.</b> Resultados obtenidos en el taller de evaluación de la calidad en uso.....	61
<b>Figura 20.</b> Riesgos y oportunidades asociados con el nivel de calidad para cada riesgo. ....	61
<b>Figura 21.</b> Términos clave para comprender el marco de trabajo.....	63
<b>Figura 22.</b> Marco de trabajo propuesto para el riesgo 01.....	64

## Índice de anexos

<b>Anexo 1.</b> Revisión Bibliográfica.....	75
<b>Anexo 2.</b> Definición de probabilidad e impacto de riesgos de calidad.....	93
<b>Anexo 3.</b> Taller de evaluación del marco de trabajo.....	100
<b>Anexo 4.</b> Versión final del marco de trabajo propuesto.....	105
<b>Anexo 5.</b> Certificado de traducción del resumen. ....	112

## **1. Título**

**Desarrollo de un marco de trabajo para la gestión de riesgos de calidad en proyectos de desarrollo de software ágil, basado en la ISO 31000 y gestionados con Scrum.**

## 2. Resumen

Se diseñó un marco de trabajo para la gestión de riesgos de calidad en proyectos de desarrollo de software que utilizan Scrum, tomando como referencia la norma ISO 31000 para la gestión de riesgos y la norma ISO/IEC 25022 para evaluar la calidad en uso. El marco se estructuró a partir de una Revisión Bibliográfica (RB), definición de riesgos prioritarios, construcción del marco de trabajo y la validación de su efectividad mediante talleres prácticos. Con la RB, se identificaron 187 riesgos que afectan directamente a la calidad o que podrían afectar a la calidad en proyectos Scrum, de los cuales, 22 se clasificaron como riesgos que afectan directamente a la calidad del software; estos fueron afinados quedando 14 y priorizados a través de una encuesta realizada a profesionales de desarrollo de software, seleccionándose los 5 riesgos más relevantes: la falta de aprendizaje y prevención de errores, gestión ineficaz de la calidad por falta de colaboración, retroalimentación incompleta de los stakeholders, aumento de defectos por inexperiencia del equipo, gestión inadecuada de la deuda técnica. El marco propuesto se compone de tres secciones: Información del Riesgo (nombre, categoría, descripción, probabilidad e impacto), Identificación del Riesgo (evento, causa, consecuencia) y Mitigación del Riesgo en los eventos de Scrum (Sprint Planning, Daily Scrum, Sprint Review, Sprint Retrospective). Para validar el marco, se realizó un taller con 17 estudiantes de ciclos superiores de la carrera Computación, utilizando las características de la norma ISO/IEC 25022 como guía para evaluar su calidad en uso; en este taller se reportaron observaciones que se aplicaron al marco para fortalecerlo, mejorando su desempeño en términos de eficacia y satisfacción. El Trabajo de Titulación demuestra que es posible diseñar un marco de trabajo para gestionar riesgos de calidad de software basado en normas internacionales como ISO 31000 e ISO/IEC 25022.

**Palabras clave:** Riesgos de calidad de software, Scrum, ISO 31000, ISO/IEC 25022, marco de trabajo.

## **Abstract**

A framework was created for managing quality risks in software development projects that use Scrum, referencing the ISO 31000 standard for risk management and the ISO/IEC 25022 standard for assessing quality in use. The framework was developed based on a bibliographic review (BR), the identification of priority risks, the construction of the framework itself, and the validation of its effectiveness through practical workshops. The bibliographic review identified 187 risks that directly impact quality or have the potential to affect quality in Scrum projects. Among these, 22 risks were classified as having a direct effect on software quality; these were refined leaving 14 and prioritized through a survey of software development professionals, selecting the 5 most relevant risks: lack of learning and error prevention, ineffective quality management due to lack of collaboration, incomplete feedback from stakeholders, increased defects due to team inexperience, inadequate management of technical debt. The proposed framework is composed of three sections: Risk Information (name, category, description, probability, and impact), Risk Identification (event, cause, consequence), and Risk Mitigation in Scrum events (Sprint Planning, Daily Scrum, Sprint Review, Sprint Retrospective). To validate the framework, a workshop was held with 17 senior students from a Computer Science program. The characteristics of ISO/IEC 25022 were used as a guide to evaluate the framework's quality in use. During the workshop, observations were gathered and applied to enhance the framework, resulting in improved performance regarding effectiveness and user satisfaction. The thesis demonstrates the possibility of designing a framework for managing software quality risks based on international standards such as ISO 31000 and ISO/IEC 25022.

**Key words:** Software Quality Risks, Scrum, ISO 31000, ISO/IEC 25022, framework.

### 3. Introducción

Debido a la naturaleza ágil de los proyectos que se gestionan con el marco de trabajo Scrum, donde los Sprints (ciclos de trabajo) son cortos y altamente dependientes de la colaboración del equipo, se presentan riesgos que pueden comprometer la calidad del producto final, una de las causas puede ser las limitaciones de tiempo dentro de los Sprints, ya que el tiempo asignado para construcción y pruebas es reducido, lo que dificulta las revisiones del software y genera una mayor carga de trabajo y presión en el equipo de desarrollo [1]. Reconocer estos riesgos de calidad en los equipos de Scrum pueden mejorar el éxito general y la eficiencia de los proyectos de desarrollo de software.

En entornos ágiles, los profesionales que usan Scrum no son plenamente conscientes de los riesgos. Sin embargo, implícitamente siguen el proceso de gestión de riesgos porque asumen que ejecutar el ciclo de vida del desarrollo del software minimiza los efectos inesperados en un proyecto de software [2], muchos de estos riesgos pueden afectar a la calidad del software de manera directa o indirecta. En base a este contexto se plantea como objetivo general “Desarrollar un marco de trabajo para la gestión de riesgos de calidad de software en proyectos de desarrollo ágil, basado en la ISO 31000 y gestionados con Scrum”. La importancia de este marco de trabajo está en que especifica directamente 5 riesgos de calidad que se pueden presentar en los proyectos con Scrum, proporcionando adicionalmente criterios para identificarlos y acciones de mitigación que se pueden aplicar durante los eventos que usa dentro de proyectos de esta índole. En base a lo indicado, no es necesario realizar un levantamiento de riesgos, únicamente se tienen que aplicar las acciones de mitigación en caso de que se determine que el riesgo está presente en el proyecto.

El presente Trabajo de Titulación (TT) propone dar una respuesta a la pregunta de investigación: “¿Qué riesgos de calidad de software son más críticos en los proyectos de desarrollo gestionados con Scrum y cómo puede un marco de trabajo basado en la norma ISO 31000 y la ISO 25022 mitigar los 5 riesgos de mayor impacto al proyecto?”. Para responder la pregunta se establecieron y desarrollaron dos objetivos específicos: “Diseñar el marco de trabajo basado en la Revisión Bibliográfica y en el diagnóstico de principales riesgos que afectan la calidad de software en proyectos que usan Scrum” y “Evaluar el marco de trabajo propuesto utilizando la norma ISO/IEC 25022”

El Trabajo de Titulación está organizado con las siguientes secciones: Marco teórico, en el que se describen los antecedentes necesarios para comprender el TT; Metodología, en la que se detalla el área de estudio y los procedimientos que se utilizaron; Resultados, en esta sección se describen los datos obtenidos en base al desarrollo de los dos objetivos propuestos, generando las siguientes actividades: Revisión Bibliográfica, caracterizar los riesgos en el desarrollo de software mediante la norma ISO 31000, proponer un marco de trabajo aplicando Design Science Research, definición del modelo de calidad en uso basado en la ISO/IEC 25022, evaluación y actualización del marco de trabajo; Discusión, en la que se contrastan los resultados obtenidos en la construcción de la propuesta del marco de trabajo con estudios o propuestas relacionados a los riesgos de calidad en proyectos que usan Scrum; Conclusiones, en esta sección se da respuesta a la pregunta de investigación y se explican temas relevantes que se presentaron en el desarrollo de los objetivos; Recomendaciones, en la que se plantean sugerencias para futuros trabajos en base a los resultados obtenidos.

La presente propuesta del marco de trabajo representa un aporte significativo para la gestión ágil de riesgos en proyectos de desarrollo de software gestionados con el marco de trabajo Scrum, promoviendo la calidad, la eficiencia y la satisfacción. Esto debido a que no solo se identifican los riesgos, sino que también proporciona lineamientos para poder mitigarlos en los eventos (Reuniones de trabajo) de Scrum.

## **4. Marco teórico**

### **4.1. Antecedentes.**

Los riesgos de calidad asociados con Scrum se derivan principalmente de los desafíos en la calidad del código debido a las limitaciones de tiempo dentro de los Sprints, lo que dificulta las revisiones exhaustivas del código y potencialmente conduce a una mayor carga de trabajo y presión en el equipo de desarrollo [1]. Para ampliar la certeza del éxito de un proyecto de Scrum debemos analizar todos los posibles riesgos que afecten a la calidad, no solo los riesgos relacionados con el código. Al reconocer estos riesgos de calidad los equipos que usan Scrum pueden mejorar el éxito general y la eficiencia de sus proyectos de desarrollo de software.

### **4.2. Bases teóricas.**

La calidad de software es uno de los procesos considerados de soporte que debe ser transversal en los proyectos de desarrollo de software [3]. La ejecución de este proceso permite reducir la cantidad de defectos que se introducen en el desarrollo de software. En proyectos con Scrum tiene una gran importancia ya que su ejecución debe ser eficiente dado que el tiempo de pruebas es limitado. Para optimizar los tiempos de pruebas es importante conocer los riesgos de calidad que se pueden presentar y las definiciones que están relacionadas con la calidad del software y que son necesarias para comprender este Trabajo de Titulación.

#### **4.2.1. Calidad de software.**

En el estudio realizado por [4] define la calidad de software en términos de dos atributos principales: la conformidad a las necesidades del negocio y la ausencia de defectos. También, destaca que la calidad de software puede ser entendida como “el grado en que el sistema satisface las necesidades declaradas e implícitas de sus diversos interesados”, proporcionando así valor.

Los enfoques tradicionales también enfatizan las buenas prácticas de ingeniería de software y control de calidad (por ejemplo, integración continua, estándares de codificación, revisión por pares y automatización). En el Manifiesto Ágil, la calidad es un objetivo concomitante, no defendido explícitamente [4].

#### **4.2.2. Calidad en USO.**

“La calidad en uso es el grado en que un producto o sistema puede ser utilizado por usuarios específicos para satisfacer sus necesidades y lograr objetivos específicos con eficacia, eficiencia, satisfacción y ausencia de riesgos en contextos de uso específicos” [5].

#### **4.2.3. Riesgos en desarrollo de software.**

Los riesgos son eventos o condiciones inciertas que, si ocurren, pueden tener un efecto positivo o negativo en los objetivos de un proyecto [6]. En el contexto del desarrollo de software, particularmente en Scrum, los riesgos pueden surgir de diversas fuentes, como cambios en los requisitos del cliente, problemas de calidad, de comunicación, la complejidad técnica de las tareas. La naturaleza flexible de Scrum permite a los equipos adaptarse rápidamente a los cambios y a los riesgos emergentes, lo que es crucial para el éxito del proyecto.

#### **4.2.4. Gestión de riesgos.**

Por otro lado, la “Gestión de Riesgos” se refiere al proceso de identificar, evaluar y mitigar estos riesgos para minimizar su impacto en el proyecto [6]. Esto implica la planificación de estrategias para abordar los riesgos, así como la implementación de medidas para monitorear y controlar los riesgos a lo largo del ciclo de vida del desarrollo de software (SDLC).

En Scrum la gestión de riesgos se integra en las ceremonias y artefactos del marco de trabajo. Por ejemplo:

- Reuniones diarias: En estas reuniones, los miembros del equipo pueden discutir los obstáculos que enfrentan, lo que permite identificar riesgos potenciales de manera temprana.
- Revisión de Sprint: Al final de cada Sprint, el equipo revisa lo que se ha logrado y lo que no, lo que ayuda a identificar riesgos que pueden haber afectado el progreso.
- Retrospectivas: Permiten al equipo reflexionar sobre el Sprint anterior, identificar lecciones aprendidas y ajustar procesos para mitigar riesgos en futuros Sprints.

La implementación de gestión de riesgos dentro de Scrum puede ayudar a los equipos a categorizar y priorizar riesgos, así como a desarrollar estrategias de mitigación efectivas. Una

gestión de riesgos efectiva puede mejorar los resultados del proyecto, reduciendo costos y retrasos, y mejorando el rendimiento general.

#### **4.2.5. Marco de gestión de riesgos.**

“Las organizaciones inicialmente pueden enfocarse en la identificación del riesgo para tomar conciencia, y reaccionar ante la materialización de estos riesgos a medida que ocurren” [3]. El área de proceso Gestión de riesgos de CMMI (Capability Maturity Model Integration) describe un marco de trabajo para planificar, prevenir y mitigar los riesgos sistemáticamente a fin de minimizar proactivamente su impacto sobre el proyecto [3]. El marco de trabajo propuesto por CMMI incluye las siguientes prácticas:

1. Preparar la gestión de riesgos.
  - 1.1 Determinar las fuentes y las categorías de riesgos.
  - 1.2 Definir los parámetros de riesgos.
  - 1.3 Establecer una estrategia de gestión de riesgos.
2. Identificar y analizar los riesgos.
  - 2.1 Identificar los riesgos.
  - 2.2 Evaluar, clasificar y priorizar los riesgos.
3. Mitigar los riesgos.
  - 3.1 Desarrollar los planes de mitigación de riesgos.
  - 3.2 Implementar los planes de mitigación de riesgos.

#### **4.2.6. Estándar ISO 31000.**

Esta normativa es una guía internacional para la gestión del riesgo, “ofrece lineamientos guía para administrar/gestionar los riesgos a los que las organizaciones se enfrentan. La aplicación de estos lineamientos puede adaptarse a cualquier organización y a su contexto” [7]. En el contexto del desarrollo de software, se utiliza para:

1. Identificación de Riesgos: Detectar posibles riesgos en cada fase del desarrollo, como riesgos técnicos, de seguridad, y de cumplimiento.
2. Evaluación y Análisis de Riesgos: Evaluar la probabilidad e impacto de los riesgos identificados y priorizarlos.

3. Tratamiento de Riesgos: Implementar estrategias para mitigar, transferir, aceptar o evitar riesgos.

4. Monitoreo y Revisión: Consiste en supervisar continuamente los riesgos para revisar la efectividad de la aplicación de las estrategias de mitigación.

5. Comunicación y Consulta: Establecer canales de comunicación claros sobre los riesgos y sus tratamientos con todas las partes interesadas.

Utilizando la norma se puede estructurar un enfoque sistemático y consistente para gestionar los riesgos asociados con el desarrollo de software, mejorando así la probabilidad de éxito de los proyectos y la calidad del producto final.

#### **4.2.7. Estándar ISO/IEC 25022.**

Determina el modelo para la evaluación de calidad en uso de sistemas y productos de software que se compone de diferentes actividades para su proceso:

1. Aplicación de la calidad en Uso: Se refiere al subconjunto de calidad en uso compuesto por efectividad, eficiencia, satisfacción y cobertura del contexto.

2. Medición de la calidad en Uso: La calidad en uso depende no sólo de la calidad del producto del software o sistema informático, sino también del contexto particular en el que se utiliza el producto. El contexto de uso incluye factores del usuario, factores de la tarea y factores ambientales físicos y sociales que pueden afectar la calidad del uso [5].

3. Interpretación de las medidas de calidad en Uso: Analizar los datos para determinar la calidad en uso. Para cumplir adecuadamente con este punto se debe incluir el análisis de datos, la interpretación de resultados y la generación de informes.

Para medir la calidad en uso del marco se aplicará una “Matriz de modelo de calidad en USO” basado en la ISO/IEC 25022, que se relacionan con el resultado de la interacción cuando un producto o artefacto de software es utilizado en un contexto particular de uso.

#### **4.2.8. Scrum.**

Es un marco de trabajo por el cual las personas pueden abordar problemas complejos adaptativos, a la vez que crean productos del máximo valor posible productiva y creativamente. Scrum no es un proceso, una técnica o método definitivo. En lugar de eso, es un marco de trabajo dentro del cual se pueden emplear varios procesos y técnica [8].

*Eventos de Scrum.* “El Sprint es un contenedor para todos los demás eventos. Cada evento en Scrum es una oportunidad formal para inspeccionar y adaptar los artefactos Scrum. No operar cualquier evento según lo prescrito resulta en la pérdida de oportunidades para inspeccionar y adaptarse” [9]. Los eventos que se utilizan en Scrum son:

- Sprint Planning
- Daily Scrum
- Sprint Review
- Sprint Retrospective

#### **4.2.9. Design Science Research (DSR).**

Design Science Research (Investigación en ciencias del diseño) es un enfoque de investigación riguroso que propone la construcción de artefactos para brindar una solución útil y efectiva a un problema de un dominio dado. El artefacto debe ser una solución innovadora a un problema no trivial. El desarrollo del artefacto implica un ciclo de actividades de diseño-construcción-evaluación, que iteran tantas veces como sean necesarias antes que el artefacto sea finalmente verificado, validado y comunicado para su utilización [10].

Según explica [11], la metodología DSR consta de varios pasos que guían a los investigadores en el desarrollo de soluciones para problemas complejos. Los pasos principales son:

*Identificación del problema:* Consiste en reconocer y comprender el problema que debe abordarse. Esto es crucial, ya que sienta las bases de todo el proceso de investigación.

*Objetivos de la solución:* En este paso, se deben establecer los objetivos que pretende alcanzar la solución. Estos objetivos permiten validar que el diseño esté alineado con las necesidades de los usuarios y el contexto en el que se aplicará.

*Diseño y desarrollo:* En este paso se crea la solución o el artefacto. Se puede usar modelos, diagramas, matrices, etc. para representar el artefacto.

*Demostración:* Una vez construido el artefacto, se debe demostrar su funcionamiento en la práctica. Esto podría implicar pruebas piloto o simulaciones para validar la solución.

*Evaluación:* En este paso se evalúa la solución comparándola con los objetivos establecidos anteriormente. Se puede usar entrevistas semiestructuradas, cuestionarios, observaciones, etc. para evaluar la aplicación del marco de trabajo.

*Comunicación:* Los resultados y la solución desarrollada deben comunicarse o compartirse con las partes interesadas.

#### **4.2.10. Las herramientas de colaboración participativa.**

Son servicios informáticos como los formularios de Google que permiten a un grupo de trabajo participar en un mismo proyecto en tiempo real o comunicarse sin estar en el mismo lugar.

## **5. Metodología**

En esta sección se indican detalles sobre el procedimiento empleado en la realización del Trabajo de Titulación denominado “Desarrollo de un marco de trabajo para la gestión de riesgos de calidad en proyectos de desarrollo de software ágil, basado en la ISO 31000 y gestionados con Scrum”.

### **5.1. Área de estudio.**

El área de estudio del Trabajo de Titulación está relacionada con los proyectos que han sido gestionados con Scrum. Específicamente se levantó la información de riesgos de calidad en bases de datos a nivel internacional y luego se priorizó los riesgos críticos con profesionales de la maestría, docentes del área de computación y gerentes de proyecto involucrados en el desarrollo de software. El Trabajo de Titulación se desarrolló en la Universidad Nacional de Loja como parte de la Maestría de Ingeniería en Software que oferta esta institución. La Universidad se encuentra ubicada en la ciudad de Loja en el sector la Argelia, entre las avenidas Pio Jaramillo Alvarado y Reinaldo Espinosa.

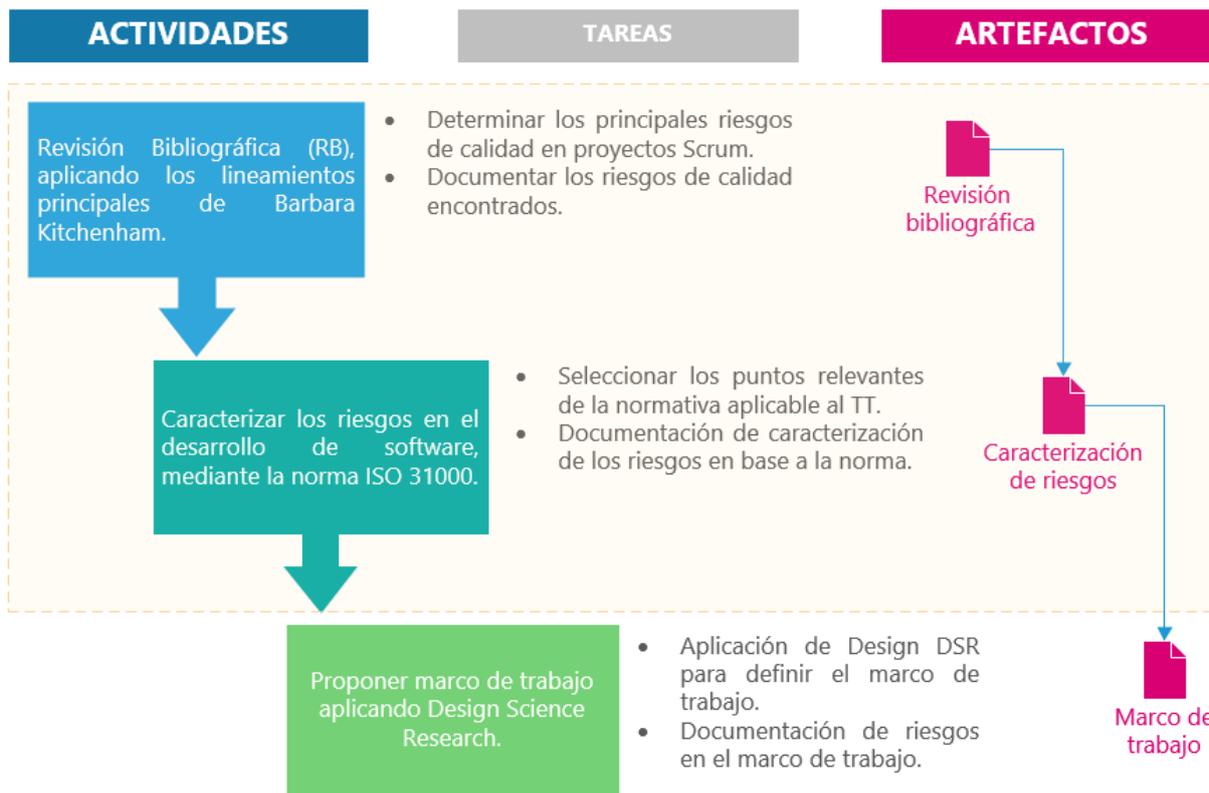
### **5.2. Procedimiento.**

En este apartado se presenta el proceso llevado a cabo para alcanzar el objetivo general del Trabajo de Titulación que pretende “Desarrollar un marco de trabajo para la gestión de riesgos de calidad de software en proyectos de desarrollo ágil, basado en la ISO 31000 y gestionados con Scrum”, donde se abordan de manera detallada como se logró cumplir con los objetivos específicos y las actividades asociadas a cada uno de ellos.

#### **5.2.1. Objetivo 1: Diseñar el marco de trabajo basado en la Revisión Bibliográfica y en el diagnóstico de principales riesgos que afectan la calidad de software en proyectos que usan Scrum.**

Para enfrentar los retos que implica desarrollar el objetivo, se ejecutan las actividades y tareas propuestas en la figura 1, así mismo, se elaboraron los artefactos correspondientes. Si bien en la figura 1 se muestran separadas las actividades de “Caracterizar los riesgos en el desarrollo de software” y la “Revisión Bibliográfica”, éstas son complementarias, ya que la caracterización propuesta en la norma ISO 31000 tiene una actividad de “Identificación de riesgos de calidad” y ésta se la realizó por medio de la Revisión Bibliográfica.

# DESIGN SCIENCE RESEARCH (DSR)



**Figura 1.** Procedimiento de ejecución del objetivo 1.  
**Fuente:** Elaborada por el tesista.

## 5.2.1.1. Revisión Bibliográfica (RB), aplicando los lineamientos principales de Barbara Kitchenham.

El objetivo principal de la RB fue identificar los riesgos relacionados con la calidad de software al aplicar Scrum en proyectos de desarrollo. Para ello se tomó como referencia los lineamientos principales propuestos por Barbara Kitchenham y se los aplico mediante las fases que propone la herramienta Parsifal. Las fases aplicadas son:

- *Revisión:* Título, descripción.
- *Planeación:* Protocolo (Objetivos, PICOC, preguntas de investigación, palabras clave y sinónimos, cadenas de búsqueda, fuentes bibliográficas, criterios de selección), lista de verificación de evaluación de calidad (preguntas, respuestas, puntuaciones de evaluación de calidad), formulario de extracción de datos [12].
- *Ejecución:* Búsqueda, importar estudios, selección de estudios, evaluación de calidad, extracción y análisis de datos.

En la figura 2 se puede observar la importación de estudios correspondiente a la fase de ejecución de la RB, en total se importaron 144 artículos para ser revisados y obtener los riesgos de calidad del software. La configuración y aplicación de cada una de las fases se encuentran en el anexo 1.



**Figura 2.** Fase de ejecución en herramienta Parsifal.  
**Fuente:** Herramienta Parsifal: <https://parsif.al/>.

#### **5.2.1.1.1. Determinar los principales riesgos de calidad en proyectos Scrum.**

Al ejecutar la fase de Planificación en la Parsifal se detectaron un total de 18 artículos, para seleccionar los riesgos se consideraron dos preguntas, la primera para determinar si en los artículos se indican riesgos que afectan directamente a la calidad del producto y la segunda para determinar riesgos que podrían influir en la calidad del producto. Para la primera pregunta se identificaron en total 22 riesgos y para la segunda 165. Los riesgos que se analizaron en este estudio corresponden a los que afectan directamente a la calidad, los nombres de riesgos detectados se encuentran en la figura 3.

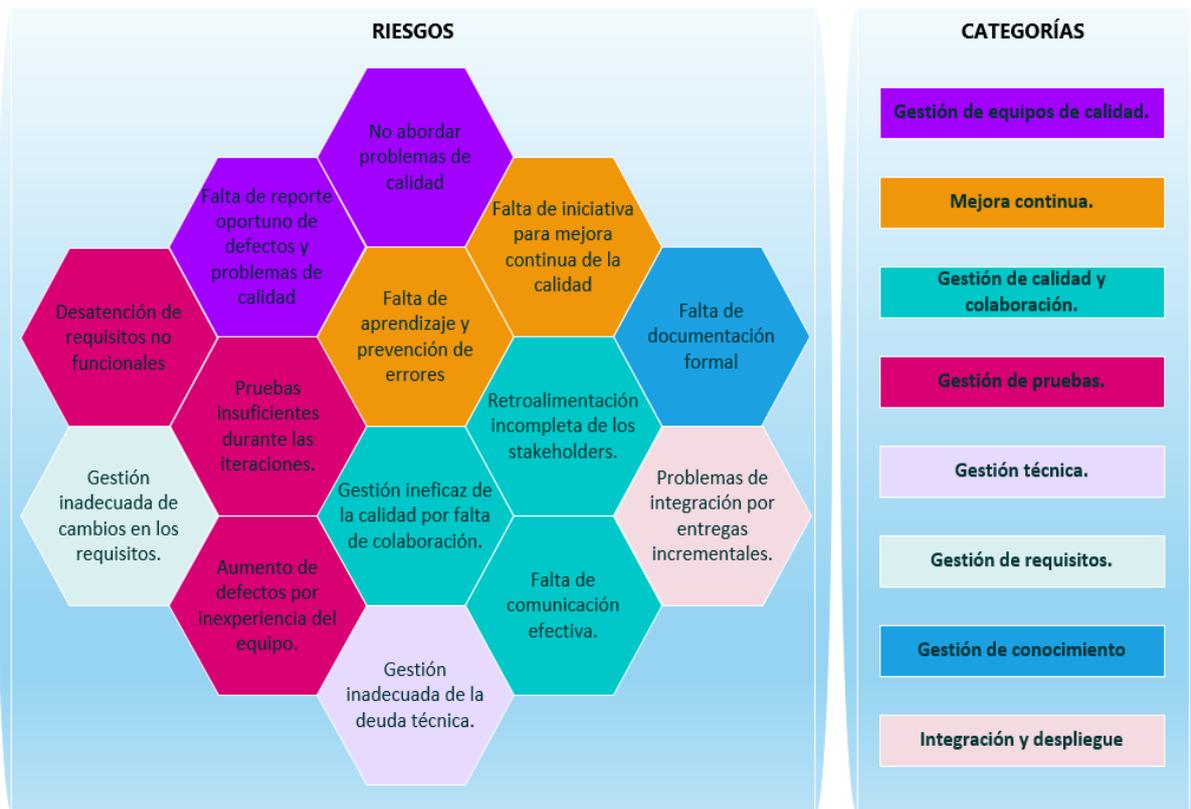
Para detectar los riesgos y responder las preguntas planteadas en la RB, se utilizó herramientas de inteligencia artificial (ChatGPT y ChatPDF), luego se procedió a revisar los aportes propuestos por los autores sobre los riesgos finales para levantar la información necesaria y posteriormente facilitar la documentación.



**Figura 3.** Riesgos de calidad detectados en la RB.  
**Fuente:** Elaborada por el tesista.

#### 5.2.1.1.2. Documentar los riesgos de calidad encontrados.

Se procedió con el análisis de los riesgos de calidad para unificar o eliminar los que estén duplicados, también se definió una categoría, esta se utilizará para realizar la alineación con el contexto definido en la caracterización de los requisitos en base a la norma ISO 31000. Luego del análisis quedaron documentados 14 riesgos, adicionalmente se redactó una descripción general de las categorías. Los riesgos finales y sus categorías se muestran en la figura 4.



**Figura 4.** Riesgos y categorías de calidad depurados.  
**Fuente:** Elaborada por el tesista.

### 5.2.1.2. Caracterizar los riesgos en el desarrollo de software, mediante la norma ISO 31000.

#### 5.2.1.2.1. Seleccionar los puntos relevantes de la normativa aplicable al TT.

El proceso propuesto en la normativa ISO 31000 plantea la ejecución de seis actividades que las organizaciones deberían ejecutar para tener una gestión de riesgo adecuada. Para desarrollar la caracterización de los requisitos, se consideran parte de estas actividades propuestas en la norma. Primero, se realizó el levantamiento de los riesgos para lo cual se usaron los procesos de “Establecer el contexto” del que obtienen los riesgos y la “Evaluación de riesgos”; luego se implementó el “Tratamiento de los riesgos”, con lo que se pretende tener mapeados los riesgos y las opciones de tratamiento que se les puede dar en proyectos de Scrum; finalmente, se caracterizarán todos los riesgos en una matriz. En la figura 5 se muestra los procesos que se utilizaron de la norma para armar la caracterización de riesgos.



**Figura 5.** Actividades ejecutadas de la norma ISO 31000.

**Fuente:** Elaborada por el tesista, basado en el proceso de la norma ISO 31000.

Las actividades “Comunicación y consulta”, “Seguimiento y revisiones” y “Registro e informes”, no se utilizaron porque son actividades que corresponden con crear un sistema de gestión de riesgos, lo cual no es el objetivo del presente Trabajo de Titulación.

