



## Trabajo Fin de Máster

ASEGURAMIENTO OFICIAL DE LA CALIDAD (AOC) EN  
PROYECTOS                    GESTIONADOS                    MEDIANTE  
METODOLOGÍAS AGILE

Raquel Medrano Montanel

Director: Dra. Dña. Marta Torralba Gracia

Centro Universitario de la Defensa-Academia General Militar

2025



## Agradecimientos

A mi familia y amigos por su constante ánimo y apoyo incondicional a lo largo de este viaje académico. Y a mí misma, por haberme embarcado en este máster y haber logrado llegar hasta aquí superando cada uno de los obstáculos.



## RESUMEN

This Master's Final Project (TFM) addresses the integration of agile methodologies in defence projects managed by the Ministry of Defence (MINISDEF) and its contractors. This study arises due to the increase in the implementation of agile methodologies in multiple areas, where the ability to quickly adjust and adaptability are essential.

It analyses how these methodologies can coexist with current quality standards, such as PECAL/AQAP (Publicación Española de Calidad/ Allied Quality Assurance Publication) standards, and proposes strategies to ensure quality in an agile environment. The differences between traditional and agile methodologies are explored, and it is analysed how the latter can improve the efficiency and quality of projects. Additionally, specific tools and practices that facilitate the integration of quality at all stages of the projects are discussed.

The methodology used in the study is qualitative and analytical, based on document analysis and the use of techniques such as SWOT analysis (Weaknesses, Threats, Strengths, and Opportunities), risk analysis, and the Analytical Hierarchy Process (AHP). These techniques allow evaluating and selecting the most suitable agile methodologies for the context of OQA (Official Quality Assurance). Furthermore, specific strategies for the implementation of these methodologies are proposed, including the adaptation of the PECAL/AQAP 2210 standard to an agile approach.

The results of the study show that it is possible to integrate agile methodologies into defence projects without compromising quality. Agile methodologies allow greater flexibility and adaptation to changes, improving user satisfaction and the efficiency of certain types of projects, such as software development.

It is concluded that the adoption of agile methodologies can significantly improve project management in the defence sector, as long as existing quality standards are adequately adapted

## Palabras clave

Agile, PECAL/AQAP (Publicación Española de Calidad/ Allied Quality Assurance Publication), Scrum, Kanban, AOC/OQA (Aseguramiento Oficial de la Calidad/ Official Quality Assurance).



# INDICE DE CONTENIDO

<b><i>Agradecimientos</i></b> .....	<b>I</b>
<b><i>RESUMEN</i></b> .....	<b>II</b>
<b><i>Palabras clave</i></b> .....	<b>II</b>
<b><i>INDICE DE CONTENIDO</i></b> .....	<b>III</b>
<b><i>INDICE DE FIGURAS</i></b> .....	<b>VI</b>
<b><i>INDICE DE TABLAS</i></b> .....	<b>VII</b>
<b><i>ABREVIATURAS, SIGLAS Y ACRÓNIMOS</i></b> .....	<b>VIII</b>
<b>1    <i>INTRODUCCIÓN</i></b> .....	<b>1</b>
<b>2    <i>OBJETIVOS Y METODOLOGÍA</i></b> .....	<b>2</b>
<b>2.1    <i>OBJETIVOS Y ALCANCE</i></b> .....	<b>2</b>
2.1.1    Objetivos .....	2
2.1.2    Alcance .....	2
<b>2.2    <i>METODOLOGÍA</i></b> .....	<b>3</b>
<b>3    <i>ANTECEDENTES Y MARCO TEÓRICO</i></b> .....	<b>5</b>
<b>3.1    <i>Marco Normativo</i></b> .....	<b>5</b>
<b>3.2    <i>Metodologías de Gestión de Proyectos: Aplicación al caso de estudio</i></b> .....	<b>6</b>
<b>3.3    <i>Metodología Tradicional (Ciclo de Vida Predictivo)</i></b> .....	<b>7</b>
<b>3.4    <i>Metodología Agile (Ágil)</i></b> .....	<b>8</b>
3.4.1    Scrum .....	11
3.4.2    Kanban .....	11



3.4.3	SAFe .....	11
3.4.4	Lean Management.....	11
3.4.5	Spotify Model .....	11
3.4.6	Feature-Driven Development (FDD) .....	11
<b>4</b>	<b>DESARROLLO: ANÁLISIS Y RESULTADOS .....</b>	<b>12</b>
<b>4.1</b>	<b>Metodología Tradicional vs Metodología Agile .....</b>	<b>12</b>
4.1.1	Análisis DAFO .....	13
4.1.2	Análisis de Agile desde el punto de vista de Calidad .....	15
<b>4.2</b>	<b>Metodologías Agile: Análisis de Alternativas .....</b>	<b>16</b>
4.2.1	Primera Etapa. Representación del Problema .....	16
4.2.2	Segunda Etapa: Evaluación de los criterios.....	17
4.2.3	Tercera Etapa: Evaluación de las Alternativas.....	17
4.2.4	Cuarta Etapa: Jerarquización de las Alternativas .....	20
<b>4.3</b>	<b>Propuesta Agile: Aseguramiento Oficial de la Calidad .....</b>	<b>21</b>
4.3.1	Scrum.....	21
4.3.2	Kanban.....	30
4.3.3	Actividades Específicas de Aseguramiento de Calidad .....	33
<b>4.4</b>	<b>Normativa PECAL/AQAP vs Enfoque Agile: análisis .....</b>	<b>33</b>
4.4.1	Adaptación PECAL/AQAP 2210 a Scrum & Kanban.....	33
4.4.2	Métricas de Calidad (KPI).....	35
<b>4.5</b>	<b>Análisis de Riesgos: implantación de Scrum y Kanban .....</b>	<b>39</b>
<b>4.6</b>	<b>Futuros Pasos .....</b>	<b>40</b>
<b>5</b>	<b>CONCLUSIONES .....</b>	<b>42</b>



<b>6</b>	<b>REFERENCIAS BIBLIOGRÁFICAS .....</b>	<b>44</b>
<b>7</b>	<b>ANEXOS.....</b>	<b>45</b>
	<i>Anexo I Técnica de Decisión Multicriterio – Metodología Analytic Hierarchy Process (AHP) .....</i>	<b>46</b>
	<i>Anexo II PECAL 2210 y Agile.....</i>	<b>49</b>
	<i>Anexo III Análisis de Riesgos: implantación de SCRUM y KANBAN en una organización tradicional desde el punto de vista de la gestión y del aseguramiento de la calidad .....</i>	<b>68</b>



## INDICE DE FIGURAS

Figura 1 Esquema de la metodología de este TFM: tareas, herramientas y fuentes de información.	4
Figura 2 Organigrama del actual MINISDEF español.	6
Figura 3 Introducción PMBOK guide (Project Management Institute Inc, 2017)	7
Figura 4 Valores del Manifiesto Ágil.	9
Figura 5 Principios del Manifiesto Ágil	9
Figura 6 Análisis de los marcos de trabajo	10
Figura 7 Metodologías Agile más utilizadas en la industria	10
Figura 8 Esquema de trabajo: Ciclo de vida PREDICTIVO vs AGILE	12
Figura 9 AHP: Primera etapa_Representación del Problema, Criterios y Alternativas	16
Figura 10 Roles de Scrum: Propuesta 1	23
Figura 11 Roles de Scrum: Propuesta 2	24
Figura 12 Elementos y Eventos de Scrum	25
Figura 13 Product backlog items (PBI)	25
Figura 14 Ejemplo de Tablero Kanban completo	30
Figura 15 Ejemplo Tablero Kanban desarrollo SW (sprint)	32
Figura 16 SAFe: Ejemplo	41
Figura 17 Anexo I: Ejemplo Etapa 1	47
Figura 18 Anexo I: Escala de Saaty	47
Figura 19 Anexo III: Matriz de índice de riesgo.	69



## INDICE DE TABLAS

Tabla 1 Relación entre las normas PECAL/AQAP en vigor a fecha de realización de este trabajo.	5
Tabla 2 Ciclos de vida Proyectos defensa (MINISDEF-SENER Aeroespacial)	7
Tabla 3 Metodología ÁGIL vs TRADICIONAL	13
Tabla 4 Metodología Agile: análisis DAFO (MINISDEF)	14
Tabla 5 Metodología Agile: análisis DAFO (Empresa Privada)	14
Tabla 6 AHP: Segunda etapa_Comparativa 2 a 2 de los criterios	17
Tabla 7 AHP: Tercera etapa_Criterio: Calidad	18
Tabla 8 AHP: Tercera etapa_Criterio: Flexibilidad	18
Tabla 9 AHP: Tercera etapa_Criterio: Implementación	19
Tabla 10 AHP: Tercera etapa_Criterio: Aprendizaje	19
Tabla 11 AHP: Tercera etapa_Criterio: Recursos	20
Tabla 12 AHP: Tercera etapa_Criterio: Escalabilidad	20
Tabla 13 AHP: Cuarta etapa_Jerarquización de Alternativas	21
Tabla 14 Scrum: Propuestas equipos y roles	23
Tabla 15 Ejemplo de Product backlog	27
Tabla 16 Ejemplo de Sprint backlog	28
Tabla 17 Ejemplo de Incremento	28
Tabla 18 Scrum: Eventos	29
Tabla 19 Resumen PECAL 2210 y Agile (Anexo II)	35
Tabla 20 Métricas de Calidad (SW)	36
Tabla 21 Resumen Análisis de Riesgos (Anexo III)	39
Tabla 22 Cumplimiento de Objetivos	43



## ABREVIATURAS, SIGLAS Y ACRÓNIMOS

<b>AII/Aii</b>	Área de Inspecciones Industriales
<b>ANAC</b>	Autoridad Nacional de Aseguramiento de la Calidad
<b>AOC</b>	Aseguramiento Oficial de la Calidad
<b>APQP</b>	<i>Advanced Product Quality Planning</i>
<b>AQAP</b>	<i>Allied Quality Assurance Publication</i>
<b>BOD</b>	Boletín Oficial de Defensa
<b>BOE</b>	Boletín Oficial del Estado
<b>DAFO</b>	Debilidades, Amenazas, Fortalezas y Oportunidades
<b>DDV</b>	Documento de Viabilidad
<b>DDP</b>	Directiva de Programa
<b>DIGAM</b>	Director General de Armamento y Material
<b>DIGENECO</b>	Dirección General de Asuntos Económicos
<b>DINFRA</b>	Dirección General de Infraestructura
<b>DGAM</b>	Dirección General de Armamento y Material
<b>ESA</b>	<i>European Space Agency</i> (Agencia Espacial Europea)



<b>FAS</b>	Fuerzas Armadas
<b>FDD</b>	<i>Feature-Driven Development</i>
<b>GQAR</b>	<i>Government Quality Assurance Representative</i>
<b>INREID</b>	Inspección, Regulación y Estrategia Industrial de Defensa
<b>ISDEFE</b>	Ingeniería de Sistemas para la Defensa de España
<b>JEMAD</b>	Jefe del Estado Mayor de la Defensa
<b>KPI</b>	<i>Key Performance Indicator</i> (Indicador Clave de Rendimiento)
<b>MINISDEF</b>	Ministerio de Defensa
<b>NATO</b>	<i>North Atlantic Treaty Organization</i>
<b>N/A</b>	No Aplica
<b>OCCAR</b>	<i>Organisation for Joint Armament Cooperation</i>
<b>OEM</b>	Objetivo de Estado Mayor
<b>OTAN</b>	Organización del Tratado del Atlántico Norte
<b>PBI</b>	Product backlog ítem
<b>PECAL</b>	Publicación Española de Aseguramiento de Calidad
<b>PLATIN</b>	Planificación, Tecnología e Innovación



<b>PMBOK</b>	<i>Project Management Body of Knowledge</i>
<b>PMI</b>	<i>Project Management Institute</i>
<b>RAC</b>	Representante de Aseguramiento de la Calidad
<b>REM</b>	Requisitos de Estado Mayor
<b>SAFe</b>	<i>Scaled Agile Framework</i>
<b>SDG</b>	Subdirección General
<b>SEDEF</b>	Secretaría de Estado de la Defensa
<b>STANAG</b>	<i>Standardization Agreement</i>
<b>SUBDEF</b>	Subsecretaría de Estado de la Defensa
<b>TFM</b>	Trabajo Fin de Máster
<b>TPS</b>	<i>Toyota Production System</i>
<b>UE</b>	Unión Europea
<b>WIP</b>	<i>Work in Progress</i>
<b>WP</b>	<i>Work Package</i>



# 1 INTRODUCCIÓN

El presente Trabajo Fin de Máster (TFM) aborda el Aseguramiento Oficial de la Calidad (AOC) que realizan los Representantes de Aseguramiento de la Calidad (RAC) en proyectos gestionados mediante metodologías Agile. La necesidad de este estudio surge de la creciente adopción de metodologías ágiles en diversos sectores, incluyendo el ámbito de la defensa, donde la flexibilidad y la capacidad de adaptación son cruciales. Tradicionalmente, los proyectos en este sector se han gestionado mediante metodologías predictivas, lo que ha generado desafíos en la integración de enfoques ágiles sin comprometer los estándares de calidad establecidos.

El ámbito de aplicación de este TFM se centra en la implementación de metodologías ágiles en proyectos de defensa, específicamente en aquellos gestionados por el Ministerio de Defensa (MINISDEF) y sus contratistas. Se analiza cómo estas metodologías pueden coexistir con las normativas de calidad vigentes, como las normas PECALE/AQAP, y se proponen estrategias para asegurar la calidad en un entorno ágil.

Los conceptos clave abordados en este estudio incluyen las metodologías ágiles más relevantes, como Scrum y Kanban, y su aplicación en el contexto del AOC. Se exploran las diferencias entre las metodologías tradicionales y ágiles, y se analiza cómo estas últimas pueden mejorar la eficiencia y la calidad de los proyectos. Además, se discuten las herramientas y prácticas específicas que facilitan la integración de la calidad en todas las etapas del proyecto.

El trabajo se estructura en 5 secciones. En primer lugar, se presentan los objetivos y la metodología del estudio. A continuación, se proporciona un marco teórico que incluye los antecedentes y la normativa relevante. Seguidamente, se realiza un análisis comparativo entre las metodologías tradicionales y ágiles, destacando sus fortalezas y debilidades. Posteriormente, se proponen estrategias específicas para la implementación de metodologías ágiles en el contexto del AOC, incluyendo ejemplos de implantación y un análisis detallado de la norma PECALE/AQAP 2210<sup>1</sup>. Finalmente, se presentan las conclusiones y recomendaciones para futuras investigaciones y aplicaciones prácticas obtenidas en este trabajo.

---

<sup>1</sup> PECALE-2210 REQUISITOS OTAN DE ASEGURAMIENTO DE LA CALIDAD DEL SOFTWARE, SUPLEMENTARIOS A LA PECALE-2110 O A LA PECALE-2310 (VERSIÓN ESPAÑOLA DE LA AQAP-2210 (Edición B, Versión 1, agosto 2022).

NATO SUPPLEMENTARY SOFTWARE QUALITY ASSURANCE REQUIREMENTS TO AQAP-2110 OR AQAP-2310.  
Edition A Version 2 (September 2015)



## 2 OBJETIVOS Y METODOLOGÍA

En esta sección se detallan los objetivos principales del estudio y la metodología empleada para alcanzarlos. Los objetivos de este TFM se centran en garantizar que la implementación de una metodología ágil en un proyecto no impacta en el aseguramiento de calidad. Para lograr estos objetivos, se sigue una metodología cualitativa, cuantitativa y analítica.

### 2.1 OBJETIVOS Y ALCANCE

En este apartado se presentan los objetivos a cumplir en detalle y el alcance del estudio, que abarca la comparativa de metodologías tradicionales y ágiles, la flexibilidad de estas últimas, el enfoque en la iteración y satisfacción con el usuario y la integración de la calidad desde el inicio hasta la completitud de los proyectos.

#### 2.1.1 Objetivos

El objetivo principal de este proyecto es el de analizar y profundizar en las estrategias para asegurar la calidad en proyectos propios de MINISDEF que requieran adoptar este enfoque ágil y no convencional, explorando las mejores prácticas y herramientas que faciliten este objetivo para el contexto considerado.

Los objetivos secundarios a conseguir tras la finalización de este TFM son:

- **Objetivo 1 (O#1):** propuesta de integración de responsables (RAC) y prácticas de calidad en proyectos y equipos ágiles.
- **Objetivo 2 (O#2):** análisis documental de la norma PECAL 2210 desde un punto de vista ágil o *Agile*.
- **Objetivo 3 (O#3):** análisis de las fortalezas y debilidades de la metodología tradicional vs metodología ágil en base al contexto en estudio.
- **Objetivo 4 (O#4):** propuesta de maneras de proceder y modificaciones necesarias (*way forward*) en los modelos tradicionales actuales hacia una metodología Agile manteniendo las buenas prácticas y garantía de calidad.
- **Objetivo 5 (O#5):** Análisis de riesgos de la implantación de metodologías Agile e identificación de futuras líneas y oportunidades de trabajo del AOC.

#### 2.1.2 Alcance

En proyectos donde los requisitos cambian con frecuencia, la adopción de una metodología ágil se presenta como la opción más adecuada. Esta metodología permite una mayor flexibilidad y adaptación a las necesidades cambiantes del proyecto, priorizando la satisfacción de los usuarios sobre la documentación exhaustiva y la aplicación rígida de los procesos de gestión y desarrollo.

Siguiendo esta línea de rapidez y flexibilidad, se propone no centrar el esfuerzo en la estricta adherencia a los planes de calidad tradicionales, sino en garantizar la implementación de un



enfoque orientado a la calidad desde el inicio. Este enfoque, conocido como "*planning for quality*" en contraposición a "*quality plan*", busca integrar la calidad de manera intrínseca en todas las etapas del proyecto.

Este Trabajo de Fin de Máster se enfocará en analizar y profundizar en las estrategias para asegurar la calidad en proyectos que adopten este enfoque ágil y no convencional, explorando las mejores prácticas y herramientas que faciliten este objetivo dentro del ámbito MINISDEF.

## 2.2 METODOLOGÍA

La metodología elegida para la ejecución del TFM ha sido cualitativa, cuantitativa y analítica.

Partiendo del análisis documental, que se concibe como una técnica que consiste en "la selección y recopilación de información, por medio de la lectura crítica de documentos y materiales bibliográficos, de bibliotecas, hemerotecas, centros de documentación e información" (Baena, 2014), se han planteado diferentes análisis e hipótesis. Para la validación de la hipótesis planteada es necesario "realizar la distinción, conocimiento y clasificación de sus elementos esenciales y las relaciones que mantienen entre sí" (Calduch, 2014), es decir, identificar la/s normativa/s aplicable/s y sus implicaciones en proyectos, los actores necesarios para desarrollos Agile, tanto en la empresa privada como en la DGAM, así como sus responsabilidades y tareas, para poder articular una propuesta que satisfaga al máximo la garantía de calidad en este tipo de proyectos tan poco convencionales.

Como punto de partida de los análisis de la implantación y uso de las metodologías Agile se ha utilizado el análisis DAFO (Debilidades, Amenazas, Fortalezas y Oportunidades) que evalúa la situación actual de la implantación de una metodología Agile, además de identificar factores internos y externos que pueden influir en el éxito de esta.

Para la selección de la metodología Agile más adecuada al contexto del AOC en proyectos ágiles, se ha utilizado la herramienta AHP (*Analytic Hierarchy Process*). Este método fue ideado en la década de 1970 (Saaty, 1980) y permite, en base a unos criterios establecidos, seleccionar una de entre varias alternativas. Se trata de un método flexible y fácil de entender puesto que se estructura en tres niveles (objetivos/criterios/alternativas). La decisión de emplear este método de decisión se fundamenta en su idoneidad para facilitar la toma de decisiones al descomponer problemas complejos en partes más sencillas y evaluar múltiples opciones y criterios de manera estructurada. Proporciona además un marco claro para comparar alternativas, asegurando que las decisiones sean coherentes y bien fundamentadas. Además, permite alcanzar resultados objetivos y fiables, basados en datos cuantificables.

La Figura 1 muestra el esquema de metodología explicado anteriormente, relacionando las tareas definidas con los objetivos establecidos, sus herramientas y las fuentes de información utilizadas.

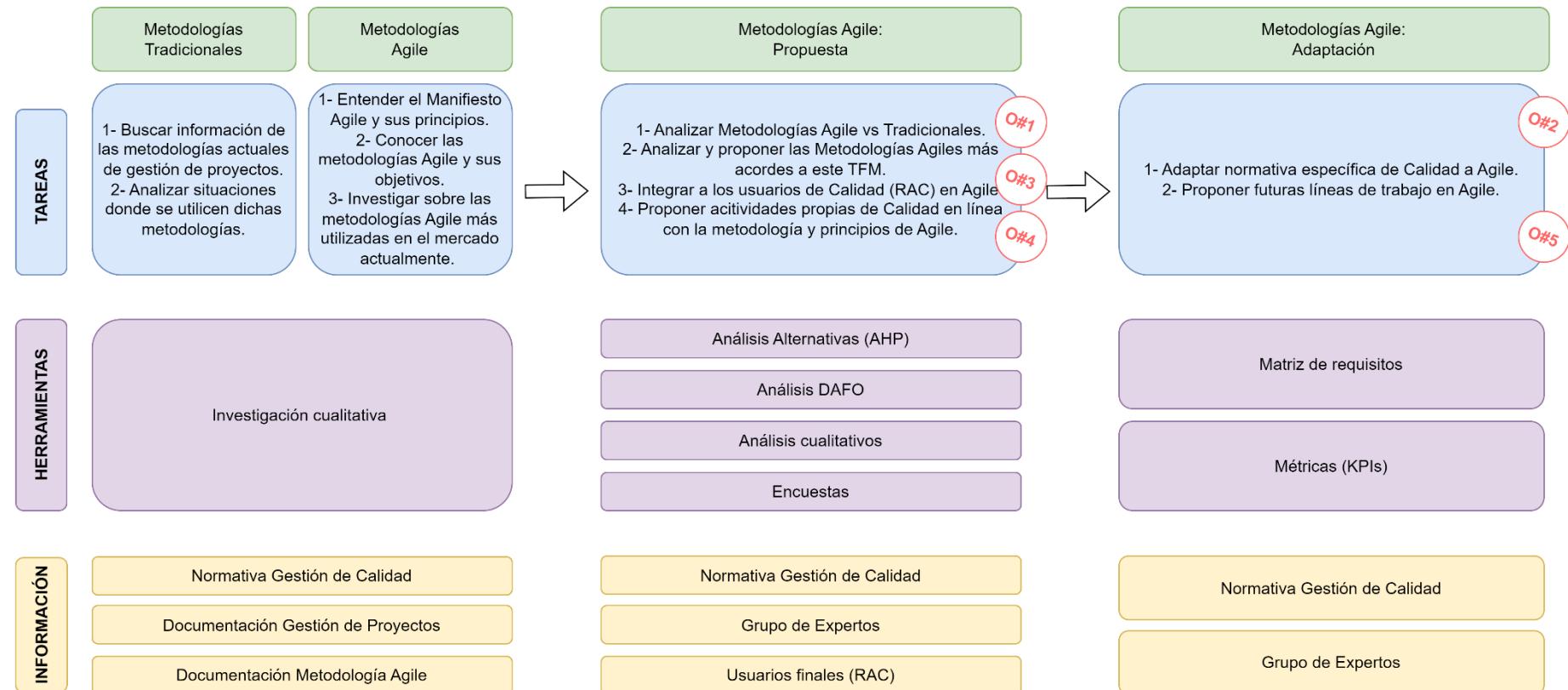


Figura 1 Esquema de la metodología de este TFM: tareas, herramientas y fuentes de información.

