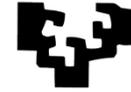




Ingeniaritza Goi Eskola Teknikoa  
Escuela Técnica Superior de Ingeniería  
Bilbao

eman ta zabal zazu



Universidad del País Vasco  
Euskal Herriko Unibertsitatea



**EURO MPM**  
**European Master**  
**in Project Management**

Trabajo de Fin de Master

***“Título del Trabajo de Fin de Master”***

**Autor: Imanol Iriondo Alzola**

**Director: Juan Ramón Arraibi Dañobeitia**

**CURSO 2021-22**

**Director**  
<Arribi, Dañobeitia>

**Autor**  
<Iriondo, Alzola>

**Curso Académico**  
2021/2022

## Resumen

Las metodologías ágiles se están convirtiendo en el modelo a seguir en cuanto a la gestión de los proyectos, esto es debido al éxito obtenido con una implementación adecuada de las metodologías. Originalmente la aplicación de este tipo de metodologías estaba enfocada a equipos pequeños capaces de desarrollar un proyecto de software de manera independiente. Debido al éxito adquirido en estos entornos, las grandes organizaciones están empezando a adoptar *Agile*.

El Trabajo Fin de Master analiza el escalado de las metodologías ágiles en grandes organizaciones. Primero se analizan dos marcos de escalado completos; *Large Scale Scrum* y *Scaled Agile Frameworks*. La segunda parte del Trabajo Fin de Master hace un análisis de los artículos publicados de la adopción de *Agile* en grandes organizaciones. El análisis se ha centrado en identificar los factores de éxito y retos que tuvieron las empresas. Por último, se ha analizado el proceso de implementación del modelo *Agile* en una gran organización actual. En base a los resultados de la segunda parte se han identificado mejoras al proceso de adopción.

Palabras Clave: *Agile*, Gestión de Proyectos, escalado, grandes organizaciones.

## Laburpena

Gaur egun proiektuen kudeaketarako *Agile* metodologiak eredu bilakatzen ari dira, hauen erabilera egoki baten ondorioz lorturiko arrakastagatik gero eta famatuagoak dira. *Agile* modeloaren lehen aplikazioak *software* garapeneko enpresetan egin ziren, hauek talde eta proiektu txikietan aplikatzen ziren. Aplikazio horietan lorturiko emaitzen ondorioz, enpresa eta proiektu handietan aplikatu nahi da *Agile* gaur egun.

Master Amaierako Lanak *Agile* metodologiaren eskalatzea aztertzen du erakunde handietan erabili ahal izateko. Lehen ata *Lean* bi eskalatze eredu aztertzen dira; *Large Scale Scrum* eta *Scaled Agile Frameworks*. Master Amaierako Lanaren bigarren zatia erakunde handietan *Agile* modeloa aplikatu izanaren inguruko artikulua aztertzen ditu. Analisi honen helburua, aurre egin beharreko erronka eta arrakasta faktoreak identifikatzea izan da. Azkenik, *Agile* eredu egungo erakunde handi batean ezartzeko prozesua aztertu da. Bigarren zatiko emaitzetan oinarrituta, adopzio prozesuan hobekuntzak identifikatu dira.

Hitz klabeak: *Agile*, Proiektuen kudeaketa, eskalazioa, erakunde handiak

## Summary

*Agile* methodologies are becoming the model to follow in terms of project management; this is due to the success obtained with an adequate implementation of the methodologies. Originally, the application of this type of methodology was focused on small *teams* capable of developing a software project independently. Due to the success achieved in these environments, large organizations are beginning to adopt *Agile*.

The Master's Thesis analyses the scaling of *Agile* methodologies in large organizations. Two complete scaling frameworks are first analysed; *Large Scale Scrum* and *Scaled Agile Frameworks*. The second part of the Master's Final Project analyses the articles published on the adoption of *Agile* in large organizations. The analysis has focused on identifying the success factors and challenges faced by the companies. Finally, the implementation process of the *Agile* model in a large current organization has been analysed. Based on the results of the second part, improvements to the adoption process have been identified.

Keywords: *Agile*, Project Management, Scaling, Large organizations

## Índice

Resumen .....	i
Laburpena .....	ii
Summary.....	iii
Índice .....	iv
Índice de imágenes y tablas .....	vi
Glosario.....	vii
1 Introducción .....	8
2 Objetivos del Trabajo Fin de Master .....	9
3 Desarrollo y plan del TFM.....	9
4 Metodologías para escalado de <i>Agile</i> .....	12
4.1 <i>Scaled Agile Framework</i> .....	12
4.1.2 Principios de SAFe.....	14
4.1.3 Nivel <i>Essential</i> .....	16
4.1.4 Nivel de <i>Portfolio</i> .....	18
4.2 Método <i>Large Scale Scrum</i> (LeSS).....	19
4.2.1 Principios de LeSS .....	19
4.2.2 Variante LeSS.....	21
4.2.3 LeSS Huge .....	23
4.3 Resumen de SAFe y LeSS .....	25
5 Método de investigación .....	26
5.1 Métodos ágiles .....	28
5.2 Prácticas de escalado .....	29
5.2.1 <i>Scrum of Scrums</i> .....	30
5.2.2 <i>Scaled Sprint Demo</i> .....	30
5.2.3 <i>Scaled Sprint Planning</i> .....	31
5.2.4 Scaled Requirement Management.....	31
5.2.5 <i>Feature Teams</i> .....	32
5.2.6 <i>Undone Department</i> .....	33
5.2.7 Comunidades de práctica .....	34
5.3 Desafíos .....	34
5.3.1 Resistencia al cambio .....	35



5.3.2	Problemas de Calidad.....	36
5.3.3	Integración con las partes no ágiles de la organización .....	36
5.3.4	Compromiso y de trabajo en equipo .....	36
5.3.5	Presión elevada o sobrecarga de trabajo.....	37
5.3.6	Falta de conocimientos, Coaching o formación.....	38
5.3.7	Scaled Requirement Management.....	39
5.3.8	Medición del progreso .....	39
5.4	Factores de éxito .....	39
5.4.1	Formación.....	40
5.4.2	Visión conjunta de los valores y las prácticas.....	41
5.4.3	Herramientas e infraestructura .....	41
5.4.4	Prácticas de ingeniería solidas .....	41
5.4.5	Precaución en la transformación.....	42
5.4.6	Apoyo al trabajo en equipo .....	42
5.4.7	Patrocinio ejecutivo .....	42
6	Estudio de la adopción de las metodologías Ágiles en un grupo empresarial..	43
7	Conclusiones .....	51
8	Referencias.....	52

## Índice de imágenes y tablas

Figura 1: Niveles de la metodología SAFe.....	13
Figura 2: Ejemplo de un Agile Release Train (ART).....	16
Figura 3: Estructura organizativa de variante LeSS.....	22
Figura 4: Estructura organizativa LeSS Huger.....	24
Figura 5: Ejemplo de un proceso de mapeo sistemático .....	26
Figura 6: Modelo de gestión más adecuado según el conocimiento y la claridad de los requisitos.....	44
Figura 7: Ejemplo organización matricial .....	45
Figura 8: Ejemplo de división de una turbina por módulos .....	46
Figura 9: Ejemplo gestión de tareas en Kanban.....	48
Figura 10: Ejemplo del modelo de escalado en Scrum of Scrums.....	48
Figura 11: Representación visual de una persona en T .....	50
Tabla 1: Bases de artículos utilizadas para el estudio .....	27
Tabla 2: Grupos y palabras clave para la búsqueda .....	28
Tabla 3: Metodologías ágiles utilizadas por las compañías .....	28
Tabla 4: Uso de prácticas de escalado .....	29
Tabla 5: Retos identificados para el escalado de Agile .....	35

## Glosario

<b>Términos</b>	<b>Definición</b>
Feedback	Retroalimentación tanto de una herramienta como de interacciones entre personas
Agile Sweet Spot	Entorno óptimo para la aplicación de <i>Agile</i>
Greenfield project	Proyectos que se llevan a cabo sobre un entorno sin limitaciones, no hace falta adaptarse al entorno.
Scrum	Es un marco de gestión de proyectos ágil, utilizado para obtener avances en 2-4 semanas.
Kanban	Palabra proveniente del japonés que significa tarjeta visual o señal visual, utilizado como nombre de un modelo de planificación de tareas actualmente
Portfolio	Cartera de proyectos que contiene una empresa
Backlog	El Backlog es un documento vivo en el que se recogen todas las tareas a realizar en un proyecto, estas se asignan a especialistas y van saliendo de la lista. La lista se actualiza constantemente añadiendo y eliminando tareas
Focal Point	Persona de referencia en la organización para tratar temas especializados, suele haber 1 focal point por especialidad habitualmente.
Product Manager	Responsable del éxito de un producto frente al cliente.
Scrum Master	Facilitador de proyectos, responsable de que los equipos alcancen los objetivos definidos para el sprint.
Product Owner	Responsable del éxito de un producto dentro del equipo.
Feature Teams	Similar a un equipo de <i>Agile</i> , es un equipo de larga duración, multidisciplinar capaz de desarrollar un incremento del proyecto de manera independiente
Undone Department	Departamento creado exclusivamente para terminar tareas inacabadas durante el avance de un proyecto, idealmente no deben existir.
Kick Off	Punto de partida de un proyecto.
Sprint	Mini proyecto de un mes de duración que tiene como objetivo incrementar el valor del producto en desarrollo.

## 1 Introducción

Las metodologías ágiles se están convirtiendo en el modelo a seguir en cuanto a la gestión de los proyectos, esto es debido al éxito obtenido con una implementación adecuada de las metodologías.

Las metodologías ágiles son un conjunto de métodos que permiten adaptar el modo de trabajo a las condiciones del proyecto, aportando flexibilidad y eficiencia. Las metodologías ágiles nacieron en el año 2001, cuando importantes desarrolladores de Software se reunieron para poner en común sus mejores métodos de desarrollo. De esta manera se crea el Manifiesto Ágil, en el que se establecieron 12 principios básicos y cuya filosofía se puede resumir en cuatro ideas:

- Valorar a los individuos sobre los procesos y herramientas:

Esta idea promueve que las personas son lo más importante, por encima de proceso y herramientas. Los procesos y herramientas deben servir para apoyar a las personas, y deben adaptarse a la organización, los equipos y las personas y no al revés.

- Valorar el software funcionando sobre la documentación excesiva:

Esta idea propone que es más importante obtener *feedback* de usuarios o cliente de un prototipo temprano a una documentación previa exhaustiva. No se renuncia a la documentación, y esta debe ser completada, pero el peso del prototipo en funcionamiento es mayor.

- Valorar la colaboración con el cliente sobre la negociación contractual:

Aquí se considera al cliente uno más del equipo, involucrándolo en mayor escala durante el proceso de desarrollo obteniendo una retroalimentación continua para cumplir a rajatabla lo pactado y deseado. Esto es especialmente útil cuando es difícil definir de antemano los requisitos del Producto.

- Valorar la respuesta ante el cambio respecto a seguir un plan:

No tiene sentido ceñirse a planes rígidos cuando el entorno es volátil. Es más valiosa la capacidad de adaptación y respuesta a la capacidad de seguir un plan preestablecido. Las metodologías ágiles promueven la anticipación y adaptación, frente a la planificación y el control tradicional.

Los métodos ágiles permiten adaptar la forma de trabajo a las condiciones del proyecto, consiguiendo flexibilidad en inmediatez para dar respuesta a las necesidades del proyecto. De esta manera, se consigue gestionar los proyectos de forma flexible, autónoma y eficaz reduciendo los costes e incrementando su Productividad.

Parte del éxito y la fama obtenida hasta el momento por las metodologías Ágiles viene asociada a su aplicación principalmente en entornos considerados como *Agile sweet spot*, que consiste en equipos de trabajo reducidos, involucrados en proyectos *greenfield* (sin condicionantes por un trabajo previo) no críticos y con una estructura de gobierno simple.

En la actualidad las metodologías ágiles están siendo implementadas en proyectos de gran envergadura y grandes compañías. Compañías como Microsoft, Amazon y Google han adoptado las metodologías ágiles en su gestión. La implementación de metodologías ágiles en grandes organizaciones supone un reto para la compañía tradicional, ya que estas suelen tener una estructura rígida en la que la introducción de cambios es lenta y tiene que hacer frente a una inercia de funcionamiento generada durante el tiempo. Por otra parte, las grandes organizaciones implican por lo general grandes proyectos con una combinación de equipos dedicados y recursos compartidos. Estos factores combinados suponen un reto para el escalado de las metodologías ágiles en estas organizaciones.

Debido a la demanda de aplicación de metodologías ágiles en grandes organizaciones, se han generado varios métodos para el escalado del modelo ágil. Según [1], los métodos de escalado más populares en la actualidad son *Scrum of Scrums*, *Scaled Agile Frameworks SAFe*, *Large Scale Scrum LeSS* y *Enterprise Scrum*. Debido a que estas metodologías son novedosas, la información obtenible en artículos o libros es en su mayoría teórica y no basada en implementaciones.

## 2 Objetivos del Trabajo Fin de Master

El TFM se centra en el problema de escalado de las metodologías Ágiles en grandes organizaciones. Los objetivos definidos son los siguientes:

- Revisión y comparación de 2 metodologías existentes para el escalado del modelo ágil en grandes organizaciones.
- Identificación de desafíos y factores de éxito para el escalado del modelo *Agile*.
- Análisis de la adopción del modelo ágil en una gran organización existente.

## 3 Desarrollo y plan del TFM

El Trabajo Fin de Master alcanzará los objetivos evaluando, en primer lugar, los métodos y marcos existentes para escalar el modelo ágil. Se seleccionarán dos de ellos que serán examinados en profundidad.

En segundo lugar, se llevará a cabo una investigación de estudios primarios que examina las experiencias de grandes empresas que adoptan métodos ágiles. En esta investigación, se identificarán los retos y los factores de éxito de estas adopciones, así como las prácticas de escalado que utilizaron estas empresas.



Universidad del País Vasco Euskal Herriko Unibertsitatea

BILBOKO INGENIARITZA ESKOLA

ESCUELA DE INGENIERÍA DE BILBAO

Por último, se analizará la adopción del modelo ágil en un grupo empresarial existente, el cual se encuentra en los primeros pasos de la adopción. Teniendo en cuenta los resultados de las dos primeras partes del TFM se propondrán sugerencias para una mejor transformación.

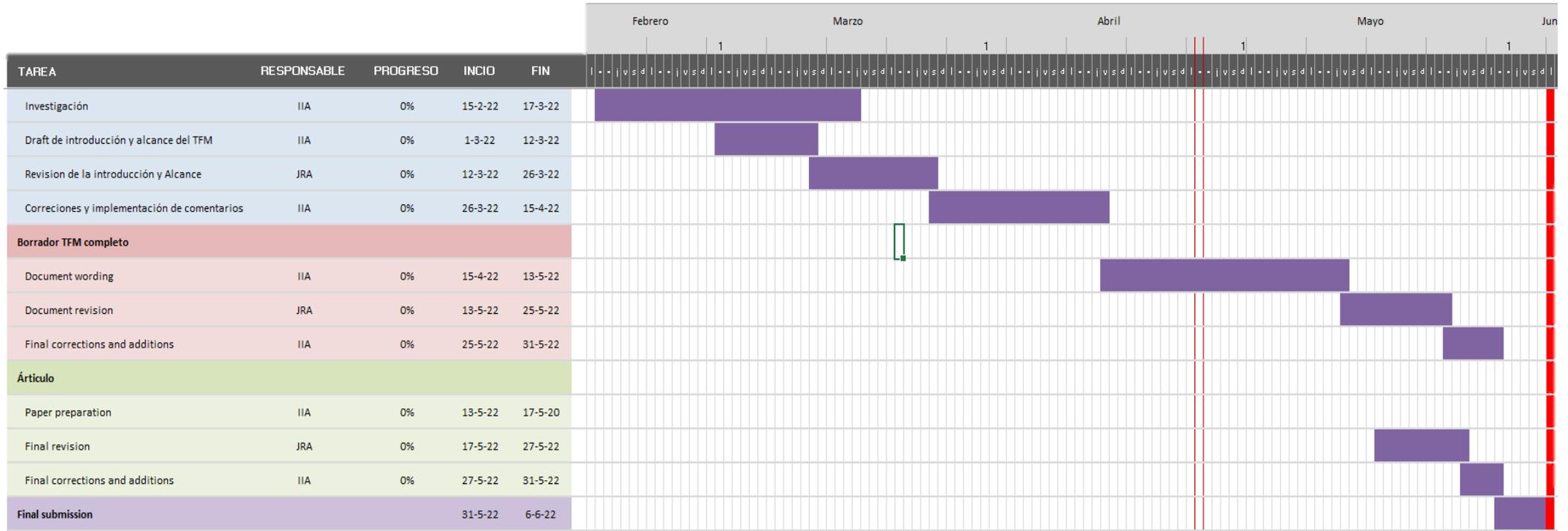


Ilustración 1: Plan inicial Trabajo Fin de Master

## 4 Metodologías para escalado de *Agile*

Introducir la metodología ágil en grandes organizaciones es un reto para la entidad [2]. Los proyectos grandes requieren una coordinación y comunicación adecuadas entre los equipos [3], hay que gestionar las dependencias entre los equipos y hay que involucrar a otras unidades no ágiles. Los métodos ágiles tradicionales se diseñaron para proyectos de un solo equipo y no se enfrentaban a estos problemas [4].