#### **5.2.1.2.2. Documentación de caracterización de los riesgos en base a la norma.**

Para realizar la caracterización de los riesgos se ejecutaron las actividades indicadas en la figura 5, fusionándolas con los riesgos detectados en la Revisión Bibliográfica. A continuación, se detallan las tareas realizadas en cada actividad:

- **Establecer el contexto para la gestión de riesgos.**

Según el proceso propuesto en la ISO 31000 para caracterizar los riesgos se debe definir el contexto. Para proyectos de desarrollo de software que usan Scrum, se definió un contexto interno general que puede ser común para proyectos que usan Scrum y que influyen en la detección de riesgos de calidad; el contexto externo no será definido ya que depende de la realidad propia de la organización, por ejemplo, se consideran los factores del mercado, las regulaciones, etc.

- **Evaluación de riesgos.**

La evaluación de riesgos incluye las actividades de identificación de riesgos, análisis de riesgo y evaluación de riesgos. Dado que los riesgos ya se documentaron en la RB, se procedió a complementarlos según los lineamientos de las actividades mencionadas.

### Identificación de los riesgos de calidad.

El propósito de la identificación de riesgos es encontrar, reconocer y describir los riesgos que pueden ayudar o impedir a una organización lograr sus objetivos. Para la identificación de los riesgos es importante contar con información pertinente, apropiada y actualizada [7]. Para cumplir con esta sección de la normativa se utilizó la Revisión Bibliográfica que facilitó la detección de 14 riesgos de calidad asociados a proyectos de software que usan Scrum.

La normativa sugiere analizar los riesgos identificados y validar que estén relacionados con las características propias de proyectos Scrum, para cumplir con esta sección se utilizó la categorización de los riesgos y el contexto interno definido como parte de la caracterización de los riesgos.

### Análisis de riesgos de calidad.

Para proyectos generales que utilizan Scrum, se analizó cada riesgo en términos de **probabilidad** (¿con qué frecuencia puede ocurrir?) y su **impacto** (¿qué tan severas son las consecuencias si ocurren?). Para realizar esta actividad se elaboró una encuesta que se aplicó a 15 profesionales relacionados con el desarrollo de software, se solicitó que en cada riesgo se determine su probabilidad e impacto en el contexto de desarrollo de software con Scrum. El detalle de los resultados se encuentra en el anexo 2.

Para facilitar el análisis y la clasificación de los riesgos, se colocaron en una matriz de riesgo en la que se puede visualizar su probabilidad e impacto.

### Evaluación de los riesgos

En el análisis de riesgos de calidad se determinó la probabilidad y el impacto de los riesgos, esto permitió clasificarlos y determinar cuáles son los más críticos. En el alcance definido para el presente TT se contempló evaluar los 5 riesgos de calidad que sean más críticos. Para realizar esta selección se evaluó la matriz de riesgos y se eligieron los riesgos que tienen mayor impacto y probabilidad de ocurrencia. Adicionalmente se mapeó otro riesgo en base a la experiencia del tesista (Entrega tardía de componentes para pruebas en cada Sprint), el cual no salió en la Revisión Bibliográfica, pero es crítico en proyectos de Scrum. Los riesgos priorizados se pueden apreciar en la siguiente figura:



**Figura 6.** Riesgos de calidad priorizados.

**Fuente:** Elaborada por el tesista.

- **Tratamiento de riesgos de calidad (ISO 31000).**

Una vez identificados los riesgos de calidad se procedió con la caracterización de riesgos, con la finalidad de entender el evento que genera el riesgo, la causa que origina el riesgo y los posibles efectos que se pueden tener en la calidad en caso de que se materialicen. Esta caracterización completa incluye la categorización, y la probabilidad e impacto.

Para identificar el evento, las causas y el impacto, se retomó los artículos de los que se extrajo la información, luego se amplió la búsqueda con herramientas de IA y se complementó con la experiencia del Tesista.

Siguiendo la normativa ISO 31000, para el tratamiento de riesgos de calidad, se proponen 3 actividades puntuales:

1. Definir generalidades de los requisitos.
2. Selección de opciones de tratamiento de los riesgos.
3. Preparación e implementación de los planes para el tratamiento de los riesgos.

Se consideran los puntos 1 y 2 para el presente Trabajo de Titulación. Estos puntos se desarrollaron en la sección 5.2.1.3 de definición del marco de trabajo, ya que es donde se plantean opciones de tratamiento a los riesgos en proyectos Scrum.

### **5.2.1.3. Proponer marco de trabajo.**

Para proponer el marco de trabajo se aplicó la metodología Design Science Research (DSR), ya que permite crear y evaluar artefactos que resuelven problemas prácticos. La DSR se conoce también como investigación constructiva, ya que tiene objetivos de investigación académica generalmente de naturaleza más práctica.

Dependiendo del problema a resolver DSR plantea varios enfoques o pasos que permiten crear los artefactos, por ejemplo [10], plantea tres ciclos principales: ciclo de relevancia, ciclo de diseño y ciclo de rigor. Cada uno de estos ciclos tiene una función específica que va desde investigar y comprender el problema, desarrollar y evaluar el artefacto, finalmente mejorar el modelo y comunicar los resultados.

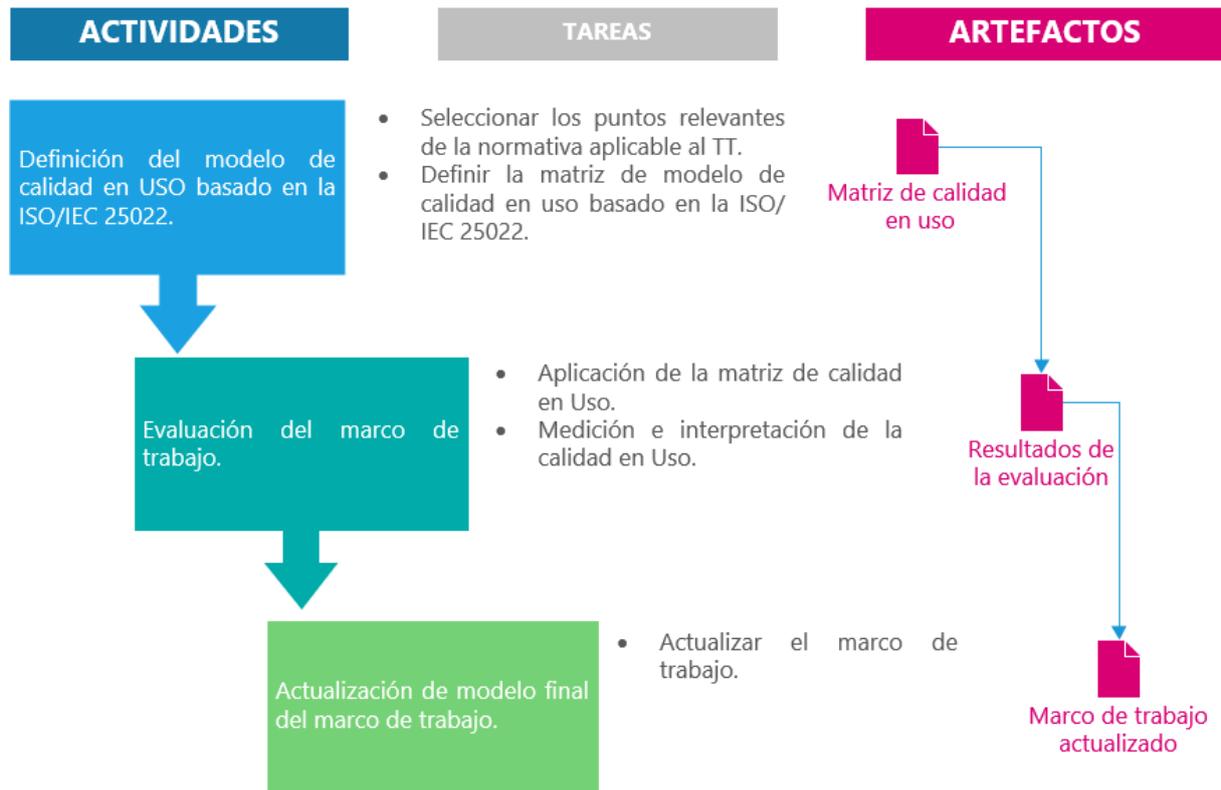
Los autores [11] [13], plantean otro enfoque el cual coincide en su mayoría en ambos artículos, indican los siguientes pasos: “Identificación del problema, objetivos de la solución, diseño y desarrollo, demostración, evaluación, comunicación”.

El alcance de este TT es definir el marco de trabajo en el objetivo 1 y validar que sea usable en el objetivo 2, por lo que se utilizará una adaptación de los pasos propuestos para la metodología DSR indicados en el párrafo anterior. Estos pasos se aplicaron a lo largo de todo el Trabajo de Titulación en las diferentes secciones, por ejemplo, la identificación del problema y los objetivos de la solución están en el planteamiento del proyecto, el diseño y desarrollo se aborda en el objetivo 1, la demostración y evaluación en el objetivo 2 y la comunicación, mediante la documentación del presente Trabajo de Titulación y la creación del artículo.

### **5.2.2. Objetivo 2: Evaluar el marco de trabajo propuesto utilizando la norma ISO/IEC 25022.**

En la figura 7 se presenta un resumen de la metodología aplicada, indicando las actividades, tareas y artefactos generados en el desarrollo del objetivo. El entregable final de este apartado fue el marco de trabajo actualizado considerando las observaciones pertinentes obtenidas como resultado de la evaluación de la calidad en uso del marco propuesto.

## DESIGN SCIENCE RESEARCH (DSR)



**Figura 7.** Procedimiento de ejecución del objetivo 2.  
**Fuente:** Elaborada por el tesista.

### 5.2.2.1. Definición del modelo de calidad en Uso basado en la ISO/IEC 25022.

#### 5.2.2.1.1. Seleccionar los puntos relevantes de la normativa aplicable al TT.

Luego de revisar la normativa se seleccionaron varias características que en conjunto miden la calidad en uso de aplicaciones o artefactos. Las características son:



**Figura 8.** Características de la normativa ISO/IEC 25022.  
**Fuente:** Elaborada por el tesista.

#### **5.2.2.1.2. Definir la matriz de modelo de calidad en uso basado en la ISO/IEC 25022.**

Para determinar la Calidad en Uso, se evaluó el marco de trabajo en base a la adaptación de la plantilla de “Definición del Modelo de Calidad en Uso”, propuesta como parte de un taller desarrollado en la asignatura Aseguramiento de Calidad, de la Maestría de Ingeniería en Software ofertada por la Universidad Nacional de Loja. Se definieron las siguientes actividades para definir el modelo de calidad a evaluar:

1. Diseño de preguntas para evaluar las subcaracterísticas.
2. Ponderación del modelo de calidad en uso en base al contexto del proyecto.

#### **5.2.2.2. Evaluación del marco de trabajo.**

La evaluación del marco de trabajo se realizó mediante un taller con la colaboración de estudiantes de la carrera de computación de la Universidad Nacional de Loja. Se aplicó la siguiente metodología:

1. Se organizó un taller en el que se desarrollaron las siguientes actividades:
  - a. Capacitación a los estudiantes para estandarizar los conocimientos respecto de la metodología Scrum y de la norma ISO/IEC 25022 y resolver cualquier duda respecto del taller.
  - b. Se explicó la estructura de la matriz, la forma de ingresar los datos que los estudiantes consideren pertinentes y se procedió con la ejecución del taller.
2. Ejecución de la evaluación de la matriz de calidad.
3. Se realizó la valoración de los resultados obtenidos, los cuales se pueden visualizar en la sección 6.2.2 de resultados.
4. Se finaliza con la interpretación de resultados.

#### **5.2.2.3. Actualización de modelo final del marco de trabajo.**

Durante el desarrollo del taller se observaron y reportaron algunas mejoras para los riesgos, ejecutando las siguientes actividades:

1. Se analizaron las observaciones y se refinaron las que eran similares.
2. Se procedió con la actualización del marco de trabajo en los casos que eran pertinentes.
3. Liberación formal del marco de trabajo propuesto.

## 6. Resultados

En esta sección se presentan los resultados obtenidos para cada objetivo específico, junto con la descripción de las actividades que se llevaron a cabo para su cumplimiento. En el primer objetivo se planteó como meta principal crear un marco de trabajo para riesgos de calidad, mientras que en el segundo se planteó evaluar el marco de trabajo propuesto en base a la norma ISO/IEC 25022, con el propósito de garantizar que esté listo para ser aplicado en proyectos con Scrum.

### 6.1. Objetivo 1: Diseñar el marco de trabajo basado en la Revisión Bibliográfica y en el diagnóstico de principales riesgos que afectan la calidad de software en proyectos que usan Scrum.

En este apartado se muestran los resultados de aplicar la metodología definida. Se inicio con la Revisión Bibliográfica, posteriormente de caracterizaron los riesgos de calidad priorizados y finalmente se propuso el marco de trabajo.

#### 6.1.1. Revisión Bibliográfica (RB), aplicando los lineamientos principales de Barbara Kitchenham.

Con el objetivo de determinar los riesgos que afectan la calidad de software en proyectos que usan Scrum se realizó una Revisión Bibliográfica (Anexo 1), en la que se evaluaron 144 artículos, luego de aplicar los criterios de exclusión (Tabla 1) quedaron 42. Como resultado de la aplicación de preguntas de calidad (Tabla 1) se encontraron 3 artículos que indican 22 riesgos que afectan directamente a la calidad de software. Adicionalmente se encontraron 15 artículos que indican 165 riesgos que posiblemente puedan afectar a la calidad del software. En este Trabajo de Titulación se analizó la información levantada en la RB para establecer los riesgos de calidad de los 3 artículos y documentarlos.

**Tabla 1.** Criterios de inclusión, exclusión y pregunta de calidad de software.

Criterios/Preguntas	Descripción
Criterios de inclusión.	<ul style="list-style-type: none"><li>Estudios que abordan riesgos de calidad del software en proyectos que utilizan Scrum.</li><li>Estudios que abordan riesgos que influyen en la calidad de software en proyectos que utilizan Scrum.</li></ul>
Criterios de exclusión.	<ul style="list-style-type: none"><li>Estudios no accesibles.</li><li>Estudios publicados antes del 2019.</li><li>Estudios que no mencionan Scrum.</li></ul>

	<ul style="list-style-type: none"> <li>• Estudios que usan Scrum en otros ámbitos distintos al desarrollo de software.</li> <li>• Estudios que mencionan Scrum, pero no abordan riesgos.</li> <li>• Publicaciones duplicadas.</li> <li>• Publicaciones en idiomas diferentes al Inglés y Español.</li> </ul>
Preguntas de calidad.	<ul style="list-style-type: none"> <li>• ¿El autor identifica o detalla riesgos de calidad en proyectos de software desarrollados con Scrum?</li> <li>• ¿El autor identifica o detalla riesgos que pueden influir en la calidad de software en proyectos de desarrollo que utilizan Scrum?</li> </ul>

### 6.1.1.1. Determinar los principales riesgos de calidad en proyectos Scrum.

Como resultado final de la RB se encontraron 22 riesgos que afectan directamente a la calidad del software en proyectos que utilizaron Scrum. Para facilitar su comprensión, se realizó una categorización y una descripción inicial, en base a la revisión de los artículos relacionados. Los riesgos se describen en la tabla 2.

**Tabla 2.** Riesgos de calidad de software.

Nro.	Categoría	Nombre	Descripción
1	Gestión de equipos y roles.	No abordar problemas de calidad.	Los equipos que no tienen un ambiente seguro psicológicamente tienden a evitar discutir problemas de calidad.
2	Gestión de calidad y comunicación.	Retraso en la identificación de problemas de calidad.	Esto incrementa el riesgo de que los defectos lleguen a producción, afectando la funcionalidad y estabilidad del producto.
3	Gestión de comunicación y calidad.	Falta de reporte de defectos y errores.	Permite que los errores pasen desapercibidos y no sean corregidos a tiempo, impactando negativamente la calidad del software entregado.
4	Mejora continua y calidad.	Falta de mejora continua.	Sin un entorno que promueva la seguridad psicológica, los equipos tienden a evitar proponer mejoras en el proceso o en el producto.
5	Gestión de calidad y mejora continua.	Repetición de Errores.	Si los errores no se discuten abiertamente y no se aprende de ellos, es probable que se repitan en futuras iteraciones, acumulando defectos y disminuyendo la calidad del producto.
6	Gestión de equipos y calidad.	Falta de Iniciativa para Mejorar la Calidad.	En un entorno donde no se fomenta la seguridad psicológica, los miembros del equipo pueden ser menos propensos a proponer iniciativas de mejora.
7	Gestión de calidad y colaboración.	Dificultades en la Gestión de la Calidad.	La falta de un enfoque colaborativo y de aprendizaje puede dificultar la implementación de prácticas efectivas de gestión de la calidad.
8	Gestión de equipos y calidad.	Defectos No Reportados.	Los miembros del equipo no se sienten cómodos reportando defectos o problemas de calidad.

9	Gestión de calidad.	de	Riesgos relacionados con el aumento de defectos (Equipos nuevos).		Los equipos ágiles, especialmente los nuevos, tienden a experimentar un aumento de defectos debido a la falta de experiencia.
10	Gestión técnica.		Riesgos relacionados con la deuda técnica.		El manejo incorrecto de la deuda técnica puede comprometer la calidad del software a largo plazo, ya que los problemas no resueltos en etapas tempranas del proyecto se acumulan y afectan la integridad del código.
11	Gestión de requisitos.	de	Riesgos de Requisitos Cambiantes.	de	Cambios frecuentes en los requisitos pueden llevar a un producto final que no cumpla con las expectativas del cliente, afectando la calidad.
12	Gestión de tiempo y calidad.	del y	Riesgos de Desempeño (Presión por cumplir plazos).	de por	La presión para cumplir con los plazos puede resultar en una calidad de código inferior.
13	Seguridad y calidad.	y	Riesgos de Seguridad.	de	La falta de pruebas adecuadas de seguridad puede comprometer la integridad del software, afectando su calidad y confiabilidad.
14	Gestión de pruebas.	de	Falta de Pruebas Adecuadas.		La ausencia de un enfoque riguroso en las pruebas puede resultar en la entrega de un producto con defectos, especialmente al no realizar pruebas suficientes en cada iteración.
15	Comunicación y colaboración.		Falta de comunicación efectiva.	de	La calidad del producto puede verse comprometida si hay fallas en la comunicación entre los miembros del equipo.
16	Gestión de requisitos.	de	Cambios frecuentes en los requisitos.		La flexibilidad de Scrum permite que los requisitos cambien a lo largo del desarrollo.
17	Gestión de conocimiento.	de	Falta de documentación formal.	de	Scrum minimiza la documentación esto puede dificultar la comprensión de los requisitos y estándares de calidad.
18	Integración y despliegue.	y	Integración insuficiente de componentes.	de	La entrega de funcionalidades de forma incremental en Scrum puede generar problemas de integración si no se realiza adecuadamente.
19	Gestión técnica.		Deuda técnica acumulada.		La necesidad de entregar productos rápidamente puede llevar a decisiones técnicas subóptimas que generan deuda técnica.
20	Gestión de pruebas.	de	Falta de pruebas exhaustivas (suficientes).		La velocidad de las iteraciones de Scrum puede resultar en pruebas insuficientes.
21	Calidad funcional.	no	Enfoque limitado en la calidad funcional.	no	Scrum tiende a priorizar la entrega de nuevas funcionalidades visibles para el usuario, descuidando aspectos no funcionales.
22	Gestión de stakeholders.	de	Retroalimentación incompleta de los stakeholders.		Si no se obtiene retroalimentación constante y precisa de los usuarios finales es posible que se desarrollen características que no cumplan con sus expectativas.

### 6.1.1.2. Documentar los riesgos de calidad encontrados.

Antes de realizar la documentación se procedió con la eliminación o unificación de riesgos que tenían un nombre distinto pero su descripción era similar o relacionada, se ejecutó esta actividad en base a la lectura de la información extraída de los artículos, en total quedan 14 riesgos de calidad de software. Luego de afinar los riesgos, así como sus nombres, se revisó las categorías para de igual manera unificarlas, eliminarlas o redactar su nombre de manera adecuada. En la tabla 3 se presenta el listado final de riesgos.

**Tabla 3.** Riesgos de calidad de software afinados y agrupados por categoría.

Nro.	Categoría	Nombre	Descripción
R1	Gestión de equipos calidad.	de No abordar de problemas de calidad.	Los equipos que no tienen un ambiente seguro psicológicamente tienden a evitar discutir problemas de calidad. Esto resulta en un deterioro del software al no identificar defectos tempranamente.
R2	Gestión de equipos calidad.	de Falta de oportuno reporte de defectos y problemas de calidad.	La falta de un ambiente de seguridad psicológica o falta de confianza en los equipos Scrum puede llevar a que los miembros no se sientan cómodos para señalar, reportar o discutir defectos y problemas de calidad del producto durante los Sprints. Esto provoca que los errores persistan sin ser identificados o corregidos a tiempo, incrementando el riesgo de que los defectos lleguen a producción. La consecuencia es un impacto negativo en la funcionalidad, estabilidad y calidad del software entregado.
R3	Mejora continua.	Falta de iniciativa para mejora continua de la calidad.	En un entorno donde no se promueve la seguridad psicológica, los miembros del equipo tienden a evitar proponer mejoras en el proceso o en el producto. Esto limita la iniciativa para innovar y mejorar de manera iterativa, lo que impacta negativamente en la calidad del software, el rendimiento y la satisfacción del cliente a largo plazo.
R4	Mejora continua.	Falta de aprendizaje y prevención de errores.	Si los errores no se discuten abiertamente y no se aprende de ellos, es probable que se repitan en futuras iteraciones, acumulando defectos y disminuyendo la calidad del producto.
R5	Gestión de calidad y colaboración.	Gestión ineficaz de la calidad por falta de colaboración.	La falta de un enfoque colaborativo y de aprendizaje puede dificultar la implementación de prácticas efectivas de gestión de la calidad, resultando en un producto que no cumple con los estándares de calidad esperados.
R6	Gestión de calidad y colaboración.	Retroalimentación incompleta de los stakeholders.	Si no se obtiene retroalimentación constante y precisa de los usuarios finales o partes interesadas durante cada Sprint, es posible que se desarrollen características que no cumplan con las expectativas o necesidades del cliente.

R7	Gestión de calidad y colaboración.	Falta de comunicación efectiva.	de	La calidad del producto puede verse comprometida si hay fallas en la comunicación entre los miembros del equipo. Malentendidos o falta de información pueden resultar en errores en la implementación de características y funcionalidades.
R8	Gestión de pruebas.	Aumento de defectos por inexperiencia del equipo.	de	Los equipos ágiles, especialmente los nuevos, tienden a experimentar un aumento de defectos debido a la falta de experiencia o a una implementación apresurada. Estos defectos pueden afectar la estabilidad y la funcionalidad del software.
R9	Gestión de pruebas.	Pruebas insuficientes durante iteraciones.	de	La falta de un enfoque riguroso en las pruebas, debido a la velocidad de las iteraciones en Scrum, puede resultar en pruebas insuficientes en cada iteración. Esto permite que los defectos persistan hasta las etapas finales o incluso después del lanzamiento, afectando la calidad del producto entregado. La presión para cumplir con los plazos puede resultar en una calidad de código inferior, ya que los desarrolladores pueden priorizar la velocidad sobre la calidad.
R10	Gestión de pruebas.	Desatención de requisitos funcionales.	de	Scrum tiende a priorizar la entrega de nuevas funcionalidades visibles para el usuario, descuidando aspectos no funcionales como el rendimiento, la seguridad o la usabilidad, lo que afecta la experiencia del usuario. La falta de pruebas adecuadas de seguridad puede comprometer la integridad del software, afectando su calidad y confiabilidad.
R11	Gestión técnica.	Gestión inadecuada de la deuda técnica.	de	La necesidad de entregar productos rápidamente puede llevar a decisiones técnicas que no son óptimas, generando deuda técnica. Al no ser gestionada adecuadamente, los problemas no resueltos en etapas tempranas del proyecto se acumulan, comprometiendo la integridad del código y afectando negativamente el rendimiento, la escalabilidad y la mantenibilidad del software.
R12	Gestión de requisitos.	Gestión inadecuada de cambios en los requisitos.	de	La flexibilidad de Scrum permite que los requisitos cambien durante el desarrollo. Si estos cambios no se gestionan adecuadamente, pueden generar confusiones y llevar a la entrega de un producto que no cumpla con las expectativas del cliente, afectando la calidad final del software.
R13	Gestión de conocimiento.	Falta de documentación formal.	de	La tendencia de Scrum a minimizar la documentación puede dificultar la comprensión de los requisitos y estándares de calidad, resultando en inconsistencias y un producto que no cumple con los estándares esperados.
R14	Integración y despliegue.	Problemas de integración entregas incrementales.	de	La entrega de funcionalidades de forma incremental en Scrum puede generar problemas de integración de componentes si no se realiza adecuadamente. Los módulos desarrollados en diferentes Sprints pueden no interactuar correctamente.

Se establecieron 8 categorías que agrupan los riesgos de calidad detectados, estas categorías ayudan a organizar los diferentes riesgos de calidad de software y a enfocar mejor las estrategias de mitigación para asegurar la calidad del producto final. A continuación, se presenta una descripción para cada categoría.

**Tabla 4.** Categorías de riesgos de calidad.

Categorías	Descripción
1. Gestión de equipos de calidad.	Se refiere a los riesgos relacionados con la dinámica del equipo Scrum, especialmente generando confianza en cuanto a la capacidad de los miembros para comunicar y abordar problemas de calidad [14]. Un ambiente de seguridad psicológica es crucial para que los equipos discutan y resuelvan defectos a tiempo [9].
2. Mejora continua.	Engloba los riesgos relacionados con la falta de iniciativa para proponer mejoras o aprender de los errores, ya que los proyectos de Scrum trabajan con una iteración constante y la mejora continua. Si los equipos no fomentan una cultura de aprendizaje y no implementan mejoras de manera regular, los errores tienden a repetirse [15].
3. Gestión de calidad y colaboración.	Esta categoría cubre los riesgos derivados de la falta de colaboración efectiva y la falta de una comunicación clara dentro del equipo y con los stakeholders, lo que puede dificultar la obtención de retroalimentación oportuna de las partes interesadas. [16].
4. Gestión de pruebas.	Esta categoría agrupa los riesgos relacionados con pruebas insuficientes (funcionales y no funcionales), ya sea por la falta de experiencia del equipo o la presión por cumplir con los plazos de los Sprints. Si las pruebas no son rigurosas, los defectos pueden afectar la estabilidad y la funcionalidad del software [17].
5. Gestión técnica.	Los riesgos en esta categoría están relacionados con las decisiones técnicas no adecuadas que afectan la calidad del software, como la gestión de la deuda técnica.
6. Gestión de requisitos.	Esta categoría agrupa los cambios frecuentes y mal gestionados en los requisitos que pueden generar confusión y afectar la calidad del producto final.
7. Gestión del conocimiento.	Sin una documentación adecuada, los equipos pueden tener dificultades para comprender los requisitos y los estándares de calidad, lo que lleva a inconsistencias en el desarrollo, esta categoría agrupa riesgos relacionados con la falta de transferencia de conocimientos y documentación. [9].
8. Integración y despliegue.	En proyectos Scrum, los riesgos en esta categoría están relacionados con fallos de integración y despliegue, ya que las entregas incrementales de funcionalidades pueden generar problemas de integración si los componentes desarrollados en diferentes Sprints no interactúan adecuadamente[16].

### **6.1.2. Caracterizar los riesgos en el desarrollo de software, mediante la norma ISO 31000.**

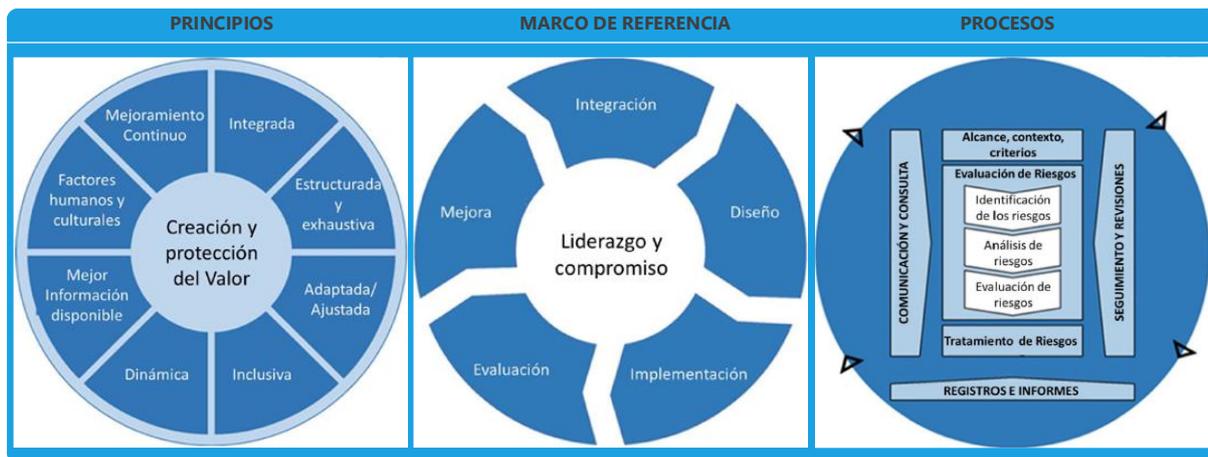
Se realizó una revisión completa de la norma con el objetivo de aplicar parte de sus directrices para la documentación de riesgos de calidad. La fusión de esta normativa con la Revisión Bibliográfica facilitó la detección, documentación y propuesta del marco de trabajo.

#### **6.1.2.1. Seleccionar los puntos relevantes de la normativa aplicable al TT.**

La normativa ISO 31000 en su edición del año 2018 provee lineamientos claros para la Administración o Gestión de Riesgos. La norma explica varias secciones, entre las que se destacan los principios, el marco de referencia y el proceso. Es importante indicar que la norma proporciona un enfoque común aplicable a cualquier tipo de riesgo, independientemente de la industria o el sector, lo que la convierte en una herramienta flexible y adaptable para diversas organizaciones.

Para este TT se utilizó específicamente la sección del “proceso” propuesta en la norma. Los principios y el marco no se consideran debido a que se requiere levantar y documentar adecuadamente los riesgos, pero no es parte de este trabajo generar un sistema que permita gestionarlos. Dada su importancia se presenta una breve descripción de las tres secciones y se desarrollará en este apartado las actividades seleccionadas (Contexto, evaluación y tratamiento de riesgos) correspondiente a la parte de procesos de la norma.

**Los principios.** “Proporcionan orientación sobre las características de una administración/gestión de riesgos efectiva y eficiente, comunicando su valor y explicando su intención y propósito. El propósito de la administración/gestión de riesgos es la creación y la protección del valor” [7]. Aunque estos principios no se implementaron en el TT es importante tomarlos en cuenta cuando se desea implementar un sistema de gestión de riesgos ya que mediante ellos se puede asegurar que toda la organización tenga una cultura orientada a reducir los riesgos, en el contexto de esta TT, los que afectan a la calidad. En la siguiente figura se pueden apreciar los principios que sugiere la norma.