Fuente: Elaboración propia.



## 3 ANTECEDENTES Y MARCO TEÓRICO

Este capítulo tercero proporcionará un contexto integral de la situación tradicional de gestión de proyectos, incluida la gestión de la calidad (normas PECALE/AQAP), que permitirá comprender mejor las bases sobre las cuales se quieren desarrollar y aplicar las metodologías ágiles.

### 3.1 Marco Normativo

Desde que España se incorpora en la OTAN el 30 de mayo de 1982, se compromete, entre otras cosas, al cumplimiento de los requisitos OTAN en materia de calidad del STANAG (*Standardization Agreement*) 4107, es decir, aceptación mutua del aseguramiento oficial de la calidad y utilización de las publicaciones aliadas de aseguramiento de la calidad (AQAP/PECAL). La relación entre las normas AQAP (*Allied Quality Assurance Publications*) y la normativa PECALE (Publicación Española de Calidad) se recoge en la Tabla 1.

Tabla 1 Relación entre las normas PECALE/AQAP en vigor a fecha de realización de este trabajo.  
Fuente: Elaboración propia

Norma PECALE	Norma AQAP
PECALE 2110  "Requisitos OTAN de aseguramiento de la calidad para el diseño, el desarrollo y la producción". Edición D, Versión 1, June 2016.	AQAP-2110  "NATO Quality assurance requirements for design, development and production". Edition D, Version 1, June 2016.
PECALE 2105  "Requisitos OTAN para planes de calidad entregables" Edición C, Versión 1.	AQAP-2105  "NATO requirements for deliverable quality plans". Edition C, Version 1
PECALE 2131  "Requisitos OTAN de aseguramiento de la calidad para inspección y pruebas finales". Edición 3 Revisión 0.	AQAP-2131  "NATO Quality assurance requirements for final inspection". Edition C, Version 1, December 2017.
PECALE 2210  "Requisitos OTAN de aseguramiento de la calidad del software, suplementarios a la PECALE 2110". (Edición B, Versión 1, agosto 2022).	AQAP-2210  "NATO supplementary software quality assurance requirements to AQAP 2110". Edition A Version 2, September 2015.
PECALE 2310  "Requisitos OTAN para los sistemas de gestión de calidad de suministradores de aviación, espaciales y de defensa". Edición B, Versión 1, diciembre 2017.	AQAP-2310  "NATO quality management system requirements for Aviation, Space and Defence Suppliers". Edition B, Version 1, December 2017.

El Aseguramiento Oficial de la Calidad (AOC) dentro del MINISDEF es gestionado por el DIGAM (Director General de Armamento y Material), que actúa como la Autoridad Nacional de Aseguramiento de la Calidad (ANAC).



El DIGAM cuenta con una estructura organizativa específica (marcada con rectángulos verdes en la Figura 2), el Área de Inspecciones Industriales (AII) de la SDGINREID, que incluye a la INTECDEF Nº1 y otras Áreas de Inspección Industrial (AII's) en España. Además, se nombra un representante encargado de ejecutar las tareas de AOC que es el RAC. El RAC actúa como delegado del DIGAM, tanto en contratos nacionales como internacionales, y se ocupa de garantizar que la calidad de los productos o servicios contratados a una empresa cumplan con los estándares y requisitos establecidos.

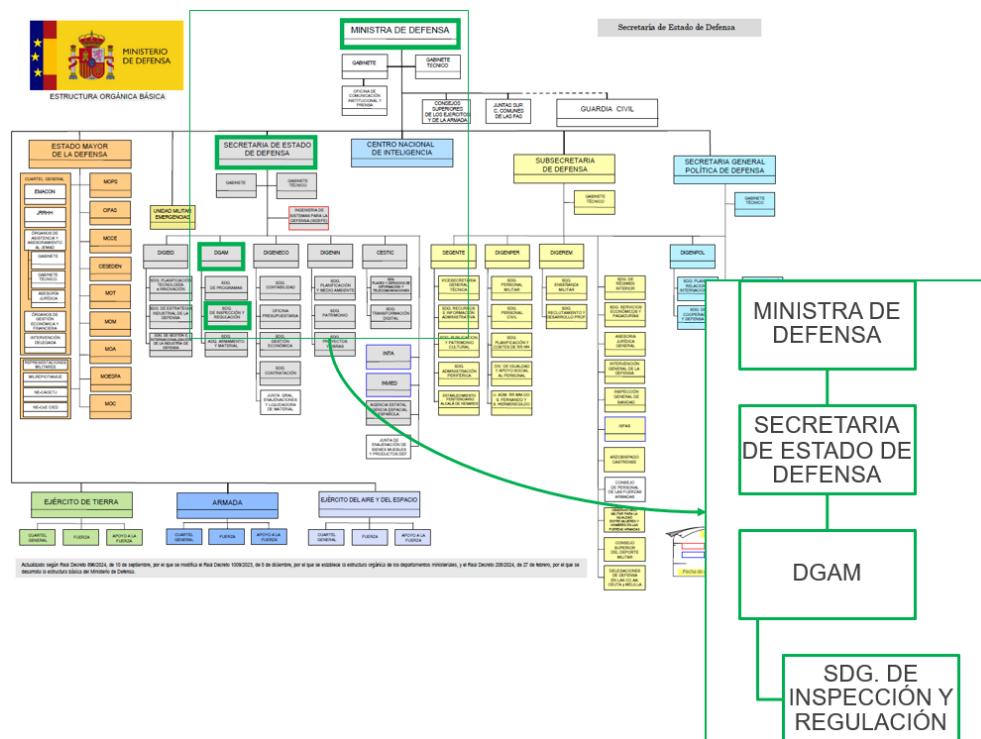


Figura 2 Organigrama del actual MINISDEF español.

Fuente: Web MINISDEF (Ministerio de Defensa, 2018), cuadro resumen: Elaboración propia.

### 3.2 Metodologías de Gestión de Proyectos: Aplicación al caso de estudio

Para entender la realidad actual de la gestión de proyectos, se ha tomado como referencia la sección 3 de la Guía Práctica de Ágil (Project Management Institute, 2017), que define 4 tipos de ciclos de vida, definidos a continuación:

- Predictivo: Enfoque tradicional y proceso secuencial.
- Iterativo: Enfoque de retroalimentación del trabajo.
- Incremental: Enfoque con entregables para uso inmediato.
- Ágil: Enfoque iterativo e incremental.



Considerando estos ciclos de vida de los proyectos, y seleccionando como muestra los contratos directos del MINISDEF con la empresa SENER (total de 5 expedientes desde 2021 hasta la actualidad), se concluye que todos ellos se gestionan mediante ciclos de vida predictivos, y ninguno mediante metodologías Agile, de acuerdo con la Tabla 2. Estos datos hacen referencia solo a esta empresa porque la autora de este TFM trabaja la misma y tiene acceso a esta información, no disponiendo de la misma sobre otros contratos del MINISDEF con otras empresas del sector.

*Tabla 2 Ciclos de vida Proyectos defensa (MINISDEF-SENER Aeroespacial)*  
Fuente: Elaboración propia

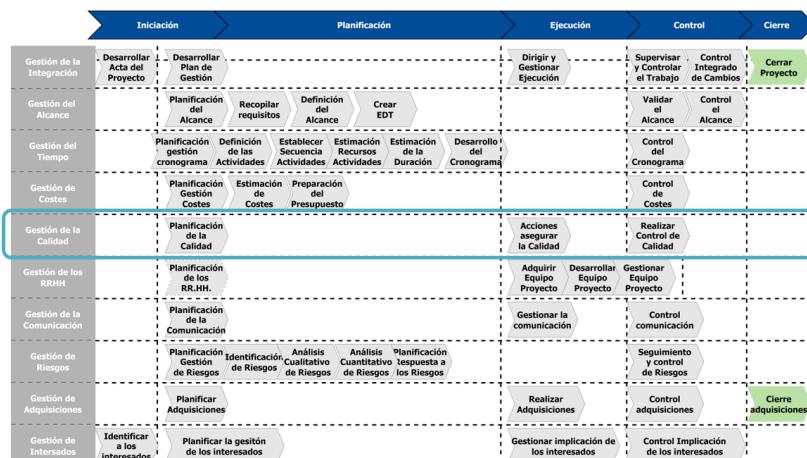
Ciclo de vida	% Proyectos
Predictivo	100%
Iterativo	0%
Incremental	0%
Ágil	0%

En los siguientes apartados se explican las metodologías tradicionales (ciclo de vida predictivo) y ágiles.

### 3.3 Metodología Tradicional (Ciclo de Vida Predictivo)

Esta metodología está recogida en el PMBOK guide (Project Management Institute Inc, 2017), el cual, contiene procesos, prácticas recomendadas, terminologías y directrices para una gestión de proyectos exitosa, lo que en el caso de la 6ed supone centrarse en proyectos con enfoque tradicional, donde la fase de planificación adquiere tanta importancia. Cuanto más planificado, más predictivo y controlado será el proyecto.

Contiene las 9+1 áreas de gestión de proyectos incluyendo entre ellas la Gestión de la Calidad, mostradas en la Figura 3.



*Figura 3 Introducción PMBOK guide (Project Management Institute Inc, 2017)*  
Fuente: Master Adquisiciones de Sistemas de Defensa (Módulo 5)



Las normas PECAL/AQAP, mencionadas en la Tabla 1, están también basadas en las metodologías de trabajo tradicionales con ciclos de vida predictivos, donde se establece que, para cumplir con los requisitos y objetivos de aseguramiento de la calidad, tanto en el ámbito del desarrollo de software (SW) como de sistemas de armas/comunicación/mando y control, se debe planificar, controlar, mejorar y mantener el proceso.

### 3.4 Metodología Agile (Ágil)

La Metodología Agile (ágil) es un enfoque de gestión y desarrollo de proyectos que promueve la flexibilidad, la colaboración y la mejora continua. Está centrada en la entrega de valor de manera incremental y en la capacidad de adaptarse rápidamente a los cambios, en lugar de seguir un plan rígido desde el inicio hasta el final del proyecto.

Los principios básicos de Agile incluyen:

- Iteraciones cortas: Los proyectos se dividen en ciclos pequeños, llamados *sprints*, que generalmente duran entre 1 y 4 semanas. Al final de cada *sprint*, se entrega una parte funcional del producto.
- Colaboración constante: Los equipos trabajan de cerca con los clientes o usuarios finales para asegurarse de que el producto se ajuste a sus necesidades, haciendo ajustes según sea necesario.
- Respuesta al cambio: Agile se enfoca en ser adaptable y flexible. A medida que los requisitos del proyecto cambian, el equipo puede ajustar el enfoque y el producto sin problemas.
- Enfoque al cliente: La satisfacción del cliente es prioritaria. Esto se logra mediante entregas frecuentes de valor y la posibilidad de recibir retroalimentación continua.
- Trabajo en equipo y comunicación: Agile promueve la colaboración efectiva entre todos los miembros del equipo, tanto dentro como fuera del equipo de desarrollo.
- Simplicidad y eficiencia: Se busca siempre la solución más simple para cumplir con los requisitos, evitando la sobrecarga innecesaria de funcionalidades.

El Manifiesto Ágil (Kent Beck et al., 2001) derivado de la crisis del software y creado en 2001 por un grupo de 17 expertos desarrolladores de ingeniería de software, establece los 4 valores (Figura 4) que luego despliegan en 12 principios (Kent Beck et al., 2001) fundamentales de la metodología, mostrados en la Figura 5.



Figura 4 Valores del Manifiesto Ágil  
Fuente: Elaboración propia



Figura 5 Principios del Manifiesto Ágil  
Fuente: Elaboración propia

De todo ello se puede extraer, tomando como referencia el 17º Informe sobre el Estado de Agile (Figura 6), que las metodologías ágiles se han consolidado como un enfoque ampliamente implementado en un gran número de organizaciones (71% de acuerdo con la Figura 6) debido a su capacidad para mejorar la eficiencia y adaptabilidad en la gestión de proyectos. Su flexibilidad y enfoque en la colaboración continua permiten una rápida respuesta a los cambios y necesidades del entorno.

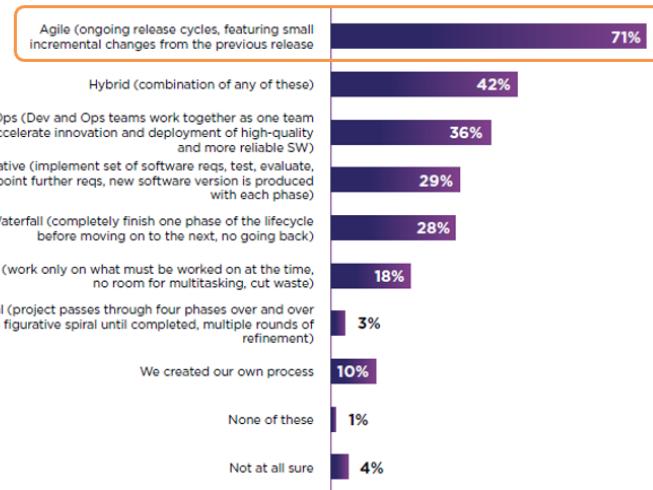


Figura 6 Análisis de los marcos de trabajo

Fuente: The 17th State of Agile Report (Digital.ai, 2024)

Integrar estas metodologías en contratos de Defensa de proyectos caracterizados por entornos complejos y cambiantes puede ser altamente beneficioso, ya que facilita la entrega de resultados de alta calidad en plazos más cortos, optimizando recursos y mejorando la capacidad de respuesta ante imprevistos y desafíos emergentes. Sirva como ejemplo el caso de estudio “Aplicación de metodologías ágiles para proyectos web” del EA (Ejército del Aire y del Espacio, 2024).

A continuación, se presentan las metodologías a ser analizadas como mejor opción para la implementación de las mismas en el marco de este TFM.

La elección de dichas metodologías se basa en los datos obtenidos del 17º Informe sobre el Estado de Agile (Figura 7), añadiendo Kanban (por ser una herramienta muy visual) y FDD (centrado en el desarrollo SW) al estudio como otras posibles opciones de metodología Agile a implementar.

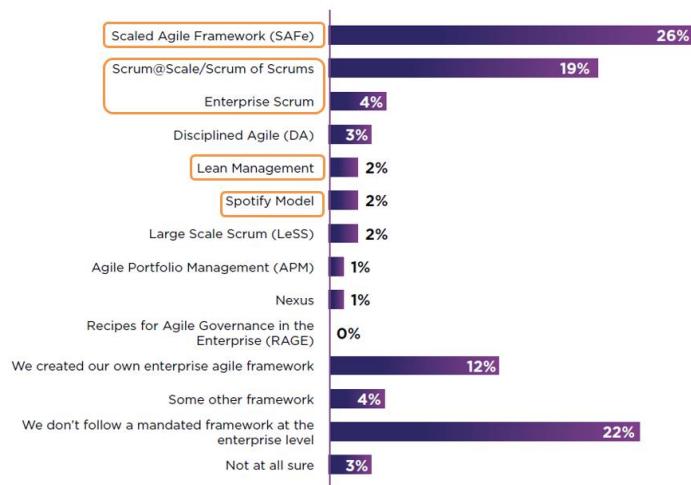


Figura 7 Metodologías Agile más utilizadas en la industria

Fuente: The 17th State of Agile Report (Digital.ai, 2024)



### 3.4.1 Scrum

Se basa en ciclos de trabajo cortos llamados *sprints*, que permiten entregas parciales y regulares del producto final. Esta metodología promueve la colaboración, la autoorganización y la mejora continua.

### 3.4.2 Kanban

Es un enfoque ágil de gestión de proyectos que se centra en la mejora continua y la eficiencia del flujo de trabajo. Utiliza tableros visuales para representar tareas y su progreso, permitiendo a los equipos gestionar y optimizar la entrega de trabajo. Originada en la industria manufacturera dentro del *Toyota Production System* (TPS), Kanban se utiliza para desencadenar una acción, y se ha adaptado para su uso en el desarrollo de software y en otros sectores.

### 3.4.3 SAFe

Son un conjunto de patrones organizacionales y flujos de trabajo diseñados para implementar prácticas ágiles a gran escala y en grandes y complejas organizaciones. SAFe sería el siguiente paso una vez esté interiorizada e implantada la metodología de trabajo Agile.

### 3.4.4 Lean Management

Aunque ofrece una base robusta para mejorar la eficiencia y reducir el desperdicio, puede ser más complicado de aplicar eficazmente en equipos que son nuevos en métodos ágiles, ya que requiere un cambio cultural considerable y un compromiso con la mejora continua, lo cual puede ser difícil de manejar al principio.

### 3.4.5 Spotify Model

Está basado en la simplicidad y aboga por la autonomía de los equipos (o escuadrones) para que cada uno de ellos elija su marco de metodología por lo que es uno de los más flexibles, pero a la vez más complejos de implementar en una organización, sobre todo que gestione proyectos complejos.

### 3.4.6 Feature-Driven Development (FDD)

Este enfoque, centrado en características y con procesos más estructurados, puede ser menos adaptable que otros métodos ágiles, lo que podría representar un desafío al cambiar desde un método tradicional.



## 4 DESARROLLO: ANÁLISIS Y RESULTADOS

En este capítulo analizará las diferentes metodologías, utilizando herramientas de análisis como DAFO y AHP, hasta llegar a una propuesta de implantación de Agile garantizando el aseguramiento de la calidad.

### 4.1 Metodología Tradicional vs Metodología Agile

En distintos foros, como el del grupo “OTAN AC/327 WG2 on Quality”, se está contemplando como adecuar la normativa existente a la realidad actual de las nuevas metodologías Agile ya que la agilidad es un factor clave en el desarrollo de productos hoy en día.

Uno de los principales factores que diferencian ambas metodologías, tradicional y Agile, es el esquema de trabajo. En la primera, se establece una planificación detallada y secuencial mientras que en la segunda se tienen iteraciones cortas, continuas y flexibles que permiten adaptarse rápidamente a los cambios. La Figura 8 muestra ambas casuísticas.

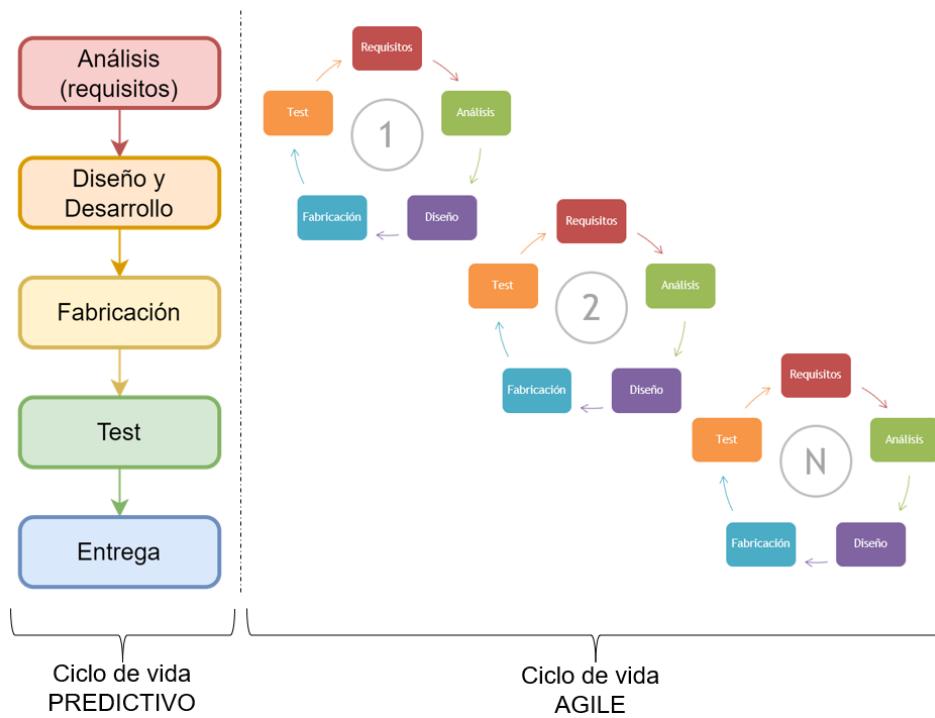


Figura 8 Esquema de trabajo: Ciclo de vida PREDICTIVO vs AGILE

Fuente: Elaboración propia

Las diferencias entre la metodología tradicional (ciclo de vida predictivo) y las metodologías de desarrollo de SW ágiles, son también evidentes en la forma en que se gestiona la calidad, pues el modelo iterativo de Agile requiere una revisión más frecuente de los criterios y problemas de calidad, así como la comprobación de los resultados.

La Tabla 3 muestra la comparativa de los ciclos de vida anteriormente mencionados.



*Tabla 3 Metodología ÁGIL vs TRADICIONAL*  
*Fuente: Elaboración propia*

	<b>Ciclo de vida: PREDICTIVO</b>	<b>Ciclo de vida: AGILE</b>
<b>Requisitos</b>	Rígidos y lineales	Iterativo, adaptable y flexible
<b>Avance</b>	Hitos de proyecto	<i>Sprints</i>
<b>Retroalimentación y detección de problemas</b>	Lenta	Rápida, continua y cooperativa
<b>Resultados y optimización</b>	De acuerdo con los requisitos iniciales	Alineados con las necesidades de cliente (mejora continua)
<b>Entrega</b>	Hitos y final del proyecto	Rápida y frecuente ( $\approx 30$ días)
<b>Equipos</b>	Por proyecto (aislados)	Multifuncionales
<b>Documentación</b>	Exhaustiva y extensa	Menor importancia y en pequeños paquetes

La parte más importante de cualquier proyecto reside en la gestión de requisitos, un área en la que la metodología Agile difiere significativamente de las tradicionales.

La identificación de requisitos en un marco Agile gira en torno a diálogos colaborativos con los clientes en lugar de documentación extensa. A través de estas iteraciones, surge lo que los usuarios finales desean del producto, información que luego será registrada a modo de “requisitos”.

#### 4.1.1 Análisis DAFO

Tras la comparativa realizada en la Tabla 3, el punto de partida del análisis de la implantación y uso de la metodología ágil reside en los análisis DAFO realizados (ver Tabla 4 y Tabla 5).

Ambos análisis DAFO consideran las Debilidades y Fortalezas (análisis interno) de utilizar la metodología Agile desde el punto de vista de implantación de la metodología, así como sus Amenazas y Oportunidades (análisis externo).

El DAFO MINISDEF (Tabla 4) se ha considerado desde el punto de vista de la implantación y uso de Agile en el Ministerio, mientras que el DAFO Empresa Privada (Tabla 5) se ha realizado en el contexto de ciertos contratos de Defensa realizados por organizaciones diferenciadas de la Administración Pública.