Por lo tanto, se crearon varios métodos, prácticas y marcos completos centrados en la ampliación de los métodos ágiles tradicionales al entorno a gran escala. Según [1] los métodos ágiles de ampliación más utilizados son *Scrum / Scrum of Scrums* (9 %), *Scaled Agile Framework* (37 %), *Lean Management* (2 %), *Enterprise Agile* (6%), *Agile Portfolio Management* (93%), *Large Scale Scrum* (3 %), *Disciplined Agile* (3 %).

Se ha seleccionado *Scaled Agile Framework* (SAFe) y *Large Scale Scrum* (LeSS) como dos representantes de estos métodos, que se analizarán con más detalle. Estos métodos se han seleccionado por varias razones. SAFe es probablemente el método ágil de escalado más complejo y completo. Es un marco completo, y muchas otras prácticas conocidas como *Lean Management* o *Agile Portfolio Management*, ya están incorporadas en él. *Large Scale Scrum* (LeSS) es también un marco completo para escalar la metodología Ágil. Sin embargo, trata de ser lo más minimalista posible. Se basa menos en los procesos y las estructuras organizativas específicas y más en la mentalidad de las personas y la comunicación ‘ad hoc’.

### 4.1 *Scaled Agile Framework*

La información sobre *Scaled Agile Framework* (SAFe) se revisó a partir del sitio web oficial de SAFe [5]. En el momento de escribir este Trabajo Fin de Master, la versión más reciente de SAFe era la 5.0.

SAFe es un marco de trabajo y una colección de buenas prácticas de desarrollo ágil para grandes empresas. El marco ha sido adoptado por grandes empresas como Intel, Hewlett-Packard Enterprise y Cisco[5].

El SAFe se basa en las ideas del desarrollo ágil, el desarrollo de Productos *Lean* y el pensamiento sistémico. Es compatible con empresas de diferentes tamaños, desde las pequeñas con menos de 100 empleados hasta las mayores empresas con más de miles de personas. Para soportar tal grado de flexibilidad, SAFe tiene extensiones opcionales para las grandes empresas.

La estructura de SAFe se divide en tres niveles básicos (Figura 1); Nivel de *Portfolio*, *Large Solution* (flujo de valor en SAFe 4.0) y nivel *Essential* (combina el nivel de equipo y programa del SAFe 4.0) considerados los fundamentales. A través de todos los niveles se extiende el nivel Fundación con elementos adicionales que sirven de guía a las organizaciones. Los elementos del nivel Fundación son, por ejemplo, los valores centrales de SAFe, la mentalidad *Lean-Agile* y los principios de SAFe.

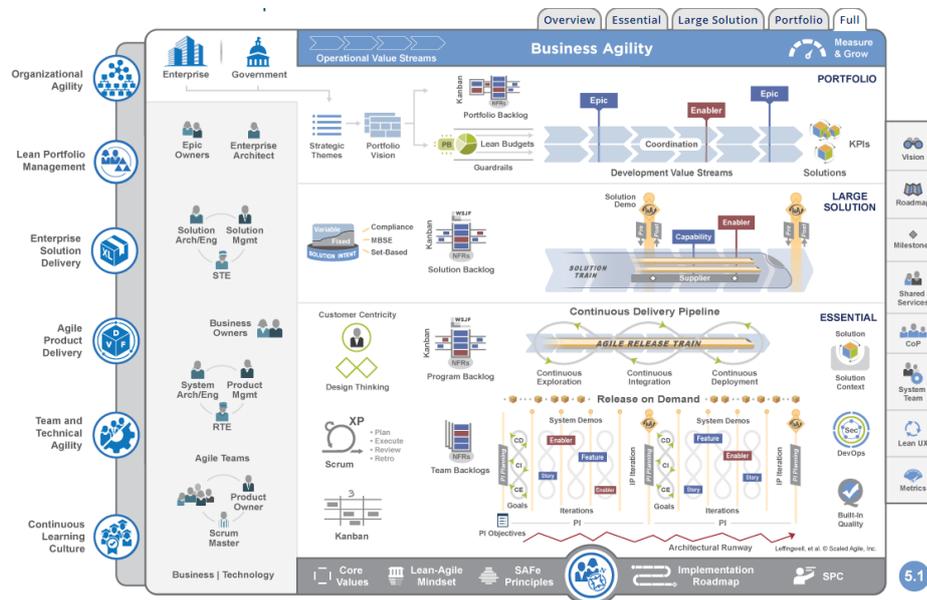


Figura 1: Niveles de la metodología SAFe

El nivel de *Portfolio* guía a la empresa en su misión, define y gobierna las decisiones estratégicas fundamentales que deben aportar valor a la organización. También gestiona el presupuesto de la organización. El nivel *Essential* contiene el mínimo número de roles, eventos y artefactos necesarios para entregar incrementos mediante el *Agile Release Train* (ART). El nivel de *Essential* es el nivel más bajo de SAFe. Describe el funcionamiento del equipo ágil. Utiliza técnicas de *Scrum*, *Kanban* y *XP* en el proceso de desarrollo definido de los equipos ágiles.

*Large Solution* describe roles adicionales para los retos más grandes, estos normalmente no se pueden definir en el objetivo de un único ART. Implementa el *Solution Train* que combina varios ART, al suministrador y la gente a un único objetivo.

#### 4.1.1.1 Alineación

La alineación se extiende a todos los niveles de SAFe. Empezando por el nivel del *Portfolio* y luego se propaga hacia abajo a través de los *backlogs* individuales en cada nivel. De este modo, los equipos de los *Agile Release Trains* (ART) valoran los objetivos del nivel de programa por encima de los objetivos de su equipo. Los ART valoran la consecución de los objetivos del *Solution Train* por encima de los objetivos del ART y, por último, los *Solution Trains* valoran su contribución a los resultados empresariales del *Portfolio*.

#### 4.1.1.2 Calidad incorporada

La sensibilidad de los grandes proyectos a la calidad es mayor que la de los subsistemas que los componen. Por lo tanto, cada fase del proyecto tiene que ser validado durante el flujo de las tareas y no después. SAFe incluye prácticas de calidad integradas

que ayudan a los equipos a garantizar que cada parte del proyecto tenga la calidad adecuada.

#### 4.1.1.3 *Transparencia*

El desarrollo de grandes proyectos es una tarea compleja y puede no salir según lo previsto. El proceso de desarrollo tiene que ser capaz de reaccionar ante estas situaciones inesperadas. En esta compleja tarea, hay que ser capaz de tomar decisiones con rapidez, por lo que es necesario implementar la toma de decisiones distribuida. La transparencia, permite a la organización adaptarse a tiempo frente a los retos y evita las decisiones tardías, las cuales tienen un alto de riesgo de no ser adecuadas o puede que ya lleguen tarde.

#### 4.1.1.4 *Ejecución del programa*

El requisito fundamental para el éxito económico es la entrega continua de valor. SAFe proporciona herramientas y guías para las funciones y responsabilidades que ayudan a los ART a cumplir el objetivo fundamental de la ejecución del programa.

#### 4.1.2 Principios de SAFe

SAFe aparte de estar definido por sus valores fundamentales, también está definido por principios. Según SAFe, estos principios se consideran verdades inmutables, fundamentales y básicas sobre las que se construye el marco de trabajo y deberían aportar más compromiso de los empleados, tiempo de comercialización, calidad de las soluciones y productividad del equipo.

El reto de desarrollar grandes proyectos que integran varios equipos de una organización es muy complejo. Los creadores de SAFe están de acuerdo en que no hay una norma única aplicable a todos los casos a los que una compañía se puede enfrentar. Es por esto que recomiendan que los principios definidos en la metodología sirvan de guía, la cual se personalice adaptándola en entornos en los que SAFe no sea del todo aplicable.

Según [5], los principios son una evolución de los principios definidos en el *Agile manifesto*, metodologías *Lean* y pensamiento sistemático, basado en la experiencia de empresas de éxito. Los principios de SAFe son los siguientes [5]:

- Tener una visión económica

Definir la estrategia para la entrega de valor incremental y el marco económico del flujo de valor, que define las compensaciones entre el riesgo, el coste de los retrasos y los costes operativos y de desarrollo, y apoya la toma de decisiones descentralizada.

- Utilizar un pensamiento de sistema

Muchos de los retos a los que se enfrenta un proyecto requieren un profundo entendimiento del sistema en el que los usuarios y los trabajadores operan. Para intentar mejorar u optimizar un proyecto requiere un entendimiento mayor del objetivo del sistema.

- Asumir la variabilidad; preservar las opciones

El modelo tradicional de diseño tiene como base la elección de una única opción para desarrollar el diseño. El descartar el resto de opciones una fase tan pronto en el proyecto puede hacer que modificar la opción seleccionada más adelante lleve más tiempo. Propone mantener múltiples opciones durante un periodo más largo del ciclo de desarrollo.

- Construir de forma incremental, integrar ciclos de aprendizaje

El desarrollo de un proyecto por incrementos da la oportunidad para que los clientes aporten su opinión en cada interacción ayuda a mitigar riesgos. Esto en proyectos de diseños complejos se implementa mediante auditorias con el cliente en las que se revise el estado del trabajo realizado.

- Basar los hitos en la evaluación objetiva de los sistemas de trabajo

La evaluación de los objetivos ofrece la oportunidad de garantizar que el Producto proporcionará un rendimiento proporcional y, por tanto, merece la pena seguir invirtiendo.

- Visualizar y limitar el WIP, gestionar y observar el “Lead time” de equipos que tengan que intervenir en el proyecto

Todas estas técnicas son medios para lograr y mantener un flujo continuo de progreso dentro del sistema.

- Aplicar la cadencia, sincronizar planes entre secciones

La cadencia aporta ritmo a los desarrollos y añade una faceta de predictibilidad a los proyectos. Conseguir una sincronización entre secciones de la organización requiere observar múltiples perspectivas e integrarlas.

- Desbloquear la motivación intrínseca del conocimiento de los trabajadores

Aumentar el compromiso de los empleados ofreciéndoles más autonomía, objetivos y minimizando las limitaciones y procesos burocráticos dentro de lo posible. Intentar motivar a los empleados exclusivamente con incentivos personales puede generar problemas de competitividad dentro del equipo.

- Descentralizar la toma de decisiones

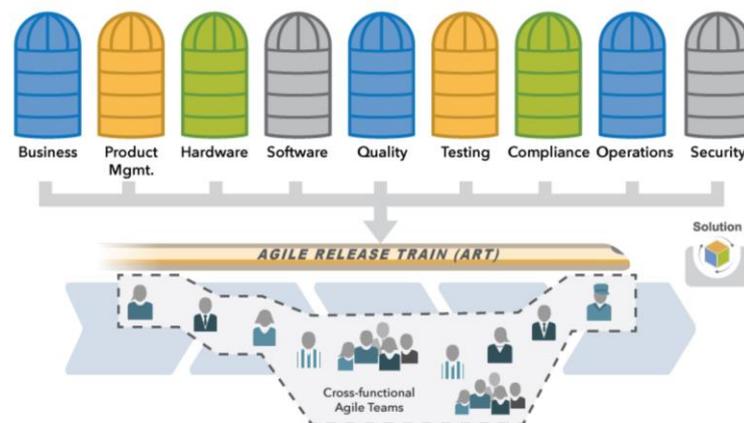
La descentralización de la toma de decisiones conduce a un menor tiempo de espera, a un contexto más local en la decisión y a soluciones más innovadoras y retroalimentadas. Sin embargo, también se toman decisiones estratégicas y globales.

#### 4.1.3 Nivel *Essential*

*Essential* SAFe es el nivel más bajo de SAFe, el ART es el corazón de este nivel. Los ART son organizaciones virtuales formadas para acelerar la entrega de valor.

##### 4.1.3.1 *Agile Release Train (ART)*

Cada ART es auto organizado, un grupo de larga duración de equipos ágiles. SAFe sugiere tener entre 50 y 125 personas organizadas en un ART. Los equipos de ART son multifuncionales e incluyen todos los roles necesarios para transformar las ideas desde el concepto, pasando por las fases de desarrollo, hasta el despliegue. Los *Focal Point* de cada disciplina asesoran a los equipos de ART para garantizar la calidad necesaria del Producto.



*Figura 2: Ejemplo de un Agile Release Train (ART)*

Los objetivos que define SAFe para cada ART son los siguientes:

- Unir a los equipos ágiles para lograr un objetivo común utilizando el *backlog* del programa.
- Producir valor a nivel de proyecto que pueda ser evaluado periódicamente (semanal, mensual...).
- Alinear las iteraciones de los equipos ágiles.

- Concentrar las iteraciones del equipo en un incremento del Programa/Proyecto.

#### 4.1.3.2 Equipos y funciones de ART

Los equipos ágiles en SAFe utilizan prácticas ágiles estándar, basadas principalmente en *Scrum*, *XP* y *Kanban*. La elección de estos métodos depende de cada equipo, pudiéndose implementar practicas adicionales en caso de considerarse necesarias.

Un equipo de SAFe está formado generalmente por entre cinco y nueve personas idealmente, este número de personas está definido por los roles necesarios para cumplir la iteración del proyecto. Los roles principales del equipo son el *Scrum Master*, responsable de asegurar que el equipo trabaja adecuadamente y sigue los procesos; el *Product Owner* responsable de la priorización y los *Agile team*, responsables de llevar a cabo las tareas e iteraciones.

En ART se identifican los siguientes roles:

- *Release train Engineer (RTE)*

RTE es el *Scrum Master* escalado. Es el *Scrum Master* principal de todo el ART. Su trabajo consiste en facilitar el proceso y la ejecución a nivel de programa, eliminar los impedimentos, gestionar el riesgo e impulsar la mejora continua a nivel de programa.

- *Product Manager*

El *Product Manager* es el dueño del *backlog* del programa. Responsables de la priorización en las iteraciones, y se aseguran de que los requisitos sean claros y se entiendan.

- *Business Owner*

Los principales *Stakeholders* responsables del resultado del proyecto/programa.

- *System Engineer*

Los *System Engineer* son responsables de diseñar y compartir una visión estructural del proyecto en el ART, que mostrará como el trabajo realizado en cada iteración aportará al objetivo final.

#### 4.1.3.3 Incremento del Programa (PI)

El Incremento del Programa PI es una iteración del *Agile Release Train*, equivalente a un *Sprint* en *Scrum*. Es el intervalo para construir y validar un incremento completo del proyecto, la duración de un PI debe ser de entre 8-12 semanas. Un

Incremento de Programa (PI) consiste en varias iteraciones de equipo más pequeñas, cada iteración del equipo tiene su planificación del equipo, refinamientos del *backlog* y stand-ups diarios.

El PI tiene sus propias reuniones, consideradas de mayor nivel que las del equipo. Hay reuniones de *Scrum* de *Scrums*, planificación de PI y de coordinación del *Product Owner*, juntos estos dos tipos de reuniones se llaman reuniones de Sincronización de ART. Además, hay Reuniones de Gestión de Lanzamientos, Demostración del Sistema, Planificación del Incremento del Programa, y reunión de Inspección y Adaptación.

La planificación del PI es la reunión más grande de todas. Esta planificación se realiza en una reunión en la que se involucran todas las partes de la organización afectadas por el PI. La duración de la planificación puede ser de varios días.

Los equipos hacen reuniones de *Scrum* de *Scrums* (SoS) deben celebrarse al menos una vez a la semana. Las reuniones tienen una duración inferior a 30 minutos. El orden del día debe ser sobre los impedimentos, las tareas realizadas, la sincronización y otras cuestiones.

Para la coordinación del programa y los siguientes incrementos, se genera una reunión de *Product Owners* y *Program Managers*. La duración de la reunión debe ser de entre 30 y 60 minutos. El orden del día de esta reunión es la preparación del siguiente PI, el refinamiento del *backlog* del programa y la priorización del *backlog*.

Las reuniones de gestión de lanzamientos sirven para controlar los lanzamientos y comunicar la información sobre ellos a la dirección.

La reunión de *System Demo* se realiza cada dos semanas del incremento del programa. El objetivo de esta reunión es revisar con los *Stakeholders* el progreso del incremento y revisar la integración entre equipos.

La preparación del siguiente Incremento de Programa no es una reunión, sino un proceso continuo. Como la planificación del PI es un evento de dos días que puede incluir hasta cientos de personas, es necesario una preparación continua. La preparación incluye la alineación de la gestión y la preparación de la organización para la planificación, la preparación del *Portfolio* de pedidos y la preparación de las instalaciones.

La reunión de Inspección y Adaptación se lleva a cabo al final del incremento del programa. La reunión funciona como una retrospectiva a escala en *Scrum*.