**Figura 9.** Principios, marco de referencia, proceso.  
**Fuente:** Basada en la norma ISO 31000.

**Marco de referencia.** Al igual que los principios, el marco de referencia permite gestionar riesgos en toda la organización. Su propósito consiste en apoyar a las organizaciones en integrar la gestión de riesgos en todas sus actividades y funciones significativas. “La efectividad de la administración/gestión de riesgos dependerá de su integración en la gobernanza de las organizaciones, incluyendo la toma de decisiones” [7]. Es decir, cada departamento o área de la institución debe participar en la gestión de riesgos, de igual manera en los proyectos de Scrum, todos los interesados en el proyecto deben participar activamente en la identificación y tratamiento de los riesgos, esto garantiza que la mayoría estén contemplados y todo el personal este al tanto del nivel de exposición que se tiene en el proyecto. En la figura 9 se muestran los componentes del marco del marco de referencia.

**Proceso.** El proceso de gestión de riesgos “implica la aplicación sistemática de políticas, procedimientos y prácticas a las actividades de comunicación y consulta, establecimiento del contexto y evaluación, tratamiento, seguimiento, revisión, registros y reportes de los riesgos” [7]. Este proceso definido en la norma se acopla a los requerimientos de este TT, permitiendo levantar y documentar de manera precisa los riesgos de calidad.

Al revisar las tres secciones de la normativa, se determinó que la sección de procesos es la más útil para el presente Trabajo de Titulación, esto debido a que una parte de sus actividades tienen como propósito principal realizar la identificación y el tratamiento de los riesgos. Las dos primeras secciones (principios y marco de referencia) tienen mayor relevancia cuando se pretende implementar un esquema de gestión de riesgos completo. Como se explicó en la metodología se eligieron las actividades del Contexto, Evaluación de Riesgos y el Tratamiento

de Riesgos, éste último se utilizó para proponer el marco de trabajo de los riesgos de calidad identificados. En los siguientes apartados se detalla cómo se aplicaron dichas actividades.

### 6.1.2.2. Documentación de caracterización de los riesgos en base a la norma.

En esta sección se indican los resultados de la aplicación de las actividades seleccionadas del proceso indicado en la normativa ISO 31000.

- **Establecer el contexto para la gestión de riesgos.**

Según lo establecido en el proceso de la norma, se documentó el contexto sobre el que deberían estar enmarcados los riesgos de proyectos que utilizan Scrum. Para ello se definieron factores comunes a los proyectos de desarrollo que se alinearan con los riesgos identificados en la Revisión Bibliográfica y que pueden afectar a la calidad del software:

**Tabla 5.** Factores comunes de proyectos Scrum.

Factores	Descripción
1. Estructura del equipo Scrum.	Los equipos son autoorganizados y multidisciplinarios, por lo que ellos mismos deciden cómo abordar las tareas asignadas. Esto fomenta la autonomía y la flexibilidad, puede generar riesgos relacionados con la falta de coordinación o desequilibrio en las habilidades dentro del equipo [9]. Roles definidos: Los roles clave en Scrum (Product Owner, Scrum Master, y Equipo de Desarrollo) también influyen en la gestión de riesgos. La calidad puede verse comprometida si el Product Owner no comunica claramente las prioridades de negocio o si el Scrum Master no facilita adecuadamente las prácticas ágiles. Tanto los equipos autoorganizados como sus roles pueden influir en la calidad si no se gestionan correctamente.
2. Proceso iterativo e incremental.	En Scrum, el desarrollo de software se realiza en ciclos cortos, llamados Sprints, con entregas incrementales de producto [8]. Este enfoque iterativo tiene ventajas, pero también puede introducir riesgos: Los plazos ajustados (cortos), ya que los desarrolladores pueden verse forzados a completar las tareas rápidamente, sacrificando las revisiones y pruebas; pruebas incompletas, el enfoque en la entrega continua de incrementos funcionales puede resultar en menos tiempo para realizar pruebas de calidad en profundidad, lo que puede llevar a la introducción de defectos. A largo plazo, este proceso puede acumular problemas técnicos que comprometen la calidad general del producto.
3. Deuda técnica.	En la búsqueda de cumplir con los plazos de los Sprints, los equipos a menudo sacrifican buenas prácticas de desarrollo, acumulando deuda técnica. Esta deuda técnica se da principalmente cuando se aplazan mejoras o refactorización del código, lo que a largo plazo genera problemas de mantenibilidad, escalabilidad y rendimiento del software debido al código de baja calidad.
4. Comunicación y colaboración.	La colaboración diaria en Scrum es clave para mantener el alineamiento dentro del equipo, ya que se realizan reuniones diarias (Daily Scrum) para discutir el progreso, obstáculos y coordinar el trabajo [9]. Sin embargo, la falta de comunicación efectiva entre el personal del equipo de Scrum o con

		los stakeholders puede generar malentendidos en la interpretación de los requisitos, lo que puede llevar a la entrega de software con baja calidad o con defectos no resueltos. La transparencia y claridad en la comunicación son esenciales para prevenir riesgos de calidad.
5. Cultura ágil y mejora continua.		Scrum promueve una cultura de mejora continua a través de las retrospectivas, donde los equipos revisan sus prácticas y buscan oportunidades de mejora, pero si estas reuniones no se gestionan correctamente o no se toman en serio, puede generarse un estancamiento en la capacidad del equipo para identificar continuamente problemas y soluciones de calidad del producto de manera proactiva.
6. Foco en la entrega funcional.		Scrum se enfoca en la entrega de funcionalidades que agregan valor para el cliente en cada Sprint, este enfoque puede dejar de lado los aspectos no funcionales como el rendimiento, la seguridad y la escalabilidad. Si el equipo prioriza solo las funcionalidades visibles para el usuario y no aborda aspectos no funcionales, el producto final podría no cumplir con los estándares de calidad técnica que aseguran un buen funcionamiento a largo plazo [17].
7. Tamaño y complejidad del proyecto.		A medida que los proyectos Scrum crecen en tamaño y complejidad, es más difícil mantener una coordinación efectiva y una visión clara del producto final entre los equipos y garantizar que todos sigan los mismos estándares de calidad [15] y los criterios de “hecho”. Las dependencias entre los equipos y la gestión de un backlog más amplio pueden aumentar la probabilidad de errores e inconsistencias, lo que afectaría la calidad general del producto.
8. Dependencia del cliente y stakeholders.		El Product Owner en Scrum juega un papel crucial en la priorización de las funcionalidades del backlog del producto basadas en el valor del negocio, es el encargado de representar los intereses del cliente y las partes interesadas, y asegurar que el equipo trabaje en las funcionalidades más valiosas. Sin embargo, si no hay suficiente involucramiento de los stakeholders o de claridad de las prioridades del cliente, los riesgos de calidad pueden aumentar ya que el equipo puede trabajar en funcionalidades que no son cruciales, entregando un producto que no satisface completamente las expectativas del cliente.

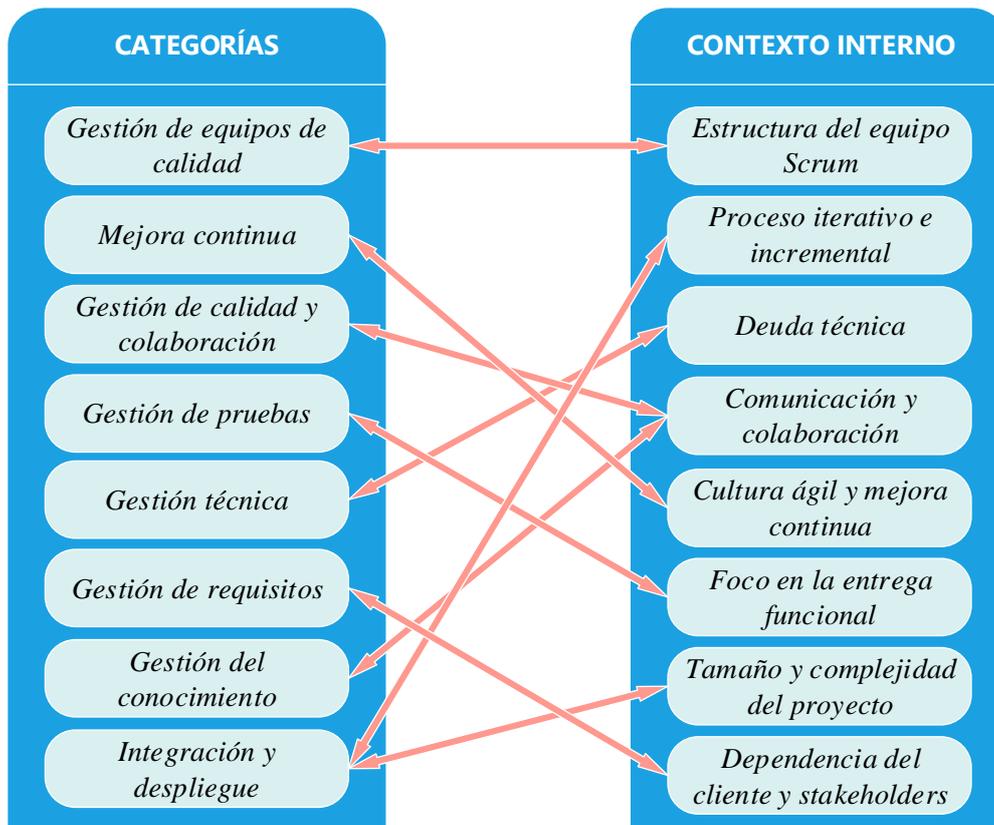
### 6.1.2.3. Análisis y evaluación de riesgos.

- **Identificación de los riesgos de calidad.**

En base a la Revisión Bibliográfica, se identificaron 14 riesgos de calidad de software. Los riesgos se presentan afinados y agrupados por categoría en la tabla 3.

- **Análisis de riesgos.**

La categorización de riesgos se comparó con el contexto definido para proyectos de desarrollo de software que usan Scrum. Esta actividad se realizó para validar que los riesgos detectados están alineados con el contexto de proyectos que usan Scrum, lo que se demuestra en la siguiente figura.



**Figura 10.** Relación entre categorías de riesgos y contexto de Scrum.  
**Fuente:** Elaborada por el tesista.

En base a la figura 10 se puede concluir que los riesgos levantados en la RB, efectivamente pueden afectar la calidad en proyectos Scrum, ya que están alineados con su contexto.

*Probabilidad e Impacto.* Los resultados de la encuesta aplicada para definir la probabilidad e impacto de los riesgos obtenidos (Anexo 2), se puede apreciar en la siguiente tabla:

**Tabla 6.** Datos tabulados de la encuesta.

<b>R1</b>	<b>R2</b>	<b>R3</b>	<b>R4</b>	<b>R5</b>	<b>R6</b>	<b>R7</b>	<b>R8</b>	<b>R9</b>	<b>R10</b>	<b>R11</b>	<b>R12</b>	<b>R13</b>	<b>R14</b>
P	I	P	I	P	I	P	I	P	I	P	I	P	I
3	4	4	3	4	5	5	5	5	5	4	5	5	4
4	3	4	5	5	5	5	5	4	4	5	4	5	3
3	4	3	4	4	3	4	4	3	4	4	4	3	3

Para facilitar la visualización de los datos, se presentan en una matriz de calor con el fin de identificar gráficamente la relación de probabilidad e impacto que tiene cada riesgo, siendo los más críticos los que están cerca de la esquina superior derecha de la imagen en color rojo:

		IMPACTO				
		INSIGNIFICANTE	MENOR	MEDIO	MAYOR	SEVERO
PROBABILIDAD	CASI SEGURO	MEDIO	MEDIO	ALTO	CRITICO R10	CRITICO R4-R5-R6-R8-R11
	PROBABLE	BAJO	MEDIO	MEDIO R2-R14	ALTO R9-R13	CRITICO R3-R7
	MODERADO	BAJO	MEDIO	MEDIO R12	MEDIO R1	ALTO
	IMPROBABLE	MUY BAJO	BAJO	MEDIO	MEDIO	MEDIO
	RARO	MUY BAJO	MUY BAJO	BAJO	BAJO	MEDIO

**Figura 11.** Análisis de probabilidad e impacto.

**Fuente:** Plantilla adaptada por el tesista.

- **Evaluación de riesgos.**

En la figura 11 se puede apreciar riesgos con diferentes niveles de clasificación. Existen 4 riesgos con clasificación media, 2 riesgos con clasificación alta y 8 riesgos con clasificación crítica. De los riesgos marcados como críticos, existen 5 que tienen una probabilidad del 100% de materializarse y un impacto marcado como severo en cuanto a su afectación hacia la calidad del software.

Considerando los aportes realizados por los profesionales se eligieron los 5 riesgos que representan mayor probabilidad e impacto. Los riesgos evaluados se presentan a continuación.

**Tabla 7.** Riesgos críticos que impactan en la calidad del producto.

Riesgo	Nombre	Probabilidad	Impacto	Clasificación
R4	Falta de aprendizaje y prevención de errores.	5	5	Crítico
R5	Gestión ineficaz de la calidad por falta de colaboración.	5	5	Crítico
R6	Retroalimentación incompleta de los stakeholders.	5	5	Crítico
R8	Aumento de defectos por inexperiencia del equipo.	5	5	Crítico
R11	Gestión inadecuada de la deuda técnica.	5	5	Crítico

En base a la experiencia del tesista, se propuso un riesgo más que no se presenta en la investigación realizada y merece atención porque afecta directamente a la calidad del producto: R15: Entrega tardía de componentes para pruebas en cada Sprint. En proyectos de desarrollo de software con Scrum, es común que se asignen 2 semanas a cada Sprint, en las cuáles un tiempo este asignado para construir el software y otro para evaluarlo. Si el tiempo de evaluación se ve reducido por entregas tardías, no se logran ejecutar todos los casos de pruebas previstos, esto podría generar deuda técnica, producto inestable, no hay certeza de que el producto cumple con los requerimientos mínimos esperados por el cliente. La probabilidad e impacto del riesgo está en los umbrales más altos, es decir es casi seguro que ocurra y su impacto es severo. Este riesgo no será incluido en el marco de trabajo final, pero es importante que los equipos de Scrum lo consideren en sus proyectos.

**Tabla 8.** Riesgo de calidad propuesto por el tesista.

Nro.	Categoría	Nombre	Descripción	Probabilidad	Impacto	Clasificación
R15	Gestión de calidad y colaboración.	Entrega tardía de componentes para pruebas en cada Sprint.	Ocurre cuando los desarrolladores no entregan los módulos o funcionalidades dentro del tiempo previsto para su construcción, disminuyendo el tiempo de pruebas; esto impide que el software se evalúe de manera adecuada.	5	5	Crítico

Se definió una tabla de equivalencias de riesgos detectados. Esta tabla permite mantener la trazabilidad del número del riesgo levantados en la revisión bibliográfica y la nueva numeración asignada a cada riesgo.

**Tabla 9.** Tabla de equivalencia de riesgos.

Riesgo	Equivalencia	Nombre
R4	R01	Falta de aprendizaje y prevención de errores.

R5	R02	Gestión ineficaz de la calidad por falta de colaboración.
R6	R03	Retroalimentación incompleta de los stakeholders.
R8	R04	Aumento de defectos por inexperiencia del equipo.
R11	R05	Gestión inadecuada de la deuda técnica.
R15	R06	Entrega tardía de componentes para pruebas en cada Sprint.

#### 6.1.2.4. Tratamiento de riesgos de calidad.

Previo a definir el tratamiento de los riesgos se adoptaron algunas definiciones que permiten caracterizar los riesgos y entender con mayor detalle su comportamiento. Estas definiciones son el evento, la causa y la consecuencia del riesgo.

- **Evento del riesgo:** ¿Qué podría salir mal?

Es una ocurrencia o incidente específico que puede afectar al proyecto o producto de software. Representa la materialización de una situación que puede impactar los objetivos de calidad de software [15]. Es decir, es una ocurrencia específica y bien definida que podría suceder durante el ciclo de vida del desarrollo de software, también se considera como la descripción de lo que podría salir mal en el desarrollo de software en un entorno ágil.

- **Causa:** ¿Por qué podría suceder el evento?

Son las condiciones, circunstancias o factores que pueden dar origen a la ocurrencia de un evento de riesgo. Representan el origen o la fuente que puede desencadenar el riesgo [18]. En proyectos Scrum, una causa común podría ser la presión por cumplir con plazos ajustados, lo que puede llevar a tomar decisiones técnicas apresuradas y a la acumulación de deuda técnica [9].

- **Consecuencia:** ¿Qué impacto tendría si el riesgo se materializa?

Describe los efectos o resultados negativos si el riesgo se materializa. Son los impactos o efectos que resultan si el evento de riesgo llega a materializarse. Pueden ser impactos negativos en términos de calidad, tiempo, costos o funcionalidad del software [19]. En proyectos Scrum, al no gestionan adecuadamente los eventos de riesgo, las consecuencias podrían ser diversas, como la entrega de un producto de baja calidad, retrasos en la entrega del

software, o la insatisfacción del cliente debido a funcionalidades que no cumplen con sus expectativas.

En base a las definiciones previas, se realizó la búsqueda de información en los artículos procesados en la Revisión Bibliográfica para determinar si están definidos la causa, evento y consecuencia de los riesgos. Para fortalecer la información se amplió la búsqueda con herramientas de IA y finalmente se elaboró la matriz de caracterización de riesgos.

Se definieron 5 tipos de eventos con sus respectiva causa e impacto para cada riesgo, esto permite que se puedan tener al alcance varios indicadores que permitan identificar si un riesgo de calidad puede presentarse en un proyecto con Scrum.

Adicionalmente se complementó la matriz en base a la experiencia en proyectos de Scrum que tiene el tesista. La matriz incluye el nuevo riesgo propuesto que hace referencia a la “Entrega tardía de componentes para pruebas en cada Sprint”. Dado que, para todos los riesgos, la probabilidad y el impacto es de 5, se omitieron las columnas correspondientes para facilitar la comprensión de la matriz.

**Tabla 10.** Caracterización de riesgos de calidad.

Nro.	Categoría	Nombre	Descripción	Evento del riesgo	Causa	Consecuencia
R01	Mejora continua.	Falta de aprendizaje y prevención de errores.	Si los errores no se discuten abiertamente y no se aprende de ellos, es probable que se repitan en futuras iteraciones, acumulando defectos y disminuyendo la calidad del producto.	Los errores cometidos en un Sprint se repiten en los siguientes Sprints.	Falta de análisis y discusión de errores en las retrospectivas. No se discuten abiertamente los errores.	Acumulación de defectos, recurrencia de defectos similares.
				No se documentan ni analizan los problemas que ocurren en los Sprints anteriores (Ausencia de análisis post-mortem de fallos).	Equipos sin un enfoque estructurado en la mejora continua. Falta de tiempo dedicado a la reflexión.	Los errores persisten, lo que lleva a problemas recurrentes que impactan el rendimiento del proyecto. Pérdida de oportunidades de mejora.
				Las lecciones aprendidas no se documentan ni se comparten entre los miembros del equipo.	Falta de cultura organizacional orientada al aprendizaje. Presión por comenzar el siguiente Sprint.	Pérdida de conocimiento que limita la capacidad de mejorar la calidad del producto a largo plazo. Repetición de problemas técnicos.
				El equipo no aborda ni corrige los defectos identificados en fases anteriores.	Prioridad en la entrega de nuevas funcionalidades sobre la corrección de errores pasados. Cultura que evita reconocer errores.	Aumento en la deuda técnica, afectando la mantenibilidad del software.
				Los problemas técnicos que surgen en cada iteración no se resuelven de manera efectiva (Omisión de análisis de causa raíz).	Ausencia de procesos de retroalimentación. Equipos con poca disciplina para la refactorización y mejora continua.	Baja calidad en las futuras versiones del producto debido a la acumulación de problemas técnicos no resueltos. Deterioro continuo de la calidad.

R02	Gestión de calidad y colaboración.	Gestión ineficaz de la calidad por falta de colaboración.	La falta de un enfoque colaborativo y de aprendizaje puede dificultar la implementación de prácticas efectivas de gestión de la calidad, resultando en un producto que no cumple con los estándares de calidad esperados.	Falta de coordinación entre los equipos para asegurar la calidad de los entregables. Generación de silos de conocimiento en el equipo	Falta de colaboración entre equipos que trabajan en paralelo. Falta de cultura colaborativa.	Disminución de la cohesión del producto, afectando su estabilidad y funcionalidad. Inconsistencias en la implementación.
				Deficiencias en la integración de módulos debido a la falta de cooperación entre los equipos.	Comunicación deficiente entre los equipos sobre las dependencias técnicas.	Errores de integración que generan defectos en el producto final.
				Los problemas de calidad no son abordados debido a la falta de alineación entre los roles clave del equipo.	Descoordinación entre los diferentes roles (Scrum Master, Product Owner, Desarrolladores).	Aumento en los tiempos de corrección de errores y retrasos en la entrega del producto.
				Falta de revisión cruzada entre los equipos para asegurar la calidad de los entregables. Ausencia de pair programming o revisiones de código superficiales.	Equipos distribuidos con poca colaboración. Resistencia al trabajo en equipo.	Problemas de calidad que afectan la percepción del cliente sobre el producto. Pérdida de oportunidades de mentoring. Deterioro de la calidad del código.
				Equipos que trabajan de manera aislada, sin compartir lecciones aprendidas. Decisiones técnicas no consensuadas.	Poca integración entre equipos en retrospectivas y reuniones de planificación. Roles excesivamente especializados	Pérdida de conocimiento y de oportunidades de mejora en el proceso de desarrollo. Fragmentación del conocimiento técnico.
R03	Gestión de calidad y colaboración.	Retroalimentación incompleta de los stakeholders.	Al no obtener retroalimentación constante y precisa de los usuarios finales o partes interesadas durante cada Sprint, es	Los stakeholders no participan activamente en las revisiones de Sprint, en el refinamiento del producto.	Falta de compromiso por parte de los stakeholders en el proyecto. Stakeholders sobrecargados.	Características desarrolladas que no cumplen con las necesidades o expectativas del cliente. Pérdida de valor del producto.

		posible que se desarrollen características que no cumplan con las expectativas o necesidades del cliente.	La retroalimentación de los stakeholders no se recibe de manera oportuna, se recibe de manera tardía.	Comunicación deficiente entre el equipo y los stakeholders. Canales de comunicación inadecuados.	El equipo trabaja en funcionalidades que no son prioritarias para el cliente. Desarrollo de funcionalidades incorrectas. Retrabajos costosos.	
			Los requerimientos de los stakeholders no están claros ni actualizados. Requisitos mal interpretados.	Los stakeholders no brindan una retroalimentación clara y detallada durante los Sprints. Barreras de comunicación.	Producto final que no refleja las verdaderas necesidades del cliente. Insatisfacción del cliente.	
			El equipo no tiene suficiente información para ajustar el backlog de acuerdo a las prioridades del cliente.	Falta de reuniones efectivas de revisión con los stakeholders.	Retrasos en la implementación de funcionalidades críticas para el cliente.	
			Las prioridades del cliente no se reflejan correctamente en las historias de usuario del backlog. Validación insuficiente con usuarios.	Desconexión entre el Product Owner y los stakeholders durante la planificación de los Sprints. Procesos de feedback no estructurados.	Insatisfacción del cliente debido a la falta de alineación con sus expectativas y necesidades reales.	
R04	Gestión de pruebas.	Aumento de defectos por inexperiencia del equipo.	Los equipos ágiles, especialmente los nuevos, tienden a experimentar un aumento de defectos debido a la falta de experiencia o a una implementación apresurada. Estos defectos pueden afectar la estabilidad y la	El equipo nuevo no sigue las mejores prácticas de desarrollo, lo que genera defectos en el producto. Implementación incorrecta de patrones.	Falta de experiencia en el equipo para implementar Scrum de manera efectiva. Falta de capacitación técnica.	Inestabilidad del sistema. Aumento de defectos en las primeras entregas del producto.
				Errores en manejo de casos de prueba que evalúan los valores límites.	Presión por entregar rápidamente.	Se presentan errores funcionales del sistema cuando se ingresa valores límite.
				Los miembros del equipo no realizan suficientes pruebas	Falta de capacitación en herramientas de pruebas y	Defectos críticos que llegan a producción, afectando la

			funcionalidad del software.	debido a la falta de experiencia con herramientas de pruebas. Código no optimizado.	automatización. Rotación alta de personal.	estabilidad del software. Deterioro del rendimiento.
				El equipo no gestiona correctamente las dependencias entre módulos, generando errores en la integración (Pruebas inadecuadas).	Falta de conocimiento sobre la planificación de dependencias técnicas en Scrum. Falta de documentación técnica.	Problemas de integración entre los diferentes módulos del sistema. Mayor costo de mantenimiento.
				Los plazos no se cumplen debido a la inexperiencia en estimaciones ágiles.	Falta de experiencia en la estimación y planificación de Sprints.	Retrasos en la entrega del producto final, afectando la satisfacción del cliente.
R05	Gestión técnica.	Gestión inadecuada de la deuda técnica.	La necesidad de entregar productos rápidamente puede llevar a decisiones técnicas que no son óptimas, generando deuda técnica. Al no ser gestionadas adecuadamente, los problemas no resueltos en etapas tempranas del proyecto se acumulan, comprometiendo la integridad del código y afectando negativamente el rendimiento, la escalabilidad y la	Postergación continua de refactoring.  Los problemas técnicos se acumulan sin ser abordados, generando defectos en fases posteriores.  Las funcionalidades implementadas en los primeros Sprints no se optimizan adecuadamente (Implementación de soluciones rápidas sin mayor análisis).	Decisiones técnicas apresuradas para cumplir con los plazos del sprint. Presión por nuevas funcionalidades.  Falta de refactorización continua y gestión de deuda técnica en cada Sprint.  Falta de revisiones técnicas en cada Sprint. Priorización incorrecta de la deuda técnica, falta de tiempo para mejoras.	El rendimiento del software se ve comprometido, afectando la experiencia del usuario. Código difícil de mantener.  Aumento de los costos de mantenimiento y mayor esfuerzo en corregir errores a largo plazo.  Dificultades para escalar y mantener el sistema en versiones futuras. Dificultad para evolucionar.  Degradación progresiva del código, lo que dificulta la mantenibilidad del

			mantenibilidad del software.	del	corrección de deuda técnica (como ignorar warnings de código, saltarse estándares de código, acumulación de bugs menores, etc.).	del código, ausencia de políticas de calidad.	software. Velocidad reducida a largo plazo.
					El equipo no tiene visibilidad sobre la deuda técnica acumulada.	Falta de herramientas para medir y rastrear la deuda técnica en el proyecto. Falta de visibilidad de deuda técnica.	Problemas técnicos acumulados que afectan la escalabilidad del sistema. Costos crecientes de desarrollo.
R06	Gestión de calidad y colaboración.	Entrega tardía de componentes para pruebas en cada Sprint.	Ocurre cuando los desarrolladores no entregan los módulos o funcionalidades dentro del tiempo previsto para su construcción, disminuyendo el tiempo de pruebas; esto impide que el software se evalúe de manera adecuada.		Los componentes necesarios para pruebas no están disponibles a tiempo en cada Sprint.	Retrasos en la finalización de las tareas de desarrollo debido a mala planificación.	Atraso en el proceso de pruebas, lo que afecta la calidad del producto final. Incumplimiento de los plazos del proyecto.
					Los desarrolladores no entregan los módulos en el tiempo acordado para ser probados.	Estimaciones inadecuadas del tiempo necesario para el desarrollo de los módulos en el Sprint.	Retrasos acumulados en el proceso de pruebas, afectando los plazos de entrega del producto.
					Las pruebas de integración se retrasan porque los componentes individuales no se completan a tiempo.	Falta de coordinación entre los equipos de desarrollo y pruebas.	Problemas de integración y defectos no detectados hasta etapas avanzadas del proyecto. No se realiza detección temprana de errores críticos y la integración efectiva de los módulos.
					Las pruebas comienzan tarde, lo que impide la detección temprana de errores críticos.	Falta de un proceso de seguimiento efectivo del progreso de desarrollo.	Aumento en la acumulación de defectos no identificados, que afectan la estabilidad y funcionalidad del producto. Las pruebas se realizan

---

		tarde o de manera apresurada.
Los Sprints finalizan sin que las pruebas se completen debido a entregas tardías de los componentes.	Priorización de la entrega rápida de funcionalidades sobre la calidad y el tiempo adecuado para pruebas.	Producto entregado con defectos no detectados debido a la falta de tiempo para realizar pruebas exhaustivas.

---

Como se indicó en la metodología la definición de generalidades de los requisitos y la selección de opciones de tratamiento de los riesgos se desarrollaron en el apartado 6.1.3, ya que es donde se proponen las opciones de tratamiento a los riesgos en proyectos Scrum al definir el marco de trabajo.

### **6.1.3. Proponer el marco de trabajo.**

Para proponer el marco de trabajo que es el entregable principal de este objetivo, se implementó la metodología Design Science Research (DSR). Esta metodología se aplicó en varias secciones del objetivo, por ello únicamente se hará referencia a dicha información en los pasos que se considere necesario.

#### **6.1.3.1. Definición de la estructura del marco de trabajo.**

La estructura del marco de trabajo para los riesgos detectados que afectan directamente a la calidad, se organizó mediante 3 componentes:

1. Información del riesgo.
2. Como identificar el riesgo.
3. Como mitigarlo en las fases de Scrum.

#### **6.1.3.2. Diseño y desarrollo del artefacto.**

El artefacto es el marco que guiará a los equipos Scrum en la identificación y mitigación de los riesgos de calidad. Para realizar el diseño del artefacto, se consideró los riesgos identificados en la Revisión Bibliográfica con la que se creó el primer componente del marco de trabajo que corresponde a la información del riesgo.

El primer componente “**Información del riesgo**” consta de la categoría, el nombre y la descripción del riesgo. Se desarrollo para que el equipo de Scrum tenga información suficiente para analizar la existencia del riesgo de calidad del software en sus proyectos de desarrollo.

El segundo componente **¿Cómo identificar el riesgo?**, está conformado por el evento del riesgo, la causa y la consecuencia, definidos en la sección “Tratamiento de riesgos de calidad”. Se desarrolló para que el equipo de Scrum tenga conocimiento de las alertas que se presentan en los proyectos y que pueden hacer que el riesgo se materialice. Con esta información podrán aplicar los controles necesarios para reducir la probabilidad de ocurrencia del riesgo.

El tercer componente **¿Cómo mitigarlo en las fases de Scrum?**, se organizó de acuerdo a los principales eventos descritos en la guía de Scrum que son el Sprint Planning, Daily Scrum, Sprint Review, Sprint Retrospective [9]. Se desarrollo para indicar mecanismos de mitigación específicos que se podrían ejecutar para detectar y reducir la materialización de los riesgos de calidad.

Un resumen del artefacto de gestión de riesgos de calidad (marco de trabajo) se presenta en la figura 12, incluye la descripción, guías para detectar los riesgos y mecanismos de mitigación que se integra en las diferentes fases de Scrum.