*Tabla 4 Metodología Agile: análisis DAFO (MINISDEF)*  
Fuente: *Elaboración propia*

<u>DEBILIDADES</u>	<u>AMENAZAS</u>
<ul style="list-style-type: none"> <li>• Esfuerzo (carga de trabajo elevada).</li> <li>• Recursos limitados (falta o escasez de personal).</li> <li>• Falta de conocimientos.</li> <li>• Organización tradicional incompatible con la flexibilidad de Agile (autoorganización de los equipos).</li> <li>• Rígidez estructural y burocracia.</li> </ul>	<ul style="list-style-type: none"> <li>• Planificación.</li> <li>• Falta de participación y <i>feedback</i> por parte del cliente (FAS, OCCAR, etc).</li> <li>• Aumento de los riesgos de proyecto.</li> <li>• Contratos y regulaciones estrictas con alcance fijo.</li> <li>• Adversidad al cambio.</li> </ul>
<u>FORTALEZAS</u>	<u>OPORTUNIDADES</u>
<ul style="list-style-type: none"> <li>• Orientación a Calidad.</li> <li>• Capacidad de adaptación a la evolución de las necesidades.</li> <li>• Gestión rápida y eficaz de requisitos.</li> <li>• Resolución rápida de problemas (retroalimentación continua).</li> <li>• Resultado final orientado a satisfacer las necesidades reales del usuario final.</li> </ul>	<ul style="list-style-type: none"> <li>• Mejora de la respuesta gubernamental a necesidades internas y ciudadanas.</li> <li>• Gestión eficaz de requisitos.</li> <li>• Flexibilidad (adaptación rápida y eficaz a cambios legislativos y tecnológicos).</li> <li>• Mejora continua y aprendizaje constante.</li> <li>• Colaboración entre departamentos.</li> </ul>

*Tabla 5 Metodología Agile: análisis DAFO (Empresa Privada)*  
Fuente: *Elaboración propia*

<u>DEBILIDADES</u>	<u>AMENAZAS</u>
<ul style="list-style-type: none"> <li>• Esfuerzo (carga de trabajo elevada).</li> <li>• Recursos limitados (falta o escasez de personal).</li> <li>• Falta de conocimientos.</li> <li>• Organización tradicional incompatible con la flexibilidad de Agile (autoorganización de los equipos).</li> <li>• Rígidez estructural y burocracia interna.</li> </ul>	<ul style="list-style-type: none"> <li>• Planificación.</li> <li>• Falta de participación y <i>feedback</i> por parte del cliente (Empresa Cliente, Jefes de Programa, DGAM/FAS).</li> <li>• Aumento de los riesgos de proyecto.</li> <li>• Contratos y regulaciones estrictas con alcance fijo.</li> <li>• Adversidad al cambio.</li> </ul>
<u>FORTALEZAS</u>	<u>OPORTUNIDADES</u>
<ul style="list-style-type: none"> <li>• Orientación a Calidad.</li> <li>• Capacidad de adaptación a la evolución de las necesidades del cliente (Empresa Cliente, Jefes de Programa, DGAM/FAS).</li> <li>• Gestión rápida y eficaz de requisitos.</li> <li>• Resolución rápida de problemas (retroalimentación continua).</li> <li>• Resultado final orientado a satisfacer las necesidades reales del usuario final (cliente).</li> </ul>	<ul style="list-style-type: none"> <li>• Mejora de la respuesta a necesidades tanto internas como externas a la empresa.</li> <li>• Gestión eficaz de requisitos.</li> <li>• Flexibilidad (adaptación rápida y eficaz a cambios legislativos y tecnológicos).</li> <li>• Mejora continua y aprendizaje constante.</li> <li>• Innovación y competitividad en el sector de la defensa.</li> </ul>



La conclusión de ambos análisis DAFO es que, aunque existen debilidades significativas como la carga de trabajo elevada, recursos limitados y la incompatibilidad de la organización tradicional con la flexibilidad de Agile, también se identifican fortalezas importantes como la orientación a la calidad y la capacidad de adaptación. Las amenazas, como la falta de participación del cliente y la adversidad al cambio, contrastan con las oportunidades de mejora continua y la gestión eficaz de requisitos.

Integrar metodologías Agile puede mitigar las debilidades y amenazas, potenciando las fortalezas y aprovechando las oportunidades para lograr una gestión de proyectos más eficiente y adaptable.

#### 4.1.2 Análisis de Agile desde el punto de vista de Calidad

Además de la información que se proporciona en los DAFO (Tabla 4 y Tabla 5), los desafíos más importantes y propios de la gestión de Calidad encontrados, a través de experiencias personales de expertos en la materia, tanto de la empresa SENER, como ejemplo de empresa privada, como del MINISDEF (RAC) y de conversaciones con directores de proyecto (SENER) y jefes de programa (MINISDEF), durante el desarrollo y análisis de la propuesta de uso de metodología Agile son:

- Adversidad al cambio: Los equipos que no están familiarizados con Agile pueden resistirse al cambio debido a la flexibilidad que estos métodos demandan. La adopción temprana de una mentalidad ágil es beneficiosa, enfatizando el objetivo final de entregar un producto valioso *versus* la rigidez de procesos específicos. Este cambio de mentalidad puede facilitar la transición y priorizar el objetivo final de creación de valor.
- Normativa asociada a metodología tradicional: La implementación de una metodología ágil siguiendo la normativa tradicional sin ninguna modificación, impacta en el cumplimiento de muchos de los requisitos incluidos en la misma.
- Documentación reducida: Agile prioriza productos funcionales sobre la documentación extensa, lo que podría llevar a una escasez de documentación a la que no se está acostumbrado en metodologías tradicionales. Esta reducción de la documentación podría aumentar la sensación de "pérdida de control" así como elevar el riesgo de aumento del alcance. Sin embargo, es vital no pasar por alto la documentación por completo. En su lugar, Agile enfatiza la creación de documentación que aporte valor tangible, enfocándose en detalles esenciales cruciales para entender y mantener el producto.
- Planificación: En Agile, los ciclos de ejecución de pruebas son breves, dejando menos tiempo para crear planes de prueba detallados y realizar pruebas extensivas. Los horarios pueden cambiar repentinamente, requiriendo que el equipo esté listo para gestionar los riesgos de manera efectiva y adaptarse rápidamente.
- Recursos: La asignación de recursos podría estar en riesgo al adaptarse a nuevas características y prioridades cambiantes.



## 4.2 Metodologías Agile: Análisis de Alternativas

El análisis de las alternativas mencionadas en la sección 3.4 se realiza mediante la técnica de decisión multicriterio AHP, explicada en el *Anexo I*. A continuación, se expone un resumen y conclusiones de dicho análisis.

### 4.2.1 Primera Etapa. Representación del Problema

La Figura 9 representa el diagrama de árbol del problema incluyendo la siguiente información:

- Objetivo: Respuesta a la cuestión de ¿Cuál/es es/son la/s mejor/es metodología/s a utilizar?
- Criterios:
  - Calidad: orientación a la excelencia de calidad en procesos y entregas finales.
  - Flexibilidad en los procesos asociados a la metodología.
  - Facilidad de implementación en la organización.
  - Aprendizaje: curva de aprendizaje necesaria para la implantación y gestión.
  - Recursos: cantidad de recursos necesarios a asignar para gestionar la metodología.
  - Escalabilidad: capacidad de satisfacer las necesidades empresariales de alto nivel.
- Alternativas: mencionadas en la sección 3.4.

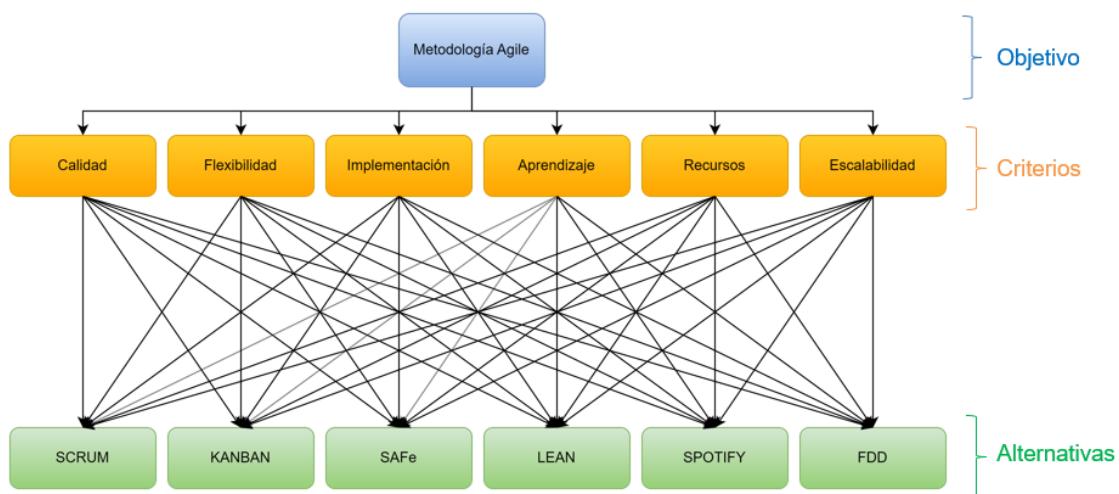


Figura 9 AHP: Primera etapa\_Representación del Problema, Criterios y Alternativas  
Fuente: Elaboración propia



#### 4.2.2 Segunda Etapa: Evaluación de los criterios

Se genera una matriz de criterios para su comparación dos a dos, utilizando la escala de Saaty (1-9) con el propósito de estimar la importancia relativa entre cada uno de ellos, incluyendo sus pesos (W). Los resultados para los criterios analizados se muestran en la Tabla 6.

*Tabla 6 AHP: Segunda etapa\_Comparativa 2 a 2 de los criterios  
Fuente: Elaboración propia*

CRITERIOS	Calidad	Flexibilidad	Implementación	Aprendizaje	Recursos	Escalabilidad	PESOS (W)
<b>Calidad</b>	1	3	3	3	5	5	0,39
<b>Flexibilidad</b>	1/3	1	1/3	1	1/3	3	0,10
<b>Implementación</b>	1/3	3	1	1	1	3	0,16
<b>Aprendizaje</b>	1/3	1	1	1	3	3	0,16
<b>Recursos</b>	1/5	3	1	1/3	1	3	0,13
<b>Escalabilidad</b>	1/5	1/3	1/3	1/3	1/3	1	0,05
<b>Razón de inconsistencia (R.I.): 0,0845 &lt; 0,1 (10%)</b>							

La conclusión de la evaluación de los criterios es que los más importantes (los que tienen más peso) son: Calidad (0,39), seguido de Implementación (0,16) y Aprendizaje (0,16). La razón de inconsistencia (R.I.) es de 0,0845, valor inferior a 0,1 (10%), por lo que el análisis realizado es consistente.

#### 4.2.3 Tercera Etapa: Evaluación de las Alternativas

En esta tercera etapa se evalúan las diferentes alternativas comparando las mismas entre sí para cada uno de los criterios considerados. Así, en la Tabla 7, Tabla 8, Tabla 9, Tabla 10, Tabla 11 y Tabla 12 se muestran los resultados de dicha evaluación en base a los criterios definidos en la sección 4.2.1.



*Tabla 7 AHP: Tercera etapa\_Criterio: Calidad  
Fuente: Elaboración propia*

CALIDAD	SCRUM	KANBAN	SAFe	LEAN	SPOTIFY	FDD	PESOS (W)
<b>SCRUM</b>	1	3	3	5	7	3	0,39
<b>KANBAN</b>	1/3	1	1	3	5	1	0,16
<b>SAFe</b>	1/3	1	1	3	5	3	0,20
<b>LEAN</b>	1/5	1/3	1/3	1	3	1/3	0,07
<b>SPOTIFY</b>	1/7	1/5	1/5	1/3	1	1/3	0,04
<b>FDD</b>	1/3	1	1/3	3	3	1	0,05
<b>Razón de inconsistencia (R.I.): 0,0414 &lt; 0,1 (10%)</b>							

La conclusión del criterio de Calidad es que las metodologías que van más en línea con este son, por orden de peso: SCRUM, SAFe y KANBAN.

*Tabla 8 AHP: Tercera etapa\_Criterio: Flexibilidad  
Fuente: Elaboración propia*

FLEXIBILIDAD	SCRUM	KANBAN	SAFe	LEAN	SPOTIFY	FDD	PESOS (W)
<b>SCRUM</b>	1	1	5	5	1	7	0,28
<b>KANBAN</b>	1	1	5	5	1	7	0,28
<b>SAFe</b>	1/5	1/5	1	1	1/5	3	0,07
<b>LEAN</b>	1/5	1/5	1	1	1/5	1	0,05
<b>SPOTIFY</b>	1	1	5	5	1	5	0,27
<b>FDD</b>	1/7	1/7	1/3	1	1/5	1	0,04
<b>Razón de inconsistencia (R.I.): 0,0190 &lt; 0,1 (10%)</b>							

La conclusión del criterio de Flexibilidad es que las metodologías que van más en línea con este son, por orden de peso: SCRUM y KANBAN seguido de SPOTIFY.



*Tabla 9 AHP: Tercera etapa\_Criterio: Implementación  
Fuente: Elaboración propia*

IMPLEMENTACIÓN	SCRUM	KANBAN	SAFe	LEAN	SPOTIFY	FDD	PESOS (W)
<b>SCRUM</b>	1	1/3	3	5	5	7	0,26
<b>KANBAN</b>	3	1	5	7	7	7	0,46
<b>SAFe</b>	1/3	1/5	1	3	3	3	0,13
<b>LEAN</b>	1/5	1/7	1/3	1	1	1	0,05
<b>SPOTIFY</b>	1/5	1/7	1/3	1	1	1	0,05
<b>FDD</b>	1/7	1/7	1/3	1	1	1	0,05
<b>Razón de inconsistencia (R.I.): 0,0248 &lt; 0,1 (10%)</b>							

La conclusión del criterio de Implementación es que las metodologías que van más en línea con este son, por orden de peso: KANBAN, SCRUM y SAFe.

*Tabla 10 AHP: Tercera etapa\_Criterio: Aprendizaje  
Fuente: Elaboración propia*

APRENDIZAJE	SCRUM	KANBAN	SAFe	LEAN	SPOTIFY	FDD	PESOS (W)
<b>SCRUM</b>	1	1/3	3	1	5	1	0,18
<b>KANBAN</b>	3	1	5	3	5	3	0,39
<b>SAFe</b>	1/3	1/5	1	1/33	1	1/3	0,06
<b>LEAN</b>	1	1/3	3	1	3	1	0,16
<b>SPOTIFY</b>	1/5	1/5	1	1/3	1	1	0,07
<b>FDD</b>	1	1/3	3	1	1	1	0,14
<b>Razón de inconsistencia (R.I.): 0,0396 &lt; 0,1 (10%)</b>							

La conclusión del criterio de Aprendizaje es que las metodologías que van más en línea con este son, por orden de peso: KANBAN, SCRUM y LEAN.



*Tabla 11 AHP: Tercera etapa\_Criterio: Recursos*  
*Fuente: Elaboración propia*

RECURSOS	SCRUM	KANBAN	SAFe	LEAN	SPOTIFY	FDD	PESOS (W)
<b>SCRUM</b>	1	1/3	5	1	5	1	0,19
<b>KANBAN</b>	3	1	7	3	5	3	0,40
<b>SAFe</b>	1/5	1/7	1	1/5	1	1	0,06
<b>LEAN</b>	1	1/3	5	1	3	1	0,17
<b>SPOTIFY</b>	1/5	1/5	1	1/3	1	1/3	0,05
<b>FDD</b>	1	1/3	1	1	3	1	0,13
<b>Razón de inconsistencia (R.I.): 0,0526 &lt; 0,1 (10%)</b>							

La conclusión del criterio de Recursos es que las metodologías que van más en línea con este son, por orden de peso: KANBAN, SCRUM y LEAN.

*Tabla 12 AHP: Tercera etapa\_Criterio: Escalabilidad*  
*Fuente: Elaboración propia*

ESCALABILIDAD	SCRUM	KANBAN	SAFe	LEAN	SPOTIFY	FDD	PESOS (W)
<b>SCRUM</b>	1	1/3	1/5	3	1/3	1	0,08
<b>KANBAN</b>	3	1	1/3	5	1	3	0,19
<b>SAFe</b>	5	3	1	7	3	5	0,42
<b>LEAN</b>	1/3	1/5	1/7	1	1/5	1/3	0,04
<b>SPOTIFY</b>	3	1	1/3	5	1	3	0,19
<b>FDD</b>	1	1/3	1/5	3	1/3	1	0,08
<b>Razón de inconsistencia (R.I.): 0,0526 &lt; 0,1 (10%)</b>							

La conclusión del criterio de Escalabilidad es que las metodologías que van más en línea con este son, por orden de peso: SAFe seguido de KANBAN y SPOTIFY.

#### 4.2.4 Cuarta Etapa: Jerarquización de las Alternativas

Finalmente, para conocer qué alternativa es la más importante de acuerdo con los criterios establecidos se trasladan a la matriz de decisión todos los pesos calculados anteriormente.



Tabla 13 AHP: Cuarta etapa\_Jerarquización de Alternativas  
Fuente: Elaboración propia

CRITERIOS	PESOS (W)	SCRUM	KANBAN	SAFe	LEAN	SPOTIFY	FDD
<b>Calidad</b>	0,39	0,39	0,16	0,20	0,07	0,04	0,13
<b>Flexibilidad</b>	0,10	0,28	0,28	0,07	0,07	0,27	0,04
<b>Implementación</b>	0,16	0,26	0,46	0,13	0,05	0,05	0,05
<b>Aprendizaje</b>	0,16	0,18	0,39	0,06	0,16	0,07	0,14
<b>Recursos</b>	0,13	0,19	0,40	0,06	0,17	0,05	0,13
<b>Escalabilidad</b>	0,05	0,08	0,19	0,42	0,04	0,19	0,08
	<b>0,28</b>	<b>0,29</b>	<b>0,14</b>	<b>0,09</b>	<b>0,08</b>	<b>0,11</b>	

Teniendo en cuenta los resultados de la Tabla 13 y el marco de referencia de este TFM, para la jerarquización de alternativas en las metodologías Agile disponibles, se enfocará en las metodologías **Scrum (0,28)** y **Kanban (0,29)**, con ciclos de vida híbridos como estrategia de transición y comenzando en proyectos pequeños o WP dentro de un contrato grande.

### 4.3 Propuesta Agile: Aseguramiento Oficial de la Calidad

Siguiendo la línea del apartado anterior, a continuación, se detalla la aplicación de ambas metodologías Agile seleccionadas. La metodología Scrum proporciona orientación sobre el uso de trabajo pendiente y un tablero Kanban aumenta más su efectividad ya que se puede visualizar el flujo de trabajo y tareas.

#### 4.3.1 Scrum

Esta metodología se basa en el empirismo y el pensamiento Lean. El empirismo sostiene que el conocimiento proviene de la experiencia y de la toma de decisiones basadas en lo observado. El pensamiento Lean busca reducir el desperdicio y centrarse en lo esencial.

Scrum es un método diseñado para abordar problemas complejos, tomando decisiones basadas en la información disponible y la experiencia adquirida. Esta metodología se sustenta en tres pilares fundamentales: transparencia, inspección y adaptación; y cuenta con dos enfoques principales, iterativo e incremental:

- Iterativo: en cada *sprint* se genera una nueva versión del producto, refinando y mejorando sus características conforme avanza. Cada nuevo producto produce una nueva retroalimentación de los interesados, así como nuevas percepciones del equipo.
- Incremental: se optimiza el trabajo con el fin de entregar en cada período de tiempo el producto con nuevas funcionalidades y/o características.



#### 4.3.1.1 Scrum: Equipos y Roles

Los equipos ágiles se componen de tres a nueve miembros y se ubican, idealmente, en el mismo lugar. Dichos miembros se autogestionan y están dedicados al 100% al proyecto en cuestión.

Particularmente en Scrum existen tres roles clave, que son las personas que están comprometidas con el proyecto y el proceso Scrum, y que deben ser definidos y entendidos por todos.

- **Product Owner:** Es el responsable de la toma de decisiones y quien realmente conoce el negocio del cliente y su visión del producto. Gestiona el *Product Backlog*, incluyendo las ideas del cliente (requisitos), ordenadas por prioridad. El *Product Owner* debe tener una visión clara del producto y las prioridades para maximizar su valor. Deben ser elaboradas por el propio alumno.
- **Scrum Master:** Comprueba que la metodología funciona, ayuda al equipo a seguir las prácticas ágiles y elimina obstáculos. El *Scrum Master* no es un jefe, sino un facilitador. Puede asemejarse al rol de director de proyecto en metodología tradicional.
- **Equipo de desarrollo (Agile Team):** Grupo de profesionales (3-9 personas) que tienen autorizar para organizar y tomar decisiones para conseguir su objetivo.

Adicionalmente, y aunque no son parte del proceso Scrum, existen otros tres actores necesarios para la retroalimentación del proceso y hacer posible la revisión y planificación de cada *sprint*.

- **Usuarios:** Destinatario final del producto.
- **Stakeholders:** Personas a las que el proyecto les producirá un beneficio. Son también participantes durante las revisiones del *sprint*.
- **Managers:** Toman las decisiones finales participando en la selección de objetivos y requisitos.

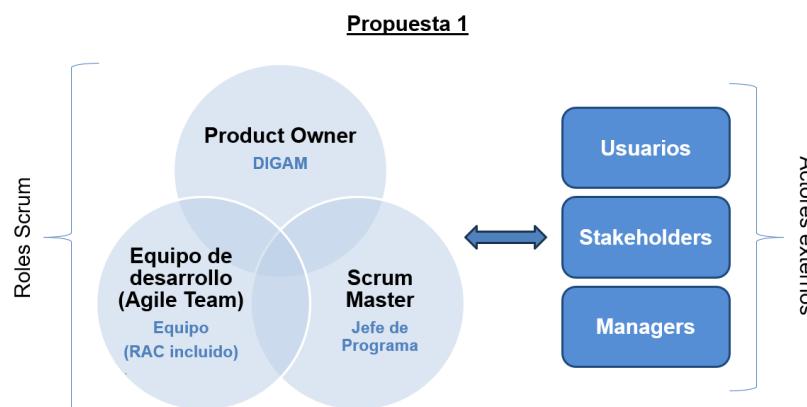
Se describen a continuación 2 propuestas de equipo Scrum de acuerdo al estudio y análisis previo mostrado en las secciones anteriores, tal y como se resumen en la Tabla 14, Figura 10 y Figura 11.