#### 4.1.4 Nivel de *Portfolio*

El nivel de *Portfolio* es el nivel más alto en SAFe. Su función es definir toda la estrategia de una organización, asignar fondos y servicios a los programas y sincronizar el esfuerzo y la arquitectura de los mismos. Las pequeñas y medianas empresas pueden tener sólo un nivel de *Portfolio*, pero para las grandes empresas, pueden ser necesarios

varios niveles de *Portfolio*. En este nivel se definen los *Strategic Themes*, *Managers* del *Portfolio* de Proyectos, *Value Streams* y se evalúan los *Epics* de *Portfolio*.

Los *Strategic Themes* definen la estrategia de una empresa. Establece los objetivos que la empresa quiere cumplir, como el mercado objetivo, la ventaja competitiva, las características principales, etc. También funcionan como filtros principales para la cartera de pedidos. Los *Strategic Themes* son definidos por la dirección de la empresa junto con la Gestión del *Portfolio* de Programas.

Los *Managers* del *Portfolio* de Proyectos (PPM) son los *Stakeholders* de todo el nivel del *Portfolio*. Algunas de las responsabilidades del PPM son el cumplimiento de los *Strategic Themes*, la gestión del retraso del *Portfolio* y la asignación de fondos.

Las *Value Streams* identifican las tareas/eventos que tienen que suceder para llevar a cabo una solución, Producto o servicio. Pueden consistir en una o varias ARTs, pero una ART puede tener también múltiples *Value Streams*. Los presupuestos se asignan a las *Value Streams* y no a las ARTs.

Los *Epics* de *Portfolio* son iniciativas de nivel empresarial y afectan a varias organizaciones e incrementos de programa. Un *Epic* de *Portfolio* afecta a varios ART.

## 4.2 Método *Large Scale Scrum* (LeSS)

La información sobre el *Large Scale Scrum* (LeSS) se ha recopilado en el sitio web oficial de LeSS [6] y en el libro de la referencia [7]. LeSS es un marco para el escalado de la metodología ágil, trata de traer toda la información necesaria sobre el escalamiento en una solución compacta. Planteamiento opuesto a SAFe, el cual aporta un marco completo en el que se generan muchos roles e introduce muchas metodologías.

Esencialmente, LeSS amplía la visión de *Scrum* a un entorno de mayor escala. La idea central de LeSS es que incluso para las grandes organizaciones no es necesario un proceso excesivamente complicado que cree la carga de la burocracia. LeSS está planteado con el objetivo de marcarse como pilar el valor de que las personas y las interacciones son más valiosas que los procesos y las herramientas.

*Large Scale Scrum* se ha dividido en dos variantes, contiene la variante LeSS; la cual está más enfocada a organizaciones que desarrollan tareas involucrando hasta 8 equipos. Por otro lado existe la variante LeSS Huge, esta última está enfocada a organizaciones que requieran un número mayor de equipos para desarrollar un proyecto.

### 4.2.1 Principios de LeSS

Similar al caso de SAFe, LeSS contiene unos principios que deben ser la base de la transición o modelo de trabajo. Estos principios deben adaptarse a las organizaciones, pero siempre manteniendo la esencia de LeSS. Los principios definidos por LeSS [6] son los siguientes:



- LeSS es Scrum

LeSS trata de ser una extensión lo más minimalista posible de *Scrum*. Considera *Scrum* como una base completa la cual es aplicable a grandes entornos con mínimos ajustes. El marco pretende ser un *Scrum* de varios equipos en vez de varios equipos de *Scrum*.

- Control del proceso empíricamente

Inspección y adaptación del producto y procesos con el fin de generar una organización adecuada con base de *Scrum* para cada caso. En vez de una formula genérica para todos los casos.

- Transparencia

Basada en objetivos claros de cumplimiento, ciclos cortos de desarrollo, con definiciones comunes y cooperación entre equipos/personas.

- Más con menos

Este principio tiene tres significados en LeSS. El primero es en el contexto del control del proceso empírico, Se prefiere hacer más aprendizaje sobre el proceso en vez de seguir procesos predefinidos. El segundo es utilizar pensamiento *Lean*, crear más valor con menos desperdicio. El último tiene que ver con el escalamiento, una propiedad y un propósito más distribuidos para impulsar a la gente y menos roles, artefactos y grupos especializados.

- Enfoque de Producto completo

No importa si hay varios equipos. Debe haber un *backlog* de producto, un *Product Owner*, un incremento de producto potencialmente alcanzable y un *Sprint*.

- Centrado en el cliente

Las organizaciones deben centrarse en el cliente, lo que significa comprender los problemas reales de sus clientes, implicarlos en el proceso de desarrollo, reducir el tiempo de espera y aumentar los circuitos de retroalimentación.

- Mejora continua hacia la perfección

Entregar Productos continuamente, sin defectos, haciendo que el cliente esté satisfecho. Hacer mejoras humildes o radicales en cada *Sprint* hacía la entrega de Productos sin defectos.

- Pensamiento de sistema

Observar y entender el sistema completo (no partes sueltas), centrar las optimizaciones y mejoras en todo el sistema en vez de equipos o procesos individuales. El cliente se preocupa sobre el resultado del sistema completo en vez de partes individuales, lo importante es aportar valor mejorando el sistema completo.

- Pensamiento esbelto

Los directivos de las organizaciones deben guiar a las personas y no dictarlas utilizando el "debería" más que el "debe". Deben enseñar y aplicar el pensamiento *Lean* y sus principios fundamentales, como el de ir y ver, la mejora continua y el respeto por las personas.

- Teoría de las colas

Utilizar la teoría de las colas. La teoría de colas ayuda a las organizaciones a comprender mejor el flujo de grandes elementos en su proceso de desarrollo. La comprensión de cómo el flujo de estos grandes elementos afectará al sistema puede detectar problemas con antelación y mostrar formas de mejorarlo.

#### 4.2.2 Variante LeSS

LeSS está pensado para empresas con hasta ocho equipos *Scrum* que desarrollan un producto. Muchas prácticas en LeSS son las mismas que en *Scrum*, por lo tanto, incluso para los proyectos de varios equipos, LeSS tiene un único *backlog* de Producto, un *Sprint*, una Definición de Hecho, un incremento de producto potencialmente enviable al final de cada *Sprint*, y un Propietario de Producto para todos los equipos.

##### 4.2.2.1 Estructura organizativa

LeSS tiene como máximo 8 equipos, un *Product Owner* y un *Undone Department*. Todo esto coordinado por el Head of *Product group*. La estructura organizativa típica de LeSS se puede ver en la Figura 3.

Los equipos en LeSS se autogestionan y son interfuncionales. Las personas se asignan a un solo equipo y no se comparten. Un mismo equipo no puede estar distribuido en varios lugares. Los miembros del equipo están ubicados en la misma sala. Los equipos son de larga duración, los equipos deben ser lo más estables posible y preferiblemente permanecer juntos durante varios años. Además, los equipos de desarrollo en LeSS deben ser también los llamados *Feature Teams*. La principal característica de los *Feature Teams* es que el equipo debe ser capaz de entregar la característica completa. El equipo individual debe tener conocimientos sobre cada componente del sistema. Esto permite especializar a los equipos individuales desde un punto de vista centrado en el cliente y no desde el punto de vista técnico y, por tanto, les permite centrarse en la característica desde los valores centrados en el cliente.

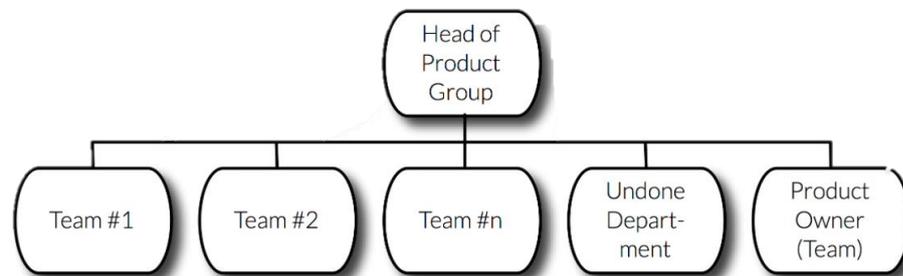


Figura 3: Estructura organizativa de variante LeSS

Como sólo hay un *Product Owner* para todo el proyecto, se centra más en una visión general de todo el producto y maximiza su retorno de la inversión. Por lo demás, el papel del *Product Owner* es el mismo que en *Scrum*. Por lo tanto, es responsable de la priorización del *backlog*, la aclaración de los elementos en el *backlog* y el apoyo a la colaboración de los equipos con los clientes. Se aplica el mismo principio de *Scrum*: el *Product Owner* es un conector y no un intermediario entre los equipos y el cliente. LeSS también menciona que el *Product Owner* tiene toda la responsabilidad y autoridad sobre el proyecto; esto significa que el cambio de contenido, las fechas de lanzamiento, las prioridades y la visión son todas responsabilidades del *Product Owner*. Sin embargo, los equipos y el *Product Owner* son compañeros como en *Scrum*.

La última parte de la estructura organizativa en LeSS es el “*Undone Department*”. El *Undone Department* ayuda a los equipos a crear un incremento alcanzable en cada *Sprint*; esto significa ayudarles a definir y seguir su definición de “Hecho”. Este departamento puede apoyar a los equipos, en tareas de garantía de la calidad, conocimiento técnico, o de negocio. LeSS no apoya la creación de este departamento, pero reconoce que puede ayudar a la organización. La propuesta es integrar el rol de este departamento en cada equipo mediante los integrantes del equipo.

#### 4.2.2.2 El Portfolio de Productos y su perfeccionamiento

Cada equipo implementa las características que ha seleccionado en cooperación con otros equipos durante un proceso de planificación multi equipo. La coordinación de las tareas se basa en un único *backlog*, el cual está priorizado y contiene las fechas esperadas de cumplimiento de cada tarea.

El *backlog* se perfecciona durante varias sesiones. Estas sesiones no son requeridas por LeSS, y depende de los equipos cómo acuerden el proceso. Sin embargo, el proceso de refinamiento del *backlog* sugerido es que todos los equipos tengan una reunión común llamada Refinamiento General del *Backlog* del Producto (RGBP). En esta reunión los equipos junto al *Product Owner*, deciden sobre los elementos más importantes y se acuerda el proceso a seguir. Después del RGBP, los equipos pueden seguir refinando el *backlog* en cooperación con otros equipos o solos. Como sólo hay un *Product Owner*, éste sólo asiste a la RGBP.



LeSS plantea las reuniones esenciales de una manera diferente a *Scrum*, la planificación del *Sprint* se divide en dos partes; la primera parte es de planificación de equipos, en esta reunión el *Product Owner* presenta los elementos que se han seleccionado para el *Sprint* y los elementos más prioritarios del *backlog*. En esta reunión el *Product Owner* debe aclarar todas las dudas referentes a los elementos seleccionados. Los elementos pueden ser asignados a los equipos o elegidos por los equipos, en el caso de que un elemento deba ser realizado por más de un equipo, es responsabilidad de los equipos coordinarse. Para tener una planificación clara entre equipos, los equipos deben generar una reunión de coordinación entre equipos en la que revise la planificación del *Sprint*. La segunda parte de la planificación del *Sprint* es la planificación en cada equipo o entre equipos, el *Product Owner* no asiste a estas reuniones.

Por otra parte, LeSS introduce una nueva reunión de revisión del *Sprint*. La revisión de *Sprint* cumple con el objetivo de seguir el control de proceso empírico. En esta reunión se revisa el estado del Producto, las partes interesadas examinan el trabajo realizado y se discute sobre los retos, oportunidades o cambios que se quieran introducir.

LeSS amplía la retrospectiva a nivel de equipo introduciendo la retrospectiva global. La retrospectiva global debería celebrarse después de las retrospectivas a nivel de equipo. Durante la reunión, los equipos intentan mejorar la productividad de las iteraciones consecutivas identificando los impedimentos entre equipos y definiendo los pasos que deben llevar a su eliminación. El objetivo de esta reunión es intentar mejorar el sistema, en vez de optimizar equipos individuales. A la reunión asisten el *Product Owner*, los *Scrum Masters* y los representantes del equipo.

LeSS utiliza el *Scrum of Scrums* como una reunión de sincronización entre equipos durante el *Sprint*. A los *Scrum of Scrums* deben asistir representantes de cada equipo. Según LeSS, los asistentes tienen que ser miembros de los equipos que participan activamente en el trabajo del equipo - y no sólo el *Scrum Master*.

LeSS identifica como un apoyo a los equipos el uso de Comunidades de Práctica. Las Comunidades de Práctica se centran en áreas específicas o disciplinas en las que se intercambie información y experiencia para la mejora de la sesión o la disciplina.

#### 4.2.3 LeSS Huger

Para el caso de los productos más complejos o que contengan un gran volumen de tareas y requisitos, en los que el número de equipos involucrados es muy alto, existe la variante Less Huger.

La idea de LeSS Huger es dividir los requisitos en las llamadas Áreas de Requisitos. Estas áreas se organizan en torno a problemas y perspectivas centrados en el cliente. Esto significa que LeSS no recomienda crear áreas basadas en la arquitectura del producto, la propiedad del código o la tecnología. Definir las Áreas de Requisitos en función del problema o perspectiva genera estructuras rígidas en la organización. Las Áreas de

Requisitos deben ser capaces de amoldarse a las necesidades de cada cliente, es por esto que se recomienda que sean temporales.

El funcionamiento de un Área de Requisitos debe ser similar a un *Feature Team* pero escalando los conceptos a un entorno más grande. El *Product Owner* asigna a cada elemento del *Product Backlog* su Área de Requisitos. Esta asignación crea grupos de elementos del *Product Backlog* con las mismas Áreas de Requisitos. Al filtrar los elementos del *Product Backlog* por sus áreas de requerimiento, se genera una vista diferente del *Product Backlog* general. Estas vistas se denominan Áreas de Requisito.

Los *Backlogs* de Área son priorizados por el Propietario de Producto de Área que se especializa en un área determinada centrada en el cliente. El Propietario de Producto de Área actúa como un *Product Owner* especializado. Prioriza los elementos de su *Area Product Backlog*, los divide y los aclara a los equipos.

En esta variante se genera un equipo de *Product Owners*, en el están el *Product Owner* principal y los *Product Owner* de cada área. Este equipo toma las decisiones de priorización de todo el producto, aunque el *Product Owner* tiene la última palabra. Las responsabilidades, como el calendario de publicación y el alcance del producto, permanecen en manos del *Product Owner*.

#### 4.2.3.1 Estructura organizativa de LeSS Huge

Según los autores de LeSS Huge, los productos más complejos pueden requerir estructuras organizativas adicionales, pero en la medida de lo posible se recomienda evitar la creación de las mismas. La estructura organizativa típica de LeSS Huge se puede ver en la Figura 4.

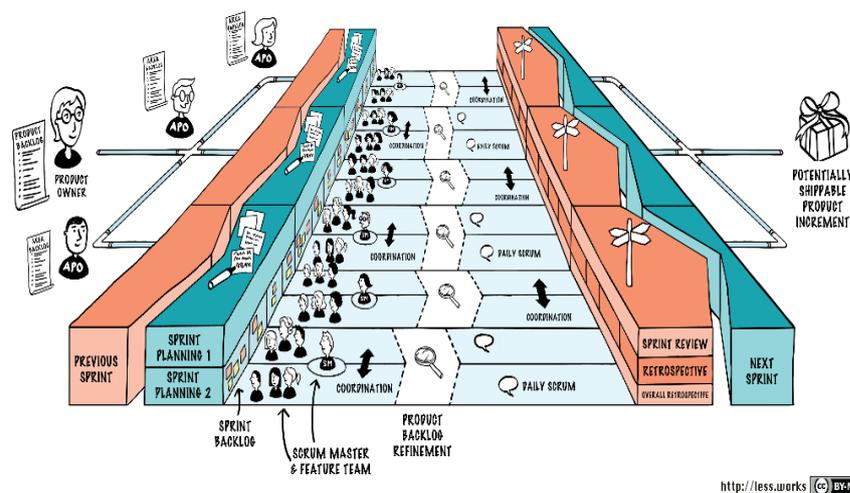


Figura 4: Estructura organizativa LeSS Huge

Los autores mencionan varias diferencias entre la estructura organizativa de LeSS Hige y la de LeSS. En primer lugar, los grandes proyectos suelen desarrollarse en varios lugares. La organización debe evitar que las Áreas de Requisitos sean la manera de organización, ya que crearía demasiadas Áreas de Requisitos permanentes.

En segundo lugar, está el nuevo equipo de *Product Owners*, ya mencionado. El equipo puede tener una subestructura adicional, como sub-equipos si el proyecto es excepcionalmente grande.

En tercer lugar, las responsabilidades del *Undone Department* siguen siendo las mismas, pero tienden a ser mayores en LeSS Hige que en LeSS. Por lo tanto, se necesita más tiempo para deshacerse de él. En los proyectos grandes, el *Undone Department* puede tener subestructuras adicionales similares a las del equipo de *Product Owners*.

Por último, LeSS Hige identifica como un valor esencial para el éxito de los proyectos la formación. Identifica un departamento exclusivo para tareas relacionadas con la formación o el coaching. LeSS Hige aboga por que las organizaciones dediquen suficientes recursos a la formación y al coaching, estableciendo un grupo de coaching independiente con profesionales cualificados.