**Figura 12.** Estructura del marco de trabajo propuesto.  
**Fuente:** Elaborada por el tesista.

### 6.1.3.3. Documentación de riesgos en el marco de trabajo.

La versión 1 del marco de trabajo propuesto se documentó considerando la información levantada durante el desarrollo del objetivo 1, esto incluye la Revisión Bibliográfica, la caracterización de riesgos, la retroalimentación del Director del Trabajo de Titulación y del tesista. En la tabla 11 se presenta documentado el marco de trabajo, considerando el componente de mitigación de riesgo en los 4 eventos de Scrum (Sprint Planning, Daily Scrum, Sprint Review, Sprint Retrospective).

**Tabla 11.** Marco de trabajo de riesgos de calidad.

INFORMACIÓN DEL RIESGO						
Nro.	Categoría.	Nombre.	Descripción.			
R01	Mejora continua.	Falta de aprendizaje y prevención de errores.	Si los errores no se discuten abiertamente y no se aprende de ellos, es probable que se repitan en futuras iteraciones, acumulando defectos y disminuyendo la calidad del producto.			
¿CÓMO IDENTIFICAR EL RIESGO?			¿CÓMO MITIGARLO EN LOS EVENTOS DE SCRUM?			
Evento	Causa	Consecuencia	Sprint Planning	Daily Scrum	Sprint Review	Sprint Retrospective
✓ Los errores cometidos en un sprint se están repitiendo en los siguientes sprints.	✓ Falta de análisis y discusión de errores en las retrospectivas. No se discuten abiertamente los errores.	✓ Acumulación de defectos, recurrencia de defectos similares.	✓ Incluir tiempo en el sprint para análisis de errores previos.  ✓ Definir responsables para asegurar la documentación de lecciones aprendidas.	✓ Revisar brevemente los errores detectados, su estatus y cómo se está aprendiendo de ellos.  ✓ Fomentar la comunicación abierta sobre cualquier error para evitar su repetición.	✓ Revisar los errores más críticos detectados en el sprint y discutir cómo evitarlos en el siguiente.  ✓ Incluir stakeholders en la discusión de cómo mejorar la calidad del producto.	✓ Establecer mejoras claras basadas en los errores cometidos en el sprint.  ✓ Crear una lista de acciones específicas que se puedan implementar para prevenir
✓ No se están documentando ni analizando los problemas que están ocurriendo en los sprints (Ausencia de análisis post-mortem de fallos).	✓ Equipos sin un enfoque estructurado en la mejora continua. Falta de tiempo dedicado al análisis de los problemas.	✓ Los problemas se arrastran de un sprint a otro, reduciendo la calidad del producto y perdiendo oportunidades de mejora.				
✓ Las lecciones aprendidas no se están documentando ni compartiendo entre los miembros del equipo.	✓ Falta de cultura organizacional orientada al aprendizaje. Presión por comenzar el siguiente sprint.	✓ Pérdida de conocimiento que limita la capacidad de mejorar la calidad del producto a largo plazo. Repetición de problemas técnicos.				

✓ El equipo no está abordando ni corrigiendo los defectos identificados en fases anteriores.	✓ Prioridad en la entrega de nuevas funcionalidades sobre la corrección de errores pasados. Cultura que evita reconocer errores.	✓ Aumento en la deuda técnica, afectando la mantenibilidad del software.	✓ Planificar revisiones técnicas periódicas.			futuros errores similares.
✓ Los problemas técnicos que están surgiendo en cada iteración no se están resolviendo de manera efectiva (Omisión de análisis de causa raíz).	✓ Ausencia de procesos de retroalimentación. Equipos con poca disciplina para la refactorización y mejora continua.	✓ Baja calidad en las futuras versiones del producto debido a la acumulación de problemas técnicos no resueltos. Deterioro continuo de la calidad.		✓ Solucionar impedimentos relacionados con la corrección de errores.		✓ Comunicar a todo el equipo las acciones de mejora.

### INFORMACIÓN DEL RIESGO

<b>Nro.</b> R02	<b>Categoría.</b> Gestión de calidad y colaboración.	<b>Nombre.</b> Gestión ineficaz de la calidad por falta de colaboración.	<b>Descripción.</b> La falta de un enfoque colaborativo y de aprendizaje puede dificultar la implementación de prácticas efectivas de gestión de la calidad, resultando en un producto que no cumple con los estándares de calidad esperados.
--------------------	---	---	--

#### ¿CÓMO IDENTIFICAR EL RIESGO?

#### ¿CÓMO MITIGARLO EN LOS EVENTOS DE SCRUM?

Evento	Causa	Consecuencia	Sprint Planning	Daily Scrum	Sprint Review	Sprint Retrospective
✓ No está existiendo coordinación entre los equipos para asegurar la calidad de los entregables. Se están generando barreras de conocimiento en el equipo.	✓ Falta de colaboración entre equipos que trabajan en paralelo, falta de cultura colaborativa.	✓ Disminución de la cohesión del producto, afectando su estabilidad y funcionalidad. Inconsistencias en la implementación.	✓ Fomentar la colaboración temprana asignando tareas conjuntas entre roles	✓ Crear un ambiente donde todos los miembros del equipo puedan expresar	✓ Incluir a todos los miembros clave en la revisión del sprint para garantizar que la	✓ Identificar problemas de colaboración específicos y definir planes para mejorar la

✓ Están ocurriendo deficiencias en la integración de módulos debido a la falta de cooperación entre los equipos.	✓ Comunicación deficiente entre los equipos sobre las dependencias técnicas.	✓ Errores de integración que generan defectos en el producto final.	(Product Owner, desarrolladores, testers).	preocupaciones sobre la calidad y colaboración.	calidad sea evaluada desde todas las perspectivas.	comunicación en el siguiente sprint.
✓ Los problemas de calidad no están siendo abordados (falta de alineación entre los roles clave del equipo).	✓ Descoordinación entre los diferentes roles (Scrum Master, Product Owner, Desarrolladores).	✓ Aumento en los tiempos de corrección de errores y retrasos en la entrega del producto.	✓ Definir sesiones de planificación compartida para asegurar alineación de expectativas.	✓ Asegurarse de que las tareas cruzadas entre equipos estén claras.	✓ Revisar el flujo de colaboración entre equipos.	✓ Planificar sesiones de trabajo colaborativas antes del sprint.
✓ Está faltando revisión cruzada entre los equipos para asegurar la calidad de los entregables. No se está practicando pair programming o las revisiones de código están siendo superficiales.	✓ Equipos distribuidos con poca colaboración. Resistencia al trabajo en equipo.	✓ Problemas de calidad que afectan la percepción del cliente sobre el producto. Pérdida de oportunidades de mentoring. Deterioro de la calidad del código.	✓ Definir tiempo para pruebas de integración del software desarrollado por diferentes equipos.			
✓ Los equipos están trabajando de manera aislada, sin estar compartiendo lecciones aprendidas o tomando decisiones técnicas no consensuadas.	✓ Poca integración entre equipos en retrospectivas y reuniones de planificación. Roles excesivamente especializados	✓ Pérdida de conocimiento y de oportunidades de mejora en el proceso de desarrollo. Fragmentación del conocimiento técnico.				

### INFORMACIÓN DEL RIESGO

Nro.	Categoría.	Nombre.	Descripción.
R03.	Gestión de calidad y colaboración.	Retroalimentación incompleta de los stakeholders.	Si no se obtiene retroalimentación constante y precisa de los usuarios finales o partes interesadas durante cada Sprint, es posible que se desarrollen características que no cumplan con las expectativas o necesidades del cliente.

¿CÓMO IDENTIFICAR EL RIESGO?			¿CÓMO MITIGARLO EN LOS EVENTOS DE SCRUM?			
Evento	Causa	Consecuencia	Sprint Planning	Daily Scrum	Sprint Review	Sprint Retrospective
✓ Los stakeholders no están participando activamente en las revisiones de sprint ni en el refinamiento del producto.	✓ Falta de compromiso por parte de los stakeholders en el proyecto o Stakeholders sobrecargados.	✓ Características desarrolladas que no cumplen con las necesidades o expectativas del cliente. Pérdida de valor del producto.	✓ Asegurarse de que el Product Owner se comunique regularmente con los stakeholders para priorizar los requisitos.	✓ Mantener informados a los stakeholders del progreso a través de actualizaciones periódicas.	✓ Asegurar la participación activa de los stakeholders en la revisión del sprint.	✓ Evaluar si la retroalimentación de los stakeholders fue clara y suficiente.
✓ La retroalimentación de los stakeholders no se está recibiendo de manera oportuna, se está recibiendo de manera tardía.	✓ Comunicación deficiente entre el equipo y los stakeholders. Canales de comunicación inadecuados.	✓ El equipo trabaja en funcionalidades que no son prioritarias para el cliente. Desarrollo de funcionalidades incorrectas, retrabajos costosos.	✓ Documentar y revisar los requisitos detalladamente durante la planificación.	✓ Discutir cualquier cambio solicitado por los stakeholders durante las reuniones diarias.	✓ Revisar si se cumplieron las expectativas del cliente y planificar mejoras.	✓ Definir un plan para mejorar la comunicación con los stakeholders en sprints futuros.
✓ Los requerimientos de los stakeholders no están siendo claros ni están siendo actualizados. Los requisitos están siendo mal interpretados.	✓ Los stakeholders no brindan una retroalimentación clara y detallada durante los sprints. Existen barreras de comunicación.	✓ Producto final que no refleja las verdaderas necesidades del cliente. Insatisfacción del cliente.	✓ Gestionar que el Product Owner disponga de un tiempo diario asignado al proyecto para resolver dudas.			
✓ El equipo no está teniendo suficiente información para ajustar el backlog de acuerdo con las prioridades del cliente.	✓ Falta de reuniones efectivas de revisión con los stakeholders.	✓ Retrasos en la implementación de funcionalidades críticas para el cliente.				
✓ Las prioridades del cliente no se están reflejando correctamente en las historias de usuario del backlog. Está	✓ Desconexión entre el Product Owner y los stakeholders durante la planificación de los sprints.	✓ Insatisfacción del cliente debido a la falta de alineación con sus expectativas y necesidades reales.				

existiendo una validación insuficiente con los usuarios.	Procesos de feedback no estructurados.					
--	--	--	--	--	--	--

### INFORMACIÓN DEL RIESGO

Nro.	Categoría.	Nombre.	Descripción.
R04.	Gestión de pruebas.	Aumento de defectos por inexperiencia del equipo.	Los equipos ágiles, especialmente los nuevos, tienden a experimentar un aumento de defectos debido a la falta de experiencia o a una implementación apresurada. Ocurre algo similar cuando se tiene poco conocimiento del giro del negocio, de metodologías ágiles o del área de pruebas. El aumento de defectos puede afectar la estabilidad y la funcionalidad del software.

#### ¿CÓMO IDENTIFICAR EL RIESGO?

#### ¿CÓMO MITIGARLO EN LOS EVENTOS DE SCRUM?

Evento	Causa	Consecuencia	Sprint Planning	Daily Scrum	Sprint Review	Sprint Retrospective
✓ El equipo nuevo no está siguiendo las mejores prácticas de desarrollo, generando defectos en el producto. Por ejemplo, están implementando patrones de manera incorrecta.	✓ Falta de experiencia en el equipo para implementar Scrum de manera efectiva. Falta de capacitación técnica.	✓ Inestabilidad del sistema. Aumento de defectos en las primeras entregas del producto.	✓ Planificar tareas de menor complejidad para los miembros con menos experiencia.	✓ Monitorear el progreso de los miembros con menos experiencia para proporcionar soporte temprano.	✓ Incluir una revisión técnica detallada de los entregables realizados por los miembros más inexpertos.	✓ Planificar capacitaciones adicionales para los miembros del equipo que necesiten mejorar sus habilidades.
Se están cometiendo errores en el manejo de casos de prueba que evalúan los valores límites.	✓ Presión por entregar rápidamente.	✓ Los defectos son encontrados por el usuario final en producción. Se presentan errores funcionales del sistema cuando se ingresa valores límite.	✓ Incluir revisiones más frecuentes de los entregables	✓ Facilitar el apoyo de miembros más	✓ Identificar conjuntamente con los involucrados en	✓ Discutir qué problemas se presentaron por

<p>✓ Los miembros del equipo no están realizando suficientes pruebas debido a la falta de experiencia con herramientas de pruebas o en conocimiento del negocio, lo que está generando código no optimizado.</p>	<p>✓ Falta de capacitación en herramientas de pruebas y automatización. Existe rotación alta de personal. Inducción no adecuada al proyecto del nuevo personal.</p>	<p>✓ Defectos críticos que llegan a producción, afectando la estabilidad del software. Deterioro del rendimiento.</p>	<p>durante el sprint.</p> <p>✓ Planificar capacitaciones o tutorías.</p>	<p>experimentados durante el desarrollo diario.</p> <p>✓ Considerar el uso de técnicas de monitoreo de equipo como: Tablero Kanban Visual, el Burn-Down Chart, el Monitoreo de Impedimentos y el Recuento Diario de Historias de Usuario Completadas.</p>	<p>el Sprint Review las oportunidades de mejora de las funcionalidades presentadas.</p>	<p>la falta de experiencia y cómo mitigarlos.</p>
<p>✓ El equipo no está gestionando correctamente las dependencias entre módulos, lo que está generando errores en la integración (pruebas inadecuadas).</p>	<p>✓ Falta de conocimiento sobre la planificación de dependencias técnicas en Scrum, falta de documentación técnica.</p>	<p>✓ Problemas de integración entre los diferentes módulos del sistema. Mayor costo de mantenimiento.</p>				
<p>✓ Los plazos no se están cumpliendo debido a la inexperiencia en estimaciones ágiles.</p>	<p>✓ Falta de experiencia en la estimación y planificación de sprints.</p>	<p>✓ Retrasos en la entrega del producto final, afectando la satisfacción del cliente.</p>				

### INFORMACIÓN DEL RIESGO

Nro.	Categoría.	Nombre.	Descripción.
R05.	Gestión técnica.	Gestión inadecuada de la deuda técnica.	La necesidad de entregar productos rápidamente puede llevar a decisiones técnicas que no son óptimas, generando deuda técnica. Al no ser gestionada adecuadamente, los problemas no resueltos en etapas tempranas del proyecto se acumulan, comprometiendo la integridad del código y afectando negativamente el rendimiento, la escalabilidad y la mantenibilidad del software.

¿CÓMO IDENTIFICAR EL RIESGO?			¿CÓMO MITIGARLO EN LOS EVENTOS DE SCRUM?			
Evento	Causa	Consecuencia	Sprint Planning	Daily Scrum	Sprint Review	Sprint Retrospective
✓ Se está postergando continuamente la refactorización.	✓ Decisiones técnicas apresuradas para cumplir con los plazos del sprint. Presión por nuevas funcionalidades.	✓ El rendimiento del software se ve comprometido, afectando la experiencia del usuario. Código difícil de mantener.	✓ Priorizar la deuda técnica como parte del backlog del sprint. ✓ Considerar tiempos de holgura para solución de deuda técnica generada en cada sprint. ✓ Asegurar que se asignen tiempos específicos para la refactorización y reducción de deuda técnica en cada sprint. ✓ Planificar la instalación y configuración	✓ Hacer seguimiento de las tareas de refactorización y reducción de deuda técnica.  ✓ Discutir cualquier impedimento que pueda retrasar la reducción de la deuda técnica durante el sprint.	✓ Evaluar los avances en la reducción de la deuda técnica. ✓ Identificar la deuda técnica más crítica, mediante la revisión del impacto funcional, la revisión de código y pruebas, retroalimentación de los stakeholders, etc.  ✓ Priorizar la deuda técnica más crítica que aún no ha sido atendida para el próximo sprint.	✓ Discutir cómo la deuda técnica afectó el sprint y planificar cómo se gestionará en el futuro.  ✓ Establecer métricas para medir la reducción de deuda técnica en cada sprint.
✓ Los problemas técnicos se están acumulando sin ser abordados, generando defectos en fases posteriores.	✓ Falta de refactorización continua y gestión de deuda técnica en cada sprint.	✓ Aumento de los costos de mantenimiento y mayor esfuerzo en corregir errores a largo plazo.				
✓ Las funcionalidades implementadas en los primeros sprints no se están optimizando adecuadamente (se están implementando soluciones rápidas sin mayor análisis).	✓ Falta de revisiones técnicas en cada sprint. Priorización incorrecta de la deuda técnica, falta de tiempo para mejoras.	✓ Dificultades para escalar y mantener el sistema en versiones futuras. Dificultad para evolucionar.				
✓ El equipo de desarrollo está priorizando la entrega de nuevas funcionalidades sobre la corrección de la deuda técnica (como ignorar warnings de código, saltarse estándares de código, o acumular bugs menores, etc.).	✓ Equipos con poca disciplina para equilibrar entre entrega de funcionalidades y calidad del código. Ausencia de políticas de calidad.	✓ Degradación progresiva del código, lo que dificulta la mantenibilidad del software. Velocidad reducida a largo plazo (rendimiento).				

<p>✓ El equipo no está teniendo visibilidad sobre la deuda técnica acumulada.</p>	<p>✓ Falta de herramientas para medir y rastrear la deuda técnica en el proyecto.</p>	<p>✓ Problemas técnicos acumulados que afectan la escalabilidad del sistema. Costos crecientes de desarrollo.</p>	<p>de herramientas que permitan gestionar la deuda técnica.</p>			
---	---	---	---	--	--	--

Aunque el marco de trabajo proporciona elementos guía para la mitigación de los riesgos en proyectos de Scrum, se recomienda complementar con un **plan de tratamiento** que aplique específicamente al proyecto en ejecución.

Para tener éxito en la aplicación del marco de trabajo se sugiere realizar un monitoreo y revisión continua [7], ya que los riesgos deben ser **monitoreados constantemente** en cada Sprint, asegurando que las medidas de mitigación aplicadas en cada evento de Scrum sean efectivas y que se identifiquen nuevos riesgos de calidad a medida que avanza el proyecto.

## **6.2. Objetivo 2: Evaluar el marco de trabajo propuesto utilizando la norma ISO/IEC 25022.**

Este objetivo pretende evaluar la calidad en el uso del marco de trabajo, para ello se utiliza parte de la norma ISO/IEC 25022. La calidad en uso puede medirse y evaluarse mediante el efecto que tiene en los usuarios cuando utilizan el marco de trabajo durante las pruebas.

El alcance de este objetivo es determinar si el marco de trabajo está documentado adecuadamente, es comprensible y fácil de usar. En base a lo indicado el artefacto queda listo para evaluar su aplicación en proyectos reales con Scrum.

### **6.2.1. Definición del modelo de calidad en Uso basado en la ISO/IEC 25022.**

La normativa ISO/IEC 25022 [5], define la Calidad en Uso como el “Grado en el que un producto o sistema puede ser utilizado por usuarios específicos para satisfacer sus necesidades y lograr objetivos específicos con eficacia, eficiencia, satisfacción y libre de riesgos en contextos de uso específicos”.

#### **6.2.1.1. Seleccionar los puntos relevantes de la normativa aplicable al TT.**

Para evaluar el uso del marco de trabajo se eligió el punto 8 “medidas de calidad en uso” de la normativa ISO/IEC 25022, el cual consta de 5 características descritas en la tabla 12. Esta normativa es ampliamente utilizada cuando se requiere evaluar la calidad que tienen las aplicaciones o artefactos al ser utilizados por el usuario final.

**Tabla 12.** Características de calidad seleccionadas para evaluar el artefacto “Norma ISO/IEC 25022”.

<b>Características</b>	<b>Subcaracterísticas</b>	<b>Definición</b>
1. Eficacia.	Tareas completas. Objetivos logrados. Los errores en una tarea.	“Las medidas de eficacia evalúan la precisión y la integridad con la que los usuarios logran objetivos específicos” [5].

2. Eficiencia.	Tiempo de tareas. Eficiencia del tiempo. La rentabilidad.	“Las medidas de eficiencia evalúan los recursos gastados en relación con la precisión e integridad con la que los usuarios logran sus objetivos” [5].
3. Satisfacción.	Utilidad. Confianza. Comodidad.	“Las medidas de satisfacción evalúan el grado en que se satisfacen las necesidades del usuario cuando un producto o sistema se utiliza en un contexto de uso específico” [5].
4. Libertad de Riesgo.	Reducción de riesgos económicos. Reducción de riesgos de seguridad y salud. Reducción de riesgos del ambiente.	“Las medidas de ausencia de riesgo evalúan el grado en que la calidad de un producto o sistema mitiga o evita riesgos potenciales para el usuario, organización o proyecto, incluidos riesgos para el estado económico, la vida humana, la salud o el medio ambiente” [5].
5. Cobertura del Contexto.	Integridad del contexto. Flexibilidad.	“Las medidas de cobertura del contexto evalúan el grado en que un producto o sistema puede usarse con eficacia, eficiencia, satisfacción y ausencia de riesgos tanto en contextos de uso específicos como en contextos más allá de los inicialmente identificados explícitamente” [5].

#### 6.2.1.2. Definir la matriz de modelo de calidad en uso basado en la ISO/IEC 25022.

Se adaptaron las preguntas que estaban orientadas a evaluar software, para que se ajusten a la evaluación del marco de trabajo. Mediante estas preguntas se evaluó las características seleccionadas de la normativa. A continuación, se presenta un resumen de las preguntas diseñadas para evaluar cada característica.

**Tabla 13.** Preguntas por subcaracterística para evaluar el artefacto.

Características	Subcaracterísticas	Preguntas
1. Eficacia.	Tareas completas.	¿Considerando los riesgos evaluados, cuantas instrucciones pudo comprender sin asistencia?
	Objetivos logrados.	¿Las instrucciones del marco de trabajo han logrado cumplir con el objetivo propuesto para cada riesgo?
	Los errores en una tarea.	¿Cuántas instrucciones de identificación y mitigación han tenido al menos 1 error en su ejecución/comprensión?
2. Eficiencia.	Tiempo de tareas.	¿Cuánto tiempo se tarda en comprender las instrucciones para gestionar un riesgo en comparación con lo planeado?
	Eficiencia del tiempo.	¿Cuánto tiempo necesita un usuario normal para comprender las instrucciones para gestionar un riesgo en comparación con un experto?
	La rentabilidad.	¿Ha notado una reducción en el tiempo y esfuerzo necesarios para identificar y mitigar riesgos de calidad desde que comenzó a usar este marco?

3. Satisfacción.	Utilidad.	¿Está satisfecho con el uso del marco de trabajo para la gestión de riesgos de calidad de software en proyectos que usan Scrum?
	Confianza.	¿Detecto al menos una inconsistencia (considerada como impedimento) en las instrucciones para mitigar riesgos en los eventos de Scrum? o ¿Existe alguna instrucción con la que no esté de acuerdo?
	Comodidad.	¿Considera que el marco de trabajo presenta facilidad y poco esfuerzo en su uso?
4. Libertad de Riesgo.	Reducción de Riesgos Económicos.	¿Considera que el marco ayuda a reducir los costos de re-trabajo al prevenir problemas de calidad desde etapas tempranas?
	Reducción de Riesgos de Seguridad y Salud.	¿Considera que el marco ayuda a reducir la presión relacionada con la entrega de productos de calidad, proporcionando instrucciones claras y anticipando los problemas?
	Reducción de Riesgos del ambiente.	¿Confía en que el marco facilita un ambiente seguro y propicio para trabajar en un entorno ágil, reduciendo riesgos que podrían afectar la productividad y calidad?
5. Cobertura del Contexto.	Integridad del contexto.	¿El marco de trabajo propuesto para proyectos que usan Scrum puede ser utilizado con facilidad de uso aceptable?
	Flexibilidad.	¿En qué medida el producto puede utilizarse en contextos adicionales de uso?, por ejemplo, puede usarse en metodologías tradicionales, híbridas.

Para realizar la ponderación (peso de cada categoría), se analizó la importancia que tiene cada característica en el contexto del uso del marco de trabajo para la gestión de riesgos de calidad en Scrum, las ponderaciones asignadas se pueden observar en la figura 13. A continuación, se explica las ponderaciones asignadas para cada característica:

**Eficacia.** A esta categoría se le otorgó un valor del 30% de peso total. Se pondero principalmente porque permitió medir como se completan las tareas relacionadas con la gestión de riesgos y que tan fácil son de alcanzar los objetivos (comprensión de instrucciones del marco). Se asignó un peso alto porque la comprensión de instrucciones y el cumplimiento de objetivos son factores clave en procesos de gestión de riesgos.

**Eficiencia.** A esta categoría se le otorgó un 30% del peso total. Esta medida se relaciona con el uso de recursos (tiempo, esfuerzo, costos) utilizados para ejecutar las instrucciones del marco de trabajo. Se asigno un peso igual a la eficacia dada la importancia que tiene cumplir las instrucciones, pero en tiempos óptimos. En proyectos Scrum son importantes tanto la eficacia como la eficiencia dado que se tienen tiempos limitados.

**Satisfacción.** A esta categoría se le otorgó un peso del 25% del total. Esto debido a la importancia de que el usuario confíe en los resultados del marco y por la relevancia de la medida de percepción de utilidad que representa esta característica. Aunque no tiene el peso más alto es importante porque la adopción del modelo depende de la percepción de los usuarios.

**Libertad de riesgo.** Para esta categoría se le otorgó un peso del 5% del total. Considerando que la característica evalúa si el marco de trabajo reduce riesgos adicionales que podrían surgir durante su uso, como por ejemplo económicos, de seguridad y salud del personal. La matriz propuesta no causa un impacto significativo en los temas mencionados, por tal razón su ponderación es menor.

**Cobertura de contexto.** Para esta categoría se le otorgó un peso del 10% del total. La característica evalúa la adaptación del marco de trabajo a los diferentes contextos de Scrum y también si es flexible para adaptarse a otros contextos. En base a lo indicado se asignó un peso moderado ya que es importante conocer si el marco es adaptable, pero sin descuidar su eficacia y eficiencia.

ISO/IEC 25010 - ISO/IEC 25022			
Un modelo de calidad en uso compuesto de cinco características (algunas de las cuales son además subdivididas en subcaracterísticas) que se relacionan con el resultado de la interacción cuando un producto es utilizado en un contexto particular de uso.			
<b>Caso de estudio:</b>	<b>Marco de trabajo para la gestión de riesgos de calidad.</b>		
MODELO DE CALIDAD EN USO			
Características	Subcaracterísticas	Peso categoría	Observación
<b>Eficacia</b>	Tareas completas	30%	
	Objetivos logrados		
	Los errores en una tarea		
<b>Eficiencia</b>	Tiempo de tareas	30%	
	Eficiencia del tiempo		
	La rentabilidad		
<b>Satisfacción</b>	Utilidad	25%	
	Confianza		
	Comodidad		
<b>Libertad de Riesgo</b>	Reducción de Riesgos Económicos	5%	
	Reducción de Riesgos de Seguridad y Salud		
	Reducción de Riesgos del ambiente		
<b>Cobertura del Contexto</b>	Integridad del contexto	10%	
	Flexibilidad		
<b>Total</b>		<b>100%</b>	

**Figura 13.** Ponderación de características para evaluar la calidad en uso.  
**Fuente:** Adaptada por el tesista.

## 6.2.2. Evaluación del marco de trabajo.

### 6.2.2.1. Aplicación de la matriz de calidad en Uso.

La evaluación se aplicó mediante un taller, ejecutado con una muestra por conveniencia de 17 estudiantes entre 8vo y 9no ciclo de la carrera computación de la Universidad Nacional de Loja (Ver anexo 3), 13 de los cuales están realizando su trabajo de titulación y el resto están cursando la asignatura Software Quality en la que están ejecutando un proyecto siguiendo los lineamientos de Scrum.

Se aplicó la matriz para evaluar los riesgos y sus instrucciones de manera individual, facilitando las pruebas y el control del taller. Los resultados de la evaluación se presentan a continuación para cada riesgo:

- R01: Falta de aprendizaje y prevención de errores.

<b>Tabla de resultados</b>			
<b>Características</b>	<b>Subcaracterísticas</b>	<b>Peso categoría</b>	<b>Resultado Característica</b>
<b>Eficacia</b>	Tareas completas	30%	27.99
	Objetivos logrados		
	Los errores en una tarea		
<b>Eficiencia</b>	Tiempo de tareas	30%	27.26
	Eficiencia del Tiempo		
	La rentabilidad		
<b>Satisfacción</b>	Utilidad	25%	22.41
	Confianza		
	Comodidad		
<b>Libertad de Riesgo</b>	Reducción de Riesgos Económicos	5%	4.81
	Reducción de Riesgos de Seguridad y Salud		
	Reducción de Riesgos del ambiente		
<b>Cobertura del Contexto</b>	Integridad del contexto	10%	9.14
	Flexibilidad		
	<b>Total</b>	<b>100%</b>	<b>↑ 91.62</b>

**Figura 14.** Ponderación de características para evaluar la calidad en uso R01.

**Fuente:** Adaptada por el tesista.

- R02: Gestión ineficaz de la calidad por falta de colaboración.

<b>Tabla de resultados</b>			
<b>Características</b>	<b>Subcaracterísticas</b>	<b>Peso categoría</b>	<b>Resultado Característica</b>
<b>Eficacia</b>	Tareas completas	30%	28.93
	Objetivos logrados		
	Los errores en una tarea		
<b>Eficiencia</b>	Tiempo de tareas	30%	26.48
	Eficiencia del Tiempo		
	La rentabilidad		
<b>Satisfacción</b>	Utilidad	25%	23.41
	Confianza		
	Comodidad		
<b>Libertad de Riesgo</b>	Reducción de Riesgos Económicos	5%	4.79
	Reducción de Riesgos de Seguridad y Salud		
	Reducción de Riesgos del ambiente		
<b>Cobertura del Contexto</b>	Integridad del contexto	10%	9.68
	Flexibilidad		
<b>Total</b>		<b>100%</b>	<b>↑ 93.30</b>

**Figura 15.** Ponderación de características para evaluar la calidad en uso R02.

**Fuente:** Adaptada por el tesista.

- R03: Retroalimentación incompleta de los stakeholders.

<b>Tabla de resultados</b>			
<b>Características</b>	<b>Subcaracterísticas</b>	<b>Peso categoría</b>	<b>Resultado Característica</b>
<b>Eficacia</b>	Tareas completas	30%	27.86
	Objetivos logrados		
	Los errores en una tarea		
<b>Eficiencia</b>	Tiempo de tareas	30%	26.91
	Eficiencia del Tiempo		
	La rentabilidad		
<b>Satisfacción</b>	Utilidad	25%	22.18
	Confianza		
	Comodidad		
<b>Libertad de Riesgo</b>	Reducción de Riesgos Económicos	5%	4.86
	Reducción de Riesgos de Seguridad y Salud		
	Reducción de Riesgos del ambiente		
<b>Cobertura del Contexto</b>	Integridad del contexto	10%	9.29
	Flexibilidad		
<b>Total</b>		<b>100%</b>	<b>↑ 91.10</b>

**Figura 16.** Ponderación de características para evaluar la calidad en uso R03.

**Fuente:** Adaptada por el tesista.

- R04: Aumento de defectos por inexperiencia del equipo.