La propuesta 1 se hace como opción de implantación de Scrum en la DGAM (MINISDEF), en proyectos que aplique la norma PECAL/AQAP 2210. Mientras que la propuesta 2 está definida desde el punto de vista de implantación de Scrum en una empresa privada que tenga contrato con el MINISDEF, y que le aplique también la norma PECAL/AQAP 2210



*Tabla 14 Scrum: Propuestas equipos y roles*  
Fuente: Elaboración propia

	Propuesta 1	Propuesta 2
<b>Producto Owner</b>	DIGAM	Director de División
<b>Scrum Master</b>	Jefe de Programa	Director de Proyecto
<b>Equipo de desarrollo Scrum</b>	Grupo de profesionales (ingenieros de las FAS, funcionarios, personal de ISDEFE...etc) incluyendo al RAC.	<p>Equipo Agile.</p> <ul style="list-style-type: none"> <li><b>RAC</b><sup>(1)</sup>: una de las opciones es que el RAC sea parte del equipo de desarrollo y participe activamente en todo el proceso Scrum. La problemática de esta propuesta es que la carga de trabajo es inasumible debido a la gran cantidad de contratos que actualmente existen en diferentes empresas y que los RAC gestionan.</li> </ul>
<b>Usuarios</b>	Ej.: FAS	Cliente/DGAM/FAS
<b>Stakeholders</b>	Ej.: Diferentes ejércitos	<p>Cliente/Propia empresa</p> <ul style="list-style-type: none"> <li><b>RAC</b><sup>(2)</sup>: la otra opción es que el RAC sea un <i>stakeholder</i> más. Esta opción se considera la más viable ya que la participación sería solamente durante los <i>sprints</i>, en el <i>sprint review</i> y opcionalmente, en las retrospectivas, tal como se indica en la <i>Tabla 18</i>.</li> </ul>
<b>Managers</b>	Jefes de programa/DGAM	Cliente/Propia empresa



*Figura 10 Roles de Scrum: Propuesta 1*  
Fuente: Elaboración propia

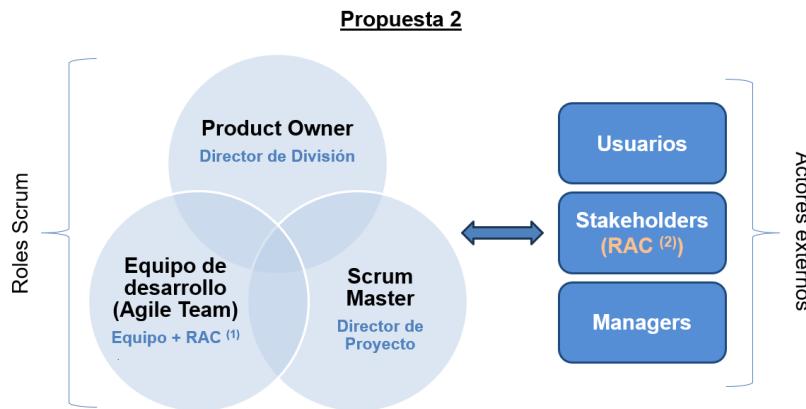


Figura 11 Roles de Scrum: Propuesta 2

Fuente: Elaboración propia

La conclusión de ambas propuestas sería la siguiente:

- La propuesta 1 sería la opción más compleja, costosa (recursos y tiempo de dedicación) y con mayor curva de aprendizaje de implementar en base a:
  - Organización del MINISDEF compleja y tradicional.
  - Normativa, regulaciones y procedimientos son muy extensos y estrictos y su modificación no sería inmediata y rápida.
  - La seguridad y confidencialidad podrían verse comprometidas en la gestión de información sensible.
  - Esfuerzo en formación específica y capacidades.
- La propuesta 2, en cambio, sería la más viable, ya que la empresa privada tiene más flexibilidad de cambios y su organización no es tan compleja, ni se caracteriza por una jerarquía tan definida. En las siguientes secciones se tomará la propuesta 2 como objeto del análisis de este TFM.

#### 4.3.1.2 Scrum: Elementos y Eventos

Esta subsección explica y analiza los elementos y eventos principales de Scrum, ambos representados en la Figura 12, de color naranja y verde, respectivamente.

Los elementos principales de Scrum son aquellos elementos físicos que resultan de aplicar la metodología. Estos son: el *Product backlog*, el *Sprint backlog* y el *Incremento*.

De la misma manera, los eventos principales son momentos clave dentro de la metodología que ayudan a estructurar el trabajo y el progreso de un proyecto. Son 5 en particular: *Sprint planning*, *Sprint*, *Daily Scrum*, *Sprint Review* y *Retrospectiva* y se desarrollan en la Tabla 18.

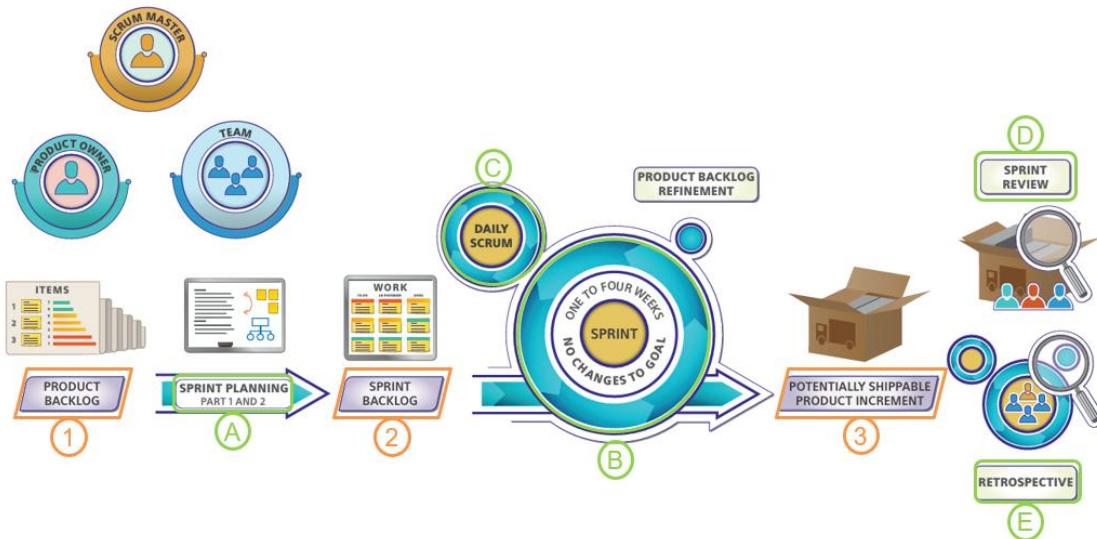


Figura 12 Elementos y Eventos de Scrum

Fuente: (Agile Carpentry, 2024). Recuperado de  
[https://agilecarpentry.com/scrum\\_guide/sp-less-scrum-guide/](https://agilecarpentry.com/scrum_guide/sp-less-scrum-guide/)

1. **Product backlog:** lista de necesidades (requisitos) del cliente priorizadas. La gestión la realiza el *Product Owner* y es su principal fuente de información del producto con los *stakeholders*. La Tabla 15 muestra un ejemplo de *Product backlog*.
  - **Product backlog ítems (PBI):** características o elementos de funcionalidad que aportarán valor tangible al usuario y al cliente sobre el producto. Los PBIs deben priorizarse (los de prioridad baja se incluyen al final de la torre y los de prioridad alta en la parte superior) y el nivel de detalle depende de su posición en la torre (parte superior están los ítems muy detallados (*features*) listos para ser incluidos en el siguiente sprint; zona media los medianamente detallados (*features*) y en la parte inferior se encuentran los poco o nada detallados (*epics*). La Figura 13 muestra lo explicado en este punto.

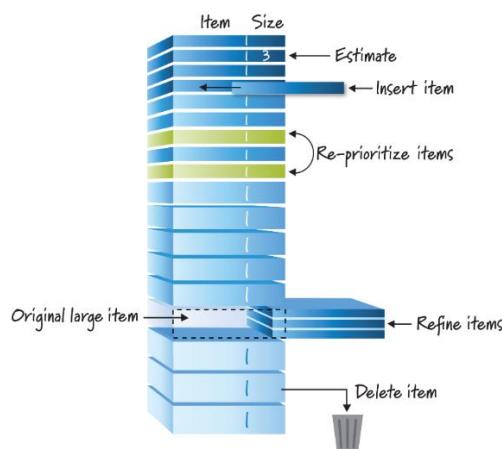


Figura 13 Product backlog items (PBI)  
 Fuente: (Kenneth S. Rubin and Innolution, 2012)



- **Features o Historias de usuario:** es cómo va a usar el producto el usuario final. Este ítem es el que más probabilidad tiene de aportar valor.
- **Errores (Bugs):** usados para mostrar errores que se han producido y han detectado los usuarios del producto. Los *bugs* se pueden clasificar por diferentes categorías (tipología, criticidad e impacto) y es esencial tener una política de resolución de los mismos que indique cuándo y cómo deben atenderse (en el momento, próximo *sprint*...).
- **Tareas técnicas:** el equipo de desarrollo es el responsable de definirlas y son clave para que el *Product Owner* entienda que ciertas labores deben realizarse, aunque el usuario no lo haya solicitado; además de ser una fuente para extraer indicadores de cada *sprint*.
- **Adquisición de conocimiento:** reunir información para cumplir con otras tareas del *Product Backlog* en el futuro.



*Tabla 15 Ejemplo de Product backlog*  
*Fuente: Elaboración propia*

PRODUCT BACKLOG							
Historias de usuario		Tarea	Estimación (días)	Responsable	Prioridad	Estado	Bugs
U 0 1	Como RAC, necesito que se haga una revisión de los nuevos requisitos (4) derivados del sprint anterior.	1- Revisión requisito 1. 2- Revisión requisito 2. 3- Revisión requisito 3. 4- Revisión requisito 4.	8	Equipo de desarrollo	Alta	80%	-
U 0 2	Como desarrollador, quiero un documento draft de especificación de diseño del interfaz de transmisión de señal.	1- Revisión requisitos transmisión. 2- Redacción documento. 3- Revisión del draft.	3	Equipo de desarrollo	Baja	0%	-
U 0 3	Como usuario, quiero una verificación de que el bug#4 se ha solucionado.	1- Prueba #1. 2- Informe de resultados. 3- Revisión calidad. 4- Implementación	5	Equipo de desarrollo	Media	20%	-
U 0 4	Como Product Owner, quiero incluir pruebas adicionales para garantizar la satisfacción del cliente.	1- Definir pruebas. 2- Incluirlos en el plan de pruebas. 3- Definir criterio de aceptación. 4- Testear	4	Product Owner	Media	50%	Bug#5 Bug#6
U 0 5	Como desarrollador, necesito una licencia adicional para el SW de validación	1- Comprar licencia. 2- Instalar licencia. 3- Probar funcionamiento de licencia.	1	Scrum Master	Alta	100%	-
BUGS		Tarea	Resolución (días)	Responsable	Prioridad	Estado	
#1	Error en el código	1- Revisar el código. 2- Probar el paquete fallido	1	Equipo de desarrollo	Alta	100%	Cerrado
#2	Error en el código: línea 156.	1- Revisar el código. 2- Probar el paquete fallido	1	Equipo de desarrollo	Baja	100%	Cerrado
#3	Fallo en pruebas.	1- Revisar el código. 2- Revisar setup de pruebas. 3- Lanzar test. 4- Modificar plan de pruebas.	7	Equipo de desarrollo	Alta	100%	Cerrado
#4	Código de error 152 durante pruebas.	1- Comprar nueva tarjeta de test. 2- Cambiarla en el setup actual. 3- Lanzar prueba.	5	Scrum Master	Media	90%	Abierto
#5	Durante la definición de las pruebas adicionales se encuentra un problema en el código fuente.	1- Revisar el código fuente. 2- Lanzar el test.	4	Equipo de desarrollo	Media	50%	Abierto
#6	Durante el lanzamiento de la segunda prueba adicional, se detecta un fallo en el setup de verificación y validación.	1- Revisar el setup de pruebas. 2- Lanzar prueba de nuevo. 3- Informar al RAC si el problema no se soluciona.	5	Equipo de desarrollo	Media	20%	Abierto



2. **Sprint backlog:** lista de necesidades (requisitos) del cliente priorizadas. La gestión la realiza el *Product Owner* y es su principal fuente de información del producto con los *stakeholders*. La Tabla 16 muestra un ejemplo de *Sprint Backlog* con la traza al *Product Backlog* anterior.

Tabla 16 Ejemplo de Sprint backlog

Fuente: Elaboración propia

SPRINT BACKLOG			
Historia Usuario	Requisito Cliente	Prioridad	Estado
U 01	C 01	Alta	Iniciado
U 05	C 02	Alta	Finalizado
U 03	C 03	Media	Iniciado
U 04	C 04	Media	Iniciado
U 02	C 05	Baja	Sin comenzar

3. **Incremento:** parte añadida o desarrollada en un *sprint* considerada parte terminada y totalmente operativa, es decir, es la suma total de los elementos del *Product Backlog* completados en el sprint actual teniendo en cuenta los *sprints* anteriores. La Tabla 17 muestra un ejemplo de Incremento.

Tabla 17 Ejemplo de Incremento

Fuente: Elaboración propia

INCREMENTO					
	Elementos Product Backlog Total	Elementos Product Backlog Terminados	Incremento	Estado	
SPRINT #1	8	8	8	100%	Finalizado
SPRINT #2	12	12	20	100%	Finalizado
SPRINT #3	5	1	21	20%	Iniciado

La Tabla 18 resume los diferentes eventos que caracterizan a la metodología Scrum, así como sus objetivos, inicio, tareas, participantes y duración.



*Tabla 18 Scrum: Eventos*  
*Fuente: Elaboración propia*

Evento	Objetivo	Inicio	Tareas	Participantes	Duración
<b>A</b> <b>(Part1)</b>	Definir el <i>sprint goal</i> y entender las necesidades del <i>Product Owner</i> (≈cliente)	Inicio del <i>sprint</i> (B)	<ul style="list-style-type: none"> <li>Revisión de elementos de máxima prioridad del <i>Product Backlog</i>.</li> <li>Objetivos y contexto de esos elementos.</li> <li>Resolución de dudas.</li> </ul>		
<b>A</b> <b>(Part2)</b>	Organizar las tareas a realizar en el <i>sprint</i> (B)	Tras Part1 del <i>sprint</i> (B)	<ul style="list-style-type: none"> <li>Previsión de la cantidad de elementos que se pueden completar en el <i>sprint</i>.</li> <li>Selección de elementos del <i>Product Backlog</i>, comenzando por los de mayor prioridad.</li> <li>Discusión de diseño y descomposición de elementos en tareas detalladas.</li> <li>Generación de tareas para un objetivo de mejora creado en la retrospectiva del <i>sprint</i> anterior</li> </ul>	<ul style="list-style-type: none"> <li><i>Product Owner</i></li> <li>Equipo de desarrollo</li> </ul>	≈ 2h para un <i>sprint</i> de 1 semana
<b>B</b>	Dar al cliente un incremento	Tras el <i>sprint</i> anterior	<ul style="list-style-type: none"> <li>Se realiza lo acordado en el <i>sprint planning</i>.</li> </ul>	<ul style="list-style-type: none"> <li>Equipo de desarrollo</li> <li><i>Stakeholders</i></li> </ul>	2-4 semanas
<b>C</b>	Actualización y coordinación entre los miembros del equipo	Diariamente	<ul style="list-style-type: none"> <li>Informar de los logros desde la última reunión para llegar al <i>sprint goal</i>.</li> <li>¿Qué trabajo se va a realizar hoy para llegar al <i>sprint goal</i>?</li> <li>Identificar los obstáculos (riesgos y problemas) para llegar al <i>sprint goal</i>.</li> </ul>	<ul style="list-style-type: none"> <li>Equipo de desarrollo</li> <li><i>Scrum Master</i></li> <li><i>Product Owner</i> (opcional)</li> </ul>	15 mins máximo (recomendación: de pie)
<b>D</b>	Revisar el <i>sprint</i> y realizar una inspección y adaptación del producto tomando las decisiones pertinentes sobre el mismo y definir futuras acciones.	Al terminar el <i>sprint</i> (B)	<ul style="list-style-type: none"> <li>El equipo de desarrollo presenta el incremento conseguido en el <i>sprint</i> a los <i>Stakeholders</i>.</li> <li>Se exponen los obstáculos encontrados y las soluciones adoptadas.</li> <li>El <i>Product Owner</i> recopila las impresiones, opiniones y respuestas de los <i>stakeholders</i> y actualiza el <i>Product Backlog</i> para el siguiente <i>sprint</i>.</li> </ul>	<ul style="list-style-type: none"> <li>Equipo de desarrollo</li> <li><i>Product Owner</i></li> <li><i>Scrum Master</i></li> <li><i>Stakeholders</i></li> </ul>	≈ 2-3h
<b>E</b>	Ánalisis del último <i>sprint</i> para identificar mejoras (desempeño, colaboración...) y aumentar la calidad y la eficacia.	Tras el <i>sprint review</i> (D)	<ul style="list-style-type: none"> <li>Inspección de todo lo ocurrido durante el <i>sprint</i>.</li> <li>Ánalisis de los obstáculos encontrados y su resolución.</li> <li>Elaboración de una lista de mejoras a implementar en el siguiente <i>sprint</i>.</li> <li>Cada miembro del equipo asume el compromiso de realizar las acciones que subsanen los obstáculos.</li> </ul>	<ul style="list-style-type: none"> <li>Equipo de desarrollo</li> <li><i>Scrum Master</i></li> <li><i>Product Owner</i> (opcional)</li> <li><i>Stakeholders</i> (optional)</li> </ul>	≈ 45 mins por <i>sprint</i> de 1 semana



#### 4.3.2 Kanban

Kanban se basa en la visualización del flujo de trabajo y la optimización de los procesos para incrementar la eficiencia. Su implementación permite a los equipos visualizar su trabajo y tareas, identificar cuellos de botella y adaptarse a los cambios de una manera más ágil.

Las prácticas clave de esta metodología son:

- Visualización del flujo de trabajo: Esto se logra a través de un tablero Kanban, donde las tareas son representadas con tarjetas (*post-it*) que se mueven a través de columnas que representan diferentes etapas del proceso. Esta visualización ayuda a identificar cuellos de botella y priorizar tareas de manera efectiva.
- Limitación del trabajo en curso (WIP o work-in-progress): al limitar el trabajo en curso, los equipos pueden concentrarse en completar las tareas antes de asumir nuevas, lo que reduce el tiempo de entrega y mejora la calidad.
- Gestión del flujo: La supervisión y gestión del flujo de trabajo son esenciales. Esto implica ajustar continuamente la cantidad de tareas en progreso y garantizar que las actividades se desplacen sin problemas a lo largo del proceso.
- Feedback loops o retroalimentación continua: Los equipos deben realizar reuniones regulares para revisar y mejorar sus procesos, incluyendo el flujo y calidad del trabajo y la colaboración del equipo.
- Mejora continua: se espera que los equipos estén en una búsqueda continua de mejoras en sus procesos. Esto puede implicar probar nuevas maneras de organizar el trabajo o implementar herramientas que incrementen la eficiencia.

Para llevar a cabo su implantación, se usará el llamado tablero Kanban (Figura 14), que es una herramienta visual que permite a los equipos de trabajo una visión general del estado actual de un proyecto. A través de él se conoce la situación de las tareas y su evolución a lo largo de las etapas del proyecto.

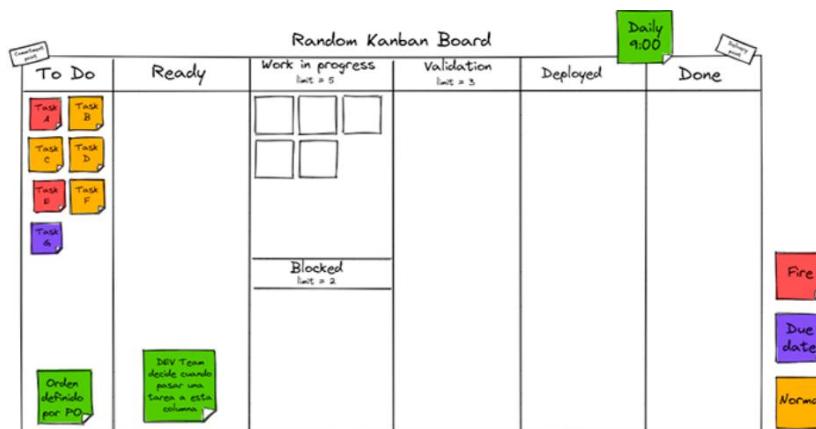


Figura 14 Ejemplo de Tablero Kanban completo

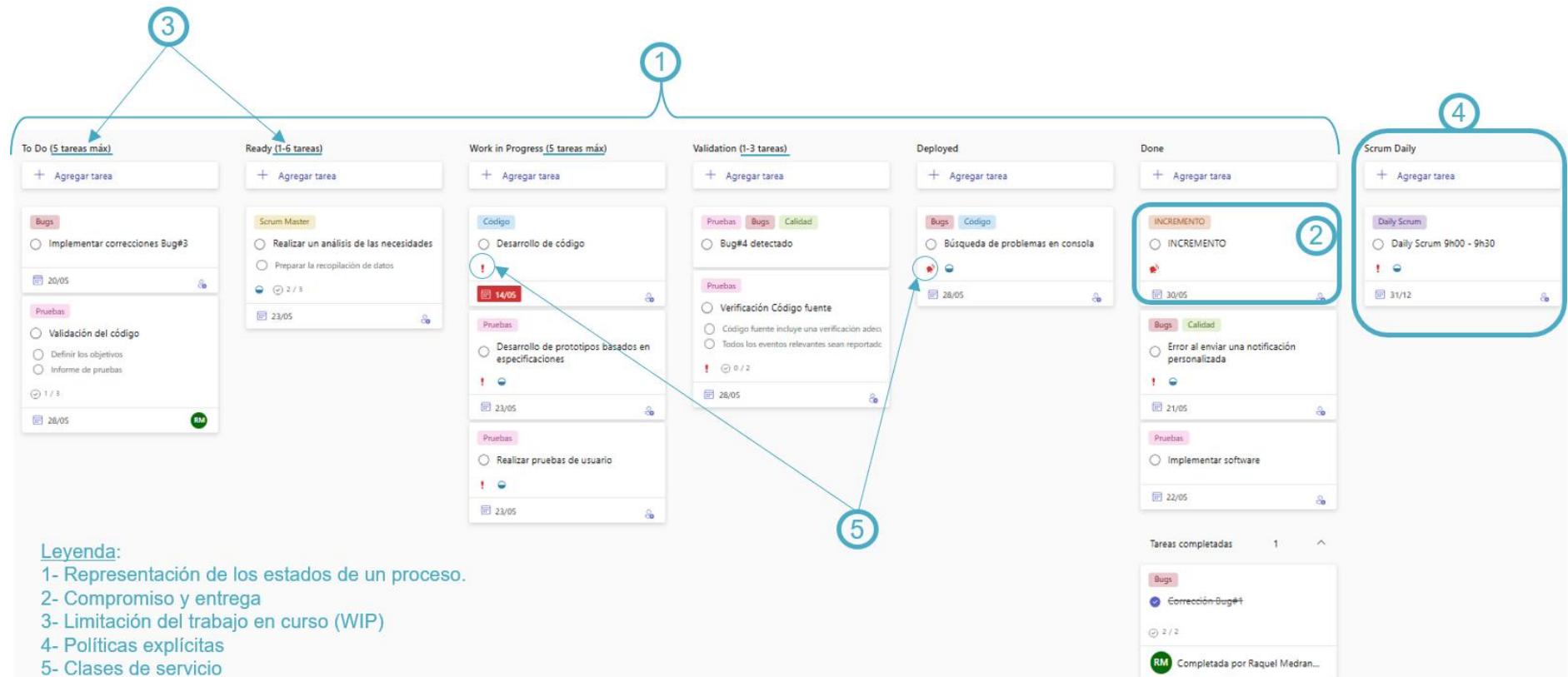
Fuente: (Garrido, 2021). Recuperado de <https://profile.es/blog/como-hacer-tablero-kanban-ejemplo/>



Para crear un tablero Kanban se deben seguir los siguientes pasos:

1. Representación de, al menos, los tres estados de un proceso: “To Do”, “WIP”, “Done”. Pudiéndose añadir también los estados: “Ready”, “Validation” y “Deployed” para que el proceso sea más realista.
2. Incluir el compromiso (commitment) y la entrega (delivery) en el tablero: con esto se medirá la eficiencia del sistema en el futuro.
3. Limitar el trabajo en curso: se puede hacer en diferentes áreas o columnas o sólo el trabajo en curso. Esto se adapta en el tiempo conforme a las capacidades y necesidades del servicio. Para limitar el trabajo:
  - Se dibujan huecos para las tareas de acuerdo con los límites acordados; por lo que, si no hay huecos libres para más tareas, se tendrán que terminar las actuales antes de comenzar las siguientes.
  - Se indica en la parte superior de cada columna o área el máximo de tareas.
4. Incluir políticas explícitas o “normas del juego”: indicaciones consensuadas entre todo el equipo como, por ejemplo, el *Daily Scrum*, priorizaciones dadas por el *Product Owner*, transiciones de tareas entre columnas, etc. Se pueden diferenciar de las tareas mediante un código de colores para que sea más visual el tablero.
5. Definir las clases de servicio: es la categorización y priorización del trabajo a realizar.