#### 4.3 Resumen de SAFe y LeSS

Los dos marcos de escalado revisados se presentan como soluciones completas que contienen las funciones necesarias para escalar el modelo ágil a grandes compañías. Entre ellas se pueden observar muchas diferencias, pero a su vez también coinciden en varios criterios.

La definición de SAFe es muy detallada y a veces contiene incluso un programa concreto de reuniones individuales. Su estructura organizativa se divide en 3 niveles, pero estos tienen una definición muy detallada de los roles necesarios.

Por otro lado, LeSS evita definir en exceso las estructuras y roles, esto se basa en ser lo más ágil posible. Se centra en la mentalidad, los valores y los principios sin introducir demasiados procesos y funciones. El marco en sí es mucho más joven y no tiene tanto apoyo empresarial como SAFe.

Atendiendo a las similitudes entre los dos marcos de escalado, ambos marcos coinciden en que la comprensión de todo el sistema es esencial para tener una adecuada definición de tareas, procesos, planes etc. Además, se pueden apreciar algunos patrones similares en la estructura organizativa y los procesos entre estos marcos. Ambos marcos utilizan el papel escalado del *Product Owner* y una jerarquía de *backlogs*. Además, ambos se inclinan por utilizar los *Feature Teams*, esto es debido a que los equipos de características permiten un escalado más fácil de la gestión de requisitos. Además, ambos marcos utilizan algunos equipos adicionales a los equipos de desarrollo, que ayudan a los equipos de desarrollo con su trabajo. Para el escalado de la metodología ágil, los marcos coinciden en el uso de las siguientes prácticas:

- *Scrums de Scrums*
- Comunidades de práctica
- Gestión de requisitos a escala
- *Scaled Sprint Planning*
- Equipos de trabajo
- *Undone Department*

## 5 Método de investigación

El método utilizado para el análisis de artículos con el objetivo de encontrar los factores de éxito y retos es el mapeo sistemático [8]. El proceso de mapeo sistemático (ver Figura 5) tiene cinco pasos; la definición de las preguntas de investigación, la búsqueda de estudios primarios, el cribado de los artículos, la redacción clave de los resúmenes y la extracción de datos. En primer lugar, se definen las preguntas de investigación y se realiza una búsqueda de estudios primarios. Posteriormente, se realiza un cribado de los estudios primarios y se seleccionan los estudios pertinentes. A continuación, se crean categorías utilizando palabras clave en los resúmenes de los estudios primarios seleccionados. Por último, los estudios primarios se clasifican en categorías individuales.



Figura 5: Ejemplo de un proceso de mapeo sistemático

El primer paso del proceso de mapeo sistemático es la definición de las preguntas para llevar a cabo la investigación. El objetivo de la investigación es conocer cómo las grandes empresas adoptan las metodologías ágiles. Las preguntas definidas son las siguientes:

- P1: ¿Qué métodos ágiles utilizan las grandes empresas?
- P2: ¿Qué prácticas de escalado ágil utilizan las grandes empresas?
- P3: ¿Qué desafíos identifican las grandes empresas cuando se adoptan a las metodologías ágiles?
- P4: ¿Qué factores de éxito identifican para adoptar y escalar el modelo ágil?

La búsqueda de artículos, casos de estudio y publicaciones es el siguiente paso del proceso. La búsqueda de publicaciones para llevar a cabo el proceso de mapeo sistemático se ha centrado en las bases de datos (en este caso artículos) definidos en la Tabla 1. Para llevar a cabo el filtrado de artículos, se han definido dos grupos y se han definido las palabras clave, esto se puede ver en la Tabla 2. A la hora de intentar definir un grupo para el tipo de organización, esto no ha sido posible en este paso, ya que la definición adecuada para el filtrado no ha sido posible definirla.

Data Base	URL
IEEE Explore	<a href="https://ieeexplore.ieee.org">https://ieeexplore.ieee.org</a>
Science Direct	<a href="https://www.sciencedirect.com/">https://www.sciencedirect.com/</a>
Research Gate	<a href="https://www.researchgate.net/">https://www.researchgate.net/</a>
Springer	<a href="https://www.springer.com/">https://www.springer.com/</a>

Tabla 1: Bases de artículos utilizadas para el estudio

El tercer paso del proceso de mapeo sistemático es la selección de los artículos, es decir, su inclusión y exclusión en la investigación. Como no ha sido posible definir un criterio de búsqueda en el apartado anterior, en este paso se han filtrado los artículos en base a la organización y su tamaño.

El número de empleados de las empresas seleccionadas oscila entre 40 y 15.000. La selección inicial abarca veinte artículos en total, de estos 20 se han seleccionado doce de ellos; seis son informes de experiencias y seis son *Case study*. En el caso de los informes de experiencias, los autores están afiliados a la empresa. Los *Case study* los realizó un investigador independiente. Dada la dificultad para encontrar artículos imparciales con alto valor científico que cumplieran los requisitos de inclusión, se ha decidido incluir los informes de experiencias en la investigación.

Grupos	Palabras Clave
Metodologías Ágiles	Ágil, <i>Lean</i> , <i>Scrum</i> , <i>Kanban</i> , <i>Backlog</i>
	<i>Agile and Sprint</i>
Metodología de Escalado	<i>Large Scale Scrum</i>
	<i>Scaled Agile Framework</i>

Tabla 2: Grupos y palabras clave para la búsqueda

### 5.1 Métodos ágiles

Este capítulo responde a la primera pregunta de investigación, P1. Se han identificado 5 metodologías ágiles utilizados por las empresas de los artículos. Como se puede ver en la Tabla 3, el método ágil más utilizado ha sido *Scrum*. El uso de *Scrum* se menciona en todos los artículos. El segundo método ágil más aplicado ha sido *Lean*, cinco artículos informaron del uso de prácticas *Lean*. Dos casos han adoptado *Kanban* y dos casos han utilizado algún marco para escalar el desarrollo ágil. Un caso adopta LeSS y otro utiliza de *Disciplined Agile Delivery* (DAD).

Metodología Ágil	# de usos
<b><i>Scrum</i></b>	12
<b><i>Lean</i></b>	5
<b><i>Kanban</i></b>	2
<b>LeSS</b>	1
<b>DAD</b>	1

Tabla 3: Metodologías ágiles utilizadas por las compañías

Los métodos y principios *Lean* se encuentran en *Scrum*, por lo que *Lean* se adoptó implícitamente en todos los casos. Sin embargo, sólo cinco trabajos mencionan explícitamente *Lean* como metodología utilizada. Varios artículos están de acuerdo en que no basta con entender e implementar un proceso, hay que comprender los valores y principios fundamentales que el proceso está aplicando.

Dos casos informan que *Kanban* es una práctica utilizada. *Kanban* se utilizó en combinación con *Scrum*. Esta combinación se llama *Scrumban* [9]. *Scrumban* es *Kanban* con retrospectivas, *groomings*, y reuniones diarias de *Scrum*, pero sin *Sprint*. La razón de esta modificación en estas empresas ha sido que las características grandes no son fáciles de dividir en pequeñas tareas [9]. Por lo tanto, *Scrumban* permite a los equipos eludir un ciclo de iteración de *Sprint*.

El caso en el que se utiliza la metodología DAD [10] se centra más en cómo se hacía el coaching que en su aplicación. El estudio informa de que los resultados iniciales de su adopción han sido satisfactorios. Se reconoce que la evaluación es un poco prematura y que para evaluar el rendimiento más veraz se necesita más tiempo.

En el caso [11] se implementa el marco LeSS. A pesar de que la adopción del marco LeSS ha sido supervisada por Greg Larman, que es uno de los co-creadores de LeSS, los resultados de la implantación no han tenido éxito. El documento identifica varias razones por las que la transformación no ha sido satisfactoria.

En primer lugar, uno de los problemas ha sido la incorrecta división de su Producto en Áreas de Producto. El no definir correctamente las Áreas de Producto conlleva a que las tareas que se identifican puedan estar repartidas en más de un Área de Producto, esto conlleva a que los equipos no son capaces de desarrollar las tareas de forma independiente y dependen de otras áreas. La necesidad de cooperación causó problemas con las reuniones, ya que asistían demasiadas personas a ellas. Esto hacía que las reuniones se alargasen, consiguiendo que los equipos dejaran de comunicar sus problemas para ceñirse a la hora de la reunión. Además, los equipos no estaban asignados a Áreas de Producto individuales. Por lo tanto, los Propietarios de Producto de Área competían por el mejor equipo.

En segundo lugar, la función de Propietario de Producto de Área la han asumido dos personas. Uno era el llamado Arquitecto de Sistemas y el otro Arquitecto de Soluciones. El Arquitecto de Sistemas se ocupa de la implementación técnica de una característica, y el Arquitecto de Soluciones se comunica con el cliente y prioriza los requisitos.

Por último, las Comunidades de Práctica han estado completamente ausentes, y no ha habido coaching ni formación regular. La falta de coaching podría afectar a la aplicación del marco, ya que muchos casos [9], [12]–[15] refuerzan la necesidad de coaching o de Comunidades de Práctica. Poner en práctica un proceso sin comprender las ideas que lo sustentan podría llevar a alternar partes del proceso de forma errónea, lo que acaba perjudicando a todo el proceso.

## 5.2 Prácticas de escalado

En esta sección se describen las prácticas de escalado que utilizaron las organizaciones, así como sus experiencias, respondiendo así a la segunda pregunta de investigación, P2. Las prácticas de escalado y el número de usos identificados se muestran en la Tabla 4.

Practica de escalado	# aplicaciones
<i>Scrums de Scrums</i>	6
<b>Comunidades de practica</b>	5
<i>Scaled Sprint Demo</i>	3
<b>Scaled Requirement Managenement</b>	3
<i>Scaled Sprint Planning</i>	3
<i>Feature Teams</i>	2
<i>Undone Department</i>	2

Tabla 4: Uso de prácticas de escalado

### 5.2.1 *Scrum of Scrums*

*Scrum of Scrums* ha sido la práctica de escalado más utilizada [9]–[11], [13], [14], [16]. Varios de los artículos utilizan *Scrum of Scrums* para la sincronización entre los equipos [11], [14], [16]. El orden del día de la reunión de *Scrum of Scrums* es informar a otros equipos sobre el estado actual del trabajo del equipo, los impedimentos que deben ser resueltos, y la discusión sobre las dependencias entre los equipos y la gestión de estas dependencias.

El número y duración de las reuniones de *Scrum* han sido un problema, esto es debido a la elevada cantidad de reuniones y en los casos de productos complejos, la duración de las mismas. El caso [9] utiliza *Scrum of Scrums* de características específicas para evitar el problema. El sobredimensionamiento de las reuniones de *Scrum of Scrum* provoca que los equipos dejen de relatar sus impedimentos, ya que sienten que no hay tiempo durante las reuniones. Al final, los equipos consideraron que las reuniones no son útiles.

Una solución parcial a este problema es utilizar representantes del equipo en la reunión, como sugiere LeSS o SAFe. Este enfoque se utiliza en el caso [11], pero otro problema surge cuando muchos equipos sólo envían a su *Scrum Master* a la reunión. El problema con este enfoque es que el *Scrum Master* no está suficientemente involucrado en el trabajo del equipo y no tiene suficiente conocimiento sobre el mismo.

### 5.2.2 *Scaled Sprint Demo*

El *Scaled Sprint Demo* es una reunión, en esta reunión los equipos muestran las características implementadas, y en el caso de ser posible probadas. Esta metodología se aplica en los 2 marcos analizados en la sección 4, SAFe lo llama *System Demo* y LeSS lo llama *Sprint Review*. Según LeSS, la *Scaled Sprint Demo* también puede ser utilizada como una reunión de inspección y adaptación para decidir la dirección del Producto, pero en todos los casos la *Scaled Sprint Demo* se utiliza para mostrar el avance de las características de todas las áreas de un Producto [11], [13], [16].

El caso [11] se identifican varios problemas con el *Scaled Sprint Demo*. Al principio, la reunión se celebraba en un gran auditorio al que asisten todos los equipos. A medida que el proyecto crece, el *Scaled Sprint Demo* se hace demasiado grande. Es por esto que los equipos empiezan a tener reuniones separadas, a las que sólo asisten los *Product Owners*, esto no daba a los equipos suficiente contexto de todo el Producto.

La organización de [13] pasa por una transformación similar de *Scaled Sprint Demo* que la organización anterior [11]. El primer intento dejar que los equipos hagan una demostración de su trabajo en un solo lugar. La segunda etapa es tener sólo una presentación de diapositivas por cada equipo. El último estado de la reunión es el llamado modelo de feria de la ciencia. El documento [13] describe el modelo de la siguiente manera: "Cada equipo monta un expositor, y los empleados visitan los expositores que les parecen

interesantes. Cada *stand* ofrece una presentación de 15 minutos sobre los esfuerzos de trabajo recientes, LeSS propone un modelo similar.

### 5.2.3 *Scaled Sprint Planning*

*Scaled Sprint Planning* es una reunión en la que se discute el futuro de un producto. Esta reunión también está integrada tanto en SAFe como en LeSS, en el caso de SAFe, llama a la reunión PI Planning. En el caso de LeSS se llama *Sprint Planning One*. La reunión es mucho más grande en el contexto de SAFe que en el de LeSS. Según SAFe, la agenda de la reunión puede ser el contexto del negocio y la visión del producto, la visión de la arquitectura, la preparación de la organización y el análisis de riesgos. LeSS sugiere sólo una reunión directa y corta, en la que el *Product Owner* presenta la lista priorizada de elementos que están listos para el desarrollo y su aclaración junto con la definición del objetivo del *Sprint*. A continuación, los equipos seleccionan los elementos en los que van a trabajar. El *Scaled Sprint Planning* se identifica en tres casos [11], [16], [17].

Varios de los artículos identifican la dificultad de hacer el *Scaled Sprint Planning* a gran escala. El *Scaled Sprint Planning* puede ser un reto en un proyecto distribuido en múltiples sitios. En el caso [16], esta reunión no es satisfactoria, las causas principales son no incluir a todos los equipos por diferencias geográficas, planificaciones tardías y malentendidos en los requisitos.

En una compañía con un planteamiento ágil a gran escala, la planificación requiere más atención y puede abarcar varias reuniones. En los casos [11] y [17] se informa que fueron necesarias dos reuniones adicionales para explicar suficientemente los requisitos a los equipos. Una reunión para aclarar los requisitos de una característica con el equipo, y otra para aclarar los detalles técnicos de la implementación. Los equipos realizan lo que se denomina investigación de características y estudio de concepto de características para cada una de ellas, con el fin de averiguar si la característica es factible, cuánto esfuerzo requiere y cómo puede implementarse.

### 5.2.4 *Scaled Requirement Management*

El *Scaled Requirement Management* es un grupo de prácticas de escalado que se ocupan de la gestión de requisitos, como la creación de la estructura jerárquica de los propietarios de productos, la estructura jerárquica de los *Backlogs* de productos y su gestión. También incluye la división de un producto en Áreas de Producto individuales. LeSS Huger utiliza Áreas de Requisitos, Propietarios de Producto de Área y *Backlogs* de Producto de Área. En SAFe esta práctica se utiliza en forma de *Value Streams*, *Agile Release Trains*, y sus *backlogs* y gestores. Se identifica el uso de esta metodología en tres casos [11], [14], [17].

El caso [11], que implementa el marco LeSS, tuvo un problema al dividir su producto en áreas de requisitos independientes. Similar al no poder dividir un producto en Áreas de Producto independientes, capítulo, esto provoca dependencias entre áreas, lo que lleva a una sobrecarga de la comunicación y a la imposibilidad de tener reuniones

separadas para las áreas del producto. Las reuniones dejan de cumplir su función al ser demasiado grandes y tener demasiados asistentes.

El caso [14] informa de problemas similares con las dependencias entre componentes, pero estos problemas se identifican antes de que empiecen a escalar su desarrollo ágil. La organización se da cuenta que, para poder escalar el desarrollo ágil de su producto, tienen que mejorar su comprensión de la arquitectura del software y ajustarla según las necesidades del desarrollo *Agile* a gran escala.

El caso [14] crea dos nuevos niveles para gestionar los requisitos. El primer nivel se llama Equipo de Producto. Es responsable del trabajo de hasta siete equipos de desarrollo. Está formada por los *Product Owner* de esos equipos y otros tantos miembros, especialistas en determinados campos. Además, pueden incluirse más expertos en el Equipo de Producto en función del área problemática del Producto.

El segundo nivel lo llaman Equipo de Producto por Área. Este nivel se utiliza para productos más grandes con más de siete equipos de desarrollo. Este nivel se sitúa entre los equipos de desarrollo y los equipos de producto. Las responsabilidades de los Equipos de Producto de Área son similares a las de los Equipos de Producto, excepto las decisiones sobre el presupuesto y el producto final. Por lo tanto, toda la responsabilidad del presupuesto y del producto recae en el Equipo de Producto, y las decisiones operativas del día a día se manejan por los Equipos de Producto de Área.