<b>Tabla de resultados</b>			
<b>Características</b>	<b>Subcaracterísticas</b>	<b>Peso categoría</b>	<b>Resultado Característica</b>
<b>Eficacia</b>	Tareas completas	<b>30%</b>	<b>28.10</b>
	Objetivos logrados		
	Los errores en una tarea		
<b>Eficiencia</b>	Tiempo de tareas	<b>30%</b>	<b>27.94</b>
	Eficiencia del Tiempo		
	La rentabilidad		
<b>Satisfacción</b>	Utilidad	<b>25%</b>	<b>22.76</b>
	Confianza		
	Comodidad		
<b>Libertad de Riesgo</b>	Reducción de Riesgos Económicos	<b>5%</b>	<b>4.81</b>
	Reducción de Riesgos de Seguridad y Salud		
	Reducción de Riesgos del ambiente		
<b>Cobertura del Contexto</b>	Integridad del contexto	<b>10%</b>	<b>9.53</b>
	Flexibilidad		
<b>Total</b>		<b>100%</b>	<b>↑ 93.14</b>

**Figura 17.** Ponderación de características para evaluar la calidad en uso R04.  
**Fuente:** Adaptada por el tesista.

- R05: Gestión inadecuada de la deuda técnica.

<b>Tabla de resultados</b>			
<b>Características</b>	<b>Subcaracterísticas</b>	<b>Peso categoría</b>	<b>Resultado Característica</b>
<b>Eficacia</b>	Tareas completas	<b>30%</b>	<b>27.86</b>
	Objetivos logrados		
	Los errores en una tarea		
<b>Eficiencia</b>	Tiempo de tareas	<b>30%</b>	<b>27.15</b>
	Eficiencia del Tiempo		
	La rentabilidad		
<b>Satisfacción</b>	Utilidad	<b>25%</b>	<b>23.35</b>
	Confianza		
	Comodidad		
<b>Libertad de Riesgo</b>	Reducción de Riesgos Económicos	<b>5%</b>	<b>6.91</b>
	Reducción de Riesgos de Seguridad y Salud		
	Reducción de Riesgos del ambiente		
<b>Cobertura del Contexto</b>	Integridad del contexto	<b>10%</b>	<b>9.27</b>
	Flexibilidad		
<b>Total</b>		<b>100%</b>	<b>↑ 94.55</b>

**Figura 18.** Ponderación de características para evaluar la calidad en uso R05.  
**Fuente:** Adaptada por el tesista.

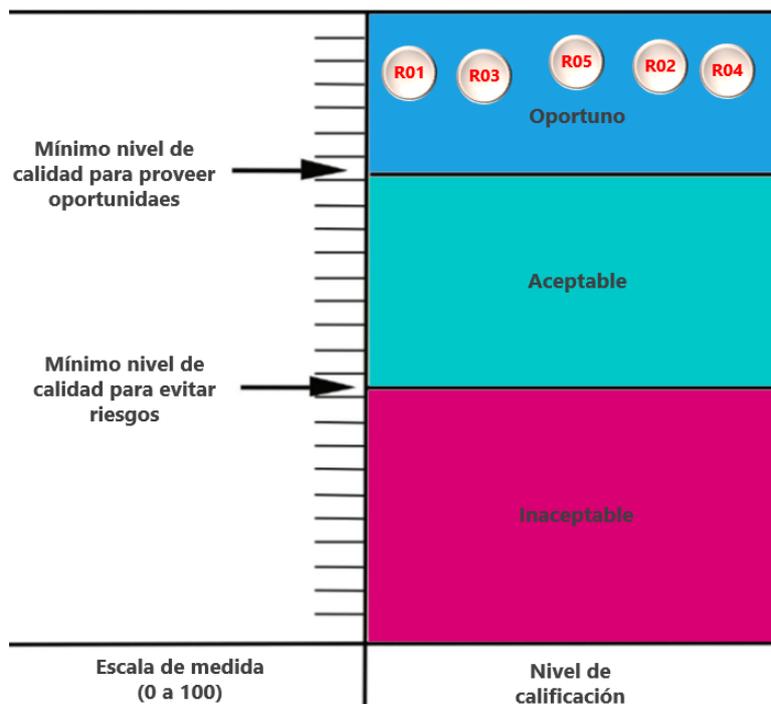
### 6.2.2.2. Medición e interpretación de la calidad en Uso.

Los resultados obtenidos muestran un nivel de calidad en uso superior al 91 % para cada uno de los riesgos evaluados. El resultado se puede visualizar en forma resumida en la siguiente figura:



**Figura 19.** Resultados obtenidos en el taller de evaluación de la calidad en uso.  
**Fuente:** Elaborada por el tesista.

En el taller se determinó que todos los riesgos se encuentran en un nivel superior a los límites definidos en la norma ISO/IEC 25022, por lo que se determina que están listos para ser evaluados en proyectos reales. En la siguiente figura se muestra el resultado de la evaluación en base a la escala de “Riesgos y oportunidades asociados con el nivel de calidad” propuesto en la norma:



**Figura 20.** Riesgos y oportunidades asociados con el nivel de calidad para cada riesgo.  
**Fuente:** Elaborada por el tesista.

En el taller se presentaron algunas observaciones reportadas por los evaluadores, las más importantes se presentan a continuación de acuerdo al análisis realizado en el anexo 3:

- Aclaración de términos en caso de que el personal que utilice el marco de trabajo no esté familiarizado o tenga poca experiencia con proyectos de Scrum.
- Escribir la sección de eventos en presente continuo, para hacer énfasis que es algo que está sucediendo actualmente.
- Se sugieren también mejora en la redacción de algunas instrucciones y agregar otras.

### **6.2.3. Actualización de modelo final del marco de trabajo.**

Se realizó las correcciones pertinentes en el marco de trabajo según las observaciones reportadas por los evaluadores, esto favoreció para que el uso del marco de trabajo sea más fluido, facilitando la comprensión de las instrucciones descritas.

Para solucionar los problemas relacionados con la terminología, se diseñó una sección complementaria al marco de trabajo llamada “Términos clave”, en la que se explica de manera resumida la definición de algunos términos importantes que deben tener claro los usuarios antes de aplicar el marco de trabajo, ver figura 21. Esta sección es considerada como opcional para los usuarios que tienen experiencia en proyectos con Scrum.

## TÉRMINOS CLAVE

**Postmortem de fallos.** Es una reunión o análisis realizado después de un fallo en un proyecto, con el fin de identificar las causas, aprender de los errores y prevenir su repetición en el futuro.

**Product Owner (PO).** En Scrum es el responsable de gestionar el backlog del producto, priorizando las tareas según el valor para el negocio y asegurar que el equipo entregue el producto deseado.

**Cultura que evita reconocer errores.** Hace referencia a un ambiente de trabajo donde los miembros del equipo no se sienten cómodos señalando o aceptando errores, obstaculizando la mejora continua y el aprendizaje colectivo.

**Sprint.** Ciclo corto de trabajo en Scrum, típicamente de 1 a 4 semanas, donde el equipo se enfoca en completar tareas específicas para entregar un incremento del producto.

**Valores límite.** Condiciones extremas o escenarios específicos que se utilizan para evaluar si el sistema funciona correctamente bajo situaciones fuera de lo común o cerca de sus límites operativos.

**Backlog.** Lista priorizada de tareas, requisitos o historias de usuario que representan el trabajo pendiente por realizar en un proyecto con Scrum.

**Refactorización.** Es el proceso de mejorar el diseño, la estructura o la legibilidad del código o cualquier artefacto sin alterar su funcionalidad, con el objetivo de aumentar su mantenibilidad o comprensión.

**Burn-Down Chart.** Gráfico que muestra el trabajo restante en un Sprint frente al tiempo disponible, ayudando al equipo a monitorear su progreso hacia los objetivos del Sprint.

**Deuda técnica.** Es la acumulación de problemas en el código por decisiones técnicas que no son óptimas, tomadas para acelerar entregas, pero que pueden generar costos adicionales y problemas de calidad.

**Dependencias técnicas.** Relaciones entre componentes, equipos o tareas de un proyecto que requieren que ciertas actividades o entregables estén completados antes de que otros puedan avanzar.

**Figura 21.** Términos clave para comprender el marco de trabajo.

**Fuente:** Elaborada por el tesista.

Se actualizó y afino el marco de trabajo con las observaciones adicionales realizadas por los usuarios. Es importante mencionar que algunas observaciones estaban redactadas como comentarios, especialmente relacionados con la flexibilidad del modelo para adaptarse a otros entornos diferentes de Scrum, estos comentarios no se implementaron en el marco de trabajo (Ver [anexo 3](#)). En la figura 22, se presenta la propuesta formal del marco de trabajo para el riesgo 01, el marco de trabajo para todos los riesgos se encuentra en el [anexo 4](#):

# MARCO DE GESTIÓN DE RIESGOS DE CALIDAD DE SOFTWARE PARA PROYECTOS CON SCRUM

INFORMACIÓN DEL RIESGO		
CATEGORÍA	NOMBRE	DESCRIPCIÓN
<b>Mejora continua.</b> Engloba los riesgos relacionados con la falta de iniciativa para proponer mejoras o aprender de los errores, ya que los proyectos de Scrum trabajan con una iteración constante y la mejora continua. Si los equipos no fomentan una cultura de aprendizaje y no implementan mejoras de manera regular, los errores tienden a repetirse.	<b>R01</b> <b>FALTA DE APRENDIZAJE Y GESTIÓN DE ERRORES.</b>	Si los errores no se discuten abiertamente y no se aprende de ellos, es probable que se repitan en futuras iteraciones, acumulando defectos y disminuyendo la calidad del producto.
	<b>PROBABILIDAD</b> Casi seguro.	

¿CÓMO IDENTIFICAR EL RIESGO?		
EVENTO	CAUSA	CONSECUENCIA
Los errores cometidos en un sprint se están repitiendo en los siguientes sprints.	Falta de análisis y discusión de errores en las retrospectivas. No se discuten abiertamente los errores.	Acumulación de defectos, recurrencia de defectos similares.
Las lecciones aprendidas no se están documentando ni compartiendo entre los miembros del equipo.	Falta de cultura organizacional orientada al aprendizaje. Presión por comenzar el siguiente sprint.	Pérdida de conocimiento que limita la capacidad de mejorar la calidad del producto a largo plazo. Repetición de problemas técnicos.
No se están documentando ni analizando los problemas que están ocurriendo en los sprints (Ausencia de análisis post-mortem de fallos).	Equipos sin un enfoque estructurado en base a la mejora continua. Falta de tiempo dedicado al análisis de los problemas.	Los problemas se arrastran de un sprint a otro, reduciendo la calidad del producto y perdiendo oportunidades de mejora.
El equipo no está abordando ni corrigiendo los defectos identificados en fases anteriores.	Prioridad en la entrega de nuevas funcionalidades sobre la corrección de errores pasados. Cultura que evita reconocer errores.	Aumento en la deuda técnica, afectando la mantenibilidad del software.
Los problemas técnicos que están surgiendo en cada iteración no se están resolviendo de manera efectiva (Omisión de análisis de la causa raíz).	Ausencia de procesos de retroalimentación. Equipos con poca disciplina para la refactorización y mejora continua.	Baja calidad en las futuras versiones del producto debido a la acumulación de problemas técnicos no resueltos. Deterioro continuo de la calidad.

¿CÓMO MITIGAR EL RIESGO?			
SPRINT PLANNING	DAILY SCRUM	SPRINT REVIEW	SPRINT REVIEW
<ul style="list-style-type: none"> <li>Incluir tiempo en el sprint para análisis de errores previos.</li> <li>Definir responsables para asegurar la documentación de lecciones aprendidas.</li> <li>Planificar revisiones técnicas periódicas.</li> </ul>	<ul style="list-style-type: none"> <li>Revisar brevemente los errores detectados, su estatus y cómo se está aprendiendo de ellos.</li> <li>Fomentar la comunicación abierta sobre cualquier error para evitar su repetición.</li> <li>Solucionar impedimentos relacionados con la corrección de errores.</li> </ul>	<ul style="list-style-type: none"> <li>Revisar los errores más críticos detectados en el sprint y discutir cómo evitarlos en el siguiente.</li> <li>Incluir a los stakeholders en la discusión de cómo mejorar la calidad del producto.</li> </ul>	<ul style="list-style-type: none"> <li>Establecer mejoras claras basadas en los errores cometidos en el sprint.</li> <li>Crear una lista de acciones específicas que se puedan implementar para prevenir futuros errores similares.</li> <li>Comunicar a todo el equipo las acciones de mejora.</li> </ul>



**Figura 22.** Marco de trabajo propuesto para el riesgo 01.  
**Fuente:** Elaborada por el tesista.

## 7. Discusión

**Objetivo 1: Diseñar el marco de trabajo basado en la Revisión Bibliográfica y en el diagnóstico de principales riesgos que afectan la calidad de software en proyectos que usan Scrum.**

El objetivo 1 permite diseñar el marco de trabajo que está fundamentado en la revisión bibliográfica, alineada con las prácticas internacionales de gestión de riesgos definidas en la norma ISO 31000 y el involucramiento de 15 profesionales relacionados con el desarrollo de software para realizar la priorización de los riesgos mediante el uso de herramientas de colaboración participativa. En base a lo indicado se comprueba que el uso de estos tres enfoques asegura un proceso completo para identificar, analizar y priorizar los riesgos de calidad en el contexto de proyectos con Scrum.

El aporte de la revisión bibliográfica es significativo para determinar la situación actual de los riesgos que podrían afectar a la calidad. Se determinó que no existen artículos que indican específicamente riesgos que afectan directamente a la calidad de software en proyectos Scrum, sin embargo, se encontró tres textos [20], [21], [14] que se acercan en este sentido, mencionando riesgos que por su descripción se asocian directamente a la calidad. Se identificaron 187 riesgos iniciales de los cuales 22 se relacionan directamente con la calidad, estos proporcionan una base para comprender los posibles riesgos, convirtiéndose en un hallazgo importante para proyectos desarrollados con Scrum, donde la calidad puede comprometerse por múltiples factores.

La aplicación de la ISO 31000 proporciona un marco sólido para el levantamiento de riesgos, lo que garantiza que el proceso de identificación sea metódico, repetible y adaptable. El uso de esta norma permite corroborar si los riesgos identificados en la Revisión Bibliográfica están relacionados con el contexto de desarrollo de software con Scrum, también facilito el establecimiento de un lenguaje común para describir eventos, causas y consecuencias, facilitando la comprensión y comunicación de los riesgos. Es importante indicar como lo explica en [7] que la ISO 31000 proporciona un enfoque común para administrar / gestionar cualquier tipo de riesgo y que puede adaptarse a cualquier proyecto y contexto, sin embargo, con el desarrollo de este Trabajo de Titulación se puede inferir que su aplicación puede presentar desafíos a los equipos de desarrollo debido a la naturaleza iterativa y flexible de los proyectos gestionados con Scrum, esto se relaciona directamente con lo explicado en [1], en donde se sostiene que “el desarrollo de proyectos Scrum de gran escala presenta desafíos únicos

en la gestión de riesgos, especialmente las dificultades que se enfrentan en la colaboración de equipos, donde no existe un procedimiento particular para comunicarse y cooperar”. Adicionalmente en [22] se explica que el desarrollo ágil de software no sigue estrictamente un estilo tradicional de gestión de proyectos; por lo tanto, las fases de la gestión de riesgos tampoco están tan bien establecidas.

La realización de una encuesta a profesionales con experiencia, involucrados en el desarrollo de software con metodologías ágiles permite priorizar los riesgos de calidad más críticos. Este paso fue fundamental para adaptar el marco de trabajo a la realidad de Scrum y permitió incluir una perspectiva más práctica de profesionales. Las respuestas de los profesionales permiten detectar los 5 riesgos más críticos que se presentan cotidianamente en los enfoques ágiles, lo que refuerza la aplicabilidad del marco en proyectos reales con Scrum. Es importante considerar que en este tipo de encuestas se presenta un posible sesgo, por las experiencias personales de los encuestados o el tamaño de la muestra, lo que podría limitar la generalización de los resultados.

En [6] se indica que muchos estudios han producido un marco de gestión de riesgos para Scrum en los últimos años, y también explica que “repetir el proceso de análisis de riesgos y seleccionar una respuesta al riesgo se convierte en una carga para las partes interesadas, por lo que se necesita un marco que pueda convertirse en un sistema de apoyo para ayudar a la toma de decisiones”. En consecuencia, el marco de trabajo propuesto proporciona 5 riesgos de calidad ya identificados y priorizados, evitando tener que asignar tiempo valioso del proyecto para levantar los riesgos. La estructura propuesta se compone de 3 partes en las que no solo se describe al riesgo, además, se indica varias formas de identificarlos en el proyecto y las posibles acciones de mitigación que se pueden aplicar en proyectos con Scrum.

Al concluir el objetivo se puede determinar que la combinación de la ISO 31000, la revisión bibliográfica y la encuesta a expertos aporta solidez tanto al enfoque como a los resultados, permitiendo generar un marco de trabajo inicial que será evaluado en base al alcance del objetivo 2.

## **Objetivo 2: Evaluar el marco de trabajo propuesto utilizando la norma ISO/IEC 25022.**

La normativa ISO/IEC 25022 indica que “Las medidas de calidad se pueden poner en práctica de diferentes maneras adaptadas a las necesidades de las partes interesadas que las utilizan”, en base a esta definición se opta por medir la calidad del marco de trabajo

considerando la escala de “Riesgos y oportunidades asociados con el nivel de calidad” [5]propuesto en la normativa, lo cual afirma que se utiliza un enfoque metodológico sólido para garantizar la calidad en uso del artefacto desarrollado (Marco de trabajo).

La medición de la calidad en uso del marco de trabajo para la gestión de riesgos, con 17 evaluadores (alumnos de ciclos superiores de la Carrera Computación de la Universidad Nacional de Loja), se realiza a través de la planificación y ejecución de un taller para aplicación de la normativa, con la finalidad de obtener una perspectiva práctica sobre la aplicabilidad y usabilidad del marco, además, de conocer cómo funcionaría en un entorno real. El aporte del grupo de alumnos, proporciona una retroalimentación valiosa debido a su conocimiento técnico y uso del framework de trabajo Scrum dentro de los modelos ágiles aplicados en trabajos académicos, su conocimiento técnico genera observaciones en los 5 riesgos evaluados; las observaciones permitieron mejorar y perfeccionar el marco de trabajo en varios aspectos clave, pero de manera relevante en la claridad y practicidad ya que se optimizó la sección de documentación de los riesgos, la sección de identificación y la sección de mitigación para que fueran más comprensibles y fáciles de ejecutar. Para la sección de mitigación en los eventos de Scrum, no solo se planteó una acción para mitigar los riesgos, se plantearon entre 2 y 4 acciones que se pueden ejecutar en cada evento y para cada riesgo, mejorando con esto otro aspecto clave que es la adaptabilidad a los diferentes contextos de Scrum. Estas mejoras garantizan que la propuesta del marco de trabajo puede ser utilizado como una herramienta confiable para administrar riesgos de calidad en proyectos gestionados a través de Scrum. Los 5 riesgos evaluados en el taller son:

- Falta de aprendizaje y prevención de errores.
- Gestión ineficaz de la calidad por falta de colaboración.
- Retroalimentación incompleta de los stakeholders.
- Aumento de defectos por inexperiencia del equipo.
- Gestión inadecuada de la deuda técnica.

Mediante este Trabajo de Titulación, se corrobora que la normativa permite realizar una evaluación basada en métricas objetivas, asegurando la confiabilidad de los resultados obtenidos en el taller. Entre las características de la normativa están la eficacia y la satisfacción, la medida de estas características garantizan que el marco abordara tanto objetivos técnicos como la percepción de satisfacción de los usuarios.

Se determina que la estructura propuesta para el marco de trabajo diseñado con 3 secciones son la respuesta a la pregunta de investigación ¿Qué riesgos de calidad de software son más críticos en los proyectos de desarrollo gestionados con Scrum y cómo puede un marco de trabajo basado en la norma ISO 31000 y la ISO/IEC 25022 mitigar los 5 riesgos de mayor impacto al proyecto? En la estructura se indica que primero se debe conocer la existencia de los riesgos, para lo que se proporciona información del riesgo (Nombre, descripción), su probabilidad de ocurrencia y su impacto; segundo, conocer como identificar los riesgos, conociendo los eventos mediante los que se manifiestan en los proyectos, las causas y las consecuencias que tendrían de materializarse; y tercero, conocer las posibles acciones que permitan mitigar los riesgos en los diferentes eventos de Scrum que son el Sprint Planning, el Daily Scrum, el Sprint Review, y la Sprint Retrospective.

Para alcanzar el objetivo de mitigación de riesgos de calidad del marco de trabajo, se ejecutan las siguientes actividades principales: en el objetivo 1 se prioriza con profesionales los cinco riesgos de calidad de mayor importancia en Scrum apoyando la factibilidad de crear el marco de trabajo en base a la norma ISO 31000; en el objetivo 2 se puso a prueba el marco de trabajo mediante la aplicación de la normativa ISO/IEC 25022 obteniendo resultados de calidad en uso entre el 91.10 y el 94.55 por ciento para los riesgos de calidad evaluados, estos porcentajes corresponden a la categoría de Oportuno en la escala de “Riesgos y oportunidades asociados con el nivel de calidad”. Según [5] la escala proporciona una vista gráfica del nivel mínimo de calidad para brindar oportunidades y del nivel mínimo de calidad para evitar riesgos. Los resultados de la evaluación del marco de trabajo están sobre el valor de los dos niveles indicados como se puede apreciar en la figura 20, dando una respuesta favorable de que la mitigación propuesta para los 5 riesgos cumple con las expectativas de los usuarios.

Finalmente, al término del Trabajo de Titulación se determina que al fusionar las metodologías y normativas para armar el marco de trabajo propuesto se pueden mitigar los riesgos de calidad en proyectos con Scrum.

## 8. Conclusiones

- Mediante la Revisión Bibliográfica se identifican 187 riesgos, de los cuales 22 afectan directamente la calidad del software en proyectos Scrum. Tras un proceso de refinamiento, se redujeron a 14 riesgos. Esto demuestra la validez de utilizar una Revisión Bibliográfica como método para identificar riesgos de calidad de software en proyectos que emplean Scrum.
- De los 14 riesgos identificados en la Revisión Bibliográfica, se priorizaron 5 riesgos de calidad clave en proyectos que utilizan Scrum, mediante una encuesta realizada a profesionales del desarrollo de software. Los riesgos priorizados fueron:
  1. Falta de aprendizaje y prevención de errores.
  2. Gestión ineficaz de la calidad por falta de colaboración.
  3. Retroalimentación incompleta de los stakeholders.
  4. Aumento de defectos por inexperiencia del equipo.
  5. Gestión inadecuada de la deuda técnica.
- La puesta en práctica del modelo para su evaluación a través de un taller permitió comprobar que cumple con las características clave de calidad en uso, como la eficacia y satisfacción que percibe el usuario mientras identifica y prioriza los riesgos de calidad en proyectos gestionados con Scrum. La evaluación presenta los siguientes resultados al medir la calidad en uso para cada riesgo: R01: 91.62%, R02: 93.30%, R03: 91.10%, R04: 93.14%, R05: 94.55%. En base a las medidas indicadas la propuesta del marco de trabajo para la gestión de riesgos presenta una aceptación que recae en el nivel de “oportuno” en base a la escala de “Riesgos y oportunidades asociados con el nivel de calidad” de la norma ISO/IEC 25022.
- Las observaciones propuestas por los evaluadores mejoran la usabilidad y comprensión del marco de trabajo, liberándolo para ser avaluado en proyectos reales con Scrum. El marco final propone acciones para afrontar los 5 riesgos más críticos que se presentan en los proyectos de Scrum, priorizados por profesionales del campo de ingeniería de software, cada riesgo está conformado por 3 secciones que facilitan la gestión de los riesgos de calidad de software: “Conozca el riesgo que puede presentarse en su proyecto”, “Comprenda ¿cómo identificarlo?” y “Comprenda ¿cómo mitigarlo?”. Cada una de estas secciones poseen información puntual que permite reducir los riesgos de calidad de software en los eventos de

Scrum (Planning, el Daily Scrum, el Sprint Review, y la Sprint Retrospective). Con la priorización de los riesgos y la estructura liberada del marco de trabajo se da respuesta a la pregunta planteada en el Trabajo de Titulación.

- El uso de la metodología Design Science Research (DSR), permite definir un marco de trabajo para la gestión de riesgos de calidad. Esto demuestra que es factible proponer un marco de trabajo que fusiona la Revisión Bibliográfica, las directrices de la normativa ISO 31000, la normativa ISO/IEC 25022 y su alineación para ser aplicados en proyectos de desarrollo de software que se gestionan con Scrum.

## 9. Recomendaciones

- Ampliar el marco de trabajo propuesto para incluir los 9 riesgos adicionales de calidad que se detectaron en la Revisión Bibliográfica y 1 riesgo propuesto por el tesista, ya que, según la evaluación la probabilidad de estos riesgos se encuentra entre moderado y casi seguro; y el impacto, entre medio y severo como se puede apreciar en la figura 11.
- Evaluar cómo la implementación práctica del marco de trabajo afecta la calidad del software en proyectos reales gestionados con Scrum, esto permitirá medir la aplicabilidad y obtener retroalimentación de los usuarios del marco.
- Aunque la Revisión Bibliográfica permitió levantar los riesgos de calidad reportados en proyectos Scrum, se recomienda ampliar el levantamiento de información con profesionales que estén aplicando procesos de pruebas. Esta actividad podría confirmar o ampliar los riesgos detectados para mejorar el marco de trabajo.
- El marco de trabajo proporciona instrucciones o elementos guía para la mitigación de los riesgos en los eventos de Scrum, pero se recomienda complementar el marco con procesos integrales de gestión de riesgos de software y analizar el contexto de cada proyecto antes de aplicar los lineamientos de mitigación. Estos complementos hacen que la aplicación del marco de trabajo sea más efectiva.
- Usar el marco de trabajo propuesto en el medio académico para fortalecer el conocimiento de los alumnos y docentes de la carrera de computación de la Universidad Nacional de Loja.

## 10. Bibliografía

- [1] S. K. Khurana, M. A. Wassay, and K. Verma, “A Review on Risk Management Framework for large scale scrum”, in *Proceedings - 2022 International Conference on Computational Modelling, Simulation and Optimization, ICCMSO 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 394–400. doi: 10.1109/ICCMSO58359.2022.00082.
- [2] S. Chaouch, A. Mejri, and S. A. Ghannouchi, “A framework for risk management in Scrum development process”, in *Procedia Computer Science*, Elsevier B.V., 2019, pp. 187–192. doi: 10.1016/j.procs.2019.12.171.
- [3] Software Engineering Institute, “CMMI ® para Desarrollo, Versión 1.3 Equipo del Producto CMMI”, 2010. [Online]. Available: <http://www.sei.cmu.edu>
- [4] A. Alami and O. Krancher, “How Scrum adds value to achieving software quality?”, *Empir Softw Eng*, vol. 27, no. 7, Dec. 2022, doi: 10.1007/s10664-022-10208-4.
- [5] International Organization for Standardization, “ISO 25022 Systems and Software Engineering-Systems and Software Quality Requirements and Evaluation (SQuaRE)-Measurement of Quality in Use”, Jun. 2016. [Online]. Available: [www.iso.org](http://www.iso.org)
- [6] M. Pilliang, Munawar, B. Tjahjono, P. Sejati, H. Akbar, and G. Firmansyah, “Criticism of the Risk Management Process in Scrum Methodology”, in *Proceedings - IEIT 2022: 2022 International Conference on Electrical and Information Technology*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 338–343. doi: 10.1109/IEIT56384.2022.9967893.
- [7] International Organization for Standardization, “Norma Internacional ISO 31000. Administración / Gestión de Riesgos”, Feb. 2018. [Online]. Available: [www.iso.org](http://www.iso.org)
- [8] K. Schwaber; J. Sutherland, “La Guía Definitiva de Scrum Tm: Las Reglas del Juego”, Nov. 2017. [Online]. Available: <http://creativecommons.org/licenses/by-sa/4.0/legalcodeandalsodescribedinsummaryformathttp://creativecommons.org/licenses/by-sa/4.0/>.Byutilizing

- [9] K. Schwaber; J. Sutherland, “La Guía de Scrum”, Nov. 2020, [Online]. Available: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-European.pdf>
- [10] L. Olsina, B. Rivera, F. Papa, and P. Becker, “Proceso de Design Science Research Aplicado a la Construcción de una Ontología de Testing de Software como Artefacto”, 2020. [Online]. Available: <http://repositorio.unlam.edu.ar/Pág:1Artículooriginal>
- [11] D. T. Absari, A. Djunaidy, and T. D. Susanto, “Design Science Research Methodology Concept and Its Application in the Development of Online Community Communication System for Selotapak Village”, in *Proceedings - 9th International Conference on Information Technology Computer and Electrical Engineering (ICITACEE)*, 2022, pp. 236–241. doi: 10.1109/ICITACEE55701.2022.9924092.
- [12] B. Kitchenham and S. M. Charters, “Guidelines for performing Systematic Literature Reviews in Software Engineering”, 2007. [Online]. Available: <https://www.researchgate.net/publication/302924724>
- [13] O. Pastor, A. Segooa, and J. I. Panach, “Teaching Design Science as a Method for Effective Research Development”, 2024.
- [14] A. Alami, M. Zahedi, and O. Krancher, “The role of psychological safety in promoting software quality in agile teams”, *Empir Softw Eng*, vol. 29, no. 5, p. 119, Sep. 2024, doi: 10.1007/s10664-024-10512-1.
- [15] I. Sommerville, *Ingeniería de Software*, Novena. México: Pearson, 2011.
- [16] D. Leffingwell, “Praise for agile software requirements”, 2010. [Online]. Available: [www.enter-agile.com](http://www.enter-agile.com)
- [17] R. S. Pressman and B. R. Maxim, *Software Engineering: A Practitioner’s Approach*, 8th ed. New York: McGraw-Hill Education, 2015.
- [18] Project Management Institute, *Practice Standard for Project Risk Management*. Filadelfia: PMI Publications, 2009.
- [19] International Organization for Standardization, “ISO/IEC/IEEE-16085 Systems and Software Engineering - Life Cycle Processes - Risk Management”, Jan. 2021. [Online]. Available: [www.iso.org](http://www.iso.org)

- [20] M. Hammad, I. Inayat, and M. Zahid, “Risk management in agile software development: A survey”, in *Proceedings - 2019 International Conference on Frontiers of Information Technology, FIT 2019*, Institute of Electrical and Electronics Engineers Inc., Dec. 2019, pp. 162–166. doi: 10.1109/FIT47737.2019.00039.
- [21] B. G. Tavares, C. E. Sanches da Silva, and A. D. de Souza, “Risk management analysis in Scrum software projects”, *International Transactions in Operational Research*, vol. 26, no. 5, pp. 1884–1905, Sep. 2019, doi: 10.1111/itor.12401.
- [22] N. Shehzad, M. Aqeel Iqbal, and M. Amjad, “A Review of Risk Management in Agile Development”, in *2022 International Conference on Digital Transformation and Intelligence, ICDI 2022 - Proceedings*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 63–68. doi: 10.1109/ICDI57181.2022.10007239.