Para la creación de estos tableros, se pueden utilizar varias herramientas como Jira, Trello, Asana o Microsoft Planner. Las tareas se mueven de izquierda a derecha y de arriba hacia abajo, dejando las más prioritarias en la parte superior. La *Figura 15* muestra un ejemplo creado para este TFM con Planner.





#### 4.3.3 Actividades Específicas de Aseguramiento de Calidad

Para mantener y garantizar el aseguramiento de la calidad en un entorno de metodología ágil, se proponen las siguientes actividades.

- **Correcta ejecución de Agile teniendo:**
  - Evidencia de que el trabajo está estructurado en *sprints*,
  - Evidencia de refinamiento,
  - Evidencia de planificación de *sprints*,
  - *Daily scrums*,
  - Evidencia de ejecución del *sprint*,
  - Retrospectivas y evidencia de aprendizaje y mejora continua,
  - Evidencia de colaboración con el cliente (*sprint review*)
- **Seguimiento de entregables:**
  - *Backlog* priorizado,
  - Historias de usuarios bien escritas,
  - Evidencia de planificación y seguimiento del progreso (tablero Kanban).
- **“Hacerse” continuamente las siguientes preguntas:**
  - ¿Todo lo que haces, y se pide a los proyectos, realmente agrega valor?
  - ¿La priorización está bien realizada?

### 4.4 Normativa PECAF/AQAP vs Enfoque Agile: análisis

Esta sección analiza en profundidad la normativa PECAF/AQAP, más específicamente la PECAF/AQAP 2210, y su posible adaptación (modificaciones a realizar y a tener en cuenta) a las metodologías Agile previamente desarrolladas.

#### 4.4.1 Adaptación PECAF/AQAP 2210 a Scrum & Kanban

Se ha realizado un análisis completo de la norma PECAF/AQAP 2210 en el que, para cada uno de los requisitos, se ha evaluado su posible adaptación a la metodología Scrum y Kanban y propuesto su forma de implantación.

Todo ello está recogido en el *Anexo II* y resumido en la Tabla 19, dividida en los siguientes 3 eventos, en los que se indica si requiere o no modificación de la norma, si aplica o no el requisito o si se deja tal como está actualmente. Para cada evento se muestra a continuación un ejemplo representativo extraído del *Anexo II*.



- Requisito sensible de modificación (52 requisitos): la adaptación de la norma a la metodología ágil implica modificaciones en los requisitos de esta. Si esta modificación no se efectuara, sería aún más complicado su adaptación a las metodologías previamente propuestas.

Ejemplo:

PECAL 2210		PROPIEDAD METODOLOGÍA AGILE (SCRUM + KANBAN)
5.2. PLAN DE CALIDAD DEL PROYECTO SW (PCPS) (8.1; 8.3.2)	6. El comprador y/o RAC se reservan el derecho de rechazar el PCPS y sus revisiones (ver también PECAL-2105).	Proceso que se realizará durante cada <i>sprint review</i> y/o retrospectiva.

- Requisito no aplicable (18 requisitos): Ya no son aplicables, y podrían eliminarse de la norma, ya que son incongruentes e incompatibles con la metodología ágil.

Ejemplo:

PECAL 2210		PROPIEDAD METODOLOGÍA AGILE (SCRUM + KANBAN)
5.1 GENERALIDADES	2. El suministrador llevará a cabo una revisión formal del contrato para garantizar que se definen todos los requisitos contractuales y para determinar los procesos técnicos y de gestión necesarios que deben planificarse e implementarse.	N/A. Proceso iterativo ( <i>sprints</i> )

- Requisito no cambia (3 requisitos): Se mantiene el original de la PECAL 2210.

Ejemplo:

PECAL 2210		PROPIEDAD METODOLOGÍA AGILE (SCRUM + KANBAN)
4.3. ADAPTACIÓN	4. Cuando se adapte la norma PECAL-2210 para su uso en un contrato, el grado de adaptación deberá ser consistente con las características del producto y del proyecto, así como con los riesgos percibidos.	Se tiene en cuenta el requisito PECAL 2210.



*Tabla 19 Resumen PECL 2210 y Agile (Anexo II)*  
Fuente: *Elaboración propia*

Evento	Propuesta Metodología Agile (SCRUM + KANBAN)	Capítulos PECL 2210	Nº Requisitos
<b>Requisito sensible de modificación</b>	Trazabilidad del requisito con la metodología.	4.1 [1,2,3]; 4.2 [1,2]; 4.3 [1,2,3,6,7,8]; 4.3.1 [1]; 4.3.2 [1,2,3]; 4.4; 5.1 [1,3,4,5]; 5.2 [1,2,4,6]; 5.3 [2]; 5.4 [3]; 5.5 [1,4]; 5.6 [4,6]; 5.7.1 [1,2]; 5.7.2; 5.7.3 [1,2,3,5,6,7,8]; 5.7.4 [1,2,4]; 5.7.5; 5.7.6; 5.8; 5.9 [1,2,3].	52
<b>Requisito no aplicable</b>	N/A. Proceso iterativo ( <i>sprints</i> )	5.1 [2]; 5.2 [3,5]; 5.3 [1]; 5.4 [1,2]; 5.5 [3,5,6,7]; 5.6 [1,2,3,5,7]; 5.7.3 [4]; 5.7.4 [3]; 5.8.2 [4];	18
<b>Requisito no cambia</b>	Se mantiene el actual de PECL 2210.	4.3 [4,5]; 5.5 [2].	3

#### 4.4.2 Métricas de Calidad (KPI)

A la hora de controlar y asegurar el éxito del proceso desde el punto de vista de gestión de la Calidad, se proponen las siguientes métricas de Calidad de SW incluidas en la Tabla 20. Como guía de referencia se ha utilizado la ECSS-Q-HB-80-04A (Agencia Espacial Europea (ESA), 2011).



Tabla 20 Métricas de Calidad (SW)

Fuente: Elaboración propia

KPI#	Métrica	Objetivo	Fórmula de cálculo	Frecuencia de evaluación	Responsable
1	Eficacia	Identificar partes del proceso donde no se logren los objetivos deseados para optimizar los recursos.	$\frac{\text{Horas Test}}{\text{Total Horas proyecto}} \times 100$	Daily scrum	Product Owner
2	Cobertura test	Evaluación de la eficiencia de los tests.	$\frac{\text{Total test creados} + \text{Total test ejecutados}}{\text{Total Horas Test}} \times 100$	Finalizar cada sprint	Scrum Master
3	Densidad de bugs	Identificar la calidad del código y la necesidad de mejoras en el proceso de desarrollo.	$\frac{\text{Total bugs}}{\text{Total Horas Test}} \times 100$	Daily scrum y Sprint review	Calidad (RAC)
4	Ratio de automatización	Mejorar la eficiencia y reducir el tiempo de pruebas.	$\frac{\text{Total Horas Test automatizados}}{\text{Total Horas Test}} \times 100$	Finalizar cada sprint	Scrum Master
5	Efectividad Test	Evaluación de la capacidad de los tests para detectar defectos	$\frac{\text{Bugs detectados en 1 test}}{\text{Total de Bugs} + \text{Total Bugs tras incremento}} \times 100$	Finalizar cada sprint	Calidad (RAC)



KPI#	Métrica	Objetivo	Fórmula de cálculo	Frecuencia de evaluación	Responsable
6	Rendimiento Test	Medir tanto velocidad como eficiencia con la que se ejecutan los tests para asegurar que estos se realizan de manera oportuna y eficiente.	$\frac{Nº \text{ Pruebas ejecutadas}}{Nº \text{ Pruebas pendientes}} \times 100$	Daily scrum y Sprint review	Equipo desarrollo
7	Cumplimiento de requisitos	Medir el grado de entendimiento y ejecución del producto final con las necesidades del cliente.	$\frac{Nº \text{ Requisitos testeados}}{\text{Total requisitos}} \times 100$	Finalizar cada sprint	Product Owner
8	Resolución bugs (días)	Mejorar la eficiencia en la gestión de errores	$\text{Fecha resolución bug} - \text{Fecha de desubrimiento del bug}$	Finalizar cada sprint	Calidad (RAC)
9	Escapes de bugs	Mejorar la calidad del proceso de pruebas midiendo los bugs detectados tras el lanzamiento del producto.	$\frac{\text{Bugs producción}}{\text{Bugs test}} \times 100$	Finalizar cada sprint	Calidad (RAC)
10	Éxito sprint goal	Controlar la planificación y ejecución del sprint.	$\frac{\text{Historias de usuario completadas}}{\text{Total de Historias en sprint}} \times 100$	Finalizar cada sprint	Scrum Master



KPI#	Métrica	Objetivo	Fórmula de cálculo	Frecuencia de evaluación	Responsable
11	Velocidad	Predicción de capacidad del equipo y planificación de futuros <i>sprints</i> .	Número promedio de puntos de historia completados por sprint.	Finalizar cada <i>sprint</i>	<i>Scrum Master</i>
12	Satisfacción de Cliente		Cubierto por ISO9000 de la organización.		



## 4.5 Análisis de Riesgos: implantación de Scrum y Kanban

Adicionalmente a los análisis y propuestas incluidas en los apartados previos, se ha realizado un análisis de riesgos en el marco de la implantación de Scrum y Kanban en una organización tradicional desde el punto de vista de la gestión y del aseguramiento de la calidad (ver *Anexo III*).

El proceso de gestión de riesgos se divide en cuatro pasos principales. Primero, se definen los requisitos de implementación, incluyendo la política y el plan de gestión de riesgos. Luego, se identifican y evalúan los riesgos mediante la identificación de escenarios y la evaluación de su impacto. Después, se decide y actúa sobre los riesgos, determinando si pueden ser aceptados, reducidos o recomendados para aceptación. Finalmente, se realiza el seguimiento, comunicación y aceptación de los riesgos, asegurando que se monitoreen y se presenten para su aceptación adecuada.

Este análisis de riesgos se ha basado en definir la clase de riesgo obtenida para los diferentes sucesos en base al impacto y su probabilidad, evaluados como 1, 4 o 9.

Inicialmente se han detectado un total de 13 riesgos (Tabla 21):

Tabla 21 Resumen Análisis de Riesgos (*Anexo III*)

Fuente: Elaboración propia

ID	Descripción	Clase de riesgo
1	Resistencia al cambio por parte de los líderes y directivos de la empresa	16
2	Resistencia al cambio por parte de los empleados (departamentos de ingeniería, calidad, desarrolladores)	81
3	Implementación ineficaz de SCRUM	81
4	Implementación ineficaz de KANBAN	36
5	Comunicación ineficaz entre los miembros del equipo de desarrollo	4
6	Identificación tardía de riesgos	36
7	Equipos de desarrollo con sobrecarga de trabajo	9
8	Adaptación a los cambios continuos	16
9	Insatisfacción del cliente	9
10	Escapes o fugas de calidad en los entregables	16
11	Falta de documentación a entregar en cada incremento	16
12	Dependencia del Scrum Master	9
13	Falta de estandarización de los procesos	16



- **Alto: 4** (ID#2, ID#3, ID#4, ID#6)
  - Relativos a la resistencia al cambio, implementación ineficaz de Scrum y Kanban e identificación de riesgos. Como medidas de mitigación, se proponen, entre otras, la realización de formaciones sobre las metodologías Scrum y Kanban, la contratación de un *Scrum Master*, el uso de tableros Kanban y la realización de revisiones periódicas para identificar y resolver bloqueos. Además de la implementación de un proceso de gestión de riesgos continuo y proactivo revisado regularmente
- **Medio: 8** (ID#1, ID#7, ID#8, ID#9, ID#10, ID#11, ID#12, ID#13)
  - Debido al desconocimiento de las metodologías y sus formas de trabajo.
- **Bajo: 1** (ID#5)
  - Relativo a la comunicación del equipo.

Todos los riesgos identificados llevan asociadas medidas de mitigación que, tras implementarlas, se estima que los mismos pasen a ser riesgos medios o bajos. El Anexo III explica la metodología que se ha seguido para hacer el análisis, así como la tabla detallada.

## 4.6 Futuros Pasos

Este TFM se ha centrado en un número limitado de metodologías, todas ellas pertenecientes al mundo Agile. Como se mencionó previamente en el capítulo 3.4, una posible línea de trabajo futura podría ser la implantación de la metodología SAFe, ya implantada en grandes empresas como GMV, y que se basa en los siguientes 5 valores y 9 principios:

- Valores: Orientación, Calidad, Transparencia, Ejecución del programa y Liderazgo.
- Principios: Visión económica, Aplicación del pensamiento sistemático, Asumir la variabilidad, Construcción de ciclos de aprendizaje rápidos, Establecimiento de hitos para la evaluación del trabajo, Visualizar y limitar el trabajo en curso, Aplicación de la cadencia, Impulsar la motivación de los miembros del equipo y Descentralizar la toma de decisiones.

La Figura 16 muestra un ejemplo de escalado de la metodología Agile a escenarios más complejos en el marco de SAFe.

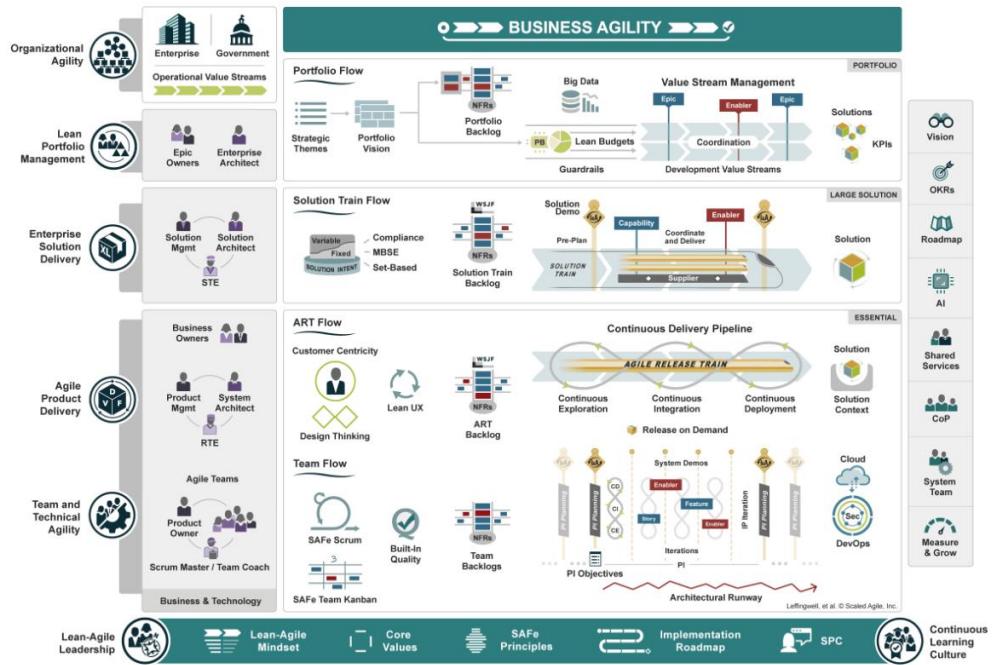


Figura 16 SAFe: Ejemplo

Fuente: (Scaled Agile, 2025). Recuperado de <https://framework.scaledagile.com/#full>



## 5 CONCLUSIONES

La adopción de las metodologías Agile tiene el potencial de mejorar significativamente la gestión de proyectos. No solo se verán beneficiadas la eficiencia operativa y la calidad del software, sino también la satisfacción y motivación del equipo. Cada metodología ofrece ventajas únicas y complementarias que, si se aplican estratégicamente, pueden llevar a un rendimiento sobresaliente de la gestión de la calidad.

Implementar una metodología ágil requiere una gran cantidad de esfuerzo, pero su transparencia permite que el equipo de desarrollo trabaje a un ritmo acelerado. Entregar un producto de SW al final de un ciclo de 30 días que pueda ser utilizado por un *stakeholder* en un entorno operativo es un desafío a medida que la funcionalidad del producto de software crece.

Uno de los mayores beneficios que las metodologías ágiles proporcionan a un *stakeholder* es un aumento en la calidad en comparación con las técnicas tradicionales de desarrollo de SW. Agile enfatiza la prueba temprana y frecuente en el ciclo de vida del desarrollo. Esto ayuda a identificar defectos en la aplicación o deficiencias arquitectónicas antes de que se conviertan en un riesgo y puedan ser fácilmente corregidas o se descubran demasiado tarde en la fase de entrega para ser corregidas sin poner en riesgo el cronograma de entrega.

El uso de aplicaciones de integración continua, que ejecute automáticamente una serie de pruebas unitarias y de integración en la aplicación con la introducción de nuevo código en la aplicación, serviría para protegerla de la introducción de defectos en áreas anteriormente funcionales, a medida que se implementan nuevas características.

El área de la gestión de riesgos y mitigación de estos, directamente relacionada con la gestión de calidad, es otra en la que se obtiene un beneficio como resultado de los ciclos de entrega acortados de un proyecto de métodos ágiles.

Típicamente, en un ciclo de desarrollo de 30 días, los *stakeholders* son informados de riesgos potenciales y problemas antes de que estos se conviertan en un problema real e irrecuperable o requiera una asignación significativa de recursos para su reparación. La identificación temprana de las limitaciones de infraestructura permite al equipo tomar acciones correctivas antes de que el riesgo tenga un gran impacto negativo en el proyecto.

En cuanto al cumplimiento de objetivos, la Tabla 22 muestra el cumplimiento y la trazabilidad al mismo de los objetivos definidos en la sección 2.1.1.



*Tabla 22 Cumplimiento de Objetivos*  
*Fuente: Elaboración propia*

Objetivo	Cumplimiento	Evidencias
O#1		Ver §4.3 y §4.4.2.
O#2		Ver §4.4.1 y Anexo II.
O#3		Ver §4.1.
O#4		Ver §4.2, §4.3.1 y §4.3.2
O#5		Ver §4.5, §4.6 y Anexo III.



## 6 REFERENCIAS BIBLIOGRÁFICAS

- Agencia Espacial Europea (ESA). (2011). *ECSS-Q-HB-80-04A\_Space product assurance: Software metrification programme definition and implementation*. ESA-ESTEC (Noordwijk, The Netherlands): European Space Agency.
- Agile Carpentry. (2024). Obtenido de Guía Scrum: [https://agilecarpentry.com/scrum\\_guide/less-scrum-guide/](https://agilecarpentry.com/scrum_guide/less-scrum-guide/)
- Baena, G. (2014). *Metodología de la investigación*. Mexico: Grupo Editorial Patria.
- Calduch. (2014). Métodos y técnicas de investigación internacional, 2<sup>a</sup> Edición electrónica revisada y actualizada. En Dr. Rafael Calduch. Madrid: Universidad Complutense de Madrid.
- Digital.ai. (2024). *The 17th State of Agile Report*. Raleigh, NC: Digital.ai.
- Ejército del Aire y del Espacio. (2024). *Aplicación de metodologías ágiles para proyectos web del EA*. Madrid: Ejército del Aire y del Espacio.
- Garrido, G. R. (8 de 11 de 2021). Obtenido de Cómo hacer un tablero Kanban en 5 pasos con un ejemplo de software: <https://profile.es/blog/como-hacer-tablero-kanban-ejemplo/>
- Kenneth S. Rubin and Innolution, L. (2012). Obtenido de Chapter 6: Product Backlog: <https://innolution.com/essential-scrum/table-of-contents/chapter-6-product-backlog>
- Kent Beck et al. (2001). *Manifiesto por el Desarrollo Ágil de Software*. Obtenido de <https://agilemanifesto.org/iso/es/manifesto.html>.
- Kent Beck et al. (2001). *Principios del Manifiesto Ágil*. Obtenido de <https://agilemanifesto.org/iso/es/principles.html>.
- Ministerio de Defensa. (6 de Junio de 2018). *Ministerio de Defensa, Organigrama*. Obtenido de <https://www.defensa.gob.es/ministerio/organigrama/>
- Project Management Institute. (2017). *Agile Practice Guide*. Pennsylvania (EEUU): Project Management Institute (PMI).
- Project Management Institute Inc. (2017). *A guide to Project Management Body of Knowledge PMBOK® Guide (6th Edition)*. Pennsylvania (EEUU): Project Management Institute (PMI).
- Saaty, T. L. (1980). *The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation*. Universidad de California: McGraw-Hill International Book Company.
- Scaled Agile, I. (2025). Obtenido de SAFe Big Picture: <https://framework.scaledagile.com/#full>



## 7 ANEXOS



## Anexo I Técnica de Decisión Multicriterio – Metodología Analytic Hierarchy Process (AHP)

El Proceso de Análisis Jerárquico, es un método basado en la evaluación de diferentes criterios que permiten jerarquizar un proceso y, su objetivo final consiste en optimizar la toma de decisiones gerenciales (Saaty, 1980). Esta metodología se utiliza para resolver problemas en los cuales existe la necesidad de priorizar distintas opciones y posteriormente decidir cuál es la opción más conveniente.

La metodología AHP es una herramienta flexible de toma de decisiones multi-criterio, utilizada en problemas en los cuales necesitan evaluarse aspectos tanto cualitativos como cuantitativos. Además, permite de manera eficiente y gráfica organizar la información respecto a un problema de decisión, descomponerla y analizarla por partes. Saaty dijo de su método que “trata de desmenuzar un problema y luego unir todas las soluciones de los subproblemas en una conclusión”. Esta técnica ayuda a los analistas a organizar los aspectos críticos de un problema en una estructura jerárquica similar a la estructura de un árbol familiar, reduciendo las decisiones complejas a una serie de comparaciones que permiten la jerarquización de los diferentes aspectos (criterios) evaluados.