Las conclusiones del caso [14] muestran que esta jerarquía de tres niveles no funciona como se esperaba debido a la gran complejidad de los productos que la empresa del caso estaba desarrollando. Por ello, la organización sigue intentando dividir sus productos en componentes más pequeños, para utilizar con una jerarquía de dos niveles sin Equipos de Producto de Área. El estudio revela que un máximo de 130 personas puede trabajar en la estructura organizativa de tres niveles.

El caso [17] utiliza el llamado Equipo de *Product Owners*; el equipo incluye a todos los Propietarios de Producto de los equipos individuales, el Gestor de *Portfolio*, el Gestor de Pruebas y el líder de UX. El caso informa de que el uso del Equipo de Propietarios de Producto clarificó las responsabilidades e hizo posible el desarrollo de requisitos funcionales claros para los equipos basados en una profunda comprensión de los requisitos de negocio desde el punto de vista del cliente.

### 5.2.5 *Feature Teams*

Un *Feature Team* (equipos de características) es un equipo multifuncional, estable y de larga duración. Este equipo debe tener los roles necesarios para poder implementar completamente una característica de un Producto e idealmente el equipo debe ser estable durante varios años.

La razón para utilizar *Feature Teams* en el desarrollo ágil a gran escala es minimizar las dependencias entre equipos no autosuficientes. Esto reduce la necesidad de sincronización y ayuda a escalar la gestión de requisitos y enfatiza los valores centrados en

el cliente. Tanto LeSS como SAFe sugieren el uso de *Feature Teams* en lugar de equipos especializados para componentes individuales.

Sólo dos casos [9], [17] mencionan el uso de equipos de características, ninguno de estos casos tuvo éxito. El caso [17] pasa por varias etapas al intentar transformar sus equipos en *Feature Teams*. Al principio, utilizan un equipo piloto interfuncional formado por voluntarios. Esto causa complicaciones, ya que los miembros del equipo piloto, tenían funciones centrales en sus equipos anteriores. Por lo tanto, la organización transforma todos sus equipos en equipos interfuncionales. Además, la organización crea la llamada reserva de competencias. La reserva de competencias proporciona miembros especializados a los equipos de componentes cruzados según las necesidades de las características que el equipo estaba implementando. El problema es que los expertos que forman el grupo de competencias pueden acabar sin trabajo, ya que los equipos no requerían su experiencia para las características que estaban implementando en ese momento. La idea de equipos totalmente interfuncionales no se hizo realidad al final. Debido a los productos desarrollados por la compañía y sus diferencias en requisitos no se cree posible que cualquier equipo pueda desarrollar características en cualquier producto. La organización crea equipos interfuncionales especializados en flujos de negocio específicos. Esta especialización permite que los equipos se centren en un área de producto más reducida, que no requiere el conocimiento de todos los componentes del producto, sino sólo de unos pocos. El conocimiento de los distintos componentes se comparte mediante Comunidades de Práctica y trabajo en parejas.

En el documento [9] la organización resalta que sus equipos ubicados en lugares diferentes desarrollan varias tareas en más de un componente. El uso de *Feature Teams* alivian estos problemas. Para poder generar *Feature Teams* la organización identifica la necesidad de ampliar el conocimiento de los distintos componentes entre los equipos, esto siempre con el fin de que cada equipo pueda hacer un desarrollo integral. La solución que aplica la organización es establecer un programa de intercambio entre equipos. Este programa incluye formación en el aula, visitas de intercambio y sesiones de videoconferencia, así como visitas de larga duración con objetivos claramente especificados. El documento afirma que este programa de intercambio fue muy útil.

#### 5.2.6 *Undone Department*

Tanto LeSS como SAFe están de acuerdo en que los equipos deben ser capaces de crear incrementos alcanzables al final de cada iteración; esto es muy desafiante en el desarrollo ágil a gran escala, especialmente cuando los equipos no tienen experiencia y se están transformando en equipos de características completas. El *Undone Department* es un grupo de equipos que apoyan a los equipos de desarrollo en la consecución de incrementos potencialmente alcanzables, incluso si la definición de hecho de los equipos no es perfecta. El *Undone Department* puede estar compuesto por equipos de pruebas, equipos de garantía de calidad, equipos de arquitectura y otros equipos de apoyo. El *Undone Department* también es necesario debido a las responsabilidades adicionales que el desarrollo ágil aporta a los equipos de desarrollo. En muchos casos [13], [16], [18], las responsabilidades adicionales crean presión en los equipos. Solo dos casos

mencionan el uso de un *Undone Department* [11], [14]. El caso [11] utilizó un equipo de arquitectura formado por 9 especialistas técnicos de alto nivel.

### 5.2.7 Comunidades de práctica

Una Comunidad de Práctica es un grupo de personas que tienen una preocupación común o una pasión por algo, que aprenden a mejorarlo y comparten sus conocimientos y experiencia sobre ello interactuando de forma continua. Esto por ejemplo puede ser una especialidad de la compañía, como pueden ser el diseño, el análisis estructural. Materiales... Tanto LeSS como SAFe sugieren encarecidamente el uso de Comunidades de Práctica en una organización ágil a gran escala.

Cinco de los artículos analizados mencionan el uso de CoP [9], [12], [13], [17], [19]. Ambos casos, [9] y [16] informaron de que las CoP resuelven la falta de información sobre el diseño del sistema, el desarrollo ágil y las herramientas. Del mismo modo, el caso [17] utiliza las CoP para el intercambio de conocimientos y la promoción de las mejores prácticas y el intercambio de conocimientos específicos de los componentes para facilitar la transición a los equipos de características. Los CoP funcionaron bien, pero hubo un problema a la hora de incorporar el conocimiento de las CoP a los *backlogs* de los equipos.

El artículo [20] estudia el uso de Comunidades de Práctica en una organización de desarrollo de software en un entorno ágil de gran tamaño. La organización utiliza CoPs para la formación, revisiones y coordinación del diseño. Las CoP de coordinación y diseño entre equipos sólo fueron útiles para los equipos distribuidos. Los productos desarrollados por un número menor de equipos ubicados en el mismo lugar no requerían estas CoP, y los equipos las consideraban inútiles. Las CoP sobre artesanía del software funcionan bien gracias al líder apasionado y a los temas interesantes. Las CoP sobre la forma de trabajar y el seguimiento del progreso de alto nivel en los equipos tuvieron éxito gracias a una amplia representación de los niveles organizativos.

### 5.3 Desafíos

Esta sección muestra los desafíos que las empresas experimentaron durante su transformación a las metodologías ágiles a gran escala y respondemos a la tercera pregunta de investigación, P3. Se han clasificado los retos identificados en siete grupos; los grupos se muestran en la Tabla 5.

Desafíos	# de casos
Falta de conocimiento, Coaching o formación	10
Resistencia al cambio	8
Problemas de calidad	6
Presión elevada o sobrecarga de trabajo	5
Integración con partes no ágiles	4
Falta de trabajo en equipo y compromiso	4
Medición del progreso	3

*Tabla 5: Retos identificados para el escalado de Agile*

### 5.3.1 Resistencia al cambio

La resistencia al cambio y el apego a los procesos anteriores son retos muy comunes. En 8 artículos [11], [12], [14], [17]–[19], [21] se informa de algún tipo de resistencia al cambio. La resistencia se produce en todos los niveles de las organizaciones, incluidos los equipos de desarrollo y los mandos intermedios y superiores.

Una de las razones por las que los miembros del equipo no quieren cambiar al desarrollo ágil es el aumento de la transparencia [13], [14], ya que la gente se siente observada y no quiere compartir sus problemas. Otra razón de la resistencia es la nueva responsabilidad que el desarrollo ágil supone para los equipos. Se espera que los equipos se auto gestionen en el desarrollo ágil, pero no todos están contentos con ello.

Además, en las grandes organizaciones, la resistencia al cambio en los niveles superiores, como los mandos, es un problema más importante [13], [17]–[19], [21]. Para que la transición tenga éxito, es necesario el apoyo de la dirección en el cambio; varios casos se enfrentan a importantes retos por la falta de apoyo.

En primer lugar, en los modelos ágiles los mandos intermedios como jefes de equipo no tienen una función clara, al promocionarse la autogestión del equipo y la jerarquía horizontal. Este cambio de responsabilidades genera resistencia al cambio ya que se puede pensar que estos mandos intermedios dejan de ser útiles.

En segundo lugar, los directivos no respetan las decisiones de los equipos ni su visión del proceso de desarrollo. En el caso [11] los directivos intentan micro-gestionar los equipos. Esta micro-gestión causa varios problemas, uno de los cuales es que los equipos pierden el interés por las reuniones y dejan de asistir a ellas, ya que no son responsables de la sincronización y la comunicación, y por tanto consideran que las reuniones son inútiles.

Por último, en el caso [21] los equipos están desmotivados por la falta de recursos y de reconocimiento de sus esfuerzos por mejorar el proceso de desarrollo.

### 5.3.2 Problemas de Calidad

Cinco estudios [9], [14], [17], [18] informan de problemas de calidad y de una pérdida de calidad en el código poco después de la transición al desarrollo ágil. La primera causa identificada para este empeoramiento de la calidad es la responsabilidad adicional y la presión elevada en los equipos. Los equipos se saltaron la corrección de errores [13], [18], las pruebas y no utilizaron la integración continua [9], por lo que la deuda técnica se acumula. Los problemas con una gran deuda técnica pueden tardar años en resolverse [9]. La pérdida de calidad provoca el desánimo de los equipos. Por el miedo a ser el foco de la atención o mostrar una baja capacidad de desarrollo, algunos equipos intentan ocultar sus problemas o la falta de avance en sus tareas [18].

Estos problemas se resuelven principalmente con la introducción de algún *Undone Department* y la formación. Por ejemplo, la organización del artículo [13] incluye a probadores y expertos en control de calidad en equipos. Los expertos definen niveles de tolerancia de la deuda técnica para productos individuales; no se pueden añadir nuevas características a un producto cuyos niveles de tolerancia no se cumplieran. Además, el documento [13] propone que los equipos corrijan los errores inmediatamente. El documento [17] sugiere que se establezca un *backlog* común, que se introduzcan roles de apoyo, como los responsables de un subsistema concreto y los arquitectos, que se construya un entorno de pruebas adecuado y que se forme al personal en integración continua. El departamento de la organización [14] incluye el equipo de arquitectura, el equipo de apoyo, que elimina los impedimentos que no podían resolver los equipos de desarrollo, y el equipo experto en *Lean*.

### 5.3.3 Integración con las partes no ágiles de la organización

El desajuste de las estructuras organizativas causa problemas en cuatro casos [9], [13], [14], [17]. Por ejemplo, el problema es que los equipos no ágiles no quieren confiar en los equipos ágiles cuando no saben si estos iban a entregar su trabajo a tiempo.

Estos problemas pueden resolverse incluyendo a las partes de la organización que trabajan en cascada en el proceso de planificación e implicando a los equipos no ágiles en las primeras fases del proceso de planificación [17]. Además, la mejora de los sistemas de integración continua y automatización de pruebas ayuda, ya que permite una integración más rápida y mejor.

### 5.3.4 Compromiso y de trabajo en equipo

Tres casos informaron de problemas de compromiso y el trabajo en equipo [14], [16], [18]. Un caso indica que los problemas con la dedicación de los equipos se deben a la incertidumbre debida a especificaciones erróneas [16].

La falta de trabajo en equipo está principalmente asociada a la especialización de los miembros, tanto previa al modelo ágil como a la desarrollada en los nuevos equipos [18]. Los integrantes de los equipos se centran en los problemas específicos a su

especialización y dejan de lado los posibles problemas del resto del equipo. La definición de "hecho" del equipo también es muy diferente entre los miembros del equipo; los miembros individuales del equipo dan prioridad a diferentes preocupaciones en función de su especialización.

La toma de decisiones sobre el conjunto de las tareas se ve obstaculizada, los diferentes miembros del equipo no quieren comprometerse con las tareas que pueda estar desarrollando otro de los integrantes. El intercambio de conocimientos no funciona porque los especialistas se dan cuenta de que son importantes para la empresa. No quieren que otros desarrolladores obtengan el conocimiento más especializado y esto pueda hacer a los especialistas menos indispensables. Todo esto da lugar a un entorno muy competitivo, por lo que algunos tienen miedo de informar de sus problemas.

En el artículo [14], se explica como la aplicación de *Scrum* hace que la plantilla deje de trabajar en equipo. Esto es debido a que, al hacer las jerarquías más planas, se cree que hay menos oportunidades de desarrollo de carrera, lo que convierte a la gente en más competitiva. Se intenta anteponer el éxito personal frente al grupal, lo que conlleva a un fracaso del modelo de trabajo en equipo.

### 5.3.5 Presión elevada o sobrecarga de trabajo

La alta presión y la carga de trabajo causaron muchos problemas. En 5 casos se señalan problemas relacionados con ella [11], [13], [16], [18], [21]. Cambiar el modo de trabajo, hacer que los integrantes de los equipos asuman nuevos roles o funciones y hacerlo mientras que la presión por la entrega de producto y mercado se mantiene o crece hace que los empleados sufran de una presión adicional. Esta tensión da lugar a un código de mala calidad y a una planificación tardía [16]. Los equipos también se saltan las pruebas, la refactorización, la corrección de defectos y las prácticas de ingeniería, creando así una gran deuda técnica [13], [18].

Además, como los equipos que son nuevos en el desarrollo ágil aún no se han dado cuenta de la importancia de la mejora de los procesos, el exceso de presión hace que la dejen fuera de sus responsabilidades [11], [18], [21]. El déficit de tiempo hace que los equipos no se preocupen por los problemas identificados en sus Retrospectivas. Por lo tanto, estos problemas nunca llegan a la cima de sus *backlogs*. En el caso del artículo [11], se identifica que los equipos sin experiencia no han tenido tiempo suficiente para generar e interiorizar la visión de *Scrum*, se concluye que el tiempo de adaptación en estos casos es mayor de lo esperado.

Además, la alta presión impide la comunicación en los equipos. Los equipos empiezan a saltarse y posponer las reuniones. Esto hace que la periodicidad de las reuniones se vaya reduciendo, hasta el punto de que las que se tienen no son suficientes para tener una adecuada comunicación.

### 5.3.6 Falta de conocimientos, Coaching o formación

El reto más común entre los artículos analizados es la falta de formación, y entendimiento de los fundamentos de los procesos. Estos problemas se señalaron en 10 casos [9], [11]–[14], [16]–[19], [21].

Varios factores causan razones para la falta de entrenamiento y conocimiento, la mayoría de las veces se debe a la subestimación de la dificultad de la transformación ágil, las limitaciones financieras o una transición apresurado. Por ejemplo, la organización [21] subestima la dificultad de la adopción de *Agile*. La organización no tiene suficientes conocimientos en sus filas, pero se niega a recurrir a un experto externo debido a las limitaciones financieras. En su lugar, los directivos que son nuevos en metodologías ágiles asumen el papel de expertos en *Agile*, lo cual puede ser perjudicial para todo el proceso.

La falta de entrenamiento y formación tiene varias consecuencias. Los equipos no pueden cambiar sus hábitos tan fácilmente. En el caso [14] se resalta que debido a la intención es que la transición sea rápida, se hace el cambio sin analizar el sistema en profundidad, lo que hace que según avanza la transición la organización hace suposiciones basadas puramente en la teoría en vez de analizar las necesidades del sistema. Además, el caso informa de que los equipos que recibieron ‘coaching’ tuvieron un mejor rendimiento que los equipos sin tutoría.

La consecuencia de la falta de conocimiento es una aplicación errónea de las prácticas ágiles. Por ejemplo, en tres casos la gente no entiende el propósito de las reuniones [11], [13], [18]. Los equipos no encuentran las causas subyacentes en las retrospectivas, sino que se limitan a identificar los síntomas. Además, los equipos no respetan los principios básicos de las prácticas que deben aplicar, como el respeto a la iteración en *Scrum* o la falta de contacto con el cliente.

Aunque los equipos tienen toda la información sobre las prácticas, no la entendieron y la utilizaron de forma inadecuada. Por ejemplo, la organización [12] experimentó una situación en la que un gerente impone las prácticas de *Scrum* de forma estricta, según el manual, y las fuerza contra su voluntad. Además, algunos equipos siguieron las prácticas demasiado religiosamente y perdieron la perspectiva.

Además, la transición a *Agile* a gran escala requiere también formación en conocimientos de dominio. El caso [9] ha tenido que formar a equipos de otras áreas de dominio para establecer equipos de características, debido a la anterior arquitectura basada en componentes de su producto.

Un entrenamiento adecuado de los directivos aumenta el apoyo de éstos a los métodos ágiles. Así, se redujeron los problemas causados por la falta de patrocinio ejecutivo.

### 5.3.7 Scaled Requirement Management

En los grandes proyectos, los requisitos no pueden ser gestionados por una sola persona (*Product Owner*), por lo que es necesario dividir la responsabilidad de su gestión. Decidir cómo se dividen los requisitos supone un reto para las compañías, similar al problema de definir Áreas de Producto, [9], [11], [14], [17].

En los artículos se ha podido observar que, para escalar el modelo ágil, es indispensable hacer un escalado de la gestión de los requisitos. Este paso ha llevado al fracaso de la adaptación a varias compañías, lo que resalta la importancia de hacer una definición y asignación correcta de los requisitos. El dividir requisitos de una manera que los equipos puedan trabajar de manera independiente, aportando a un requisito de mayor nivel global supone un gran reto para la compañía.