## **11. Anexos**

### **Anexo 1. Revisión Bibliográfica.**

# Revisión Bibliográfica

Francisco Javier Alvarez Pineda

22 de septiembre de 2024

### **Resumen**

Este documento presenta una revisión bibliográfica realizada con Parsifal. La revisión proporciona una visión general de riesgos de calidad del producto asociados a proyectos de desarrollo de software que utilizaron Scrum. Para realizar la revisión se tomó como referencia los lineamientos propuestos por Barbara Kitchenham, pero no se los ejecutó en su totalidad. Como resultado de la investigación se detectaron 18 documentos que presentan información relevante sobre los riesgos. De manera general se pudo determinar que Scrum no tiene un proceso formal para gestionar riesgos, también se encontraron especificados varios riesgos que son comunes en diferentes proyectos que usan Scrum.

# Índice

<b>1. Introducción</b>	<b>3</b>
<b>2. Metodología</b>	<b>3</b>
2.1. PICOC	3
2.2. Pregunta(s) de Investigación	3
2.3. Estrategia de Búsqueda	4
2.4. Criterios de Inclusión y Exclusión	5
2.5. Proceso de Selección de Estudios	5
2.6. Evaluación de la Calidad de los Estudios	5
2.7. Extracción de Datos	6
2.8. Síntesis de Resultados	6
<b>3. Resultados</b>	<b>7</b>
3.1. Estudios Incluidos	7
3.2. Resultados de las preguntas de Investigación	9
3.3. Riesgos detectados de calidad de software	13
3.4. Riesgos detectados que podrían afectar la calidad de software	15
3.5. Limitaciones de los Hallazgos	16
<b>4. Conclusiones</b>	<b>16</b>
<b>5. Referencias</b>	<b>17</b>

# 1. Introducción

Mediante la revisión bibliográfica se selecciona la información relevante relacionada con la aplicación de Scrum en proyectos de desarrollo de software, particularmente los riesgos asociados al usar este marco de trabajo. El objetivo principal es identificar los riesgos relacionados con la calidad de software al aplicar Scrum en proyectos de desarrollo. Los resultados de la revisión son importantes porque permitirán estructurar una base de información científica que será analizada para mejorar el nivel de calidad del software mediante el marco de trabajo que se propondrá

# 2. Metodología

Para la revisión bibliográfica se optó por adaptar los principales lineamientos de la metodología propuesta por Barbara Kitchenham para realizar revisiones sistemáticas de literatura, acompañada de la herramienta Parsifal para el registro de los resultados. Se consideraron los siguientes puntos.

## 2.1. PICOC

En esta revisión sistemática, el enfoque PICOC se define de la siguiente manera:

- **Población (Population):** [Proyectos de desarrollo de software.]
- **Intervención (Intervention):** [Determinar riesgos de calidad en la aplicación de metodologías ágiles con Scrum.]
- **Comparador (Comparator):** [No se aplican comparaciones]
- **Resultados (Outcomes):** [Identificación de riesgo de calidad en el desarrollo de software con Scrum.]
- **Contexto (Context):** [Industria de tecnología de software.]

## 2.2. Pregunta(s) de Investigación

La pregunta de investigación que guía esta revisión sistemática es:

- P1: ¿Cuáles son los riesgos de calidad de software identificados en proyectos de desarrollo que utilizan Scrum?
- P2: ¿Cuáles son los riesgos que pueden influir en la calidad de software en proyectos de desarrollo que utilizan Scrum?

## 2.3. Estrategia de Búsqueda

La estrategia de búsqueda consiste en la definición de términos relacionados con las preguntas de investigación, armado de la cadena de búsqueda y la aplicación en las diferentes bases de datos bibliográficas:

- IEEE Digital Library (<http://ieeexplore.ieee.org>)
- Science@Direct (<http://www.sciencedirect.com>)
- Scopus (<http://www.scopus.com>)
- Springer (<http://link.springer.com>)

Los términos se unificaron para formar cadenas de búsqueda específicas para cada base de datos:

- *IEEE Digital Library*

("projects software development.ºR "management projects.ºR "project.ºR "software development.ºR .agile methodologies") AND ("scrum.ºR "scrum agile development.ºR "scrum methodology.ºR "scrum project") AND ("scrum risks.ºR "development risk.ºR "project risk.ºR risk management.ºR "software risks").

- *Science@Direct*

("projects software development.ºR "software development") AND ("scrum.ºR "Software Quality Risks.ºR "scrum project") AND ("scrum risks.ºR"development risk.ºR "Quality Risks.ºR "software risks").

- *Scopus*

TITLE-ABS-KEY ( "projects software development.ºR "management projects.ºR "project.ºR "software development.ºR .agile methodologies") AND TITLE-ABS-KEY ( "scrum.ºR "scrum agile development.ºR "scrum methodology.ºR "scrum project") AND TITLE-ABS-KEY ( "scrum risks.ºR "scrum methodology.ºR "scrum project") AND TITLE-ABS-KEY ( "development risk.ºR "project risk.ºR "project risk.ºR risk management.ºR "software risks .ºR "quality risks").

- *Springer*

(software AND quality AND scrum AND risk), adicionalmente se seleccionó los siguientes filtros: Solo artículos, computer science, software engineering/Programing and Operating System, Programing Languages, Compilers, Interpreters.

## 2.4. Criterios de Inclusión y Exclusión

Se definió criterios de selección para obtener estudios primarios. El propósito es excluir estudios ajenos a la pregunta de investigación.

### Criterios de inclusión:

- Estudios que abordan riesgos de calidad del software en proyectos que utilizan Scrum.
- Estudios que abordan riesgos que influyen en la calidad de software en proyectos que utilizan Scrum.

### Criterios de exclusión:

- Estudios no accesibles.
- Estudios publicados antes del 2019.
- Estudios que no mencionan Scrum.
- Estudios que usan Scrum en otros ámbitos distintos al desarrollo de software.
- Estudios que mencionan Scrum, pero no abordan riesgos.
- Publicaciones duplicadas.
- Publicaciones en idiomas diferentes al Inglés y Español.

## 2.5. Proceso de Selección de Estudios

El proceso de selección de estudios involucró las siguientes etapas:

1. Evaluación de títulos y resúmenes.
2. Aplicación de los criterios de exclusión.
3. Revisión de textos completos.
4. Selección de estudios relevantes.

## 2.6. Evaluación de la Calidad de los Estudios

En la evaluación de calidad de los estudios se aplicaron los criterios de evaluación de Kitchenham, los cuales se pueden visualizar en la siguiente tabla:

<b>Criterio</b>	<b>Pregunta</b>	<b>Evaluación</b>
QA1	¿El autor identifica o detalla riesgos calidad en proyectos de software desarrollados con Scrum?	SI= 1, NO=0, Probablemente=0.5
QA2	¿El autor identifica o detalla riesgos que pueden influir en la calidad de software en proyectos de desarrollo que utilizan Scrum?	SI= 1, NO=0, Probablemente=0.5

Cuadro 1: Preguntas y parámetros usados en la evaluación de calidad

## 2.7. Extracción de Datos

Se extrajeron los siguientes datos de cada estudio seleccionado:

- Información general (autor, año, título).
- Riesgos de calidad de software en proyectos de desarrollo con Scrum.
- Riesgos que afectan la calidad de software en proyectos desarrollados con Scrum.

## 2.8. Síntesis de Resultados

Los resultados generales de la aplicación de la planificación realizada mediante la herramienta parsifal, presentaron un total de 144 estudios relacionados con la pregunta de investigación. Un resumen de la cantidad de estudios analizados se presenta a continuación:

<b>BD Científicas</b>	<b>Documentos incluidos</b>	<b>Documentos excluidos</b>	<b>Estudios seleccionados para revisión</b>
IEEE Xplorer	22	6	16
ScienceDirect	24	11	13
Scopus	25	22	3
Springer	73	63	10
<b>TOTAL</b>	<b>144</b>	<b>102</b>	<b>42</b>

Cuadro 2: Resultado de aplicación de criterios de inclusión y exclusión

En la selección de los estudios se tomó en cuenta los criterios de inclusión y exclusión, en base al título, resumen y palabras claves de los documentos obtenidos.

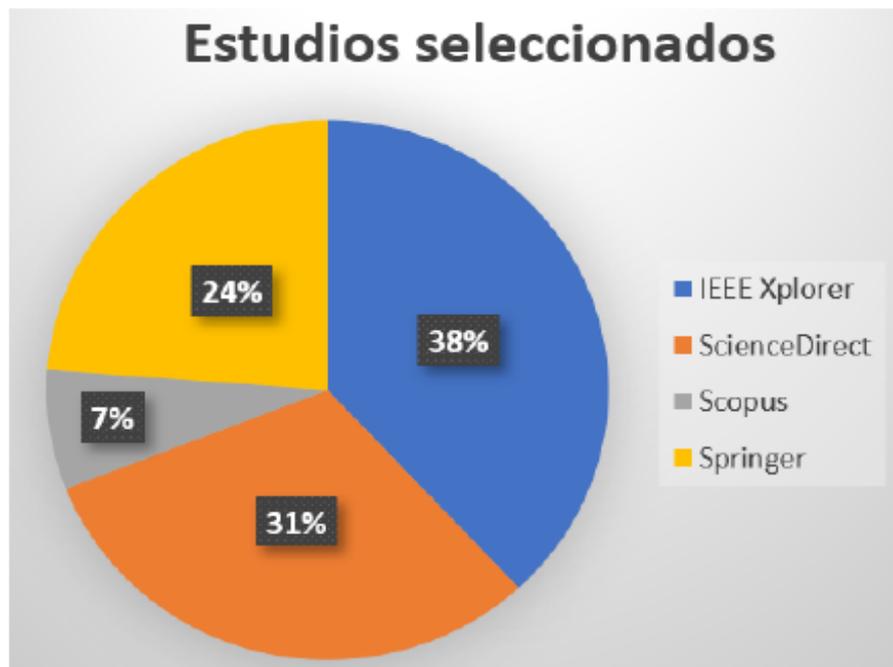


Figura 1: Porcentaje de artículos seleccionados previo a validar las preguntas de calidad

### 3. Resultados

#### 3.1. Estudios Incluidos

Para determinar los resultados finales se consideran solo los artículos que cumplen con la puntuación permitida (Mayor o igual a 0.9) para las preguntas de calidad de la revisión bibliográfica. Como resultado de la revisión se seleccionaron 18 artículos que se presentan en la siguiente tabla.

Nro.	Documento	Autor (es)	Año	QA1	QA2	Puntuación
1	A Review of Risk Management in Agile Development	Shehzad, Naveed and Iqbal, M. Aqeel and Amjad, Maria	2022	1	0	1
2	A Review on Risk Management Framework for large scale scrum	Khurana, Simran Kaur and Wassay, Md Abdul and Verma, Karunendra	2022	1	0	1
3	A framework for risk management in Scrum development process	Syrine Chaouch and Asma Mejri and Sonia Ayachi Ghannouchi	2019	1	0	1

4	A risk management framework for distributed scrum using prince2 methodology	Esteki, Mohammad and Gandomani, Taghi Javdani and Farsani, Hadi Khosravi	2020	1	0	1
5	Agile software development and UX design: A case study of integration by mutual adjustment	John Stouby Persson and Anders Bruun and Marta Kristín Lárusdóttir and Peter Axel Nielsen	2022	1	0	1
6	Criticism of the Risk Management Process in Scrum Methodology	Pilliang, Marzuki and Munawar and Tjahjono, Budi and Sejati, Puteri and Akbar, Habibullah and Firmansyah, Gerry	2022	1	0	1
7	Do scaling agile frameworks address global software development risks? An empirical study	Sarah Beecham and Tony Clear and Ramesh Lal and John Noll	2021	1	0	1
8	Factors Affecting Agile Software Project Success	Binboga, Burcu and Altin Gumussoy, Cigdem	2024	1	0	1
9	Game-based Sprint retrospectives: multiple action research	Przybyłek, Adam and Albecka, Marta and Springer, Olga and Kowalski, Wojciech	2021	1	0	1
10	How Scrum adds value to achieving software quality?	Alami, Adam and Krancher, Oliver	2022	1	0	1
11	How agile teams make self-assignment work: a grounded theory study	Masood, Zainab and Hoda, Rashina and Blincoe, Kelly	2020	1	0	1
12	Improving requirements-test alignment by prescribing practices that mitigate communication gaps	Bjarnason, Elizabeth and Sharp, Helen and Regnell, Björn	2019	1	0	1
13	Inter-team communication in large-scale co-located software engineering: a case study	Bjarnason, Elizabeth and Gislason Bern, Baldvin and Svedberg, Linda	2022	1	0	1

14	Prediction of Risk Percentage in Software Projects by Training Machine Learning Classifiers	Gouthaman P and Suresh Sankaranarayanan	2021	1	0	1
15	Risk Management in Agile Software Development: A Survey	Hammad, Muhammad and Inayat, Irum and Zahid, Maryam	2019	1	1	2
16	Risk management analysis in Scrum software projects	Tavares, Breno Gontijo and da Silva, Carlos Eduardo Sanches and de Souza, Adler Diniz	2019	1	0,5	1,5
17	Survive IT! Survival analysis of IT project planning approaches	Zsolt T. Kosztyán and Róbert Jakab and Gergely Novák and Csaba Hegedűs	2020	1	0	1
18	The role of psychological safety in promoting software quality in agile teams	Alami, Adam and Zahedi, Mansooreh and Krancher, Oliver	2024	1	1	2

Cuadro 3: Resultado final luego de pasar las preguntas de calidad

### 3.2. Resultados de las preguntas de Investigación

Luego de haber implementado los lineamientos para realizar la revisión bibliográfica, se encontraron en total 187 riegos, de los cuales 22 están directamente relacionados con la calidad de software y 165 podrían afectar a la calidad.

Se encontraron tres artículos que responden a la pregunta **¿Cuáles son los riesgos de calidad de software identificados en proyectos de desarrollo que utilizan Scrum?**, estos artículos presentan información puntual sobre riesgos de calidad. Los riesgos encontrados en cada artículo se indican en la siguiente tabla:

BD Científicas	Artículo	Riesgos de calidad de software
IEEE Xplorer	Risk Management in Agile Software Development: A Survey	- Riesgos relacionados con el aumento de defectos. - Riesgos relacionados con la deuda técnica. - Riesgos de Requisitos Cambiantes. - Riesgos de Desempeño (Presion por cumplir plazos). - Riesgos de Seguridad. - Falta de Pruebas Adecuadas. [1].
Scopus	Risk management analysis in Scrum software projects	- Falta de comunicación efectiva. - Cambios frecuentes en los requisitos. - Falta de documentación formal. - Integración insuficiente de componentes. - Deuda técnica acumulada. - Falta de pruebas exhaustivas. - Enfoque limitado en la calidad no funcional. - Retroalimentación incompleta de los stakeholders. [2].
Springer	The role of psychological safety in promoting software quality in agile teams	- No abordar problemas de calidad. - Retraso en la identificación de problemas de calidad. - Falta de reporte de defectos y errores. - Falta de mejora continua. - Repetición de Errores. - Falta de Iniciativa para Mejorar la Calidad. - Dificultades en la Gestión de la Calidad. - Defectos No Reportados. [3].

Cuadro 4: Riesgos que afectan directamente a la calidad de software

Se encontraron 15 artículos que aportan con lineamientos para dar respuesta a la pregunta **¿Cuáles son los riesgos que pueden influir en la calidad de software en proyectos de desarrollo que utilizan Scrum?**, si bien los autores no siempre especifican que los riesgos afecten directamente a la calidad, si mencionan una gran cantidad de riesgos que puede tener un impacto significativo en la calidad del producto. Para extraer los riesgos de los artículos se utilizó las herramientas ChatGpt, ChatPDF y el criterio del autor.

BD Científicas	Artículo	Riesgos de calidad
IEEE Xplorer	A Review of Risk Management In Agile Development	- Falta de habilidades adecuadas en el equipo. - Ambigüedad en la asignación de responsabilidades. - Desconocimiento de la cultura Agile. - Ineficacia en las actividades de Scrum. - Desafíos en la estimación de tareas. - Baja participación del cliente. - Requisitos poco claros - Retrasos en la comunicación. - Incorporación de nuevos miembros al equipo. [4].

IEEE Xplorer	A Review on Risk Management Framework for Large Scale Scrum	- Incertidumbre en los requisitos. - Entrega rápida sin pruebas exhaustivas. - Problemas de integración. - Comunicación ineficaz. - Coordinación y cooperación deficientes. - Obsolescencia de la documentación. - Desalineación con los requisitos del cliente. - Dificultades en la gestión del conocimiento. [5].
ScienceDirect	A framework for risk management in Scrum development process	- Incertidumbre en los requisitos. - Entrega rápida sin pruebas exhaustivas. - Problemas de integración. - Definición inadecuada de requerimientos. - Falta de pruebas efectivas. - Cambios frecuentes en el alcance. - Inconsistencias en la comunicación. - Presión por cumplir plazos. - Falta de documentación adecuada. - Falta de experiencia del equipo. - Riesgos técnicos. [6].
Scopus	A risk management framework for distributed scrum using PRINCE2 methodology	- Problemas con la integración de código. - Desalineación de los requisitos y las funcionalidades. - Disparidades en las capacidades técnicas entre equipos. - Falta de pruebas coordinadas. - Requisitos incompletos o mal definidos. - Falta de involucramiento del cliente. - Problemas de comunicación. - Diferencias culturales y de zona horaria. - Falta de capacitación y conocimiento - Gestión inadecuada de riesgos. - Iteraciones rápidas sin revisión adecuada. - Dependencias técnicas no gestionadas. [7].
ScienceDirect	Agile software development and UX design: A case study of integration by mutual adjustment	- Falta de pruebas de usuario adecuadas. - Implementación incompleta o incorrecta del diseño UX. - Priorización incorrecta de características de calidad. - Inconsistencias en la experiencia del usuario. - Falta de atención a aspectos no funcionales. - Insuficiente atención a las actividades de UX. - Comunicación deficiente entre diseñadores UX y desarrolladores. - Limitaciones de tiempo para realizar diseño y pruebas. - Conflictos de poder y colaboración ineficaz. - Dificultades en la priorización de actividades de UX. [8].
IEEE Xplorer	Criticism of the Risk Management Process in Scrum Methodology	- Falta de coordinación entre desarrollo y Product Owner. - Falta de un enfoque formal de gestión de riesgos. - Presión por cumplir los tiempos de Sprint - Requisitos incompletos o ambiguos. - Insuficiente enfoque en pruebas. - Problemas de comunicación. - Dependencia de herramientas o tecnologías no probadas. - Falta de retroalimentación continua. - Cambios frecuentes en los requisitos. - Presión por cumplir plazos. - Falta de documentación adecuada. [9].

ScienceDirect	Do Scaling Agile Frameworks Address Global Software Development Risks?	- Falta de interacción cara a cara. - Desconfianza entre los equipos. - Desalineación de procesos. - Retrasos en la resolución de problemas debido a zonas horarias. - Problemas de comunicación y colaboración. - Definición ambigua de "hecho". - Dependencias no gestionadas. - Falta de pruebas adecuadas. - Estimaciones inadecuadas de recursos y tiempo. [10].
IEEE Xplorer	Factors Affecting Agile Software Project Success	- Cambios constantes en los requisitos. - Falta de pruebas exhaustivas. - Falta de atención a aspectos no funcionales. - Integración de componentes inadecuada. - Dependencia en la capacidad del equipo. - Complejidad técnica. - Cambios en las especificaciones. - Tamaño del proyecto. - Incertidumbre tecnológica. - Presión por urgencia. - Falta de análisis de riesgos. [11].
Springer	Game-based Sprint retrospectives: multiple action research	- Falta de seguimiento a problemas identificados. - Disminución de la innovación debido a la monotonía. - Inequidad en la participación. - Uso ineficaz de las retrospectivas para identificar problemas de calidad. - Desviaciones de las prácticas de SCRUM. - Falta de reuniones de retrospectiva. - Problemas de comunicación. - Compromisos múltiples de los miembros del equipo. - Reuniones ineficaces. - Dificultades en la transformación de lecciones en acciones. [12].
Springer	How Scrum Adds Value to Achieving Software Quality	- Implementación inconsistente o incompleta de Scrum. - Falta de retroalimentación continua de los usuarios finales. - Falta de colaboración y tensiones en el equipo. - Restricciones culturales y falta de seguridad psicológica. - Desalineación con las necesidades del negocio. - Falta de enfoque en la calidad interna. - Inaccessibilidad de los usuarios finales. - Complejidad del proyecto. [13].
Springer	How agile teams make self-assignment work: a grounded theory study	- Auto-asignación inadecuada de tareas. - Desigual distribución del trabajo. - Falta de coordinación en el equipo. - Ambigüedad en los criterios de "hecho". - Cambios frecuentes en las prioridades del backlog. - Interrupciones por trabajo urgente. - Falta de responsabilidad y propiedad. - Deficiencias en la comunicación. - Elección incorrecta de tareas. [14].

Springer	Inter-team communication in large-scale co-located software engineering: a case study	- Distancia psicológica y cognitiva. - Falta de coordinación entre equipos. - Falta de conocimiento compartido. - Falta de conciencia de dependencias. - Retrasos y rework. - Enfoque en tareas individuales. - Ineficiencia en la coordinación. - Cultura organizacional. - Tamaño del equipo y experiencia. [15].
ScienceDirect	Survive IT! Survival analysis of IT project planning approaches	- Cambios frecuentes en los requisitos del cliente. - Sobrecostos y reducción de tiempos de entrega. - Dependencias y cambios en la estructura del proyecto. - Incertidumbre en los requisitos. - Falta de enfoque en la documentación. - Problemas de integración. - Compromiso de calidad por la velocidad. - Dependencia de la colaboración del equipo. [16].
Springer	Improving requirements-test alignment by prescribing practices that mitigate communication gaps	- Falta de alineación entre requisitos y pruebas. - Ambigüedad en los requisitos. - Errores en la interpretación de los requisitos. - Ambigüedad en los requisitos. - Desafíos en la definición de requisitos de calidad. - Dificultades en la comunicación. - Percepción diferente de la calidad. - Carga de trabajo del propietario del producto. - Dificultades en la priorización de aspectos de calidad. - Desalineación entre requisitos y pruebas. [17].
ScienceDirect	Prediction of Risk Percentage in Software Projects by Training Machine Learning Classifiers	- Acumulación de defectos no resueltos. - Problemas de integración entre componentes. - Falta de pruebas exhaustivas. - Falta de enfoque en aspectos no funcionales. - Falta de retroalimentación oportuna del cliente. - Incertidumbre en los requisitos. - Deficiencias en la documentación. - Problemas de comunicación y colaboración. - Presión por cumplir plazos. - Falta de enfoque en la gestión de riesgos. - Retrasos en la entrega de funcionalidades. - Errores acumulados por iteraciones rápidas. [18].

Cuadro 5: Riesgos que podrían afectar a la calidad de software

### 3.3. Riesgos detectados de calidad de software

En general los documentos evaluados mencionan una falencia en la gestión de riesgos en los proyectos gestionados con Scrum, ya que no se propone un proceso formal para tratar los riesgos. La revisión bibliográfica facilitó la identificación de algunos riesgos propuestos por los autores de los documentos seleccionados, que están relacionados con el objetivo de la revisión.

En el artículo [1], los autores identifican varios riesgos que afectan a la calidad del software. Uno de los principales es el **aumento de defectos**, que se puede presentar por el ritmo acelerado de las iteraciones y la falta de tiempo para realizar pruebas, esto permite que los defectos persistan y se acumulen, comprometiendo la calidad del producto. También destaca los **riesgos relacionados con la deuda técnica**, que ocurren cuando se priorizan entregas rápidas a costa de soluciones temporales o código de baja calidad, lo que afecta la mantenibilidad y escalabilidad del software a largo plazo. **La falta de pruebas adecuadas** es un riesgo significativo en metodologías ágiles, donde la rapidez del ciclo de desarrollo puede reducir el tiempo destinado a ejecutar diversos tipos de pruebas, afectando la fiabilidad y funcionalidad del software. Además, los autores subrayan la importancia de abordar los **riesgos de seguridad**, ya que la falta de enfoque en pruebas de seguridad puede exponer el software a vulnerabilidades críticas. También mencionan dos riesgos adicionales, los **requisitos cambiantes** y el **desempeño bajo presión**.

Los autores del artículo [2] identifican varios riesgos que afectan la calidad del software, uno de los más importantes es la **Integración insuficiente de componentes**, los módulos desarrollados en diferentes Sprints pueden no interactuar correctamente, lo que afecta la calidad final del producto en términos de funcionalidad y estabilidad. **La falta de pruebas exhaustivas**, este riesgo hace referencia a que la velocidad de las iteraciones de Scrum puede resultar en pruebas insuficientes, lo que permite que los defectos persistan hasta las etapas finales del desarrollo o incluso después del lanzamiento, afectando la calidad del producto al introducir errores que no se detectaron a tiempo. Otro riesgo es el **enfoque limitado en la calidad no funcional**, el cual puede presentarse frecuentemente, ya que al aplicar Scrum se tiende a priorizar la entrega de nuevas funcionalidades visibles para el usuario; esto puede llevar a que se descuiden aspectos no funcionales como el rendimiento, la seguridad o la usabilidad, lo que afecta la experiencia del usuario y la calidad técnica del producto final. Los autores también manifiestan otro riesgo clave que afecta a la calidad, siendo este la **retroalimentación incompleta de los stakeholders**, sin la retroalimentación constante y precisa de los usuarios finales o partes interesadas durante cada Sprint, es posible que se desarrollen características que no cumplan con las expectativas o necesidades del cliente, afectando la calidad percibida del producto. Adicionalmente menciona la **falta de comunicación efectiva** y la **falta de documentación formal** como riesgos que afectan a la calidad; también, indica dos riesgos similares a los propuestos por [1], que son los **cambios frecuentes en los requisitos** y la **deuda técnica acumulada**.

En [3] los autores identifican varios riesgos desde el punto de vista de seguridad psicológica que afectan la calidad. El primer riesgo se refiere a **no abordar problemas de calidad**, se presenta principalmente cuando los equipos no tienen un ambiente seguro psicológicamente o un ambiente independiente de pruebas, por lo que tienden a evitar discutir problemas de calidad; esto resulta en un deterioro del software al no identificar defectos tempranamente. El **retraso en la identificación de problemas de calidad**, debido a la falta de confianza para comunicar problemas o preocupaciones sobre la calidad del producto durante los sprints puede llevar a la identificación tardía de errores críticos, incrementando el riesgo de que los defectos lleguen a producción. Los autores hacen referencia a dos riesgos similares, los **defectos no reportados** y la **falta de reporte de defectos y errores**, se dan principalmente en equipos donde no existe un ambiente de seguridad psicológica, donde los miembros no se sienten cómodos reportando defectos o problemas de calidad, o en el caso del segundo riesgo que puedan evitar

señalar, documentar o reportar defectos en el producto; permitiendo que los errores no se registren y pasen desapercibidos u olvidados por lo que no serán corregidos a tiempo. **Repetición de errores**, si los errores no se discuten abiertamente y no se aprende de ellos, es probable que se repitan en futuras iteraciones, generando una acumulación de defectos y a una disminución del tiempo de pruebas por el reporte de errores repetitivos. **Dificultades en la gestión de la calidad**, se presenta principalmente cuando existe falta de un enfoque colaborativo y de aprendizaje lo que puede dificultar la implementación de prácticas efectivas de gestión de la calidad, esto sumado al hecho de que la calidad se logra con la ejecución de un proceso que permite la liberación de errores en el software. Finalmente mencionan la **falta de mejora continua** y la **falta de iniciativa para mejorar la calidad** estos dos riesgos afectan significativamente la calidad del producto, ya que al no mejorar los procesos de pruebas se mantendrán las mismas prácticas erradas, en caso de que existieran.

### 3.4. Riesgos detectados que podrían afectar la calidad de software

En la revisión bibliográfica se encontraron 15 artículos que identifican riesgos que pueden afectar a la calidad del software en proyectos gestionados con Scrum. Parte de estos riesgos se presentan a continuación:

Por ejemplo, [5] plantea riesgos relacionados con la **calidad de código**, siendo este un problema recurrente debido a la velocidad requerida en los sprints de Scrum, esto tiene mucho sentido por la falta de tiempo suficiente para realizar revisiones de código lo cual puede llevar a una disminución en la calidad del código, introduciendo defectos y problemas de mantenimiento a largo plazo.

La **falta de comunicación** es otro riesgo propuesto por [9], [6]. En sus estudios sostienen que se presenta cuando cuando los equipos no son coordinados adecuadamente o cuando la comunicación es ineficaz entre los miembros del equipo y los interesados externos. Esto también afecta a la calidad ya que puede resultar en errores en la implementación, además Scrum requiere de interacciones constantes entre todos los interesados para que el proyecto sea exitoso.

La **falta de claridad en los requisitos**, la falta de claridad en los requisitos iniciales, y los cambios frecuentes es otro riesgo significativo propuesto por [6], [11]. Este riesgo puede desestabilizar la planificación inicial del proyecto, dificultando la ejecución de pruebas eficientes por la inestabilidad de los requisitos.

La **falta de gestión de riesgos** tienen algunas implicaciones según lo exponen los autores en sus publicaciones, en [19] se plantea la mejora en la gestión de calidad para mejorar la calidad del código y reducir el tiempo necesario para las revisiones, esto soportado con una gestión de riesgo adecuada, puede disminuir los tiempos de pruebas y la cantidad de errores de código.

La importancia de la **comunicación y la coordinación** propuesta por [9], implica que para reducir este riesgo, es crucial fomentar una cultura de transparencia y colaboración, reuniones diarias efectivas y

el uso de herramientas de comunicación adecuadas.

Riesgo de **fracaso del Proyecto** [6], aumenta la probabilidad de que los proyectos no cumplan con los objetivos de costo, tiempo y calidad ofrecidos al cliente.

Estos riesgos tienen varias implicaciones importantes para la gestión de proyectos Scrum. La falta de claridad en los requisitos y la comunicación puede llevar a una mala alineación entre las expectativas de los interesados y los entregables del proyecto, afectando la satisfacción del cliente y la calidad del producto. Además, **la resistencia al cambio** y **la dependencia de habilidades individuales** pueden limitar la flexibilidad y la capacidad de adaptación del equipo, aspectos críticos en un entorno ágil. La falta de gestión de riesgos en proyectos de Scrum, dificulta el tener una visibilidad clara de las consecuencias que se podría tener al no mitigar los riesgos de calidad asociados al software.

### 3.5. Limitaciones de los Hallazgos

Una limitación clave de estos hallazgos está relacionada con que muchos estudios tienden a ser de naturaleza exploratoria y están basados en encuestas o estudios de caso limitados, lo que puede no reflejar completamente la diversidad de contextos en los que se aplican las prácticas Scrum. Además, la variabilidad en la implementación de Scrum entre diferentes organizaciones y equipos significa que los riesgos identificados pueden no ser universales.

Otra limitación es la falta de datos empíricos robustos que cuantifiquen el impacto de estos riesgos en términos de métricas como el costo, el tiempo y la calidad del proyecto. Esto hace que sea difícil evaluar la gravedad real de los riesgos y la efectividad de las estrategias de mitigación propuestas por algunos autores.