La metodología AHP se divide en 4 etapas, las cuales son:

1. Formulación del problema.
2. Evaluación de los criterios y subcriterios (si los hubiera).
3. Evaluación de las alternativas.
4. Jerarquización

### Etapa 1: Formulación del problema

Se representa el problema mediante un diagrama de árbol. En él existe una jerarquía implícita de los componentes del problema. Esta Jerarquía implica una clasificación ordinal con al menos 3 niveles subordinados entre sí:

1. Objetivo
2. Criterios – Subcriterios (1, 2... " n")
3. Alternativa

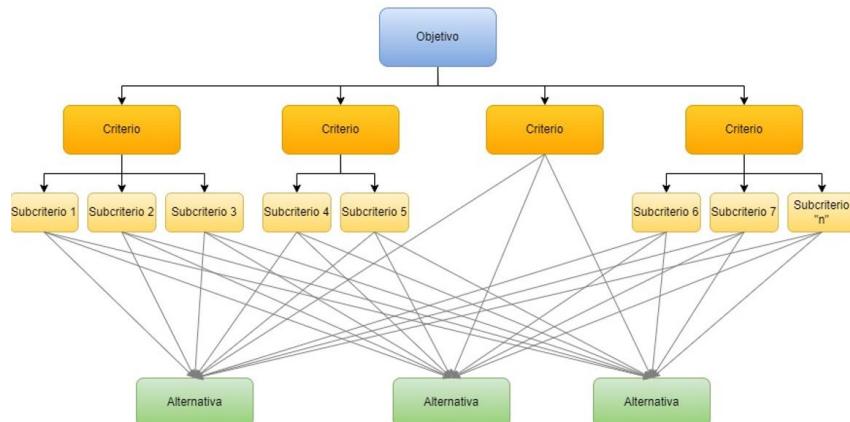


Figura 17 Anexo I: Ejemplo Etapa 1

Fuente: Elaboración propia

#### Etapa 2: Evaluación de los criterios

En esta etapa, se contará con grupos de expertos de cada materia que analizarán, tanto los criterios ya definidos como sus pesos.

Los pesos de los criterios y subcriterios se calculan siguiendo el mismo proceso que se describe de forma genérica a continuación.

Se construye una matriz A a partir de la comparación de los diferentes criterios dos a dos, con el propósito de estimar la importancia relativa entre cada uno de ellos. A cada comparación se le asignará una calificación según la escala de Saaty.

Valor	Definición	Comentario
1	Igual importancia	A y B tienen la misma importancia
3	Importancia moderada	A es ligeramente más importante que B
5	Importancia grande	A es más importante que B
7	Importancia muy grande	A es mucho más importante que B
9	Importancia extrema	A es extremadamente más importante que B

Figura 18 Anexo I: Escala de Saaty

Fuente: Master Adquisiciones de Sistemas de Defensa (Módulo 2)

Una vez normalizada la matriz, se procede a estimar los correspondientes pesos relativos de los criterios W. De esta forma obtenemos los pesos definitivos asignados a cada criterio. El cálculo de los pesos se realiza calculando la media aritmética de las filas de la matriz normalizada A'.



Existe también un indicador que controla el grado de consistencia de las puntuaciones realizadas en cada fase del proceso al establecer la importancia relativa entre los elementos de cada nivel. Controla la subjetividad, coherencia y consistencia de los expertos. Se denomina Razón de **Inconsistencia (R.I.)**, e indica, de manera numérica, el grado de incoherencia que se comete al calificar la importancia relativa de los criterios y alternativas de un problema:

$$R.I. = \frac{IC}{CA}$$

IC: índice de consistencia

CA: consistencia aleatoria

La RI verifica además que se cumpla la:

- Transitividad: si A es mejor que B y B es mejor que C, entonces A debería ser mejor que C.
- Proporcionalidad: si A es 2 veces más importante que B y B es 2 veces más importante que C, entonces A debería ser 4 veces más importante que C.

#### Etapa 3: Evaluación de las alternativas

En esta etapa, se construyen tantas matrices como subcriterios y criterios que no tengan subcriterios haya. En cada matriz se van a comparar entre sí las alternativas de acuerdo al correspondiente criterio o subcriterio.

La completitud de las matrices, el cálculo de los pesos relativos y el cálculo de la razón de inconsistencia es similar al apartado anterior.

#### Etapa 4: Jerarquización

Finalmente, para conocer qué alternativa es la más importante de acuerdo a los criterios establecidos se trasladan a la matriz de decisión todos los pesos calculados en los 3 pasos anteriores.

La herramienta utilizada para realizar el análisis es el SW informático que se utilizó en el tema 8 del módulo 2 del Master de Adquisiciones de sistemas de defensa (Coronel Caballería Carlos Ruiz López, CUD Zaragoza).



## Anexo II PECAL 2210 y Agile



PECAL 2210		¿REQUISITO CAMBIA?	PROPIEDAD METODOLOGÍA AGILE (SCRUM + KANBAN)
<b>4 REQUISITOS GENERALES DE ASEGURAMIENTO DE LA CALIDAD DEL SOFTWARE</b>			
4.1. REQUISITOS DE ASEGURAMIENTO DE LA CALIDAD SUPLEMENTARIOS A LA NORMA PECAL-2110	1. Si esta publicación es utilizada como Requisitos Suplementarios de Aseguramiento de la Calidad del Software a la PECAL-2110, el Sistema de Gestión de la Calidad también deberá estar de acuerdo con esta publicación, que incluye los requisitos de la norma AS9115 como necesarios para satisfacer los requisitos del contrato.	SI	Modificación de requisito ya que la AS9115 no es compatible con las metodologías ágiles.
	2. La norma AS9115 (estándar de software relacionado con la norma AS9100) será aplicable para los Requisitos de Aseguramiento de la Calidad del software, con las siguientes aclaraciones.	SI	Propuesta de modificación: Integrar los requisitos de calidad en las iteraciones de desarrollo, asegurando que cada <i>sprint</i> incluya actividades de aseguramiento de calidad.
	3. En el contexto de la PECAL-2210: a. Las referencias a AS9100 se reemplazan por ISO 9001. b. El párrafo 8.1.2 (Gestión de la configuración) de AS9115 se relaciona con el párrafo 5.4.1.2 de la PECAL-2110. c. El párrafo 8.1.4 (Prevención de material falsificado) de AS9115 se relaciona con el párrafo 5.4.6.2(2) de PECAL-2110.	SI	Propuesta de modificación: Integrar los requisitos de calidad en las iteraciones de desarrollo, asegurando que cada <i>sprint</i> incluya actividades de aseguramiento de calidad.
4.2. REQUISITOS DE ASEGURAMIENTO DE LA CALIDAD SUPLEMENTARIOS A LA NORMA PECAL-2310	1. Si esta publicación es utilizada como Requisito Suplementario de Aseguramiento de la Calidad del software de la PECAL-2310, el Sistema de Gestión de la Calidad también deberá estar de acuerdo con esta publicación, que incluye los requisitos de la norma AS9115 que sean necesarios para satisfacer los requisitos del contrato.	SI	Modificación de requisito ya que la AS9115 no es compatible con las metodologías ágiles.
	2. La norma AS9115 (estándar de software relacionado con la norma AS9100) será aplicable para los requisitos de Aseguramiento de la Calidad del software.	SI	Propuesta de modificación: Utilizar revisiones frecuentes y retrospectivas para evaluar la calidad del software y ajustar los procesos según sea necesario.
4.3. ADAPTACIÓN	1. El propósito del proceso de Adaptación es la adaptación de los requisitos (y los datos de ciclo de vida relacionados) de la PECAL-2210 para satisfacer las características específicas del proyecto de software, y asegurar que el alcance de los requisitos (y los datos del ciclo de vida) seleccionados es consistente con los riesgos asociados con el proyecto de software.	SI	Será el paquete de requisitos iniciales (generales) al que luego se le irán añadiendo requisitos conforme se realicen <i>sprints</i> , incrementos, retrospectiva y <i>sprint reviews</i> .
	2. El proceso de Adaptación es la eliminación (total o parcial), disminución, modificación o adición de procesos (actividades y tareas) de la PECAL-2210, con el fin de optimizar la eficacia y eficiencia de los procesos técnicos y de gestión para el proyecto de software específico; la adaptación se aplicará también a los datos del ciclo de vida producidos por cada proceso (actividad y tarea).	SI	Por ciclo de vida se entiende que es cada uno de los diferentes <i>sprints</i> . En estos <i>sprints</i> , se analizará tanto la criticidad (A-D) como el nivel de desarrollo (1-4) y se modificará según el estado del SW.
	3. El proceso de Adaptación estará determinado por las características del proyecto y del producto (controladores de la adaptación). Los controladores pueden ser una parte implícita de la experiencia colectiva del equipo del proyecto o pueden definirse explícitamente para un proyecto específico o para una aplicación dentro de un proyecto; existen varios tipos de controladores relacionados con diferentes aspectos, como la confiabilidad y la seguridad, las restricciones de desarrollo del software, la calidad del producto o los objetivos comerciales.	SI	La palabra "controlador" se traduce en indicaciones consensuadas en el <i>Daily Scrum</i> por el equipo. Se incluirán en el Tablero Kanban.
	4. Cuando se adapte la norma PECAL-2210 para su uso en un contrato, el grado de adaptación deberá ser consistente con las características del producto y del proyecto, así como con los riesgos percibidos.	NO	Requisito no cambia. Se mantiene el actual de PECAL 2210.
	5. El proceso de adaptación puede ser realizado: a. A nivel de la aplicación (ECS Elemento de Configuración Software); b. A nivel del proyecto (Sistema Software).	NO	Se tiene en cuenta el requisito PECAL 2210.



PECAL 2210		¿REQUISITO CAMBIA?	PROPIEDAD METODOLOGÍA AGILE (SCRUM + KANBAN)
	6. En general, deberán aplicarse diferentes niveles de adaptación en función de la criticidad y del alcance del desarrollo.	SI	Las adaptaciones estarán de acuerdo a los procesos de Scrum y Kanban
	7. El enfoque inicial de la adaptación deberá ser comprender el nivel de calidad solicitado para el proyecto de software, con la caracterización del proyecto, la identificación de los procesos necesarios y la caracterización del producto.	SI	<p>El enfoque inicial de la adaptación se incluirá en el Product backlog, y por ende en el Sprint backlog y tablero Kanban.</p> <p>Los roles quedan definidos en la sección 4.3.1.1..</p> <p>El nivel de calidad, la caracterización del proyecto, la identificación de los procesos necesarios y la caracterización del producto son sensibles de modificaciones a lo largo del ciclo de vida. Todo ello se revisará en cada <i>Sprint Review</i>.</p>
	8. Con el fin de aplicar el proceso de adaptación, el software deberá ser clasificado en «niveles de criticidad», según la gravedad de las consecuencias de los fallos del sistema, y en «niveles de desarrollo», según el alcance de las actividades de desarrollo que se requerirán para entregar dicho software.	SI	<p>Los niveles de criticidad se traducirán en políticas explícitas en el tablero Kanban.</p> <p>Los niveles de desarrollo serán parte del <i>product backlog</i>. Se incluirán como parte de los requisitos generales al inicio del programa. Durante el ciclo de vida del mismo, es sensible de cambios durante los diferentes <i>sprints</i>.</p>



PECAL 2210		¿REQUISITO CAMBIA?	PROPIUESTA METODOLOGÍA AGILE (SCRUM + KANBAN)
4.3.1. NIVELES DE CRITICIDAD	<p>1. Se definen los siguientes cuatro «niveles de criticidad» del software, en función de la gravedad de las consecuencias de los fallos del sistema (incluyendo el impacto en la disponibilidad, integridad, confidencialidad, negocio, seguridad de las personas o daños a la propiedad), también en particular se utilizarán para determinar el alcance necesario de pruebas de integración o calificación a aplicar:</p> <ul style="list-style-type: none"> <li>- <b>A:</b> Software que si no se ejecuta o no se ejecuta correctamente, o cuyo comportamiento anómalo o integración incorrecta en un sistema, puede causar o contribuir a un fallo del sistema que resulte en Consecuencias Catastróficas, en términos de efectos sobre la seguridad o impacto en el éxito de la misión.</li> <li>- <b>B:</b> Software que si no se ejecuta o no se ejecuta correctamente, o cuyo comportamiento anómalo o integración incorrecta en un sistema, puede causar o contribuir a un fallo del sistema que resulte en Consecuencias Críticas o Peligrosas, en términos de efectos sobre la seguridad o impacto en el éxito de la misión.</li> <li>- <b>C:</b> Software que si no se ejecuta o no se ejecuta correctamente, o cuyo comportamiento anómalo o integración incorrecta en un sistema, puede causar o contribuir a un fallo del sistema que resulte en Consecuencias Mayores, en términos de efectos sobre la seguridad o impacto en el éxito de la misión.</li> <li>- <b>D:</b> Software que si no se ejecuta o no se ejecuta correctamente, o cuyo comportamiento anómalo o integración incorrecta en un sistema, puede causar o contribuir a un fallo del sistema que resulte en Consecuencias Menores o Insignificantes, sin efectos sobre la seguridad o impacto en el éxito de la misión.</li> </ul>	SI	Los niveles de criticidad se traducirán en políticas explícitas en el tablero Kanban.



PECAL 2210		¿REQUISITO CAMBIA?	PROPIEDAD METODOLOGÍA AGILE (SCRUM + KANBAN)
4.3.2. NIVELES DE DESARROLLO	1. Se definen los siguientes cuatro «niveles de desarrollo» del software, según el nivel de desarrollo, prueba, inspección o control requerido para satisfacer los requisitos del contrato: <ul style="list-style-type: none"> <li>- 1:               <ul style="list-style-type: none"> <li>• Todo el desarrollo o mantenimiento de software a medida.</li> <li>• Software reutilizado, de código abierto o proporcionado por el cliente con una integración significativa.</li> </ul> </li> <li>- 2:               <ul style="list-style-type: none"> <li>• Software reutilizado, de código abierto o proporcionado por el cliente sin una integración significativa.</li> <li>• Software estándar militar o modificado con una integración significativa.</li> <li>• Software comercial listo para usar con una integración significativa.</li> <li>• Software a medida no entregable con una integración significativa.</li> </ul> </li> <li>- 3:               <ul style="list-style-type: none"> <li>• Software estándar militar o modificado sin una integración significativa.</li> <li>• Software comercial listo para usar sin una integración significativa.</li> <li>• Software a medida no entregable sin integración significativa.</li> </ul> </li> <li>- 4:               <ul style="list-style-type: none"> <li>• Software comercial listo para usar con integración mínima.</li> </ul> </li> </ul>	SI	Los niveles de desarrollo serán parte del <i>product backlog</i> . Se incluirán con parte de los requisitos generales al inicio del programa. Durante el ciclo de vida del mismo, es sensible de cambios durante los diferentes <i>sprints</i> .
	2. Todas las decisiones sobre la adaptación deberán ser acordadas entre el comprador y el suministrador y reflejadas en la documentación contractual.	SI	Las decisiones serán recogidas en cada <i>sprint review</i> e incluidas en una actualización del <i>Product Backlog</i> por el <i>Product Owner</i> . Esto ya no se reflejará en la documentación contractual.
	3. El suministrador, tras la correspondiente categorización del software y la aplicación del proceso de adaptación, deberá preparar el plan de calidad del proyecto software con la particularización final de la norma PECAL-2210; todas las decisiones de la adaptación deberán quedar documentadas conjuntamente con la justificación de las decisiones tomadas.	SI	Se realizará un plan de calidad al inicio del proyecto teniendo en cuenta los requisitos generales definidos entonces, asumiendo que habrá modificaciones y nuevos requisitos tras la finalización de cada sprint, y los procesos de aseguramiento de la calidad que se vayan a realizar en cada evento Scrum. Las decisiones serán recogidas en cada <i>sprint review</i> e incluidas en una actualización del <i>Product Backlog</i> por el <i>Product Owner</i> .
4.4. CLÁUSULAS ADAPTABLES/NO ADAPTABLES	Solo puede ser objeto de adaptación el capítulo 5 de esta norma. Los siguientes puntos del capítulo 5 no pueden ser adaptados: <ul style="list-style-type: none"> <li>- 5.1. Generalidades.</li> <li>- 5.2. Plan de Calidad del Proyecto Software (PCPS).</li> <li>- 5.3. Análisis de Criticidad del software.</li> <li>- 5.4. Modelo de Calidad del software (únicamente puede adaptarse el punto 2).</li> <li>- 5.5. Identificación y Revisión de los Requisitos del software (únicamente puede adaptarse el punto 4).               <ul style="list-style-type: none"> <li>- 5.7.1. Roles en la organización, responsabilidades y autoridades.</li> <li>- 5.7.2. Gestión de subsuministradores.</li> <li>- 5.7.3. Gestión de la Configuración del software (GCS).</li> <li>- 5.7.5. Infraestructura.</li> </ul> </li> </ul>	-	Ver respuestas en cada subapartado.



PECAL 2210		¿REQUISITO CAMBIA?	PROPIEDAD METODOLOGÍA AGILE (SCRUM + KANBAN)
<b>4 REQUISITOS GENERALES DE ASEGURAMIENTO DE LA CALIDAD DEL SOFTWARE</b>			
5.1. GENERALIDADES	1. Para lograr la visibilidad y el control del proyecto de desarrollo de software, el suministrador deberá planificar e implementar actividades eficaces de gestión de la calidad del software.	SI	El equipo planifica y replanifica a medida que se obtiene información tras cada <i>sprint review</i> . Se incorporarán prácticas de calidad en el <i>product backlog</i> asegurándose de que se aborden en cada <i>sprint</i> .
	2. El suministrador llevará a cabo una revisión formal del contrato para garantizar que se definen todos los requisitos contractuales y para determinar los procesos técnicos y de gestión necesarios que deben planificarse e implementarse.	No Aplica (N/A)	N/A. Proceso iterativo ( <i>sprints</i> )
	3. Basándose en los requisitos del contrato, las reglas y procedimientos del Sistema de Gestión de la Calidad de la organización y los requisitos específicos del proyecto, las actividades de gestión de la calidad del software deberán: a. Establecer/identificar, precisar y asignar requisitos a productos de software y elementos de configuración (véase el apartado 5.5). b. Establecer e implementar procesos administrativos y técnicos para diseñar, desarrollar y construir calidad en el software (véanse los apartados 5.6 y 5.7). c. Establecer e implementar procedimientos para verificar y validar la calidad de los productos de software y para evaluar procesos y actividades, incluido el software no entregable, que impacten en la calidad de los productos de software (véase apartado 5.8). d. Asegurarse de que el riesgo específico del software se haya incluido en el plan de gestión de riesgos. El suministrador deberá identificar, analizar, priorizar y monitorizar las áreas del proyecto de software que impliquen un riesgo potencial de software. e. Documentar e implementar una metodología segura de desarrollo de software basada en ISO/IEC 27001 u otro estándar reconocido acordado con el comprador.	SI	La propuesta de aseguramiento de calidad que expone en un entorno de metodología ágil se divide en: a. Aseguramiento de que Agile se ejecuta bien de acuerdo a las siguientes actividades: — Evidencia de que el trabajo está estructurado en <i>sprints</i> . — Evidencia de refinamiento. — Evidencia de planificación de <i>sprints</i> . — <i>Daily scrums</i> . — Evidencia de ejecución del <i>Sprint</i> . — Retrospectivas y evidencia de aprendizaje y mejora continua. — Evidencia de colaboración con el cliente ( <i>sprint review</i> ). b. Y en términos de entregables: — <i>Backlog</i> priorizado — Historias de usuarios bien escritas. — Evidencia de planificación y seguimiento del progreso (tablero Kanban).
	4. Las actividades de gestión de la calidad del software del proyecto se basarán en los estándares existentes y en los procedimientos del Sistema de Gestión de la Calidad de la organización. Cuando este no sea el caso, se deberá proporcionar una justificación al comprador y/o al RAC.	SI	El RAC podrá ser parte del equipo o un stakeholder por lo que tendrá visibilidad total de la gestión de la calidad del SW y no será necesaria una justificación formal. El comprador podrá ser: - Usuario (e.j: FAS), - Stakeholder, - Manager (e.j: Jefe de Programa). Y participará en las <i>sprint reviews</i> .
	5. Las actividades de gestión de la calidad del software del proyecto se documentarán en el Plan de Calidad del Proyecto Software (PCPS) (véase apartado 5.2).	-	Ver respuestas al apartado 5.2



PECAL 2210		¿REQUISITO CAMBIA?	PROPIUESTA METODOLOGÍA AGILE (SCRUM + KANBAN)
5.2. PLAN DE CALIDAD DEL PROYECTO SOFTWARE (PCPS) (8.1; 8.3.2)	1. El suministrador deberá documentar las actividades de gestión de la calidad del software, relacionadas con el proyecto, en un Plan de Calidad del Proyecto Software (ver también PECAL-2105). El Plan de Calidad del Proyecto Software (PCPS) llevará la firma de aprobación de aquellas personas de la organización que tengan responsabilidades identificadas en el PCPS y estará sujeto a control de configuración.	SI	<p>Se realizará un plan de calidad al inicio del proyecto teniendo en cuenta los requisitos generales definidos entonces, asumiendo que habrá modificaciones y nuevos requisitos tras la finalización de cada <i>sprint</i>, y los procesos de aseguramiento de la calidad que se vayan a realizar en cada evento Scrum. Estos últimos también serán sensibles de modificación/actualización dependiendo de cada <i>sprint review</i> y <i>retrospective</i>. El PCPS será un documento vivo que se actualizará en cada <i>sprint</i>. El control de configuración se cumplirá siempre.</p>
	2. El suministrador deberá presentar al RAC y/o comprador un PCPS aceptable basado en los requisitos contractuales, en un plazo mutuamente acordado, pero siempre antes del comienzo de las actividades (lo cual puede definirse como una reunión de lanzamiento del proyecto o contrato, o como se haya determinado en el contrato o pedido). El PCPS puede ser un documento independiente y claramente identificado, o parte de cualquier otro documento que se haya preparado para el contrato.	SI	<p>El suministrador presentará la primera versión con los requisitos generales, pero no una versión actualizada tras cada uno de los sprints (implicaría un incremento de trabajo considerable y un aporte de valor ínfimo), ya que toda la información actualizada (requisitos, procesos, pruebas, resultados, lecciones aprendidas...etc) quedaría reflejado e incluido en el <i>Product Backlog</i>. Además, el RAC estará involucrado en todos los procesos.</p> <p>Si se quisiera, aunque no aportaría valor añadido, podría actualizarse incluyendo todos los <i>Product Backlogs</i>, como evidencia de la gestión de la calidad, en la mitad del proyecto (o tras "X" sprints - a acordar entre el suministrador y el RAC) y una vez finalizado como evidencia de cierre y conteniendo toda la información y lecciones aprendidas.</p>
	3. El PCPS deberá documentar y mantener la trazabilidad de los requisitos desde el proceso de planificación, incluyendo una matriz de cumplimiento de requisitos y soluciones que justifique el cumplimiento de todos los requisitos contractuales (a los que se hará referencia cuando sea aplicable).	No Aplica (N/A)	<p>N/A. Proceso iterativo (<i>sprints</i>). Solamente se tendrá trazabilidad de los requisitos generales definidos al comienzo del proyecto. Los nuevos requisitos, o modificaciones de los ya establecidos, quedarán recogidos en los <i>Product Backlogs</i>.</p>
	4. El PCPS también deberá incluir: a. Métodos de análisis y criterios para determinar la criticidad del software (véase apartado 5.3 «Análisis de la Criticidad del Software»). b. Modelos de calidad, derivados de la norma ISO/IEC 25010 u otro estándar apropiado (véase apartado 5.4 Modelo de Calidad de Software) para cada producto de software, con las reglas de control y garantía de calidad para obtener y evaluar las subcaracterísticas. c. La determinación del alcance apropiado de la planificación y de control para apoyar las actividades de software requeridas por el contrato.  Nota: Los informes de calidad deben ubicarse dentro del ciclo del contrato (por ejemplo, revisiones, reuniones de progreso, fase de pruebas, punto clave del ciclo de desarrollo, etc.).	SI	<p>Modificar el alcance de este requisito para que sea compatible con la metodología Agile:</p> <p>En vez de incluir alcances, procesos, procedimientos, reglas...etc, se ha de garantizar confianza/seguridad en la metodología:</p> <p>a. Aseguramiento de que Agile se ejecuta bien de acuerdo a las siguientes actividades:</p> <ul style="list-style-type: none"> <li>— Evidencia de que el trabajo está estructurado en <i>sprints</i>.</li> <li>— Evidencia de refinamiento.</li> <li>— Evidencia de planificación de <i>sprints</i>.</li> <li>— <i>Daily scrums</i>.</li> <li>— Evidencia de ejecución del Sprint.</li> <li>— Retrospectivas y evidencia de aprendizaje y mejora continua.</li> <li>— Evidencia de colaboración con el cliente (<i>sprint review</i>).</li> </ul> <p>b. Y en términos de entregables:</p> <ul style="list-style-type: none"> <li>— Backlog priorizado</li> </ul>