### 5.3.8 Medición del progreso

Hacer cambios en el proceso de desarrollo suele estar relacionado a una necesidad de la mejora en la compañía, desde el momento en el que se decide someter a la compañía a un cambio de este tipo se desea medir el progreso del cambio y su efecto. Sin embargo, esto es un reto en el desarrollo ágil a gran escala, ya que tres casos [12], [16], [17] experimentan dificultades a la hora de supervisar y medir el progreso de su desarrollo ágil. El mayor problema es la complicación para encontrar las métricas adecuadas. Las organizaciones no saben qué medir para obtener resultados significativos.

La medición del progreso de la adaptación y el impacto de este en la organización es importante incluso para el proceso de adaptación. Obtener resultados satisfactorios puede derivar en una mayor inversión por parte de la dirección en cosas esenciales como la formación. Por otra parte, ser capaz de medir el progreso en caso de no ser satisfactorio permite redirigir la metodología en caso de ser necesario.

## 5.4 Factores de éxito

Esta sección responde a la cuarta pregunta de investigación: P4. Se identifican los factores de éxito en siete grupos. Todos los grupos, junto con el número de casos que se identificaron, se muestran en la Tabla 5.

Factores de éxito	# de casos
Formación	11
Trabajo en equipo	7
Visión única en valores y practicas	6
Transformación cautelosa	6
Herramientas e infraestructura	5
Patrocinio ejecutivo	5
Prácticas de ingeniería solidas	2

*Tabla 5: Factores de éxito identificados*

#### 5.4.1 Formación

Dado que el reto más señalado es la falta de conocimientos, no es de extrañar que el factor de éxito más señalado sea la obtención de los mismos. La adquisición y el intercambio de conocimientos de forma exhaustiva, profunda y sistemática se declaran como factores de éxito en 11 casos [9], [10], [12]–[19], [21].

La forma más recomendable de adquirir conocimientos y experiencia es según los artículos contratar a un experto externo con una amplia y profunda familiaridad con desarrollo ágil [10], [12]–[14], [19], [21]. El experto externo comparte sus conocimientos con varios empleados internos, que luego los difundieron por toda la organización. Una de las razones por las que resulta más beneficioso traer a un experto que invertir en la adquisición de conocimientos de otras fuentes, es que las organizaciones se dan cuenta de que las sugerencias generales y teóricas no son fácilmente aplicables en su contexto particular, y que se necesita una comprensión más profunda de los métodos [10], [16].

Otra recomendación es la de hacer un coaching profundo y amplio a las personas. Por ejemplo, en el caso [10] se afirma que el coaching tiene que referirse tanto a los valores como a los procesos, pero, los valores suelen omitirse. La falta de comprensión de los fundamentos básicos del proceso a implementar puede hacer que no se sepa responder a situaciones inesperadas o desconocidas. *Agile* debe ser flexible, pero su adaptación requiere un profundo conocimiento de sus valores. Es mejor que las personas se comprometan con el aprendizaje que con un método o principio ágil concreto [18].

Además, es valioso enseñar a los equipos a aprender, por ejemplo, utilizando la formación de doble bucle, para que sean mejores a la hora de encontrar la causa subyacente de un problema y puedan ajustar sus procesos de equipo correctamente [18]. Según dos casos [12], [19], es importante entrenar a las personas y no dictarlas. Se recomienda utilizar un lenguaje que deje margen a los equipos a adaptar la transformación con el uso del "debería" en lugar del "debe", por ejemplo. Para que la transformación ágil tenga éxito, es imprescindible contar con el apoyo de los empleados.

Además, para que la adopción de las metodologías Ágiles a gran escala tenga éxito, se requiere una formación sistemática de los entrenadores y otras personas dentro de la organización [3], [10], [19], [21]. La formación y el entrenamiento sistemáticos significan tener eventos o despliegues periódicos, en los que un experto tanto interno como externo, o las comunidades de práctica puedan entrenar a los empleados o resolver las posibles dudas existentes. Además, la formación sistemática garantiza la disponibilidad de suficientes entrenadores en una organización. Centrar toda la transformación en pocos expertos hace que estos puedan estar saturados por las dudas de los equipos, lo que puede hacer que la adopción se descontrole.

Además, el llamado coaching por parejas, que combina a los expertos en *Agile* con un experto con conocimientos del dominio para permitir un coaching más eficiente y preciso, demostró funcionar bien. Una forma muy útil de difundir conocimientos de forma no invasiva en una organización fueron las Comunidades de Práctica. Esto se identifica como un factor de éxito en dos de los artículos analizados [12], [17].

#### 5.4.2 Visión conjunta de los valores y las prácticas

Para que la transición ágil tenga éxito, es necesario definir una visión común sobre el cambio. En total, seis casos consideraron beneficiosa la unificación de valores, definiciones, forma de trabajar y entendimiento [9], [10], [13]–[15], [17].

Es necesario definir adecuadamente las funciones, sus responsabilidades y las definiciones comunes. Es importante aclarar qué conceptos son equivocados y que tipo de cosas se consideran errores. Tener una visión unificada sobre este tipo de cosas hace que los equipos estén predispuestos a seguir la visión unificada sobre *Scrum*. El documento [14] afirma que es beneficioso utilizar un lenguaje común entre los equipos, los arquitectos y los *Product Owners*, como las definiciones ágiles establecidas, el Lenguaje Unificado de Modelos (UML) o los Conceptos Fundamentales de Modelado (FMC).

#### 5.4.3 Herramientas e infraestructura

Las organizaciones tienen que estar preparadas para proporcionar recursos suficientes a los equipos durante la transformación al desarrollo ágil. Cinco casos [9], [14]–[17] afirman que las herramientas y la infraestructura decentes son útiles.

Las herramientas incluyen, por ejemplo, entornos de desarrollo y pruebas locales, integración continua y pruebas automatizadas, e infraestructura de comunicación que incluye equipos de videoconferencia. Las herramientas y la infraestructura comunes resultan especialmente beneficiosas en entornos distribuidos, donde los equipos necesitan toda la ayuda posible para paliar sus impedimentos de comunicación.

#### 5.4.4 Prácticas de ingeniería sólidas

Como se identifica en la sección 5.3, la pérdida de la calidad es un problema identificado en varias compañías durante la transición al modelo ágil. Esto se puede evitar

teniendo prácticas y procesos de ingeniería sólidos, que obliguen a mantener la calidad durante todo el proceso [12], [13].

El caso [13] sugiere incluir a los responsables de control de calidad y a los probadores en los equipos de desarrollo. Además, sugiere una buena gestión de la deuda técnica, como niveles de tolerancia e indicaciones visuales para los niveles de tolerancia.

#### 5.4.5 Precaución en la transformación

Cambiar una organización hacia el desarrollo ágil significa cambiar la mentalidad de las personas. Este cambio puede llevar mucho tiempo. Por ello, en seis casos fue necesaria una transformación lenta y cuidadosa [12]–[15], [18], [21].

Es mejor percibir la transformación como un cambio organizativo a largo plazo. La organización debe mejorar su desarrollo gradualmente y tomarse al menos tres meses antes de cambiar sus métodos y prácticas. Es esencial que las organizaciones antes de lanzar la transición dispongan de un plan de despliegue, disponibilidad de recursos y medición del posible impacto en cada entorno de la compañía.

Es importante asumir por parte de la organización que el tiempo necesario por los equipos para adaptarse puede variar, no presionar en exceso a los equipos ayuda a reducir la tensión y hacer que la transición se pueda dar con más éxito.

#### 5.4.6 Apoyo al trabajo en equipo

Siete casos informaron [9], [11], [12], [15], [16], [18], [19] de que las conexiones estrechas y la comunicación constante entre los equipos y sus miembros son necesarias para el éxito del desarrollo ágil.

La organización debe establecer un entorno transparente que permita la apertura del equipo sin miedo a discutir los problemas para mejorar el trabajo en equipo. Además, es mejor que los equipos sean pequeños.

La comunicación entre los equipos y los miembros del equipo puede mejorarse mediante las reuniones de *Scrum of Scrums* y las Comunidades de Práctica.

#### 5.4.7 Patrocinio ejecutivo

El patrocinio ejecutivo es un factor fundamental para el éxito, si no existe este patrocinio es muy probable que se dé un menosprecio de la complejidad de la transformación y que la asignación de los recursos no sea la suficiente. Sin un patrocinio ejecutivo adecuado, no se podrían realizar otros factores de éxito, ya que no se asignarían suficientes recursos. En cinco casos se afirma que el patrocinio ejecutivo y el apoyo de la dirección son imprescindibles [9], [13], [15], [18], [19].

Obtener el patrocinio de los ejecutivos puede ser problemático si la transición a *Agile* es un incentivo de abajo hacia arriba.

## 6 Estudio de la adopción de las metodologías Ágiles en un grupo empresarial

La organización seleccionada es un grupo empresarial del sector aeronáutico centrada en el diseño y fabricación de turbinas de gas. Esta es la primera diferencia respecto a la mayoría de artículos existentes sobre metodologías ágiles. La mayor parte de la bibliografía y procesos definidos en las metodologías ágiles están enfocadas para empresas desarrolladoras de software. Esto supone un reto a la hora de adoptar *Agile* ya que hay partes de los procesos que se tienen que adaptar al tipo de proyectos del grupo.

Aunque el *Portfolio* esté dividido por sectores, proyectos y tipos de trabajos (diseños propios, subcontrataciones, soporte en servicio...), simplificando, se han clasificado las actividades de la empresa en tres grandes grupos, el I+D+I, el diseño de componentes y desarrollo de procesos (tanto de fabricación como ingeniería) y la producción y soporte en servicio.

Aunque el I+D+I tiene partes que requieren de una gestión de proyectos, debido a que la actividad principal es un proceso creativo difícil de planificar, esta sección no analizará la introducción de *Agile* en este entorno.

Por otra parte, la optimización y gestión de la producción en este grupo empresarial se controla mediante prácticas de *Lean manufacturing*, es por esto que esta sección del Trabajo Fin de Master se centrará en la adopción de las metodologías ágiles enfocadas en la actividad de diseño de componentes y desarrollo de procesos.

Este grupo empresarial está en las fases iniciales de adopción del modelo ágil, la motivación para implementar *Agile* se basa en gran parte en la necesidad de adaptarse a entornos cambiantes continuamente. La pandemia del COVID19 tuvo un impacto directo sobre el sector de la aviación y obligó a las empresas a adaptarse. Aunque la adaptabilidad no es algo novedoso y siempre se dice que los que no se adaptan desaparecerán, eventos como este remarcan la necesidad de la capacidad para adaptarse.

La primera fase de la adopción se ha centrado en dar formación a los mandos sobre el modelo ágil. La formación es el factor de éxito y reto con más menciones identificados en el estudio de factores de éxito y retos del apartado 5.3 y 5.4. No entender las bases de las metodologías puede conllevar a una inadecuada adopción del modelo ágil y fracaso del mismo. Aunque actualmente la formación se haya dado únicamente a los mandos, se encuentra disponible para cualquier empleado. Siendo la intención de *Agile* que los equipos se autogestionen es esencial que todos los integrantes de los equipos tengan formación. Por otra parte, se considera oportuno formar primero a los “líderes” de los equipos ya que esto es un grupo reducido, que permite formar en un tiempo menor para que luego puedan hacer una función de ‘Coach’ dentro de los equipos.

En base al análisis que ha llevado a cabo la organización sobre su cartera de proyectos, se ha decidido que aplicar exclusivamente el modelo ágil puede no ser la mejor opción. Es por esto que la propuesta es aplicar un modelo híbrido. Como se puede

ver en la Figura 6 en función de la definición de los requisitos y conocimiento de la tecnología algunas metodologías de gestión son más apropiadas que otras. Para poder aplicar varias metodologías en una misma compañía es esencial que todos los integrantes de los equipos entiendan los fundamentos de las metodologías.

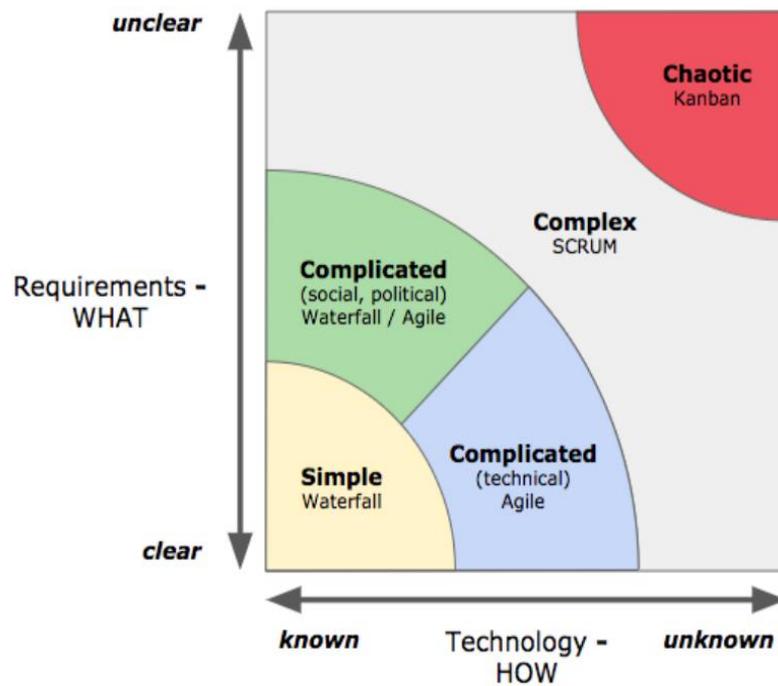
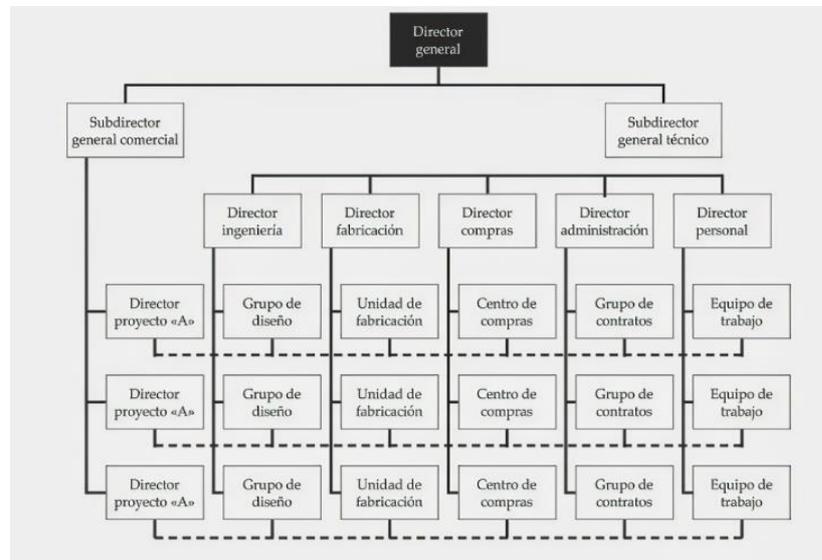


Figura 6: Modelo de gestión más adecuado según el conocimiento y la claridad de los requisitos

#### 6.1.1.1 Magnitud y distribución de la organización

El grupo seleccionado actualmente está compuesto por más de 4000 personas que están repartidas en 17 centros de trabajo. Estos centros de trabajo a su vez están repartidos en 5 países, lo que hace que los equipos de trabajo aparte de multidisciplinares sean multiculturales.

La base sobre la que se organiza el grupo empresarial es una estructura de organización matricial como la que se puede observar en la Figura 7. En una organización matricial los recursos se comparten desde cada disciplina. Este tipo de organización obtiene grupos altamente especializados de los que las personas pueden trabajar simultáneamente en varios proyectos. Jerárquicamente la organización posee 4 niveles principales, directores, Head, Manager (jefes de Producto o sección) y técnicos (tanto ingeniería como taller).



*Figura 7: Ejemplo organización matricial*

Una organización matricial promueve la creación de equipos multidisciplinares y ayuda a evitar los silos. Esto es debido a que al no organizarse por departamentos y promover la movilidad de los recursos, se facilita la comunicación fluida. Como punto negativo, una organización matricial corre el riesgo de sufrir luchas de poder entre proyectos, poniendo en riesgo el cumplimiento de las tareas más beneficiosas para la organización por el afán de conseguir objetivos personales.

Debido a que una turbina de gas cuenta con un alto número de componentes, con funciones y requisitos completamente diferentes, la organización matricial se divide especializada por tipos de componentes. Al segmentar la organización en estos grupos se corre el riesgo de generar silos, lo que en teoría es algo que tiende a evitar una organización matricial. Esto supone otro reto para la adopción de la metodología ágil en este grupo empresarial.

#### 6.1.1.2 Tipos de proyectos

Centrados en la actividad de Diseño de componentes y Desarrollo de procesos, el grupo tiene mayormente 2 tipos de proyectos, proyectos en los que la empresa es un socio a riesgo y beneficio compartido con otras entidades para el diseño de una turbina de gas y proyectos de subcontratación.