Para superar estas limitaciones, futuras investigaciones deberían enfocarse en estudios más amplios y cuantitativos, que puedan proporcionar una visión más integral y empírica de los riesgos en proyectos Scrum. Además, sería beneficioso desarrollar métricas estandarizadas para evaluar el impacto de estos riesgos y la efectividad de las estrategias de mitigación en diversos contextos organizacionales.

## 4. Conclusiones

- La identificación y gestión de riesgos en proyectos de desarrollo de software utilizando Scrum es crucial para el éxito del proyecto. Si bien se han identificado varios riesgos de calidad, es esencial considerar las limitaciones de estos hallazgos y la necesidad de adaptar las prácticas a las circunstancias específicas de cada proyecto.
- Respondiendo a la primera pregunta de investigación (¿Cuáles son los riesgos de calidad de software identificados en proyectos de desarrollo que utilizan Scrum?), se obtuvieron 22 riesgos que impactan directamente la calidad de software, dichos riesgos están identificados en el Cuadro 4.

- Respondiendo a la segunda pregunta de investigación (¿Cuáles son los riesgos que pueden influir en la calidad de software en proyectos de desarrollo que utilizan Scrum?), se detectaron 165 riesgos que podrían afectar o repercutir en la calidad de software, dichos riesgos están identificados en el Cuadro 5.
- Las implicaciones de estos hallazgos sugieren que existe un gran número de riesgos de calidad que pueden afectar a la calidad en proyectos que usan Scrum. Si los riesgos se materializan o no son gestionados, estos riesgos disminuirían la calidad del software, la utilidad del producto y la satisfacción del usuario final.

## 5. Referencias

### Referencias

- [1] M. Hammad, I. Inayat, and M. Zahid, "Risk management in agile software development: A survey," in *Proceedings - 2019 International Conference on Frontiers of Information Technology, FIT 2019*, pp. 162–166, Institute of Electrical and Electronics Engineers Inc., 12 2019.
- [2] B. G. Tavares, C. E. S. da Silva, and A. D. de Souza, "Risk management analysis in scrum software projects," *International Transactions in Operational Research*, vol. 26, pp. 1884–1905, 9 2019.
- [3] A. Alami, M. Zahedi, and O. Krancher, "The role of psychological safety in promoting software quality in agile teams," *Empirical Software Engineering*, vol. 29, p. 119, 9 2024.
- [4] N. Shehzad, M. A. Iqbal, and M. Amjad, "A review of risk management in agile development," in *2022 International Conference on Digital Transformation and Intelligence, ICDI 2022 - Proceedings*, pp. 63–68, Institute of Electrical and Electronics Engineers Inc., 2022.
- [5] S. K. Khurana, M. A. Wassay, and K. Verma, "A review on risk management framework for large scale scrum," in *Proceedings - 2022 International Conference on Computational Modelling, Simulation and Optimization, ICCMSO 2022*, pp. 394–400, Institute of Electrical and Electronics Engineers Inc., 2022.
- [6] S. Chaouch, A. Mejri, and S. A. Ghannouchi, "A framework for risk management in scrum development process," in *Procedia Computer Science*, vol. 164, pp. 187–192, Elsevier B.V., 2019.
- [7] M. Esteki, T. J. Gandomani, and H. K. Farsani, "A risk management framework for distributed scrum using prince2 methodology," *Bulletin of Electrical Engineering and Informatics*, vol. 9, pp. 1299–1310, 6 2020.
- [8] J. S. Persson, A. Bruun, M. K. Lárusdóttir, and P. A. Nielsen, "Agile software development and ux design: A case study of integration by mutual adjustment," *Information and Software Technology*, vol. 152, 12 2022.

- [9] M. Pilliang, Munawar, B. Tjahjono, P. Sejati, H. Akbar, and G. Firmansyah, "Criticism of the risk management process in scrum methodology," in *Proceedings - IEIT 2022: 2022 International Conference on Electrical and Information Technology*, pp. 338–343, Institute of Electrical and Electronics Engineers Inc., 2022.
- [10] S. Beecham, T. Clear, R. Lal, and J. Noll, "Do scaling agile frameworks address global software development risks? an empirical study," *Journal of Systems and Software*, vol. 171, 1 2021.
- [11] B. Binboga and C. A. Gumussoy, "Factors affecting agile software project success," *IEEE Access*, 2024.
- [12] A. Przybyłek, M. Albecka, O. Springer, and W. Kowalski, "Game-based sprint retrospectives: multiple action research," *Empirical Software Engineering*, vol. 27, 1 2022.
- [13] A. Alami and O. Krancher, "How scrum adds value to achieving software quality?," *Empirical Software Engineering*, vol. 27, 12 2022.
- [14] Z. Masood, R. Hoda, and K. Blincoe, "How agile teams make self-assignment work: a grounded theory study," *Empirical Software Engineering*, vol. 25, pp. 4962–5005, 11 2020.
- [15] E. Bjarnason, B. G. Bern, and L. Svedberg, "Inter-team communication in large-scale co-located software engineering: a case study," *Empirical Software Engineering*, vol. 27, 3 2022.
- [16] Z. T. Kosztyán, R. Jakab, G. Novák, and C. Hegedűs, "Survive it! survival analysis of it project planning approaches," *Operations Research Perspectives*, vol. 7, 1 2020.
- [17] E. Bjarnason, H. Sharp, and B. Regnell, "Improving requirements-test alignment by prescribing practices that mitigate communication gaps," *Empirical Software Engineering*, vol. 24, pp. 2364–2409, 8 2019.
- [18] G. P and S. Sankaranarayanan, "Prediction of risk percentage in software projects by training machine learning classifiers," *Computers and Electrical Engineering*, vol. 94, 9 2021.
- [19] I. I. Hadyan, D. S. Kusumo, and J. H. Husen, "Implementing continuous code quality for code quality development in the scrum team," 2022.

## Anexo 2. Definición de probabilidad e impacto de riesgos de calidad.

Para determinar la probabilidad e impacto de los riesgos, se definió una acuesta realizada a personal relacionado con el proceso de desarrollo de software en entornos ágiles, específicamente con Scrum.

Se considero personal docente de la carrera de computación, personal externo que trabaja en desarrollo de software, alumnos y docentes de la maestría, en total se realizaron 15 encuestas.

El levantamiento de información se realizó por medio de la herramienta de formularios de Google, puntualmente solicitando a los encuestados que en base a su experiencia analicen el riesgo de calidad y definan su probabilidad e impacto. En la siguiente figura se muestra el encabezado principal de la encuesta y el número de encuestados.

Preguntas Respuestas **15** Configuración

### Análisis de probabilidad e impacto de riesgos de calidad de software en proyectos que usan Scrum

**B I U ↻ ✕**

A continuación, encontrará una lista de riesgos identificados en proyectos de desarrollo de software que usaron Scrum. Por favor, evalúe cada riesgo en términos de:

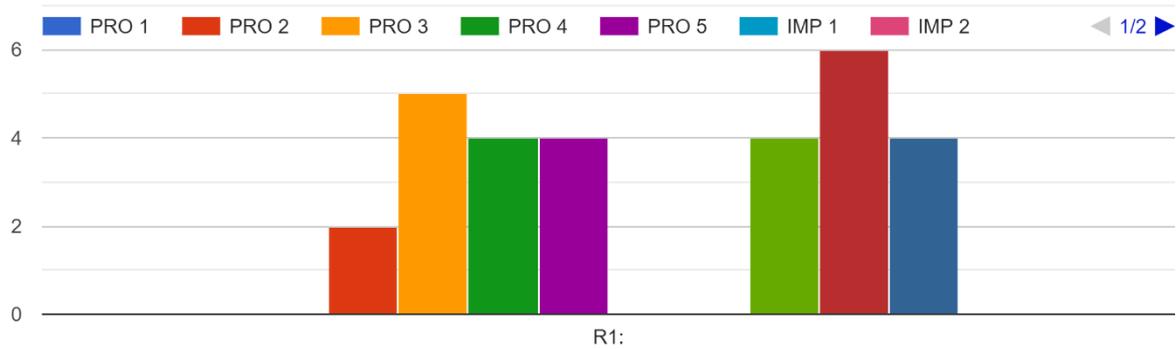
- **Probabilidad:** Según su experiencia, ¿Qué tan probable podría ser que ocurra este tipo de riesgo en los proyectos de desarrollo de software en los que ha trabajado? (Escala del PRO 1 al PRO 5, siendo PRO 1 muy poco probable y PRO 5 muy probable).
- **Impacto:** Si este riesgo llegara a materializarse, ¿Qué tan significativo sería su impacto en la calidad del producto o en el éxito del proyecto? (Escala del IMP 1 al IMP 5, siendo IMP 1 un impacto bajo y IMP 5 un impacto muy alto).

**PARA CADA RIESGO TIENE QUE MARCAR ÚNICAMENTE DOS CELDAS DE SU ELECCIÓN, UNA PARA MEDIR LA PROBABILIDAD O OTRA PARA MEDIR EL IMPACTO. DESDE YA GRACIAS POR SU COLABORACIÓN.**

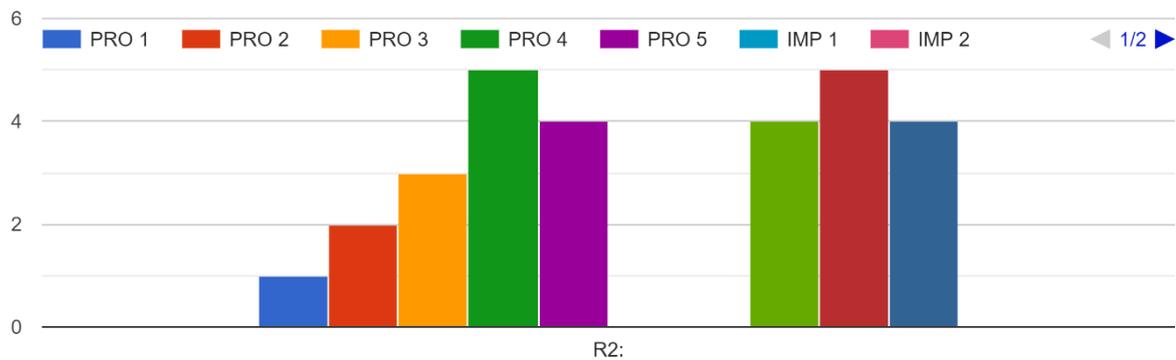
Nota. La información que proporcione se utilizará únicamente para fines académicos y se analizará de manera reservada.

A continuación, se muestra las respuestas proporcionadas por los encuestados para cada uno de los riesgos.

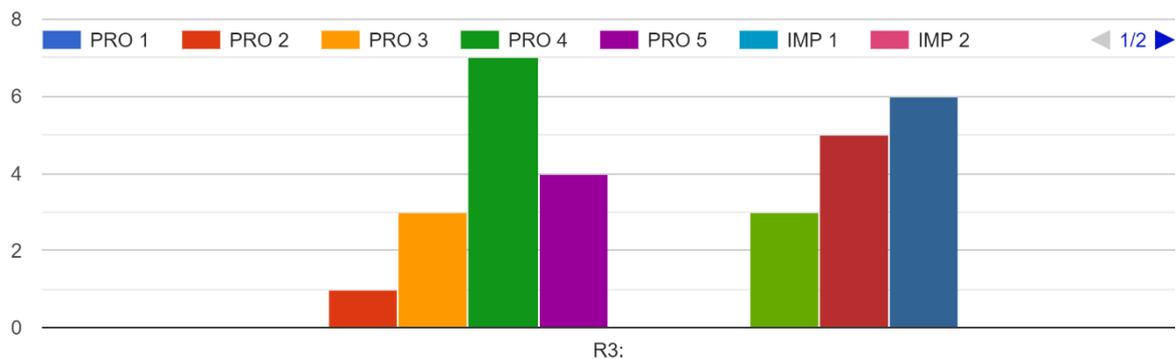
R1.No abordar problemas de calidad. Los equipos que no tienen un ambiente seguro psicológicamente tienden a evitar discutir problem...are al no identificar defectos de manera temprana.



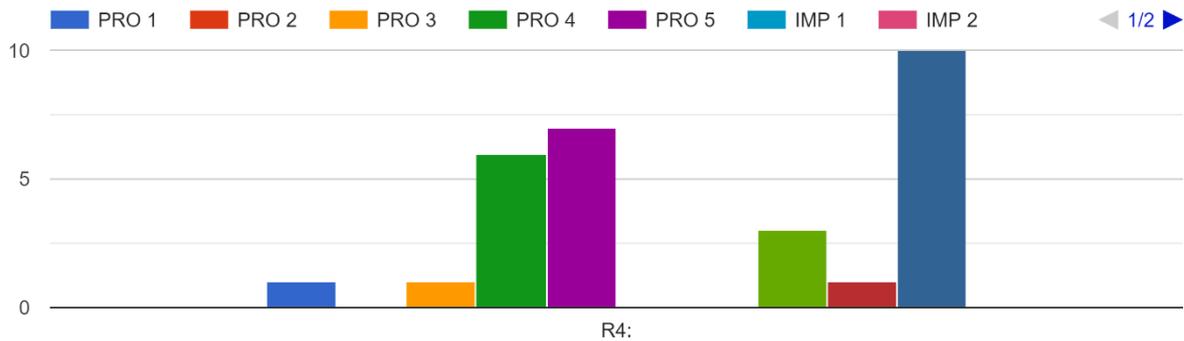
R2. Falta de reporte oportuno de defectos y problemas de calidad. La falta de un ambiente de seguridad psicológica o falta de confianza en los e...lemas de calidad del producto durante los sprints.



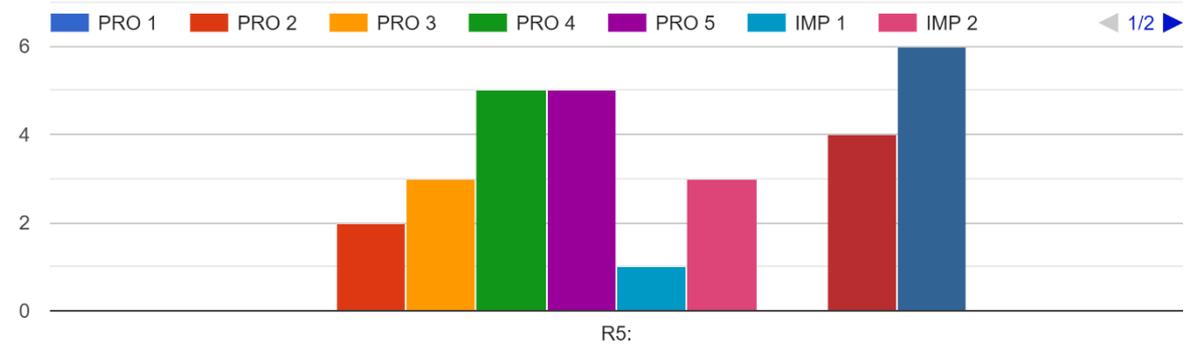
R3. Falta de iniciativa para mejora continua de la calidad. En un entorno donde no se promueve la seguridad psicológica, los miembros del equipo tien...iativa para innovar y mejorar de manera iterativa.



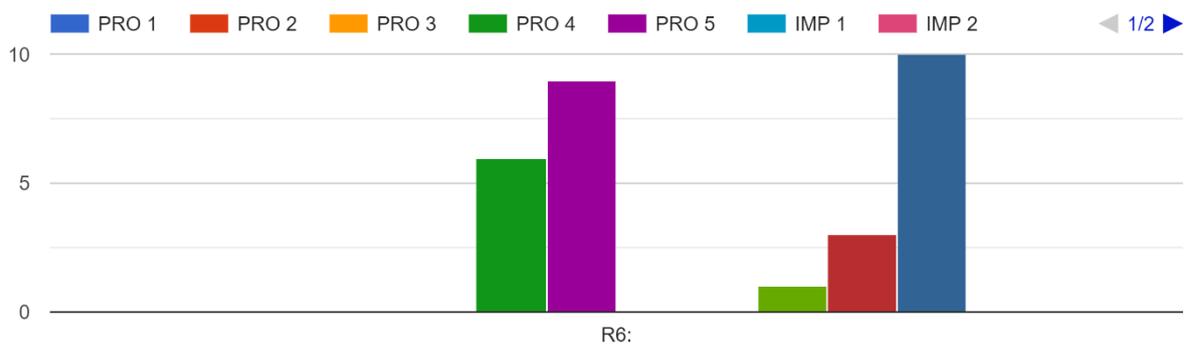
R4. Falta de aprendizaje y prevención de errores. Si los errores no se discuten abiertamente y no se aprende de ellos, es probable que se repitan en fut...o defectos y disminuyendo la calidad del producto.



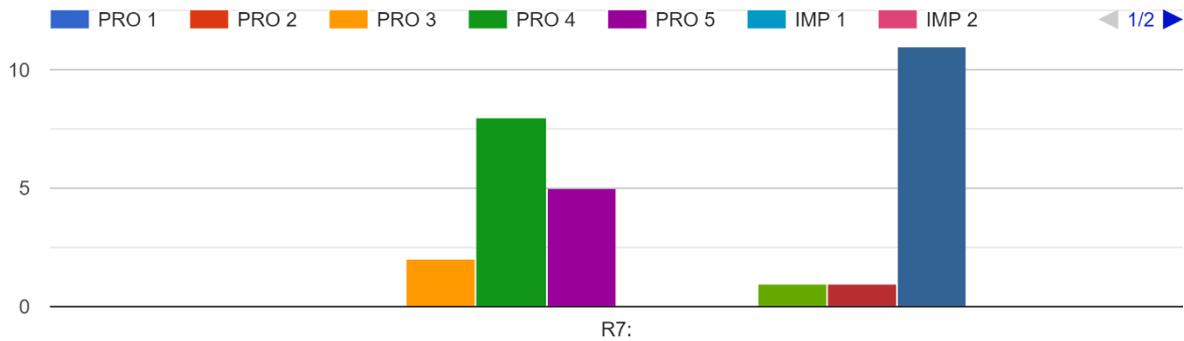
R5. Gestión ineficaz de la calidad por falta de colaboración. La falta de un enfoque colaborativo y de aprendizaje puede dificultar la implementación de...o cumple con los estándares de calidad esperados.



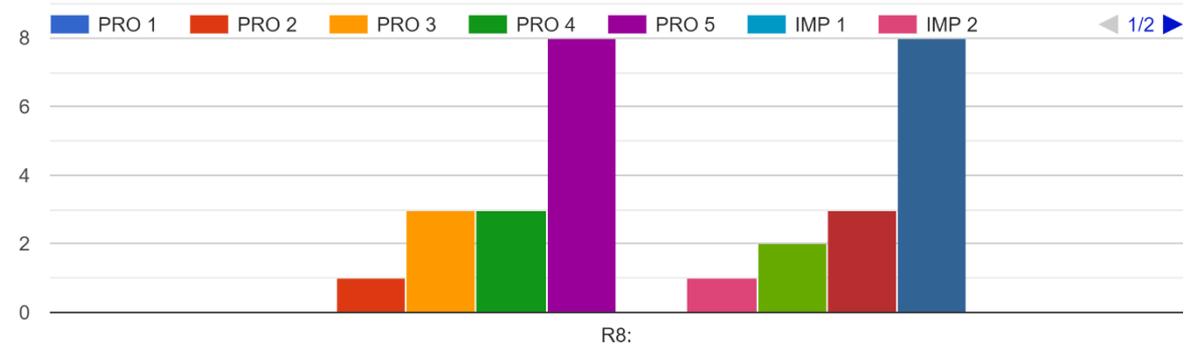
R6. Retroalimentación incompleta de los Stakeholders. Si no se obtiene retroalimentación constante y precisa de los usuarios finales o parte...lan con las expectativas o necesidades del cliente.



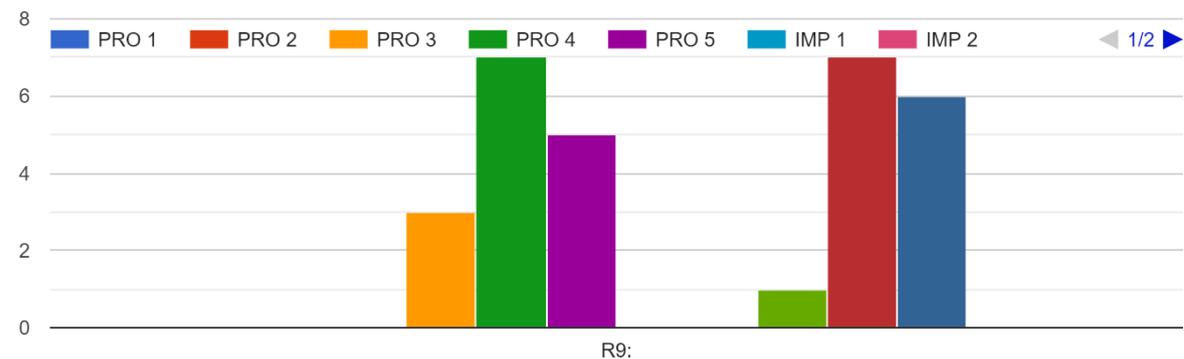
R7. Falta de comunicación efectiva. La calidad del producto puede verse comprometida si hay fallas en la comunicación entre los miembros del equipo. ...lementación de características y funcionalidades.



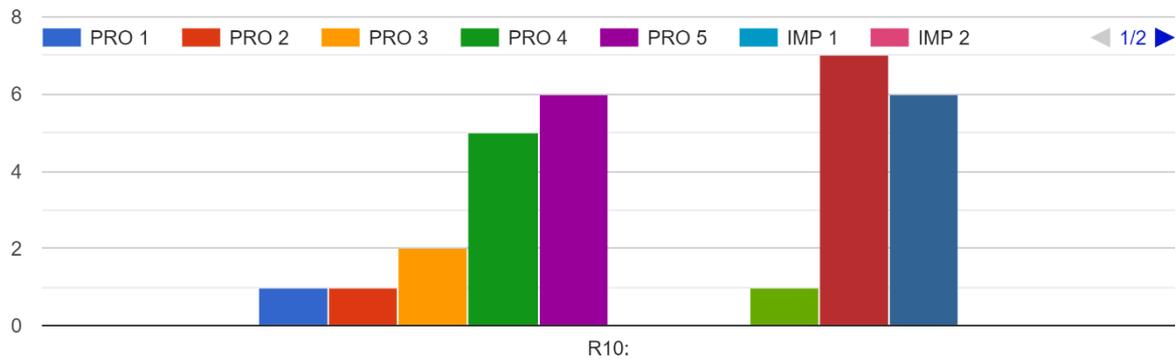
R8. Aumento de defectos por inexperiencia del equipo. Los equipos ágiles, especialmente los nuevos, tienden a experimentar un aumento de defect...ar la estabilidad y la funcionalidad del software.



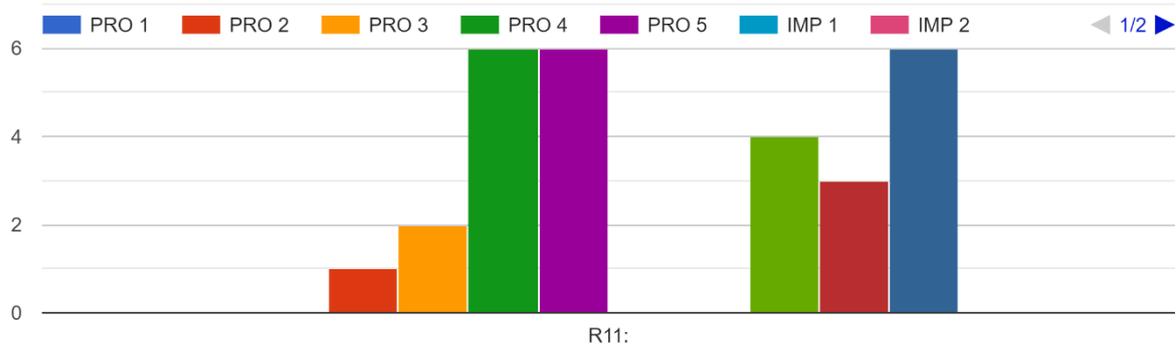
R9. Gestión inadecuada de la deuda técnica. La presión por entregar productos rápidamente puede causar decisiones técnicas no óptimas que generan ...la escalabilidad y la mantenibilidad del software.



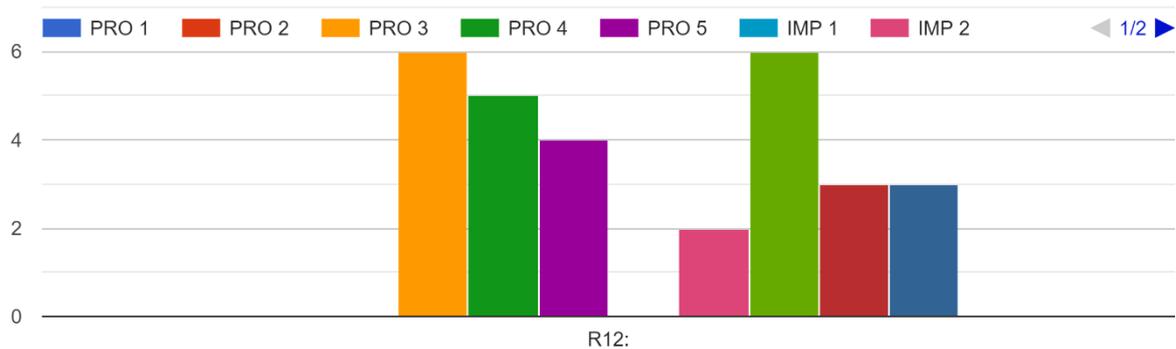
R10. Gestión inadecuada de cambios en los requisitos. La flexibilidad de Scrum permite que los requisitos cambien durante el desarrollo. Si estos ca...el cliente, afectando la calidad final del software.



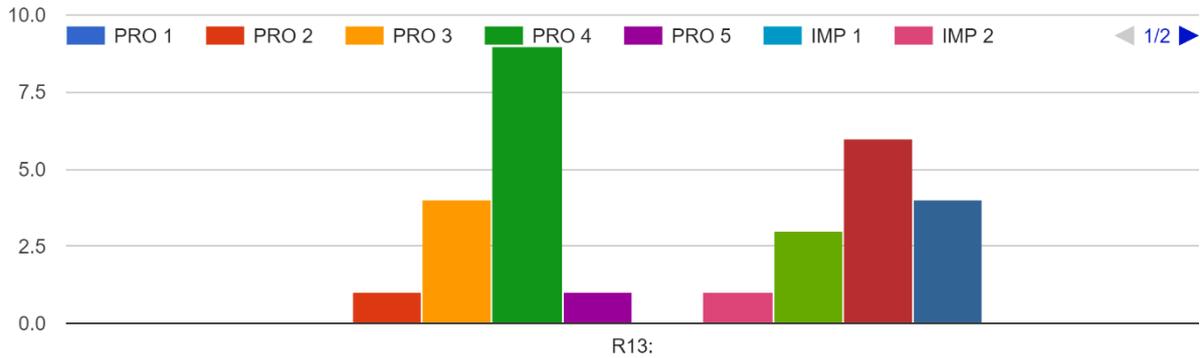
R11. Pruebas insuficientes durante las iteraciones. La presión para cumplir con los plazos puede resultar en una calidad de código inferior, ya que el...podría dar prioridad a la velocidad sobre la calidad.



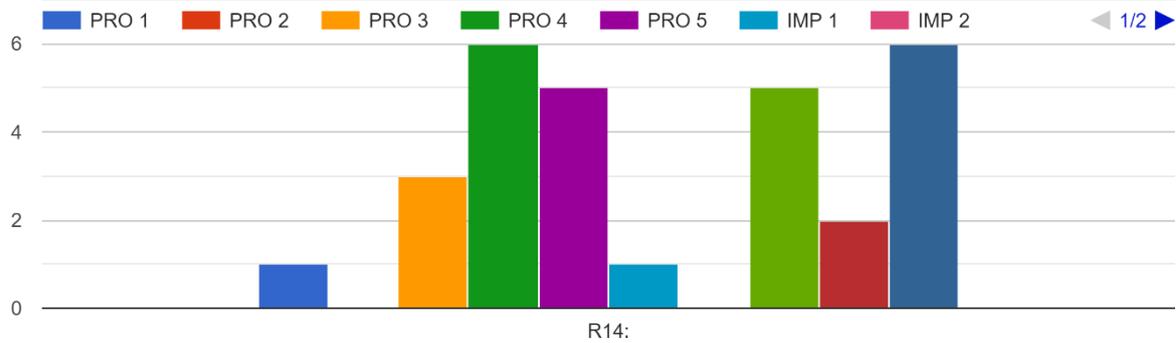
R12. Falta de documentación formal. La tendencia de Scrum a minimizar la documentación puede dificultar la comprensión de los requisitos y está...ducto que no cumple con los estándares esperados.



R13. Problemas de integración en entregas incrementales. La entrega de funcionalidades de forma incremental en Scrum puede generar problemas de i...tes sprints pueden no interactuar correctamente.



R14. Desatención de requisitos no funcionales. Scrum tiende a priorizar la entrega de nuevas funcionalidades visibles para el usuario, descuidand...abilidad, lo que afecta la experiencia del usuario.



**Tabulación de la información.** Los datos ingresados por los encuestado, se descargaron en una matriz de Excel para que puedan ser evaluados. En la matriz se usa la siguiente nomenclatura: “R1” significa riesgo 1, “P” significa probabilidad, “I” significa impacto.

**Tabla 1.** Datos recolectados de la encuesta.

R1.		R2.		R3.		R4.		R5.		R6.		R7.		R8.		R9.		R10.		R11.		R12.		R13.		R14.				
P	I	P	I	P	I	P	I	P	I	P	I	P	I	P	I	P	I	P	I	P	I	P	I	P	I	P	I	P	I	
2	4	2	5	4	4	4	3	4	4	4	4	4	3	3	3	4	4	4	4	3	4	4	5	3	4	4	5	3	4	
5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	4	5	5	5	5	5	5	
3	3	3	3	4	3	4	4	3	2	5	5	4	5	4	4	4	5	5	5	4	5	5	5	3	4	4	4	5	5	
5	5	5	5	3	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	3	3	3	3	4	3	4	3	5	
4	4	4	4	4	4	4	3	4	4	5	5	4	5	4	4	4	4	5	5	5	3	4	3	4	4	5	5	5	5	
3	5	3	3	3	5	4	5	4	4	4	5	3	5	3	5	3	4	3	4	3	3	3	3	3	4	4	4	5	5	
4	4	3	3	3	3	5	5	5	5	4	4	4	3	2	2	3	3	5	5	3	3	4	4	3	3	3	3	3	3	
5	3	4	4	5	4	3	3	4	5	5	5	4	5	5	5	4	4	2	4	2	3	4	4	2	2	1	1	1	1	
5	5	5	5	5	5	5	5	3	1	5	5	5	5	5	5	4	4	5	5	5	5	4	2	4	4	5	5	5	5	
2	4	1	3	4	5	1	5	2	4	4	3	4	5	3	3	5	5	1	4	5	5	4	3	4	3	4	3	4	3	
3	3	4	4	2	3	4	5	2	2	4	4	3	5	4	4	4	5	4	4	4	4	3	3	4	5	4	4	4	4	
4	3	5	3	4	3	5	3	5	3	5	3	4	3	5	3	4	3	4	3	4	3	5	3	5	3	5	3	5	3	
3	4	2	3	5	4	5	5	3	2	5	5	4	4	5	5	5	4	3	3	4	4	3	2	4	5	4	3	4	3	
4	4	4	4	4	5	4	5	4	5	4	5	5	5	5	5	5	5	4	4	5	5	5	5	4	4	5	5	5	5	
3	3	4	4	4	4	5	5	5	5	5	5	5	5	5	5	4	4	4	4	4	3	3	4	4	3	3	4	4	3	3

Para obtener el nivel (calificación de 1 a 5) de la probabilidad y el impacto, se realizó un conteo y se eligió los que tenían mayor cantidad de votos. En caso de que el número de votos coincidía para varios niveles, se procedió a tomar el que estaba en el nivel más alto. El resultado se muestra en la siguiente tabla.