PECAL 2210		¿REQUISITO CAMBIA?	PROPIUESTA METODOLOGÍA AGILE (SCRUM + KANBAN)
			<ul style="list-style-type: none"> <li>— Historias de usuarios bien escritas.</li> <li>— Evidencia de planificación y seguimiento del progreso (tablero Kanban).</li> </ul>
	<p>5. El PCPS será utilizado por el suministrador como línea de base para definir las actividades para supervisar y controlar la calidad del proyecto de software. El PCPS se revisará y actualizará en hitos predefinidos durante el proyecto a medida que se conozcan nuevas definiciones y detalles de desarrollo.</p> <p>6. El comprador y/o RAC se reservan el derecho de rechazar el PCPS y sus revisiones (ver también PECALE-2105).</p>	No Aplica (N/A)	<p>N/A. Proceso iterativo (<i>sprints</i>). Los nuevos requisitos, o modificaciones de los ya establecidos, quedarán recogidos en los <i>Product Backlogs</i>.</p>
5.3. ANÁLISIS DE CRITICIDAD DEL SOFTWARE (4.3; 7.2; 8.1; 8.3.2; 8.3.4)	<p>1. El suministrador deberá proporcionar un análisis de criticidad para cada producto software del sistema por separado y el nivel de criticidad asociado, utilizando las categorías de análisis de criticidad requeridas o derivadas del contrato y siendo consistentes con los niveles de criticidad establecidos en el proceso de adaptación (véase apartado 4.3).</p> <p>2. El análisis de criticidad se revisará y actualizará en hitos predefinidos durante el proyecto, y a medida que se conozcan nuevas definiciones y detalles del desarrollo.</p>	No Aplica (N/A)	<p>N/A. Esta información estará incluida en el <i>backlog</i> y tablero Kanban.</p>
5.4. MODELO DE CALIDAD DEL SOFTWARE (8.1; 8.3)	<p>1. El suministrador deberá identificar y proporcionar un modelo de calidad de software, para cada producto software, basado en la norma ISO/IEC 25010 u otro modelo apropiado. Este modelo de calidad se adaptará a cada producto de software según su nivel de criticidad, su análisis preliminar de riesgos y su contexto de desarrollo y uso.</p> <p>2. Los aspectos de mantenibilidad deberán ser considerados para aquellos productos software que sean susceptibles de evolucionar, como el software que requiere un mantenimiento regular.</p>	SI	<p>El análisis de criticidad se revisará:</p> <ul style="list-style-type: none"> <li>- Internamente: durante el <i>sprint planning</i> y <i>daily scrum</i> (cuando aplique),</li> <li>- Externamente: durante cada <i>sprint</i>, <i>sprint review</i> y/o retrospectiva.</li> </ul> <p>Serán el <i>Product Owner</i> y/o <i>Scrum Master</i> quienes lo actualicen en los <i>Product Backlogs</i>.</p>
		No Aplica (N/A)	<p>N/A. Las 9 características del calidad que define la norma ISO/IEC 25010 se tendrán en cuenta en los diferentes <i>sprints</i> a lo largo del proceso.</p>
		No Aplica (N/A)	<p>N/A. La metodología ágil ya contempla una evolución del producto.</p>



PECAL 2210		¿REQUISITO CAMBIA?	PROPIUESTA METODOLOGÍA AGILE (SCRUM + KANBAN)
5.5. IDENTIFICACIÓN Y REVISIÓN DE LOS REQUISITOS DEL SOFTWARE (8.2)	<p>3. El suministrador deberá identificar, con la conformidad del comprador, los medios necesarios de aseguramiento y control de calidad para el software bajo prueba. Estos podrán incluir pero no estarán limitados a:</p> <ul style="list-style-type: none"> <li>a. Mediciones estáticas o métricas registradas durante el desarrollo del software.</li> <li>b. Tablas de trazabilidad de requisitos de software vinculadas a los elementos del ciclo de vida del software.</li> <li>c. Técnicas de codificación y reglas o estándares.</li> <li>d. Inspección de artículos producidos durante el desarrollo.</li> <li>e. Métricas dinámicas basadas en pruebas.</li> <li>f. Método de aseguramiento de no regresión (p. ej., integración continua).</li> <li>g. Comprobaciones de los riesgos técnicos (p. ej., ausencia de pérdida de memoria, etc.).</li> <li>h. Revisiones de código fuente.</li> <li>i. Pruebas reforzadas con integración de casos límite y fuera de límite.</li> <li>j. Actualización e inspección de la documentación producida.</li> <li>k. Medios apropiados de Gestión de la Configuración.</li> </ul> <p>Nota: Los productos software clasificados como «críticos» tendrán umbrales más restrictivos para las métricas empleadas, la trazabilidad de los requisitos para el diseño, la codificación y las pruebas, y la cobertura de pruebas impuesta a partir de las pruebas unitarias.</p>	SI	<p>Los medios necesarios de aseguramiento y control de calidad de cada uno de los puntos, siguiendo Scrum, se facilitan a continuación:</p> <ul style="list-style-type: none"> <li>a. <i>Sprint planning, sprint review</i> y retrospectiva.</li> <li>b. Evidencia de que el trabajo está estructurado en <i>sprints</i> y de refinamiento.</li> <li>c. Técnicas de codificación y reglas o estándares.</li> <li>d. Retrospectivas y evidencia de aprendizaje y mejora continua.</li> <li>e. <i>Backlog</i> actualizado e historias de usuarios bien escritas.</li> <li>f. N/A. Scrum ya lo contempla.</li> <li>g. <i>Daily scrums</i>.</li> <li>h. Contenido en los <i>sprint</i>.</li> <li>i. Contenido en los <i>sprint</i>.</li> <li>j. <i>Sprint review</i> y retrospectiva.</li> <li>k. Medios apropiados de Gestión de la Configuración.</li> </ul>
	1. El suministrador identificará los requisitos de software y las restricciones del desarrollo.	SI	Al asumir que los requisitos evolucionan o cambian, estos, junto con las restricciones se incluirán en el tablero Kanban.
	2. Las especificaciones de los requisitos del software incluirán una definición clara y precisa de las restricciones de diseño y de las características esenciales de calidad del software.	NO	Requisito no cambia. Se mantiene el actual de PECAL 2210. Los requisitos se incluirán como historias de usuario y, cuanto más detalladas, mejor.
	3. El PCPS deberá identificar qué estándares o guías aplican al formato y contenido de las especificaciones de requisitos del software, ya sea directamente o por referencia a procedimientos o documentos.	No Aplica (N/A)	N/A.
	4. El suministrador identificará las especificaciones funcionales, de desempeño, de interfaz, de seguridad y de protección para cada producto software. Las especificaciones del software se derivarán de las especificaciones del sistema o subsistema.	SI	Incluido en Scrum.
	5. El suministrador deberá proporcionar matrices de trazabilidad para mostrar la trazabilidad de los requisitos de software a partir de las especificaciones del sistema o subsistema.	No Aplica (N/A)	N/A. Proceso iterativo ( <i>sprints</i> ).
	6. Si no se ha realizado una revisión de requisitos de software como parte del desarrollo del sistema, esta constituirá el paso inicial en el proceso de diseño y desarrollo del software y se contemplará en el PCPS.	No Aplica (N/A)	N/A. Proceso iterativo ( <i>sprints</i> ) con actualización del <i>Product Backlog</i> tras cada incremento.
	7. La revisión deberá verificar que los requisitos del software sean completos, consistentes, inequívocos, trazables, factibles y validables. Una vez finalizada la revisión de los requisitos del software, las especificaciones de los requisitos del software deberán ser aprobadas formalmente por las autoridades responsables del suministrador y estarán sujetas a Gestión de Configuración.	No Aplica (N/A)	N/A. Proceso iterativo ( <i>sprints</i> ) con actualización del <i>Product Backlog</i> tras cada incremento.



PECAL 2210		¿REQUISITO CAMBIA?	PROPIUESTA METODOLOGÍA AGILE (SCRUM + KANBAN)	
5.6. PROCESO DE DISEÑO Y DESARROLLO DEL SOFTWARE (8.3)	1. El suministrador deberá aplicar un modelo de diseño y desarrollo (modelo de ciclo de vida) que desglose el diseño en actividades de codificación, verificación y validación. Se debe documentar la justificación del modelo elegido y cualquier riesgo que presente.	No Aplica (N/A)	N/A. En metodología ágil, cada iteración o <i>sprint</i> contiene requisitos, análisis, diseño, fabricación y test.	
	2. El suministrador producirá un diseño de la arquitectura y un diseño detallado para cada producto software. El diseño deberá presentar los módulos o clases, las funciones, los datos internos y externos, y la descripción de las entradas y salidas de cada módulo.	No Aplica (N/A)	N/A. Se entregarán pequeños paquetes de SW para que Usuario, Stakeholders y Manager den feedback.	
	3. La arquitectura y el diseño detallado deberán de ser actualizados o para cada modificación correctiva o progresiva del software.	No Aplica (N/A)	N/A. Proceso iterativo ( <i>sprints</i> ) con actualización del <i>Product Backlog</i> tras cada incremento. En el <i>sprint planning</i> y <i>daily scrum</i> se analizarán también los posibles bugs.	
	4. El PCPS (ver también PECALE-2105) deberá contemplar el Desarrollo de Software teniendo en cuenta los requisitos de AS9115 (punto 8.3.2) y las adiciones siguientes: a. Una lista exhaustiva de los productos software a los que se aplica. b. El ciclo de vida del software elegido y la justificación de su uso. c. El cronograma incluyendo la planificación inicial para cada producto software con: — Fechas de inicio y finalización del desarrollo con respecto al ciclo de desarrollo del sistema. — Fechas de integración en un subsistema y/o fecha de entrega e instalación. — Revisiones o hitos y puntos clave identificados por el suministrador para el desarrollo de cada producto software.	SI	El equipo planifica y replanifica a medida que se obtiene información tras cada <i>sprint review</i> .	
	5. Cuando un elemento no sea requerido, el PCPS deberá justificar su omisión.	No Aplica (N/A)	N/A. En el <i>Product Backlog</i> se pueden eliminar funcionalidades y/o requisitos si es necesario (documento vivo)	
	6. Para productos COTS, para los que el suministrador no tiene control sobre el diseño, el PCPS deberá identificar cómo el suministrador se asegurará de que el producto cumple con los criterios de aceptación.	SI	En cada <i>sprint</i> , en la fase de test se probarán los COTS (cuando sean aplicables)	
	7. Cualquier cambio en los modelos de desarrollo, adoptado durante el proyecto, deberá quedar registrado en el PCPS.	No Aplica (N/A)	N/A.	
5.7. GESTIÓN	5.7.1. ROLES EN LA ORGANIZACIÓN, RESPONSABILIDADES Y AUTORIDADES (5.3; 7.1.2)	1. El personal que realiza las tareas específicas asignadas (subcontratados o empleados de la compañía) debe estar cualificado en relación con un nivel apropiado de estudios, formación y/o experiencia requerida; deben mantenerse los registros apropiados.	SI	Cada miembro del equipo de desarrollo, de acuerdo con la Tabla 4-2 de la Agile Practice Guide, tendrán todas las habilidades necesarias para producir un producto funcional; tendrán miembros tanto en forma de "I" (especialistas en una materia) como de "T" (especialidad central y amplia experiencia en múltiples habilidades).
		2. Se nombrará un representante con la autoridad necesaria para garantizar que se cumplan todos los requisitos de esta publicación.	SI	<i>Product Owner</i>



PECAL 2210		¿REQUISITO CAMBIA?	PROPIEDAD METODOLOGÍA AGILE (SCRUM + KANBAN)
5.7.2. GESTIÓN DE SUBSUMINISTRADORES (8.4)	Cuando lo solicite el RAC y/o comprador, se deben tomar las disposiciones para el aseguramiento oficial de la calidad en las instalaciones del subsuministrador. Cuando el RAC y/o comprador consideren que es necesario realizar por su parte una verificación/validación/evaluación de los artículos/procesos del subsuministrador, el suministrador debe incluirlo en los documentos de compra. A petición del RAC y/o comprador, se le suministrarán copias de los documentos de compra junto con los datos técnicos relevantes.	SI	El RAC podrá ser parte del equipo o un stakeholder por lo que tendrá visibilidad total de la gestión de la calidad del SW en todo momento.
5.7.3. GESTIÓN DE LA CONFIGURACIÓN DEL SOFTWARE (GCS) (8.1.2; 8.5.1; 8.5.2; 8.5.6; 8.6)	1. El suministrador debe definir e implantar un proceso de GCS que permita mantener la integridad y la trazabilidad del (de los) producto(s) software durante el desarrollo. Las actividades y procedimientos de la GCS deben asegurar que no se produzcan cambios no controlados y deben proporcionar líneas de referencia planificadas y formalmente aceptadas como condición previa para la verificación, seguimiento y control de la calidad del software.	SI	<p>Propuesta a seguir:</p> <p>Aseguramiento de que Agile se ejecuta bien de acuerdo a las siguientes actividades:</p> <ul style="list-style-type: none"> <li>— Evidencia de que el trabajo está estructurado en <i>sprints</i>.</li> <li>— Evidencia de refinamiento.</li> <li>— Evidencia de planificación de <i>sprints</i>.</li> <li>— <i>Daily scrums</i>.</li> <li>— Evidencia de ejecución del <i>Sprint</i>.</li> <li>— Retrospectivas y evidencia de aprendizaje y mejora continua.</li> <li>— Evidencia de colaboración con el cliente (<i>sprint review</i>).</li> </ul> <p>Y en términos de entregables:</p> <ul style="list-style-type: none"> <li>— <i>Backlog</i> priorizado</li> <li>— Historias de usuarios bien escritas.</li> <li>— Evidencia de planificación y seguimiento del progreso (tablero Kanban).</li> </ul>



PECAL 2210		¿REQUISITO CAMBIA?	PROPIUESTA METODOLOGÍA AGILE (SCRUM + KANBAN)
	<p>2. El suministrador, específicamente, debe definir e implantar:</p> <ul style="list-style-type: none"> <li>a. Procedimientos para identificar, nombrar y registrar las características físicas, funcionales y de la calidad de los elementos intermedios y finales que serán controlados (p. ej., documentación, código ejecutable, código fuente, listado de programas, bases de datos, especificaciones, casos de prueba, planes) y sus estructuras en cada punto de control del proyecto. Los elementos del entorno de desarrollo y soporte (compiladores, herramientas de desarrollo, sistemas operativos, condiciones de pruebas) también deben formar parte de la estructura del elemento de configuración del software (ECS).</li> <li>b. Procedimientos para solicitar, evaluar, aprobar/rechazar e implantar los cambios (corrección de errores y mejora) de los ECS de referencia.</li> <li>c. Procedimientos para registrar e informar del estado de los ECS del proyecto.</li> <li>d. Auditorías y revisiones para determinar hasta qué punto los ECS reflejan los requisitos físicos, funcionales y las características de calidad, y para establecer una línea de referencia.</li> <li>e. Procedimientos para controlar las interfaces de los ECS del proyecto con los elementos que no forman parte del alcance directo del desarrollo del software (sistema, hardware, factor humano, software de apoyo).</li> <li>f. Procedimientos para coordinar los cambios realizados en los elementos de software desarrollados externamente e incorporar estos cambios al proyecto.</li> </ul> <p>3. Los cambios en las especificaciones de requisitos del software deben evaluarse por el impacto en coste, técnico y en el plazo de entrega, y deben ser comunicados a todas las partes afectadas. Los cambios que afecten a la funcionalidad solo deben implementarse con la aprobación del comprador.</p>	SI	<p>a. Se realiza a través de las actualizaciones del Product Backlog y los diferentes PBIs.</p> <p>b. Bugs o errores (PBI).</p> <p>c. Internamente (durante el <i>sprint planning</i> y <i>daily scrum</i>) y Externamente (durante cada <i>sprint</i>, <i>sprint review</i> y/o <i>retrospectiva</i>)</p> <p>d. Es un proceso iterativo en el que en cada sprint se corrigen bugs, se definen los nuevos objetivos y las tareas que se van a realizar (generalmente sobre el ECS).</p> <p>e. N/A.</p> <p>f. Se realiza a través de las actualizaciones del <i>Product Backlog</i>.</p>
		SI	La revisión se hará en los diferentes <i>sprint reviews</i> .



PECAL 2210		¿REQUISITO CAMBIA?	PROPIEDAD METODOLOGÍA AGILE (SCRUM + KANBAN)
	<p>4. El suministrador deberá establecer trazabilidad bidireccional entre los siguientes elementos:</p> <ul style="list-style-type: none"> <li>a. Requisitos del sistema/requisitos software/requisitos de componentes software.</li> <li>b. Requisitos software/pruebas de validación/conjuntos de pruebas de validación/resultados de pruebas.</li> </ul>	No Aplica (N/A)	N/A. Proceso iterativo que se realiza a través de las actualizaciones del Product Backlog y los diferentes PBIs.
	<p>5. Esta trazabilidad deberá quedar documentada en una matriz y/o en una herramienta específica.</p> <p>Nota: Este requisito es aplicable a todo el software. En el caso de COTS y Software Parcialmente Modificado, parte de la trazabilidad puede no ser posible o solo serlo parcialmente.</p>	SI	La trazabilidad se hará a través de las diferentes actualizaciones del <i>Product Backlog</i> .
	<p>6. El suministrador también debe identificar las herramientas del software, técnicas y equipos necesarios para implantar las actividades de la GCS y asignar las responsabilidades y autoridades para las actividades de la GCS a los organismos y personas que forman parte de la estructura del proyecto.</p>	SI	Las responsabilidades están definidas en la sección 4.3.1.1.
	<p>7. El suministrador deberá proporcionar, para cada revisión, a partir de la fase de validación interna, y para cada entrega, un informe de configuración de la versión del software, que incluya al menos lo siguiente:</p> <ul style="list-style-type: none"> <li>a. Identificación completa y única de la versión del software.</li> <li>b. Identificación de los medios de entrega;</li> <li>c. Lista de datos aplicables: especificaciones y documentos de interfaz, solicitudes de actualización.</li> <li>d. Procedimiento para generar código ejecutable y los archivos de configuración si procede (instrucciones de compilación, enlaces, etc.).</li> <li>e. Documentación asociada producida durante el ciclo de desarrollo;</li> <li>f. Identificación completa de los elementos de configuración para el producto software entregado (incluyendo COTS); los elementos que hayan cambiado de una versión a la siguiente deberán ser fácilmente identificables.</li> <li>g. Solicitudes de desviación o concesión con su estado (en curso, aceptado).</li> <li>h. Lista de fallos/no conformidades corregidas desde la versión anterior.</li> </ul>	SI	El suministrador proporcionará incrementos (parte añadida o desarrollada en un sprint considerada parte terminada y totalmente operativa, es decir, es la suma total de los elementos del Product Backlog completados en el sprint actual teniendo en cuenta los sprints anteriores) hasta conseguir el producto final. Tras cada incremento, se realiza un <i>sprint review</i> con el usuario, stakeholders y manager en el que se revisa el <i>sprint</i> y se realiza una inspección y adaptación del producto tomando las decisiones pertinentes sobre el mismo y definir futuras acciones.



PECAL 2210		¿REQUISITO CAMBIA?	PROPIEDAD METODOLOGÍA AGILE (SCRUM + KANBAN)
	<p>i. Lista de fallos/no conformidades en curso con su clasificación (mayor bloqueante, mayor, menor).</p> <p>8. La práctica de parchear el software debe restringirse a situaciones muy excepcionales y no se podrá hacer sin el conocimiento y acuerdo del comprador. El control de la configuración de parches debe estar establecido en un procedimiento específico. Cualquier parche tendrá carácter temporal y deberá incluirse en la versión definitiva, con datos de ciclo de vida actualizados. El suministrador deberá proporcionar un informe de configuración de parches que incluya:</p> <ul style="list-style-type: none"> <li>a. El estudio de impacto al que se hace referencia para el incidente técnico (descripción de las actualizaciones a realizar, a los datos del ciclo de vida de origen, para obtener un producto de software idéntico al software parcheado).</li> <li>b. Identificación del parche en la versión del software.</li> <li>c. Lista de fallos/no conformidades corregidas.</li> <li>d. Procedimiento de instalación.</li> <li>e. Lista de pruebas realizadas para validar el parche, incluidas las pruebas de no regresión.</li> <li>f. Lista exhaustiva de elementos de configuración de software contenidos en el parche.</li> </ul>	SI	Se acordará con el Cliente y se incluirá, cuando aplique, en el tablero Kanban.



PECAL 2210		¿REQUISITO CAMBIA?	PROPIUESTA METODOLOGÍA AGILE (SCRUM + KANBAN)
5.7.4. SOFTWARE PREEEXISTENTE (OFF-THE-SHELF) (8.4)	1. Si el suministrador utiliza software preexistente, debe asegurar que:	SI	Se revisará durante el <i>sprint planning</i> y/o <i>daily scrum</i> .
	a. Su uso no está afectado por ningún derecho de protección de datos (propiedad intelectual).		
	b. Existe evidencia objetiva, antes de su uso, de que el software cumplirá las funciones requeridas.		
	c. El software está sujeto a gestión de la configuración.		
5.7.5. INFRAESTRUCTURA (7.1.3)	d. El software está documentado de acuerdo con los requisitos del contrato y de esta publicación.		
	2. Si el software preexistente es entregable y se modifica durante el proceso de desarrollo debe, en ese caso, tratarse como software de desarrollo y está sujeto a los requisitos de esta publicación.	SI	Se seguirá el flujo nominal que dicta scrum.
	3. Si el suministrador considera que el software preexistente proporcionado por el comprador no es aceptable para el uso, debe informar al comprador lo más pronto posible de las razones de su rechazo y debe negociar con él las acciones a tomar para remediarlo.	No Aplica (N/A)	N/A. Si el SW no es aceptable para su uso, se desecha (PBIs)
	4. El suministrador debe informar al comprador cuando se prevea incorporar software preexistente en un producto software.	SI	Se incluirá en el incremento.
5.7.6. REVISIÓN POR LA DIRECCIÓN (9.3)	El PCPS deberá documentar la determinación de la infraestructura apropiada para soportar las actividades de software requeridas por el contrato, ya sea directamente o por referencia a procedimientos y documentos.	SI	El PCPS incluirá la descripción de la metodología scrum y tablero kanban, de manera genérica.
	La revisión por la Dirección del sistema de gestión de la calidad incluirá los aspectos específicos de software del sistema de gestión de la calidad.	SI	No se realizará revisión por la Dirección como tal, el Product Owner estará involucrado en el <i>sprint planning</i> , <i>daily scrum</i> , <i>sprint review</i> y retrospectiva.