La mayor parte del *Portfolio* de la compañía está compuesta por proyectos en los que el grupo es un socio a riesgo y beneficio compartido. Una turbina de gas se divide en diferentes módulos (ver Figura 8), cada uno de ellos posee una función específica y es altamente compleja. Desarrollar todas las partes por una misma compañía requiere una alta cantidad de recursos y conocimiento, por lo que es habitual que varias compañías colaboren en este tipo de proyectos.

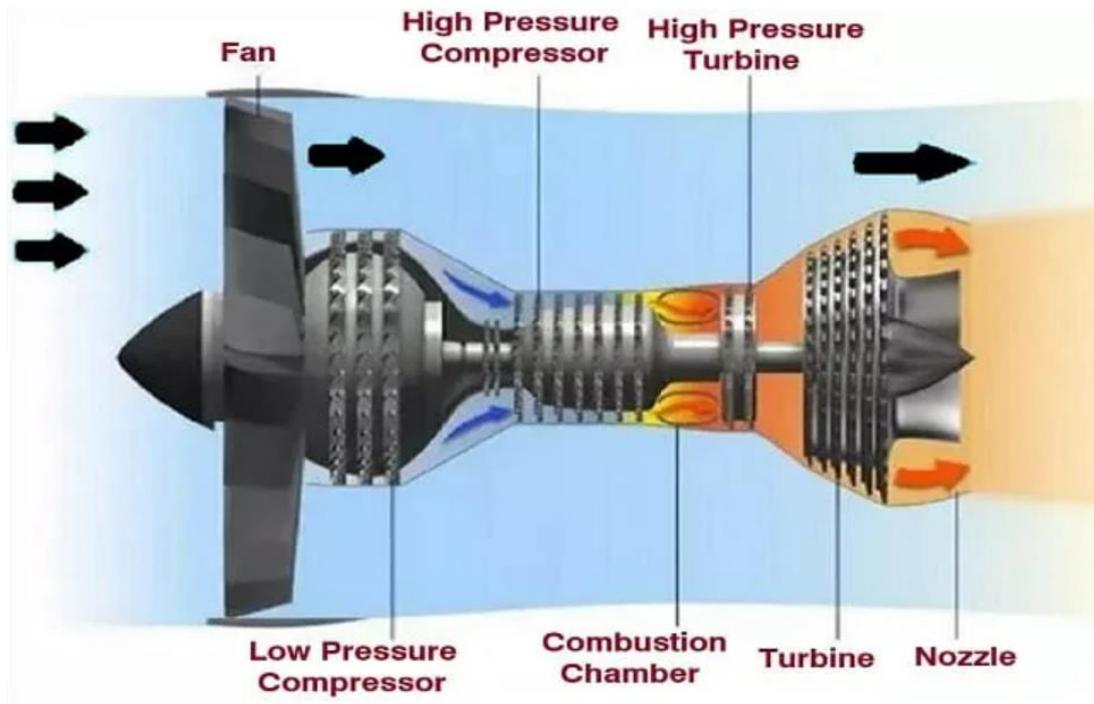


Figura 8: Ejemplo de división de una turbina por módulos

Desarrollar una turbina en conjunto con otras organizaciones convierte en un pilar esencial para el éxito del proyecto a la comunicación. Durante el desarrollo del diseño, los requisitos pueden cambiar tanto por una petición del cliente (aerolíneas), como por petición de otras partes del motor. Es por esto que las organizaciones involucradas en este tipo de proyectos, deben ser capaces de asumir cambios de requisitos en cualquier fase del proyecto.

Por otra parte, la organización también desarrolla proyectos de subcontratación. Estos pueden ser el diseño de un único componente, desarrollar el proceso de fabricación de un componente altamente complejo, o incluso fabricar componentes diseñados por otros. Este tipo de proyectos exigen a la organización una alta capacidad de respuesta y flexibilidad. Al ser contratos de subcontratación, se compite con otras organizaciones para obtenerlos, esto obliga a que los “lead time” ofertados (aparte del precio) tengan que ser competitivos, implica una flexibilidad en los recursos de la compañía para poder asumir un proyecto de este tipo en cualquier momento.

Un problema de intentar integrar proyectos de subcontratación en cualquier momento, es que se suelen asignar recursos que están en ese momento involucrados en otros proyectos, pudiendo llegar a saturar los recursos, lo que acaba siendo un problema para la flexibilidad de los mismos y su priorización.

Debido a las implicaciones que tiene trabajar en el sector aeronáutico, la exigencia en todas las fases del proyecto es máxima. Las implicaciones que puede tener

cualquier error durante cualquier fase del proyecto pueden llegar a ser catastróficas; es por esto que todas las actividades siguen unos procedimientos exhaustivos para asegurar la calidad, el desarrollo de un diseño se audita constantemente (con procedimientos como APQP [22] por ejemplo) y los recursos deben cumplir unos elevados requisitos de cualificación hasta poder ser independientes.

#### 6.1.1.3 Planificación de proyectos

La definición y planificación de los proyectos más grandes comienza con la definición de los pilares fundamentales, estos pilares marcan el camino hacia los resultados que aportan valor al proyecto / *Portfolio*. Los pilares suelen cubrir todo el proyecto, 1 pilar puede ser por ejemplo entregar la primera turbina del proyecto. Como desde el *kick off* del proyecto hasta que se entrega la primera turbina pueden pasar años, se genera un nivel inferior a los pilares que es el de los hitos.

En base a los pilares definidos, se definen los hitos del proyecto, los hitos son las tareas a gran nivel que aportan valor a los pilares previamente definidos. La definición de hitos es anual, se pone una fecha límite para alcanzarlos en el año. Los hitos se clasifican por niveles, hitos de compañía, hitos de proyecto, de sección... Cada hito se puede considerar como un proyecto en sí mismo, para cada hito se hace una planificación de las tareas principales que se tienen que llevar a cabo para cumplirlo. Estas tareas pueden afectar a varios departamentos y equipos por lo que en base a la planificación de los hitos cada equipo desarrolla los planes propios para cumplir sus tareas.

Los equipos acaban planificando de manera independiente (dentro de los límites establecidos por el hito) sus tareas y para ello se asume una planificación con ritmo semanal, similar a un *Sprint* de *Scrum* pero con una duración inferior. Los equipos hacen revisiones semanales del avance y planificación de la próxima semana siendo está transparente.

Para llevar la planificación semanal la organización deja abierto el uso de herramientas como MS Project, Excel, Gantt, Planner... la decisión de la herramienta por lo general la toma el manager de cada sección o proyecto, la recomendación general es utilizar *Planner*.

Planner es una herramienta que planifica las tareas utilizando la metodología de *Kanban*. Se genera un *backlog* donde se recogen todas las tareas a realizar durante el proyecto, después semanalmente se cogen del *backlog* las tareas a completar esa semana. Este tipo de planificación tiene como objetivo principal generar una representación visual de la planificación fácilmente revisable (ver Figura 9).

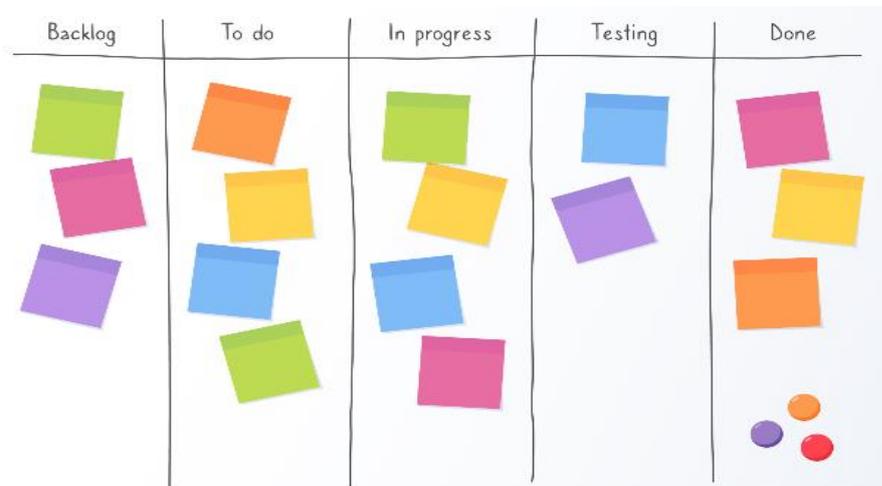


Figura 9: Ejemplo gestión de tareas en Kanban

Debido a la organización actual y el alto número de especialistas compartidos entre proyectos, la planificación de las tareas entre equipos se convierte en un problema. Debido al deseo de cumplimiento de indicadores personales de cada *Product Owner*, se corre el riesgo de cumplir tareas que no sean las más beneficiosas para la organización.

Para solventar esto, la organización debería implementar una metodología similar a *Scrum of Scrums*, esta ha sido la metodología más aplicada por las compañías en la sección 5.2, la Figura 10 muestra un ejemplo de *Scrum of Scrums*. Otras metodologías como el Steering Committee también pueden ser válidas. Estas metodologías organizan los *Sprints* por niveles, el nivel de equipo se planifica su semana en función del *backlog* que tienen y las necesidades que han recibido desde cada proyecto. Después un representante de cada equipo asiste

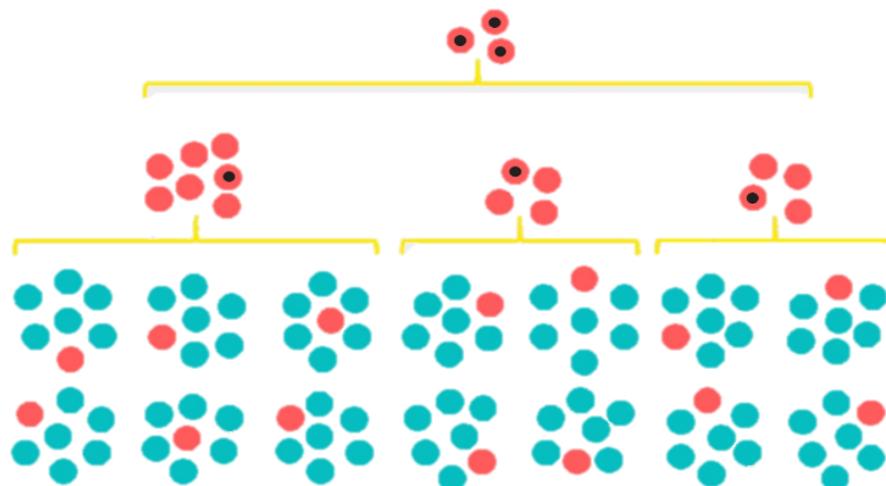


Figura 10: Ejemplo del modelo de escalado en Scrum of Scrums

#### 6.1.1.4 *Personas y equipos*

El nivel de Técnicos de este grupo empresarial está compuesto en su mayoría por personas con formaciones de ingeniería y formación profesional (FP). Aunque en el día a día estos perfiles trabajan juntos en muchos casos, por lo general los perfiles de FP están más enfocados a las tareas de producción, y son los perfiles de ingeniería los más utilizados en las tareas de Diseño de componentes y desarrollo de procesos. Es por esto que esta sección se centra en los perfiles de ingeniería.

Las personas están agrupadas en secciones de ingeniería, estas secciones están definidas por el tipo de Producto y son independientes respecto a los proyectos. Cada sección contiene diferentes perfiles de especialistas; como pueden ser diseñadores, analistas, ingenieros de fabricación etc. Como se menciona en la sección 6.1.1.2, los especialistas van progresando en un sistema de cualificaciones que les permite asumir diferentes partes del proyecto de manera independiente. El primer nivel 'L1' de cualificación permite al especialista redactar documentación, pero debe ser revisado por una persona con una cualificación superior (L2 por ejemplo). Conseguir que un especialista sea independiente conlleva años de experiencia (2 como mínimo), esto hace que la organización sea reacia al cambio, lo que puede impactar en la motivación de los empleados.

Los recursos de cada sección se asignan a diferentes proyectos en función de la necesidad, trabajar en diferentes proyectos puede ser un aliciente para la motivación personal de un especialista, ya que aporta diferentes entornos y novedades al día a día.

En las fases iniciales de un proyecto, donde la carga de diseño es más elevada, la dedicación de los especialistas puede ser exclusiva a un único proyecto, con el avance del proyecto esto se va modificando y los especialistas acaban siendo compartidos. Los equipos se crean y se deshacen en función de los proyectos, en vez de cambiar un equipo por completo de proyecto se van metiendo y sacando los recursos.

Este tipo de gestión de recursos tiene 2 inconvenientes con el planteamiento ágil; no permite generar equipos de larga duración capaces desarrollar un incremento del proyecto independientemente. Por otra parte, hace que los recursos sean compartidos entre proyectos, lo que supone un reto a la hora de planificar y priorizar tareas. Esto va en contra de los pilares generales de *Agile* donde los equipos son multidisciplinares de larga duración y no se comparten entre proyectos.

Debido a la alta exigencia del sistema de cualificación, cuando un especialista consigue una delegación superior, es habitual que sus responsables no quieren que cambien de proyecto o especialidad. Esto puede impactar el nivel de motivación de los empleados, proyecta una idea de que la promoción siendo especialista es muy difícil y que incluso cambiar de puesto dentro de la organización puede ser un reto. Lo ideal para una organización que pretende ser ágil es tener las denominadas personas en 'T' [23] (ver Figura 11). Estas personas tienen un área de especialización (idealmente alineada

con sus intereses personales) profunda, además, tienen una base de conocimientos generales que les permiten estar más involucrados en los proyectos.

## Cross Functional Knowledge

**T SHAPED EMPLOYEE**



*Deep Expertise*

*Figura 11: Representación visual de una persona en T*

Aunque la obtención de los conocimientos generales tiene una base persona dependiente, suele estar ligada a personas con curiosidad por el conocimiento, lo que hace que se forme en varios campos. La organización que busca tener este tipo de empleados debe tener un sistema que lo fomente. Similar al factor de éxito de las comunidades de práctica, esto se puede obtener mediante un sistema de rotaciones por especialidades entre empleados y un buen sistema de formación.

Debido a la alta carga laboral de la organización y la optimización de los recursos, a día de hoy las rotaciones en este grupo son difíciles de hacer. Por otra parte, debido a que los recursos pueden estar saturados, la formación de un nuevo integrante puede ser lenta, ya que las personas con el conocimiento no disponen del tiempo para transmitirlo correctamente. Para solucionar esto, se propone crear una mentalidad de comunidad de práctica, se propone liberar a un empleado (quitarlo del presupuesto de horas) altamente especializado por área, su función principal es transmitir el conocimiento a nuevos integrantes, y revisar el trabajo realizado por ellos hasta que progresen en el sistema de cualificaciones. Esto crea un entorno en el que se fomenta la rotación, se dispone de un recurso dispuesto a formar a la gente y encima con el tiempo para hacerlo. Por otra al estar la persona liberada, en el caso de imprevistos que necesiten una respuesta rápida, se dispone de una persona altamente cualificada para hacerle frente.

## 7 Conclusiones

El TFM ha estado enfocado al escalado de las metodologías ágiles en grandes organizaciones, para ello inicialmente se definieron los siguientes objetivos:

- Revisar y comparar 2 metodologías existentes para el escalado del modelo ágil en grandes organizaciones.
- Identificar desafíos y factores de éxito para el escalado del modelo *Agile*.
- Análisis de la adopción del modelo ágil en una gran organización existente

La primera parte del TFM se ha centrado en el análisis de las 2 metodologías de escalado del modelo *Agile*. Las dos metodologías seleccionadas han sido ‘*Scaled Agile Frameworks*’ SAFe y ‘*Large Scale Scrum*’. SAFe tienen una definición muy detallada de todo el proceso y organización. Define una organización con 3 niveles de jerarquía en los que se hace una alta definición de cargos. LeSS por lo contrario, presenta un planteamiento más sencillo, en el que no se hace una definición completa de la organización y procesos. Define unos valores a utilizar como base para la organización, pero permite la adaptación de todo el proceso a las necesidades de cada empresa.

El apartado 5 analiza los marcos de escalado utilizados por las grandes organizaciones y los desafíos y factores de éxito identificados durante el proceso de adopción. Se ha visto que el intentar implementar modelos de escalado demasiado complicados ha resultado en fracaso en las organizaciones que lo han intentado. Por otra parte, las organizaciones también resaltan que por lo general han necesitado hacer adopciones a los modelos de escalado seleccionados. Estos dos argumentos refuerzan el planteamiento que tiene LeSS sobre el escalado, haciéndolo una mejor opción frente a SAFe. En cuanto a los retos más mencionados por las compañías son la falta de formación o coaching, la resistencia al cambio y los problemas de calidad. Los factores de éxito más repetidos han sido la formación, el trabajo en equipo y hacer las transiciones con cautela.