**Tabla 2.** Datos tabulados de la encuesta.

R1.		R2.		R3.		R4.		R5.		R6.		R7.		R8.		R9.		R10.		R11.		R12.		R13.		R14.			
P	I	P	I	P	I	P	I	P	I	P	I	P	I	P	I	P	I	P	I	P	I	P	I	P	I	P	I	P	I
3	4	4	3	4	5	5	5	5	5	5	5	4	5	5	5	4	4	5	4	5	5	3	3	4	4	4	4	4	3

La información restante se analiza en la sección de resultados (6.1.1.2.3 Evaluación de riesgos) del presente trabajo de titulación.

### Anexo 3. Taller de evaluación del marco de trabajo.

#### 3.1 Hoja de asistencia de 17 evaluadores que participaron en el taller de validación.

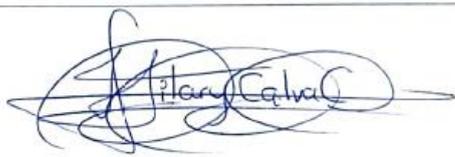
# Maestría en Ingeniería en Software

## Maestría académica con trayectoria profesional

### HOJA DE ASISTENCIA

#### Taller de evaluación de riesgos de calidad de software en proyectos con Scrum

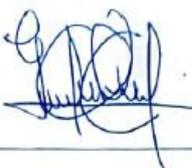
<b>Proyecto:</b> "Desarrollo de un marco de trabajo para la gestión de riesgos de calidad en proyectos de desarrollo de software ágil, basado en la ISO 31000 y gestionados con Scrum"	<b>Fecha de la reunión :</b> 15/nov/2024
<b>Moderador :</b> Francisco Alvarez Pineda	<b>Lugar/Sala:</b> Aula 421

Nombre	Firma
ANDRADE ZUÑIGA JONATHAN JAVIER	
CALVA CAMACHO HILARY MADELEY	
CAPA RODRIGUEZ CRISTIAN RAMIRO	
CASTILLO ESTRELLA JUAN FRANCISCO	
CUEVA RIVAS WILLIAN ENRIQUE	
GONZAGA RIVAS KAREN BRIGITH	

MARTINEZ GRANDA DANNY AUGUSTO	
MARTINEZ HERRERA JEAN CARLO	

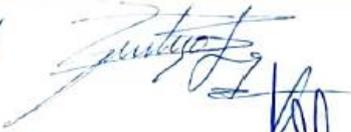
# Maestría en Ingeniería en Software

Maestría académica con trayectoria profesional

Nombre	Firma
PINTA SALAZAR JAINER SEBASTIAN	
QUIZHPE CHOCHO GERARDO MANUEL	
QUIZHPE HURTADO JENNIFER ELIZABETH	
ROMERO MAZA JAMES ARIEL	
SAUCA PUCHA JOSUE ALEJANDRO	
TADAY SANCHEZ ANDERSON MANUEL	

TRUEBA REYES CECILIA FERNANDA	
URREGO GOMEZ YOVIN STIVEN	

Toro Castillo Santiago Emanuel



Cristian Gabriel Aguirre Gonzalez



### 3.2 Observaciones reportadas por los evaluadores.

En la tabla solo se presentan las características, subcaracterísticas y preguntas sobre las que se indicó alguna observación por parte de los evaluadores. También se presentan las observaciones filtradas unificando las que están duplicadas o se refieren a la misma circunstancia.

Características	Subcaracterísticas	R4	R5	R6	R8	R11
1.Eficacia.	Tareas completas.	<p>1.Definir: postmortem de fallos, cultura que evita reconocer errores, Sprint. (APLICA).</p> <p>2. De pronto las 3 columnas pueden tornarse en solo (NO APLICA).</p> <p>3. Como sugerencia, redactar la columna Evento en presente continuo. (APLICA).</p>		<p>1. Definir: PO (Colocar palabras completas o especificar abreviaturas), backlog. (OK).</p> <p>2. En las técnicas de mitigación Sprint planing, la última instrucción esta confusa. (OK).</p>	<p>1. Definir: valores límite. (OK).</p> <p>2. El evento relacionado con los valores límite debería revisarse y confirmar si va acorde al contexto del riesgo. (OK).</p> <p>3. En la descripción del riesgo considerar que el equipo debe tener en cuenta un periodo de adaptación para personas que no tengan mucha experiencia en el giro de negocio que se está evaluando. (OK).</p> <p>4. En el Daily Scrum, se podría especificar alguna técnica de monitoreo que ayude para revisar el progreso del equipo. (OK).</p>	<p>1. Las empresas se rigen mucho a los tiempos de entrega y no consideran un tiempo de holgura sometiendo a presión a los equipos de desarrollo por lo que no se puede dar una gestión técnica adecuada. (OK).</p> <p>2. Sprint Review: ¿Cómo se podría identificar la deuda técnica más crítica? (OK).</p> <p>3. Definir: refactorización, backlog, deuda técnica. (OK).</p>
	Los errores en una tarea.	<p>1. Definir: pérdida de conocimiento (NO APLICA), deuda técnica (OK).</p>	<p>1. El término "silos", puede generar confusión. (OK).</p>	<p>1. Definir PO para referirse al Product Owner, (OK).</p>	<p>1. Definir: Existe rotación alta de personal (NO APLICA), dependencias técnicas (OK).</p> <p>3. En la sección Sprint Review en el segundo punto "Identificar oportunidades ..." evaluar su ubicación, de pronto hay conflicto entre Review y Retrospective. (OK).</p>	

3. Satisfacción.	Confianza.	1. Agregar tilde PO para que (este) en el... (OK).
	Comodidad.	1. Necesita cambiar los procesos y la mentalidad de las personas para adaptarse a las mejoras propuestas. (COMENTARIO).
5. Cobertura del Contexto.	Integridad del contexto.	1. Más que nada, se necesita integrar a los stakeholders para avanzar en las revisiones. (COMENTARIO).
	Flexibilidad.	1. Se puede adaptar, teniendo en cuenta la cuestión de tiempos y el flujo de trabajo que se aplicará. (COMENTARIO). 2. Se considera que ciertas instrucciones pueden utilizarse. (COMENTARIO). 3. En las metodologías tradicionales no se adapta. (COMENTARIO).



# MARCO DE TRABAJO PROPUESTO PARA:

## GESTIÓN DE RIESGOS DE CALIDAD DE SOFTWARE PARA PROYECTOS CON SCRUM

Elaborado por:  
**Francisco Alvarez**  
Autor



Aprobado por:  
**Wilman Chamba**  
Director de TT

**2024**

El marco de trabajo propuesto incluye una sección de términos clave en los que se estandariza la definición de algunos términos que se utilizan en el marco de trabajo; la segunda sección está compuestas por 3 partes: la información del riesgo, ¿Cómo identificar el riesgo? y ¿Cómo mitigar el riesgo?, en cada parte, se dan instrucciones que permiten conocer, identificar y dar solución a cada riesgo.

<b>TÉRMINOS CLAVE</b>	
<p><b>Postmortem de fallos.</b> Es una reunión o análisis realizado después de un fallo en un proyecto, con el fin de identificar las causas, aprender de los errores y prevenir su repetición en el futuro.</p>	<p><b>Product Owner (PO).</b> En Scrum es el responsable de gestionar el backlog del producto, priorizando las tareas según el valor para el negocio y asegurar que el equipo entregue el producto deseado.</p>
<p><b>Cultura que evita reconocer errores.</b> Hace referencia a un ambiente de trabajo donde los miembros del equipo no se sienten cómodos señalando o aceptando errores, obstaculizando la mejora continua y el aprendizaje colectivo.</p>	<p><b>Sprint.</b> Ciclo corto de trabajo en Scrum, típicamente de 1 a 4 semanas, donde el equipo se enfoca en completar tareas específicas para entregar un incremento del producto.</p>
<p><b>Valores límite.</b> Condiciones extremas o escenarios específicos que se utilizan para evaluar si el sistema funciona correctamente bajo situaciones fuera de lo común o cerca de sus límites operativos.</p>	<p><b>Backlog.</b> Lista priorizada de tareas, requisitos o historias de usuario que representan el trabajo pendiente por realizar en un proyecto con Scrum.</p>
<p><b>Refactorización.</b> Es el proceso de mejorar el diseño, la estructura o la legibilidad del código o cualquier artefacto sin alterar su funcionalidad, con el objetivo de aumentar su mantenibilidad o comprensión.</p>	<p><b>Burn-Down Chart.</b> Gráfico que muestra el trabajo restante en un Sprint frente al tiempo disponible, ayudando al equipo a monitorear su progreso hacia los objetivos del Sprint.</p>
<p><b>Deuda técnica.</b> Es la acumulación de problemas en el código por decisiones técnicas que no son óptimas, tomadas para acelerar entregas, pero que pueden generar costos adicionales y problemas de calidad.</p>	<p><b>Dependencias técnicas.</b> Relaciones entre componentes, equipos o tareas de un proyecto que requieren que ciertas actividades o entregables estén completados antes de que otros puedan avanzar.</p>

# MARCO DE GESTIÓN DE RIESGOS DE CALIDAD DE SOFTWARE PARA PROYECTOS CON SCRUM

## INFORMACIÓN DEL RIESGO

CATEGORÍA	NOMBRE	DESCRIPCIÓN	
<b>Mejora continua.</b> Engloba los riesgos relacionados con la falta de iniciativa para proponer mejoras o aprender de los errores, ya que los proyectos de Scrum trabajan con una iteración constante y la mejora continua. Si los equipos no fomentan una cultura de aprendizaje y no implementan mejoras de manera regular, los errores tienden a repetirse.	 <b>FALTA DE APRENDIZAJE Y GESTIÓN DE ERRORES.</b>	Si los errores no se discuten abiertamente y no se aprende de ellos, es probable que se repitan en futuras iteraciones, acumulando defectos y disminuyendo la calidad del producto.	
	<b>PROBABILIDAD</b>		<b>IMPACTO</b>
	Casi seguro.		Severo.

### ¿CÓMO IDENTIFICAR EL RIESGO?

EVENTO	CAUSA	CONSECUENCIA
Los errores cometidos en un sprint se están repitiendo en los siguientes sprints.	Falta de análisis y discusión de errores en las retrospectivas. No se discuten abiertamente los errores.	Acumulación de defectos, recurrencia de defectos similares.
Las lecciones aprendidas no se están documentando ni compartiendo entre los miembros del equipo.	Falta de cultura organizacional orientada al aprendizaje. Presión por comenzar el siguiente sprint.	Pérdida de conocimiento que limita la capacidad de mejorar la calidad del producto a largo plazo. Repetición de problemas técnicos.
No se están documentando ni analizando los problemas que están ocurriendo en los sprints (Ausencia de análisis post-mortem de fallos).	Equipos sin un enfoque estructurado en base a la mejora continua. Falta de tiempo dedicado al análisis de los problemas.	Los problemas se arrastran de un sprint a otro, reduciendo la calidad del producto y perdiendo oportunidades de mejora.
El equipo no está abordando ni corrigiendo los defectos identificados en fases anteriores.	Prioridad en la entrega de nuevas funcionalidades sobre la corrección de errores pasados. Cultura que evita reconocer errores.	Aumento en la deuda técnica, afectando la mantenibilidad del software.
Los problemas técnicos que están surgiendo en cada iteración no se están resolviendo de manera efectiva (Omisión de análisis de la causa raíz).	Ausencia de procesos de retroalimentación. Equipos con poca disciplina para la refactorización y mejora continua.	Baja calidad en las futuras versiones del producto debido a la acumulación de problemas técnicos no resueltos. Deterioro continuo de la calidad.

### ¿CÓMO MITIGAR EL RIESGO?

SPRINT PLANNING	DAILY SCRUM	SPRINT REVIEW	SPRINT REVIEW
<ul style="list-style-type: none"> <li>Incluir tiempo en el sprint para análisis de errores previos.</li> <li>Definir responsables para asegurar la documentación de lecciones aprendidas.</li> <li>Planificar revisiones técnicas periódicas.</li> </ul>	<ul style="list-style-type: none"> <li>Revisar brevemente los errores detectados, su estatus y cómo se está aprendiendo de ellos.</li> <li>Fomentar la comunicación abierta sobre cualquier error para evitar su repetición.</li> <li>Solucionar impedimentos relacionados con la corrección de errores.</li> </ul>	<ul style="list-style-type: none"> <li>Revisar los errores más críticos detectados en el sprint y discutir cómo evitarlos en el siguiente.</li> <li>Incluir a los stakeholders en la discusión de cómo mejorar la calidad del producto.</li> </ul>	<ul style="list-style-type: none"> <li>Establecer mejoras claras basadas en los errores cometidos en el sprint.</li> <li>Crear una lista de acciones específicas que se puedan implementar para prevenir futuros errores similares.</li> <li>Comunicar a todo el equipo las acciones de mejora.</li> </ul>

### GUÍA DE USO DEL MARCO DE TRABAJO



# MARCO DE GESTIÓN DE RIESGOS DE CALIDAD DE SOFTWARE PARA PROYECTOS CON SCRUM

## INFORMACIÓN DEL RIESGO

CATEGORÍA	NOMBRE	DESCRIPCIÓN		
<b>Gestión de calidad y colaboración.</b>  Esta categoría cubre los riesgos derivados de la falta de colaboración efectiva y la falta de una comunicación clara dentro del equipo y con los stakeholders, lo que puede dificultar la obtención de retroalimentación oportuna de las partes interesadas.	 <b>GESTIÓN INEFICAZ DE LA CALIDAD POR FALTA DE COLABORACIÓN.</b>	La falta de un enfoque colaborativo y de aprendizaje puede dificultar la implementación de prácticas efectivas de gestión de la calidad, resultando en un producto que no cumple con los estándares de calidad esperados.		
	<table border="1"> <tr> <th>PROBABILIDAD</th> <th>IMPACTO</th> </tr> <tr> <td>Casi seguro.</td> <td>Severo.</td> </tr> </table>		PROBABILIDAD	IMPACTO
PROBABILIDAD	IMPACTO			
Casi seguro.	Severo.			

### ¿CÓMO IDENTIFICAR EL RIESGO?

EVENTO	CAUSA	CONSECUENCIA
Los problemas de calidad no están siendo abordados (falta de alineación entre los roles clave del equipo).	Descoordinación entre los diferentes roles (Scrum Master, Product Owner, Desarrolladores).	Aumento en los tiempos de corrección de errores y retrasos en la entrega del producto.
Están ocurriendo deficiencias en la integración de módulos debido a la falta de cooperación entre los equipos.	Comunicación deficiente entre los equipos sobre las dependencias técnicas.	Errores de integración que generan defectos en el producto final.
Falta revisión cruzada entre los equipos para asegurar la calidad de los entregables. No se está practicando pair programming o las revisiones de código son superficiales.	Equipos distribuidos con poca colaboración. Resistencia al trabajo en equipo.	Problemas de calidad que afectan la percepción del cliente sobre el producto. Pérdida de oportunidades de mentoring. Deterioro de la calidad del código.
No está existiendo coordinación entre los equipos para asegurar la calidad de los entregables. Se están generando barreras de conocimiento en el equipo.	Falta de colaboración entre equipos que trabajan en paralelo, falta de cultura colaborativa.	Disminución de la cohesión del producto, afectando su estabilidad y funcionalidad. Inconsistencias en la implementación.
Los equipos están trabajando de manera aislada, sin estar compartiendo lecciones aprendidas o tomando decisiones técnicas no consensuadas.	Poca integración entre equipos en retrospectivas y reuniones de planificación. Roles excesivamente especializados.	Pérdida de conocimiento y de oportunidades de mejora en el proceso de desarrollo. Fragmentación del conocimiento técnico.

### ¿CÓMO MITIGAR EL RIESGO?

SPRINT PLANNING	DAILY SCRUM	SPRINT REVIEW	SPRINT REVIEW
<ul style="list-style-type: none"> <li>Fomentar la colaboración temprana asignando tareas conjuntas entre roles (Product Owner, desarrolladores, testers).</li> <li>Definir sesiones de planificación compartida para asegurar alineación de expectativas.</li> <li>Definir tiempo para pruebas de integración del software desarrollado por diferentes equipos.</li> </ul>	<ul style="list-style-type: none"> <li>Crear un ambiente donde todos los miembros del equipo puedan expresar preocupaciones sobre la calidad y colaboración.</li> <li>Asegurarse de que las tareas cruzadas entre equipos estén claras.</li> </ul>	<ul style="list-style-type: none"> <li>Incluir a todos los miembros clave en la revisión del sprint para garantizar que la calidad sea evaluada desde todas las perspectivas.</li> <li>Revisar el flujo de colaboración entre equipos.</li> </ul>	<ul style="list-style-type: none"> <li>Identificar problemas de colaboración específicos y definir planes para mejorar la comunicación en el siguiente sprint.</li> <li>Planificar sesiones de trabajo colaborativas antes del sprint.</li> </ul>

### GUÍA DE USO DEL MARCO DE TRABAJO



# MARCO DE GESTIÓN DE RIESGOS DE CALIDAD DE SOFTWARE PARA PROYECTOS CON SCRUM

## INFORMACIÓN DEL RIESGO

CATEGORÍA	NOMBRE	DESCRIPCIÓN	
<b>Gestión de calidad y colaboración.</b>  Esta categoría cubre los riesgos derivados de la falta de colaboración efectiva y la falta de una comunicación clara dentro del equipo y con los stakeholders, lo que puede dificultar la obtención de retroalimentación oportuna de las partes interesadas.	 <b>RETROALIMENTACIÓN INCOMPLETA DE LOS STAKEHOLDERS.</b>	La falta de retroalimentación constante y precisa de los usuarios finales o partes interesadas durante cada Sprint, puede llevar al desarrollo de características que no cumplan con las expectativas o necesidades del cliente.	
	<b>PROBABILIDAD</b>		<b>IMPACTO</b>
	Casi seguro.		Severo.

### ¿CÓMO IDENTIFICAR EL RIESGO?

EVENTO	CAUSA	CONSECUENCIA
Los stakeholders no están participando activamente en las revisiones de sprint ni en el refinamiento del producto.	Falta de compromiso por parte de los stakeholders en el proyecto o Stakeholders sobrecargados.	Características construidas que no cumplen con las necesidades o expectativas del cliente. Pérdida de valor del producto.
La retroalimentación de los stakeholders no se está recibiendo de manera oportuna, se recibe de manera tardía.	Comunicación deficiente entre el equipo y los stakeholders. Canales de comunicación inadecuados.	El equipo trabaja en funcionalidades que no son prioritarias para el cliente. Desarrollo de funcionalidades incorrectas, retrabajos costosos.
Los requerimientos de los stakeholders no están siendo claros ni están siendo actualizados. Los requisitos están siendo mal interpretados.	Los stakeholders no brindan una retroalimentación clara y detallada durante los sprints. Existen barreras de comunicación.	Producto final que no refleja las verdaderas necesidades del cliente. Insatisfacción del cliente.
Las prioridades del cliente no se están reflejando correctamente en las historias de usuario del backlog. Está existiendo una validación insuficiente con los usuarios.	Desconexión entre el Product Owner y los stakeholders durante la planificación de los sprints. Procesos de feedback no estructurados.	Insatisfacción del cliente debido a la falta de alineación con sus expectativas y necesidades reales.
El equipo no está teniendo suficiente información para ajustar el backlog de acuerdo con las prioridades del cliente.	Falta de reuniones efectivas de revisión con los stakeholders.	Retrasos en la implementación de funcionalidades críticas para el cliente.

### ¿CÓMO MITIGAR EL RIESGO?

SPRINT PLANNING	DAILY SCRUM	SPRINT REVIEW	SPRINT REVIEW
<ul style="list-style-type: none"> <li>Asegurarse de que el Product Owner se comunique regularmente con los stakeholders para priorizar los requisitos.</li> <li>Documentar y revisar los requisitos detalladamente durante la planificación.</li> <li>Gestionar que el Product Owner disponga de un tiempo diario asignado al proyecto para resolver dudas.</li> </ul>	<ul style="list-style-type: none"> <li>Mantener informados a los stakeholders del progreso a través de actualizaciones periódicas.</li> <li>Discutir cualquier cambio solicitado por los stakeholders durante las reuniones diarias.</li> </ul>	<ul style="list-style-type: none"> <li>Asegurar la participación activa de los stakeholders en la revisión del sprint.</li> <li>Revisar si se cumplieron las expectativas del cliente y planificar mejoras.</li> </ul>	<ul style="list-style-type: none"> <li>Evaluar si la retroalimentación de los stakeholders fue clara y suficiente.</li> <li>Definir un plan para mejorar la comunicación con los stakeholders en sprints futuros.</li> </ul>

### GUÍA DE USO DEL MARCO DE TRABAJO



# MARCO DE GESTIÓN DE RIESGOS DE CALIDAD DE SOFTWARE PARA PROYECTOS CON SCRUM

## INFORMACIÓN DEL RIESGO

CATEGORÍA	NOMBRE		DESCRIPCIÓN
<b>Gestión de pruebas.</b> Esta categoría agrupa los riesgos relacionados con pruebas insuficientes (funcionales y no funcionales), ya sea por la falta de experiencia del equipo o la presión por cumplir con los plazos de los sprints. Si las pruebas no son rigurosas, los defectos pueden afectar la estabilidad y la funcionalidad del software	 <b>AUMENTO DE DEFECTOS POR INEXPERIENCIA DEL EQUIPO.</b>		Los equipos ágiles, especialmente los nuevos, tienden a experimentar un aumento de defectos debido a la falta de experiencia o a una implementación apresurada. Ocurre algo similar cuando se tiene poco conocimiento del giro del negocio, de metodologías ágiles o del área de pruebas. El aumento de defectos puede afectar la estabilidad y la funcionalidad del software.
	PROBABILIDAD	IMPACTO	
	Casi seguro.	Severo.	

### ¿CÓMO IDENTIFICAR EL RIESGO?

EVENTO	CAUSA	CONSECUENCIA
Se están cometiendo errores en el manejo de casos de prueba que evalúan los valores límites.	Presión por entregar rápidamente.	Los defectos son encontrados por el usuario final en producción. Se presentan errores funcionales del sistema cuando se ingresa valores límite.
Los plazos no se están cumpliendo debido a la inexperiencia en estimaciones ágiles.	Falta de experiencia en la estimación y planificación de sprints.	Retrasos en la entrega del producto final, afectando la satisfacción del cliente.
No se están realizando suficientes pruebas por la falta de experiencia con herramientas de pruebas o falta de conocimiento del negocio, lo que genera código no optimizado.	Falta de capacitación en herramientas de pruebas y automatización. Existe rotación alta de personal. Inducción no adecuada al proyecto del nuevo personal.	Defectos críticos que llegan a producción, afectando la estabilidad del software. Deterioro del rendimiento.
El equipo nuevo no está siguiendo las mejores prácticas de desarrollo, generando defectos en el producto. Por ejemplo, están implementando patrones de manera incorrecta.	Falta de experiencia en el equipo para implementar Scrum de manera efectiva. Falta de capacitación técnica.	Inestabilidad del sistema. Aumento de defectos en las primeras entregas del producto.
El equipo no gestiona correctamente las dependencias entre módulos, lo que está generando errores en la integración.	Falta de conocimiento sobre la planificación de dependencias técnicas en Scrum, falta de documentación técnica.	Problemas de integración entre los diferentes módulos del sistema. Mayor costo de mantenimiento.

### ¿CÓMO MITIGAR EL RIESGO?

SPRINT PLANNING	DAILY SCRUM	SPRINT REVIEW	SPRINT REVIEW
<ul style="list-style-type: none"> <li>Planificar tareas de menor complejidad para los miembros con menos experiencia.</li> <li>Incluir revisiones más frecuentes de los entregables durante el sprint.</li> <li>Planificar capacitaciones o tutorías.</li> </ul>	<ul style="list-style-type: none"> <li>Monitorear el progreso del personal con menos experiencia para proporcionar soporte temprano.</li> <li>Facilitar el apoyo de miembros más experimentados durante el desarrollo diario.</li> <li>Considerar el uso de técnicas de monitoreo de equipo como: tablero kanban, burn-down chart, monitoreo de impedimentos y el recuento diario de historias de usuario completadas.</li> </ul>	<ul style="list-style-type: none"> <li>Incluir una revisión técnica detallada de los entregables realizados por los miembros más inexpertos.</li> <li>Identificar conjuntamente con los involucrados en el Sprint Review las oportunidades de mejora de las funcionalidades presentadas.</li> </ul>	<ul style="list-style-type: none"> <li>Planificar capacitaciones adicionales para los miembros del equipo que necesiten mejorar sus habilidades.</li> <li>Discutir qué problemas se presentaron por la falta de experiencia y cómo mitigarlos.</li> </ul>

### GUÍA DE USO DEL MARCO DE TRABAJO



# MARCO DE GESTIÓN DE RIESGOS DE CALIDAD DE SOFTWARE PARA PROYECTOS CON SCRUM

## INFORMACIÓN DEL RIESGO

CATEGORÍA	NOMBRE	DESCRIPCIÓN	
<b>Gestión técnica.</b>  Los riesgos en esta categoría están relacionados con las decisiones técnicas no adecuadas que afectan la calidad del software, como la gestión de la deuda técnica.	 <b>GESTION INADECUADA DE LA DEUDA TÉCNICA.</b>	La necesidad de entregar productos rápidamente puede llevar a decisiones técnicas que no son óptimas, generando deuda técnica. Al no ser gestionada adecuadamente, los problemas no resueltos en etapas tempranas del proyecto se acumulan, comprometiendo la integridad del código y afectando negativamente el rendimiento, la escalabilidad y la mantenibilidad del software.	
	<b>PROBABILIDAD</b>		<b>IMPACTO</b>
	Casi seguro.		Severo.

### ¿CÓMO IDENTIFICAR DEL RIESGO?

EVENTO	CAUSA	CONSECUENCIA
Se está postergando continuamente la refactorización.	Decisiones técnicas apresuradas para cumplir con los plazos del sprint. Presión por nuevas funcionalidades.	El rendimiento del software se ve comprometido, afectando la experiencia del usuario. Código difícil de mantener.
Los problemas técnicos se están acumulando sin ser abordados, generando defectos en fases posteriores.	Falta de refactorización continua y gestión de deuda técnica en cada sprint.	Aumento de los costos de mantenimiento y mayor esfuerzo a largo plazo en la corrección de errores.
Las funcionalidades implementadas en los primeros sprints no se están optimizando adecuadamente (se están implementando soluciones rápidas sin mayor análisis).	Falta de revisiones técnicas en cada sprint. Priorización incorrecta de la deuda técnica, falta de tiempo para mejoras.	Dificultades para escalar y mantener el sistema en versiones futuras. Dificultad para evolucionar.
El equipo de desarrollo está priorizando la entrega de nuevas funcionalidades sobre la corrección de la deuda técnica (ignorar warnings de código, omitir estándares de código, acumular bugs menores, etc.).	Equipos con poca disciplina para equilibrar entre entrega de funcionalidades y calidad del código. Ausencia de políticas de calidad.	Degradación progresiva del código, lo que dificulta la mantenibilidad del software. Velocidad reducida a largo plazo (rendimiento).
El equipo no tiene visibilidad sobre la deuda técnica acumulada.	Falta de herramientas para medir y rastrear la deuda técnica en el proyecto.	Problemas técnicos acumulados que afectan la escalabilidad del sistema. Costos crecientes de desarrollo.

### ¿CÓMO MITIGAR EL RIESGO?

SPRINT PLANNING	DAILY SCRUM	SPRINT REVIEW	SPRINT REVIEW
<ul style="list-style-type: none"> <li>Priorizar la deuda técnica como parte del backlog del sprint.</li> <li>Considerar tiempos de holgura para solución de deuda técnica generada en cada sprint.</li> <li>Asegurar que se asignen tiempos específicos para la refactorización y reducción de deuda técnica en cada sprint.</li> <li>Planificar la instalación y configuración de herramientas que permitan gestionar la deuda técnica.</li> </ul>	<ul style="list-style-type: none"> <li>Hacer seguimiento de las tareas de refactorización y reducción de deuda técnica.</li> <li>Discutir cualquier impedimento que pueda retrasar la reducción de la deuda técnica durante el sprint.</li> </ul>	<ul style="list-style-type: none"> <li>Evaluar los avances en la reducción de la deuda técnica.</li> <li>Identificar la deuda técnica más crítica, mediante la revisión del impacto funcional, la revisión de código y pruebas, retroalimentación de los stakeholders, etc.</li> <li>Priorizar para el próximo sprint la deuda técnica más crítica que aún no ha sido atendida.</li> </ul>	<ul style="list-style-type: none"> <li>Discutir cómo la deuda técnica afectó el sprint y planificar cómo se gestionará en el futuro.</li> <li>Establecer métricas para medir la reducción de deuda técnica en cada sprint.</li> </ul>

### GUÍA DE USO DEL MARCO DE TRABAJO



1  
Conozca el riesgo que puede presentarse en su proyecto.



2  
Comprenda ¿cómo identificarlo?



3  
Comprenda ¿cómo mitigarlo?

## Anexo 5. Certificado de traducción del resumen.

### CERTIFICACIÓN DE TRADUCCIÓN

Loja, 19 de diciembre de 2024

Lic. Viviana Valdivieso Loyola Mg. Sc.

**DOCENTE DE INGLÉS**

A petición verbal de la parte interesada:

#### **CERTIFICA:**

Que, desde mi legal saber y entender, como profesional en el área del idioma inglés, he procedido a realizar la traducción del resumen, correspondiente al Trabajo de Integración Curricular titulado **Desarrollo de un marco de trabajo para la gestión de riesgos de calidad en proyectos de desarrollo de software ágil, basado en la ISO 31000 y gestionados con Scrum**, de la autoría de: **Francisco Javier Alvarez Pineda**, portador de la cédula de identidad número **1103635643**

Para efectos de traducción se han considerado los lineamientos que corresponden a un nivel de inglés técnico, como amerita el caso.

Es todo cuanto puedo certificar en honor a la verdad, facultando al portador del presente documento, hacer uso del mismo, en lo que a bien tenga.

Atentamente. -



Lic. Viviana Valdivieso Loyola Mg. Sc.

1103682991

Nº Registro Senescyt 4to nivel **1031-2021-2296049**

Nº Registro Senescyt 3er nivel **1008-16-1454771**