PECAL 2210		¿REQUISITO CAMBIA?	PROPIEDAD METODOLOGÍA AGILE (SCRUM + KANBAN)	
5.8. EVALUACIÓN, VERIFICACIÓN Y VALIDACIÓN (EVV)	5.8.1. GENERALIDADES	<p>1. El suministrador debe planificar, definir e implantar:</p> <ul style="list-style-type: none"> <li>a. Un proceso para la evaluación de los métodos, técnicas, procedimientos, herramientas y actividades relacionadas con el software.</li> <li>b. Un proceso para la verificación y validación de los elementos y productos software.</li> <li>c. Un proceso que permita efectuar las acciones de seguimiento que aseguren que se realizan los cambios necesarios.</li> <li>d. Un proceso para determinar el nivel requerido de reevaluación en caso de corrección de errores o cambios incorporados, tanto a los requisitos como al diseño.</li> </ul> <p>2. El proceso de EVV debe definir:</p> <ul style="list-style-type: none"> <li>a. Las actividades de EVV y su secuencia en relación con las fases, hitos y programación en el tiempo.</li> <li>b. Los perfiles organizativos, responsabilidades y autoridad para la ejecución de las actividades de EVV.</li> <li>c. Los objetos de la EVV (p. ej., documentos de requisitos/desarrollos, productos software, procesos de desarrollo, métodos, procedimientos, código fuente, código ejecutable).</li> <li>d. Los criterios para realizar la EVV.</li> <li>e. Los métodos, normas, técnicas, herramientas e instalaciones específicos de EVV.</li> <li>f. Los tipos de métodos de EVV a utilizar (p. ej., prueba, revisión, auditoría).</li> <li>g. La documentación de EVV a generar (planes y procedimientos específicos, registros e informes de EVV).</li> </ul> <p>3. Como una parte integrada en el proceso de EVV, el suministrador debe desarrollar/seleccionar e implantar medidas cuantitativas y/o cualitativas para evaluar/verificar/validar las características de calidad del software descritas en las especificaciones de requisitos o en el PCPS.</p>	SI	<p>En cada sprint, se contempla el proceso de EVV:</p> <p><b>1- Definición del alcance de la prueba:</b> Se identifican las características y funciones a probar en cada <i>sprint</i>, alineadas con las historias de usuario y criterios de aceptación. Se consideran riesgos, dependencias y suposiciones, y se prioriza el alcance de manera clara y concisa, revisándolo regularmente.</p> <p><b>2- Elección de los métodos de prueba:</b> Se seleccionan técnicas y herramientas de prueba según el alcance, tipo de software, recursos y estándares de calidad. Los métodos pueden incluir pruebas manuales, automatizadas, unitarias, de integración, de sistema, de rendimiento, de usabilidad y de seguridad, equilibrando aspectos funcionales y no funcionales.</p> <p><b>3- Planificación de las actividades de prueba:</b> Se planifican tareas y responsabilidades en colaboración con el equipo, alineadas con los objetivos del <i>sprint</i>. Las actividades se estiman, programan y monitorean usando el tablero Kanban, revisándose regularmente.</p> <p><b>4- Definición del entorno de prueba:</b> Se configura el hardware y software para ejecutar las pruebas, asegurando consistencia y fiabilidad con el entorno de producción. El entorno debe ser accesible, manejable y compatible con los métodos y herramientas de prueba, verificándose antes de cada sesión.</p> <p><b>5- Establecer los criterios de prueba:</b> Se definen métricas y medidas para evaluar la calidad y resultados del proceso de prueba, basadas en historias de usuario, criterios de aceptación y estándares de calidad. Los criterios incluyen número de casos de prueba, cobertura, densidad de defectos, resolución de defectos, tiempo de ejecución y satisfacción del usuario.</p> <p><b>6- Revisión y actualización del plan de pruebas:</b> Se verifica y modifica el plan de prueba según cambios y comentarios durante el proyecto, revisándolo al final de cada <i>sprint</i> y ante cambios significativos en requisitos, alcance, métodos, entorno o criterios. El plan debe ser flexible y reflejar el estado y objetivos del proyecto.</p>



PECAL 2210		¿REQUISITO CAMBIA?	PROPIUESTA METODOLOGÍA AGILE (SCRUM + KANBAN)	
	<p>4. También deben aplicarse medidas cuantitativas/cualitativas (métricas) para la gestión y control del proceso de desarrollo software para el producto software objeto del contrato. Tales medidas deben permitir la identificación del nivel real de prestaciones obtenido, la toma de acciones correctivas y el establecimiento de objetivos de mejora.</p> <p>5. El personal que realice evaluaciones, verificaciones y validaciones de la calidad del software deberá contar con recursos, responsabilidad, autoridad y experiencia técnica. También deberán tener un nivel adecuado de independencia orgánica de la(s) persona(s) que desarrolle(n) el producto software o realicen la actividad objeto de la evaluación/verificación/validación para actuar con objetividad y promover el inicio de las correspondientes acciones correctivas.</p> <p>6. El suministrador debe ser consciente de que las actividades de EVV del RAC y/o comprador no constituirán una aceptación, ni reemplazarán de ninguna manera las actividades de EVV por parte del suministrador ni liberarán al suministrador de sus responsabilidades contractuales.</p>			
5.8.2. PRUEBAS (8.3.4)	<p>1. Como una parte integrada del proceso de EVV, el suministrador debe planificar, definir e implantar un programa de pruebas. Se deben tener en cuenta:</p> <p>a. Las pruebas de los elementos de software, de integración, de sistema y de aceptación del software.</p> <p>b. El entorno, las herramientas y el software de pruebas.</p> <p>c. Calificación de las herramientas de prueba de software y banco de pruebas.</p> <p>d. La documentación del usuario. e. El personal requerido y la formación asociada.</p> <p>2. El suministrador debe llevar a cabo una revisión de los requisitos y de los criterios de prueba para verificar la idoneidad, viabilidad, trazabilidad y la ausencia de ambigüedad. Las especificaciones de prueba deben incluir los escenarios de prueba, los datos requeridos para las pruebas y los resultados esperados.</p>	SI	Ver 5.8.1	
		SI	Ver 5.8.1	



PECAL 2210		¿REQUISITO CAMBIA?	PROPIUESTA METODOLOGÍA AGILE (SCRUM + KANBAN)
	<p>3. El suministrador debe definir e implantar medidas para controlar actividades de las pruebas, que incluyan:</p> <ul style="list-style-type: none"> <li>a. Cuando sea necesario, el establecimiento, documentación y verificación de la configuración del software a probar, junto con cualquier hardware asociado.</li> <li>b. La actualización de la documentación relativa a las pruebas que permita su repetitividad.</li> <li>c. La confirmación de que las pruebas se realizan de acuerdo con los planes, las especificaciones y los procedimientos aprobados.</li> <li>d. Disposiciones que permitan certificar que los resultados de la prueba son reales y válidos.</li> <li>e. Disposiciones para la revisión y certificación de los informes de prueba.</li> </ul> <p>4. El suministrador debe informar al RAC y/o Comprador de las dificultades inusuales encontradas durante las pruebas.</p>	SI	Ver 5.8.1
	<p>5.8.3. REVISIONES (8.2.3; 8.3.4)</p> <p>1. Los procedimientos de revisión deben incluir disposiciones para:</p> <ul style="list-style-type: none"> <li>a. Describir los objetivos de cada revisión.</li> <li>b. Identificar las funciones, autoridades y responsabilidades de personal involucrado en las revisiones.</li> <li>c. Registrar los resultados de las revisiones.</li> <li>d. Asegurar que las acciones resultantes de las revisiones se supervisan a fin de garantizar su finalización en plazo.</li> </ul> <p>2. Toda la documentación del software generada bajo el contrato debe, antes de su publicación, ser revisada y aprobada en cuanto a su adecuación por el personal autorizado.</p>	SI	Revisión en el sprint review.
	<p>1. Si después de la entrega inicial e instalación el mantenimiento del software es un requisito especificado, el suministrador debe definir e implantar los procedimientos para realizar esta actividad. El suministrador deberá proporcionar, antes de que comience la fase de mantenimiento, un Plan de Mantenimiento (PM) que aborde todos los requisitos definidos en este capítulo. El PM puede ser un documento independiente o parte de otro plan que se prepare bajo el contrato.</p> <p>2. Estos procedimientos deben incluir las disposiciones para verificar e informar que el mantenimiento realizado satisface los requisitos especificados.</p>	SI	Incluido en la sprint review y retrospectiva. No se entregará un PM como tal.
5.9. MANTENIMIENTO		SI	La verificación de que el mantenimiento realizado satisface los requisitos especificados se evidenciará en el sprint siguiente y por ende, sprint review y retrospectiva.



PECAL 2210	¿REQUISITO CAMBIA?	PROPIUESTA METODOLOGÍA AGILE (SCRUM + KANBAN)
<p>3. Se debe tener en cuenta:</p> <ul style="list-style-type: none"> <li>a. El trabajo a realizar.</li> <li>b. Los procedimientos a emplear.</li> <li>c. Los registros e informes a generar.</li> <li>d. Las responsabilidades del suministrador y su interfaz con el RAC y/o Comprador;</li> <li>e. Las actividades de gestión de configuración, incluida la identificación del estado inicial del producto a mantener.</li> <li>f. Los métodos para el tratamiento de problemas, su análisis y resolución.</li> <li>g. Las pruebas y aceptación de los cambios.</li> </ul>	SI	<ul style="list-style-type: none"> <li>a. <i>Sprint planning</i> -&gt; <i>Product Backlog</i> -&gt; PBI.</li> <li>b. Diferentes tareas definidas en el <i>Product Backlog</i> y tablero Kanban.</li> <li>c. Incrementos.</li> <li>d. N/A. Definido desde el principio.</li> <li>e. <i>Sprint planning</i> -&gt; <i>Product Backlog</i> -&gt; PBI.</li> <li>f. PBI -&gt; <i>Bugs</i>.</li> <li>g. <i>Normal work</i> en cada <i>sprint</i>.</li> </ul>



## **Anexo III Análisis de Riesgos: implantación de SCRUM y KANBAN en una organización tradicional desde el punto de vista de la gestión y del aseguramiento de la calidad**

El enfoque de gestión de riesgos, y más específicamente el de las acciones resultantes, debe ser un compromiso entre el riesgo general para los objetivos del proyecto y los recursos y el presupuesto disponibles. Esto significa que es esencial utilizar los recursos disponibles de la manera más rentable posible para mantener el riesgo del proyecto dentro de niveles aceptables.

Para aprovechar al máximo los recursos disponibles de manera rentable, se considerarán los siguientes aspectos:

- Comunicación clara con los miembros del equipo: Se comunicará de manera clara y en sesiones dedicadas el enfoque de gestión de riesgos y los canales para comunicar y evaluar los riesgos.
- Medidas preventivas: Se aplicarán medidas preventivas siempre que sea posible, especialmente desde niveles más bajos hacia arriba, para aislar los riesgos.
- Detección temprana de riesgos: Se proporcionarán medios para detectar los riesgos lo antes posible mediante encuestas periódicas. El tema de riesgo formará parte de las *daily scrum* y los *sprint review*.
- Asignación de márgenes: Se asignarán márgenes tanto como sea posible en los aspectos de diseño y programación, como el Schedule y costes.
- Ánalisis de riesgos como herramienta de toma de decisiones: Se utilizará el análisis de riesgos como una herramienta para la toma de decisiones, y priorización de PBI, en las *daily scrums*, en los compromisos que afectan el diseño, el rendimiento, el cronograma, las pruebas, etc.

### Categorías de riesgos

Los riesgos están asociados a las siguientes categorías:

- Organizativo: riesgos asociados a la posible dificultad del equipo para cumplir con las especificaciones técnicas o los requisitos del contrato. En esta categoría, están incluidos los riesgos asociados al cumplimiento de la planificación y costes.
- Calidad: riesgos relacionados con la gestión de la calidad y la calidad propia de los sistemas a fabricar.

### Estrategias de mitigación de riesgos

Los riesgos estarán asociados a las siguientes estrategias de mitigación de riesgos predefinidas:



- Evitar: No utilizar esta opción (rediseñar).
- Ignorar: Aceptar las consecuencias si ocurren.
- Contener: Tomar medidas específicas para minimizar la gravedad y/o probabilidad.
- Contingencia: Reservar fondos para usar cuando ocurra el riesgo o cuando se considere apropiado contenerlo más adelante.
- Transferir: Transferir todo o parte del riesgo a otra parte.

#### Clasificación de Riesgo, Probabilidad, Impacto y Criterios de Aceptación

Para cada riesgo identificado, se establecerá un impacto y una probabilidad de ocurrencia.

El impacto se definirá en función de los requisitos, los criterios de éxito/aceptación, los recursos y las restricciones de costes/programa. En cuanto a las clasificaciones de probabilidad, estas tienen como objetivo proporcionar una estimación de orden de magnitud basada en datos cuantitativos disponibles y experiencia cualitativa.

La Figura 19 muestra el esquema de puntuación utilizado para el análisis de riesgos del proyecto.

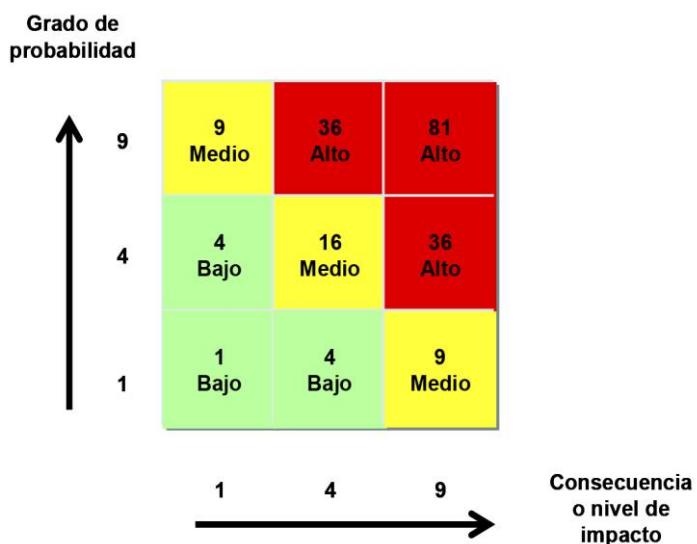


Figura 19 Anexo III: Matriz de índice de riesgo  
Fuente: Master Adquisiciones de Sistemas de Defensa (Módulo 5)



### Identificación de los riesgos: Responsables y Fechas

Cada riesgo identificado apunta a un responsable. En según qué riesgos, se puede contar también con una o más personas co-responsables o que participen en la ejecución de acciones de mitigación.

Y todos los riesgos deben tener fechas:

- Fecha planificada: siendo esta como la fecha de inicio prevista para la implementación de las medidas de mitigación.
- Fecha de realización: entendida como la fecha prevista de finalización o cierre del riesgo.

### Descripción del proceso de gestión de riesgos:

El proceso de Gestión de Riesgos es una secuencia iterativa de cuatro pasos; cada paso tiene las siguientes tareas.

PASO 1: Definir los requisitos de implementación de la gestión de riesgos

Tarea 1: Definir la política de gestión de riesgos

Tarea 2: Preparar el plan de gestión de riesgos

PASO 2: Identificar y evaluar los riesgos

Tarea 3: Identificar escenarios de riesgo

Tarea 4: Evaluar los riesgos

PASO 3: Decidir y actuar

Tarea 5: Decidir si los riesgos pueden ser aceptados

Tarea 6: Reducir los riesgos

Tarea 7: Recomendar aceptación

PASO 4: Seguimiento, comunicación y aceptación de los riesgos

Tarea 8: Seguimiento y comunicación de los riesgos

Tarea 9: Presentar el riesgo para su aceptación



ID	Descripción riesgo	Categoría riesgo	Causa del riesgo	Impacto (1,4,9)	Probabilidad (1,4,9)	Clase riesgo	Efectos riesgo	Medida	Clase riesgo tras implementar medida	Responsable	Fecha planificada	Fecha realización	Status
1	Resistencia al cambio por parte de los líderes y directivos de la empresa	Organizativo	Falta de conocimiento de las metodologías SCRUM y KANBAN, así como los beneficios que aportan a la organización.	4	4	16	Planning: -Retraso en la ejecución de los sprints -Retraso en las entregas (incrementos)	Realizar formaciones y talleres de capacitación sobre SCRUM y KANBAN.	4	CEO	A definir por la organización	A definir por la organización	Open
2	Resistencia al cambio por parte de los empleados (departamentos de ingeniería, calidad, desarrolladores)	Organizativo	Falta de conocimiento de las metodologías SCRUM y KANBAN, así como los beneficios que aportan a la organización.	9	9	81	Planning: -Retraso en la ejecución de los sprints -Retraso en las entregas (incrementos)	Realizar formaciones y talleres de capacitación sobre SCRUM y KANBAN.	9	Director de división	A definir por la organización	A definir por la organización	Open
3	Implementación ineficaz de SCRUM	Organizativo	Falta de experiencia trabajando en un entorno Agile	9	9	81	Coste: -Aumento de horas estimadas en la realización de los sprints. Planning: -Retraso en la ejecución de los sprints -Retraso en las entregas (incrementos)	Contratar o formar a un Scrum Master certificado y proporcionar formación continua al equipo.	4	Product Owner	A definir por la organización	A definir por la organización	Open
4	Implementación ineficaz de KANBAN	Organizativo	Sin una correcta implementación, puede haber falta de visibilidad sobre el progreso y los bloqueos.	4	9	36	Planning: -Retraso en la ejecución de los sprints -Retraso en las entregas (incrementos)	1- Utilizar tableros Kanban bien diseñados (ejemplo en sección 4.3.2) 2- Realizar revisiones periódicas para identificar y resolver bloqueos (Daily Scrum)	4	Scrum Master	A definir por la organización	A definir por la organización	Open



ID	Descripción riesgo	Categoría riesgo	Causa del riesgo	Impacto (1,4,9)	Probabilidad (1,4,9)	Clase riesgo	Efectos riesgo	Medida	Clase riesgo tras implementar medida	Responsable	Fecha planificada	Fecha realización	Status
5	Comunicación ineficaz entre los miembros del equipo de desarrollo	Organizativo	Falta de claridad en los roles y responsabilidades y/o ausencia de canales de comunicación adecuados	4	1	4	Planning: -Retraso en la ejecución de los sprints -Retraso en las entregas (incrementos) Calidad: -Entregables con estándares de calidad bajos. Equipo Agile: -Frustración y desmotivación que desencadene en una pérdida de talento.	Fomentar la transparencia y la comunicación abierta mediante reuniones diarias (Daily Scrum) y retrospectivas.	1	Scrum Master	A definir por la organización	A definir por la organización	Open
6	Identificación tardía de riesgos	Calidad	Debido a la naturaleza iterativa de SCRUM puede que algunos no se identifiquen a tiempo.	9	4	36	-Impacto en coste y planning. -Pérdida de confianza de los stakeholders y/o clientes.	Implementar un proceso de gestión de riesgos continuo y proactivo, revisando y ajustando regularmente (sprint planning, sprint review)	9	Product Owner	A definir por la organización	A definir por la organización	Open
7	Equipos de desarrollo con sobrecarga de trabajo	Organizativo	No priorizar correctamente las tareas y asumir más de las que se pueden manejar	9	1	9	Planning: -Retraso en la ejecución de los sprints -Retraso en las entregas (incrementos) Calidad: -Disminución de la calidad en los incrementos. Equipo Agile: -Agotamiento y desmotivación que afecte al rendimiento y productividad.	Establecer límites de trabajo en progreso (WIP) claros y respetarlos estrictamente.	1	Scrum Master	A definir por la organización	A definir por la organización	Open



ID	Descripción riesgo	Categoría riesgo	Causa del riesgo	Impacto (1,4,9)	Probabilidad (1,4,9)	Clase riesgo	Efectos riesgo	Medida	Clase riesgo tras implementar medida	Responsable	Fecha planificada	Fecha realización	Status
8	Adaptación a los cambios continuos	Organizativo	La flexibilidad de Kanban puede hacer que los equipos se desvíen de los objetivos principales.	4	4	16	-Impacto en coste y planning.	Mantener un enfoque claro en los objetivos del proyecto y ajustar las políticas de Kanban según sea necesario.	1	Scrum Master / Product Owner	A definir por la organización	A definir por la organización	Open
9	Insatisfacción del cliente	Calidad	Expectativas del cliente no alineadas con la metodología y resultados; ausencia de feedback; entregables que no cumplen con los estándares de calidad entregados; retrasos en los incrementos y sprint reviews.	9	1	9	Possible pérdida de contratos actuales y futuros	Analizar feedback de los managers, usuarios y stakeholders en cada sprint review.	1	Product Owner	A definir por la organización	A definir por la organización	Open
10	Escapes o fugas de calidad en los entregables	Calidad	La naturaleza iterativa de SCRUM puede llevar a diferencias en el criterio de calidad de los entregables entre sprints.	4	4	16	Planning: -Retraso en la ejecución de los sprints -Retraso en las entregas (incrementos) Calidad: -Entregables con estándares de calidad bajos. Costes: -Aumento de horas estimadas en la realización de los sprints.	Implementar revisiones de calidad al final de cada sprint y establecer criterios de aceptación claros	1	Responsable de Calidad	A definir por la organización	A definir por la organización	Open



ID	Descripción riesgo	Categoría riesgo	Causa del riesgo	Impacto (1,4,9)	Probabilidad (1,4,9)	Clase riesgo	Efectos riesgo	Medida	Clase riesgo tras implementar medida	Responsable	Fecha planificada	Fecha realización	Status
11	Falta de documentación a entregar en cada incremento	Calidad	SCRUM puede enfocarse más en la entrega rápida que en la documentación detallada.	4	4	16	<p>Planning:</p> <ul style="list-style-type: none"> <li>-Retraso en la ejecución de los sprints</li> <li>-Retraso en las entregas (incrementos)</li> </ul> <p>Calidad:</p> <ul style="list-style-type: none"> <li>-Entregables con estándares de calidad bajos.</li> </ul> <p>Costes:</p> <ul style="list-style-type: none"> <li>-Aumento de horas estimadas en la realización de los sprints.</li> </ul>	Definir desde el inicio la documentación a entregar en cada sprint e incremento.	1	Scrum Master	A definir por la organización	A definir por la organización	Open
12	Dependencia del Scrum Master	Calidad	La calidad del proceso puede depender en gran medida de la habilidad y experiencia del Scrum Master	9	1	9	Variabilidad en la calidad del proceso y posibles problemas si el Scrum Master no es competente.	Capacitar continuamente al Scrum Master y tener un plan de contingencia para su reemplazo	1	Product Owner	A definir por la organización	A definir por la organización	Open
13	Falta de estandarización de los procesos	Calidad	Tanto SCRUM como KANBAN son muy flexibles a la hora de trabajar lo que puede llevar a no estandarizar los procesos ya que las iteraciones son muy cortas y, en algunos casos, los procesos no son repetitivos.	4	4	16	Variaciones en la calidad de los productos y procesos ineficientes.	Definir políticas claras y estándares de calidad para todos los procesos	4	Responsable de Calidad	A definir por la organización	A definir por la organización	Open



## **FIN DEL DOCUMENTO**