En el apartado 6 se ha analizado la transición que está teniendo lugar en un gran grupo empresarial actualmente. La primera observación es que no se ha adoptado un marco de escalado, al encontrarse el proceso en las fases iniciales, se ha optado por la difusión de formación sobre el modelo ágil. Vistos los resultados del análisis sistémico, este planteamiento tiene potencial, ya que la primera causa de fallo ha sido una mala selección del marco de escalado y la falta de formación. En cuanto a seleccionar un marco para un escalado en el futuro la recomendación sería utilizar LeSS. El mayor reto con el que se va a encontrar la empresa es el alto número de recursos compartidos, no tiene equipos multidisciplinares de larga duración, lo que es un pilar de *Agile*. Por otra parte, debido a los diferentes tipos de proyectos que contiene en su cartera, aunque el planteamiento de la organización es dejar abierto a los equipos del modelo de organizarse, se recomienda elegir unas herramientas y prácticas generales de

organización. Esto está enfocado en que los empleados no tengan que formarse en varios sistemas y de esta manera ser más versátiles entre proyectos.

## 8 Referencias

- [1] «15th Annual State Of *Agile* Report | Digital.ai». <https://digital.ai/resource-center/analyst-reports/state-of-Agile-report> (accedido 12 de mayo de 2022).
- [2] T. Dybå y T. Dingsøy, «What Do We Know about *Agile* Software Development?», *Softw. IEEE*, vol. 26, pp. 6-9, nov. 2009, doi: 10.1109/MS.2009.145.
- [3] D. Mishra y A. Mishra, «Complex software project development: *Agile* methods adoption», *J. Softw. Maint. Evol. Res. Pract.*, vol. 23, pp. 549-564, dic. 2011, doi: 10.1002/smr.528.
- [4] B. Boehm y R. Turner, «Management challenges to implementing *Agile* processes in traditional development organizations», *IEEE Softw.*, vol. 22, n.º 5, pp. 30-39, sep. 2005, doi: 10.1109/MS.2005.129.
- [5] «SAFe 5.0 Framework», *Scaled Agile Framework*. <https://www.scaledagileframework.com/> (accedido 12 de mayo de 2022).
- [6] «Overview», *Large Scale Scrum (LeSS)*. <https://less.works/> (accedido 12 de mayo de 2022).
- [7] A. Centre, «Practices for Scaling *Lean & Agile* Development», *Agile Centre - Certified Scrum, Agile Leadership, Scaled Agile Training & Consultancy*, 27 de marzo de 2018. <https://www.agilecentre.com/resources/book/practices-for-scaling-lean-agile-development/> (accedido 12 de mayo de 2022).
- [8] K. Petersen, R. Feldt, S. Mujtaba, y M. Mattsson, «Systematic Mapping Studies in Software Engineering», presentado en 12th International Conference on Evaluation and Assessment in Software Engineering (EASE), jun. 2008. doi: 10.14236/ewic/EASE2008.8.
- [9] M. Paasivaara, C. Lassenius, V. T. Heikkilä, K. Dikert, y C. Engblom, «Integrating Global Sites into the *Lean* and *Agile* Transformation at Ericsson», en *2013 IEEE 8th International Conference on Global Software Engineering*, ago. 2013, pp. 134-143. doi: 10.1109/ICGSE.2013.25.
- [10] S. Hanly, L. Wai, L. Meadows, y R. Leaton, «*Agile* coaching in British Telecom: making strawberry jam», en *AGILE 2006 (AGILE'06)*, jul. 2006, p. 9 pp. - 202. doi: 10.1109/AGILE.2006.13.
- [11] M. Paasivaara y C. Lassenius, «Scaling *Scrum* in a Large Globally Distributed Organization: A Case study», en *2016 IEEE 11th International Conference on Global Software Engineering (ICGSE)*, ago. 2016, pp. 74-83. doi: 10.1109/ICGSE.2016.34.
- [12] G. Benefield, «Rolling Out *Agile* in a Large Enterprise», en *Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008)*, ene. 2008, pp. 461-461. doi: 10.1109/HICSS.2008.382.
- [13] K. Long y D. Starr, «*Agile* Supports Improved Culture and Quality for Healthwise», en *Agile 2008 Conference*, ago. 2008, pp. 160-165. doi: 10.1109/Agile.2008.61.



- [14] J. Schnitter y O. Mackert, «Large-Scale *Agile* Software Development at SAP AG», en *Evaluation of Novel Approaches to Software Engineering*, Berlin, Heidelberg, 2011, pp. 209-220. doi: 10.1007/978-3-642-23391-3\_15.
- [15] B. Schatz y I. Abdelshafi, «Primavera Gets *Agile*: A Successful Transition to *Agile* Development», *Softw. IEEE*, vol. 22, pp. 36-42, jun. 2005, doi: 10.1109/MS.2005.74.
- [16] «Inter-organizational Co-development with *Scrum*: Experiences and Lessons Learned from a Distributed Corporate Development Environment | Semantic Scholar». <https://www.semanticscholar.org/paper/Inter-organizational-Co-development-with-Scrum%3A-and-Vallon-Strobl/ae9f5e36df6df88b90fe08c20f0f4707df9fe4b7> (accedido 12 de mayo de 2022).
- [17] M. Paasivaara, B. Behm, C. Lassenius, y M. Hallikainen, «Towards Rapid Releases in Large-Scale XaaS Development at Ericsson: A *Case study*», en *2014 IEEE 9th International Conference on Global Software Engineering*, ago. 2014, pp. 16-25. doi: 10.1109/ICGSE.2014.22.
- [18] N. Moe, «Key Challenges of Improving *Agile Teamwork*», jun. 2013. doi: 10.1007/978-3-642-38314-4\_6.
- [19] A. Atlas, «Accidental Adoption: The Story of *Scrum* at Amazon.com», sep. 2009, pp. 135-140. doi: 10.1109/AGILE.2009.10.
- [20] M. Paasivaara y C. Lassenius, «Communities of practice in a large distributed *Agile* software development organization – Case Ericsson», *Inf. Softw. Technol.*, vol. 56, n.º 12, pp. 1556-1577, dic. 2014, doi: 10.1016/j.infsof.2014.06.008.
- [21] H. Hajjdiab, A. Taleb, y J. Ali, «An Industrial *Case study* for *Scrum* Adoption», *JSW*, vol. 7, pp. 237-242, ene. 2012, doi: 10.4304/jsw.7.1.237-242.
- [22] H. Gutiérrez Pulido y A. Quirarte, «Planeación avanzada de la calidad del Producto (apqp): conceptos básicos y un caso práctico», *E-Gnos. Num005*, ene. 2007.
- [23] «New research: The shift between T-Shaped & Key-shaped employees», *The Oxford Review*, 20 de marzo de 2021. <https://oxford-review.com/shift-between-t-key-shaped-people/> (accedido 12 de junio de 2022).

## Anexo 1

### USO DE METODOLOGÍAS ÁGILES EN GRANDES ORGANIZACIONES

Imanol Iriondo Alzola

University of the Basque Country (UPV/EHU), Department of Project Management  
Bilbao, Spain [iriondo016@ikasle.ehu.eus](mailto:iriondo016@ikasle.ehu.eus)

**Abstract:** Las metodologías ágiles se están convirtiendo en el modelo a seguir en cuanto a la gestión de los proyectos, esto es debido al éxito obtenido con una implementación adecuada de las metodologías. Originalmente la aplicación de este tipo de metodologías estaba enfocada a equipos pequeños capaces de desarrollar un proyecto de software de manera independiente. Debido al éxito adquirido en estos entornos, las grandes organizaciones están empezando a adoptar Agile.

El Trabajo Fin de Master analiza el escalado de las metodologías ágiles en grandes organizaciones. Primero se analizan dos marcos de escalado completos; Large Scale Scrum y Scaled Agile Frameworks. La segunda parte del Trabajo Fin de Master hace un análisis de los artículos publicados de la adopción de Agile en grandes organizaciones. El análisis se ha centrado en identificar los factores de éxito y retos que tuvieron las empresas. Por último, se ha analizado el proceso de implementación del modelo Agile en una gran organización actual. En base a los resultados de la segunda parte se han identificado mejoras al proceso de adopción.

**Keywords:** *Agile, Gestión de Proyectos, escalado, grandes organizaciones.*

#### 1. INTRODUCCIÓN

Las metodologías ágiles se están convirtiendo en el modelo a seguir en cuanto a la gestión de los proyectos, esto es debido al éxito obtenido con una implementación adecuada de las metodologías.

Las metodologías ágiles son un conjunto de métodos que permiten adaptar el modo de trabajo a las condiciones del proyecto, aportando flexibilidad y eficiencia. Las metodologías ágiles nacieron en el año 2001, cuando importantes desarrolladores de Software se reunieron para poner en común sus mejores métodos de desarrollo. De esta manera se crea el Manifiesto Ágil, en el que se establecieron 12 principios básicos y cuya filosofía se puede resumir en cuatro ideas:

Valorar a los individuos sobre los procesos y herramientas:

Esta idea promueve que las personas son lo más importante, por encima de proceso y herramientas. Los procesos y herramientas deben servir para apoyar a las personas, y deben adaptarse a la organización, los equipos y las personas y no al revés.

Valorar el software funcionando sobre la documentación excesiva:

Esta idea propone que es más importante obtener feedback de usuarios o cliente de un prototipo temprano a una documentación previa exhaustiva. No se renuncia a la documentación, y esta debe ser completada, pero el peso del prototipo en funcionamiento es mayor.

Valorar la colaboración con el cliente sobre la negociación contractual:

Aquí se considera al cliente uno más del equipo, involucrándolo en mayor escala durante el proceso de desarrollo obteniendo una retroalimentación continua para cumplir a rajatabla lo pactado y deseado. Esto es especialmente útil cuando es difícil definir de antemano los requisitos del Producto.

Valorar la respuesta ante el cambio respecto a seguir un plan:

No tiene sentido ceñirse a planes rígidos cuando el entorno es volátil. Es más valiosa la capacidad de adaptación y respuesta a la capacidad de seguir un plan preestablecido. Las metodologías ágiles promueven la anticipación y adaptación, frente a la planificación y el control tradicional.

Los métodos ágiles permiten adaptar la forma de trabajo a las condiciones del proyecto, consiguiendo flexibilidad en inmediatez para dar respuesta a las necesidades del proyecto. De esta manera, se consigue gestionar los proyectos de forma flexible, autónoma y eficaz reduciendo los costes e incrementando su Productividad.



Parte del éxito y la fama obtenida hasta el momento por las metodologías Ágiles viene asociada a su aplicación principalmente en entornos considerados como Agile sweet spot, que consiste en equipos de trabajo reducidos, involucrados en proyectos greenfield (sin condicionantes por un trabajo previo) no críticos y con una estructura de gobierno simple.

En la actualidad las metodologías ágiles están siendo implementadas en proyectos de gran envergadura y grandes compañías. Compañías como Microsoft, Amazon y Google han adoptado las metodologías ágiles en su gestión. La implementación de metodologías ágiles en grandes organizaciones supone un reto para la compañía tradicional, ya que estas suelen tener una estructura rígida en la que la introducción de cambios es lenta y tiene que hacer frente a una inercia de funcionamiento generada durante el tiempo. Por otra parte, las grandes organizaciones implican por lo general grandes proyectos con una combinación de equipos dedicados y recursos compartidos. Estos factores combinados suponen un reto para el escalado de las metodologías ágiles en estas organizaciones.

Debido a la demanda de aplicación de metodologías ágiles en grandes organizaciones, se han generado varios métodos para el escalado del modelo ágil. Según [1], los métodos de escalado más populares en la actualidad son Scrum of Scrums, Scaled Agile Framework SAFe, Large Scale Scrum LeSS y Enterprise Scrum. Debido a que estas metodologías son novedosas, la información obtenible en artículos o libros es en su mayoría teórica y no basada en implementaciones.

Este artículo se centra en el problema de escalado de las metodologías Ágiles en grandes organizaciones. Los objetivos definidos son los siguientes:

- Revisión y comparación de 2 metodologías existentes para el escalado del modelo ágil en grandes organizaciones.
- Identificación de desafíos y factores de éxito para el escalado del modelo Agile.

## 2. MARCOS DE ESCALADO DE AGILE

Introducir la metodología ágil en grandes organizaciones es un reto para la entidad [2]. Los proyectos grandes requieren una coordinación y comunicación adecuadas entre los equipos [3], hay que gestionar las dependencias entre los equipos y hay que involucrar a otras unidades no ágiles. Los métodos ágiles tradicionales se diseñaron para proyectos de un solo equipo y no se enfrentaban a estos problemas [4].

Por lo tanto, se crearon varios métodos, prácticas y marcos completos centrados en la ampliación de los métodos ágiles tradicionales al entorno a gran escala. Según [1] los métodos ágiles de ampliación más utilizados son Scrum / Scrum of Scrums (9 %), Scaled Agile Framework (37 %), Lean Management (2 %), Enterprise Agile (6%), Agile Portfolio Management (93%), Large Scale Scrum (3 %), Disciplined Agile (3 %).

Se ha seleccionado Scaled Agile Framework (SAFe) y Large Scale Scrum (LeSS) como dos representantes de estos métodos, que se analizarán con más detalle. Estos métodos se han seleccionado por varias razones. SAFe es probablemente el método ágil de escalado más complejo y completo. Es un marco completo, y muchas otras prácticas conocidas como Lean Management o Agile Portfolio Management, ya están incorporadas en él. Large Scale Scrum (LeSS) es también un marco completo para escalar la metodología Ágil. Sin embargo, trata de ser lo más minimalista posible. Se basa menos en los procesos y las estructuras organizativas específicas y más en la mentalidad de las personas y la comunicación 'ad hoc'.

Los dos marcos de escalado revisados se presentan como soluciones completas que contienen las funciones necesarias para escalar el modelo ágil a grandes compañías. Entre ellas se pueden observar muchas diferencias, pero a su vez también coinciden en varios criterios.

La definición de SAFe es muy detallada y a veces contiene incluso un programa concreto de reuniones individuales. Su estructura organizativa se divide en 3 niveles, pero estos tienen una definición muy detallada de los roles necesarios.

Por otro lado, LeSS evita definir en exceso las estructuras y roles, esto se basa en ser lo más ágil posible. Se centra en la mentalidad, los valores y los principios sin introducir demasiados procesos y funciones. El marco en sí es mucho más joven y no tiene tanto apoyo empresarial como SAFe.

Atendiendo a las similitudes entre los dos marcos de escalado, ambos marcos coinciden en que la comprensión de todo el sistema es esencial para tener una adecuada definición de tareas, procesos, planes etc. Además, se pueden apreciar algunos patrones similares en la estructura organizativa y los procesos entre estos marcos. Ambos marcos utilizan el papel escalado del Product Owner y una jerarquía de backlogs. Además, ambos se inclinan por utilizar los Feature Teams, esto es debido a que los equipos de características permiten un escalado más fácil de la gestión de requisitos. Además, ambos marcos utilizan algunos equipos adicionales a los equipos de desarrollo, que ayudan a los equipos de desarrollo con su trabajo. Para el escalado de la metodología ágil, los marcos coinciden en el uso de las siguientes prácticas; Scrums de Scrums, Comunidades de práctica, Gestión de requisitos a escala, Scaled Sprint Planning, Equipos



de trabajo y Undone Department.

### 3. MÉTODO DE INVESTIGACIÓN

El método utilizado para el análisis de artículos con el objetivo de encontrar los factores de éxito y retos es el mapeo sistemático [5]. El proceso de mapeo sistemático tiene cinco pasos; la definición de las preguntas de investigación, la búsqueda de estudios primarios, el cribado de los artículos, la redacción clave de los resúmenes y la extracción de datos. En primer lugar, se definen las preguntas de investigación y se realiza una búsqueda de estudios primarios. Posteriormente, se realiza un cribado de los estudios primarios y se seleccionan los estudios pertinentes. A continuación, se crean categorías utilizando palabras clave en los resúmenes de los estudios primarios seleccionados. Por último, los estudios primarios se clasifican en categorías individuales.

El primer paso del proceso de mapeo sistemático es la definición de las preguntas para llevar a cabo la investigación. El objetivo de la investigación es conocer cómo las grandes empresas adoptan las metodologías ágiles. Las preguntas definidas son las siguientes:

- P1: ¿Qué métodos ágiles utilizan las grandes empresas?
- P2: ¿Qué prácticas de escalado ágil utilizan las grandes empresas?
- P3: ¿Qué desafíos identifican las grandes empresas cuando se adoptan a las metodologías ágiles?
- P4: ¿Qué factores de éxito identifican para adoptar y escalar el modelo ágil?

La búsqueda de artículos, casos de estudio y publicaciones es el siguiente paso del proceso. La búsqueda de publicaciones para llevar a cabo el proceso de mapeo sistemático se ha centrado en las bases de datos (en este caso artículos) definidos en la Tabla 1. Para llevar a cabo el filtrado de artículos, se han definido dos grupos y se han definido las palabras clave, esto se puede ver en la Tabla 2. A la hora de intentar definir un grupo para el tipo de organización, esto no ha sido posible en este paso, ya que la definición adecuada para el filtrado no ha sido posible definirla.

Data Base	URL
IEEE Explore	<a href="https://ieeexplore.ieee.org">https://ieeexplore.ieee.org</a>
Science Direct	<a href="https://www.sciencedirect.com/">https://www.sciencedirect.com/</a>
Research Gate	<a href="https://www.researchgate.net/">https://www.researchgate.net/</a>
Springer	<a href="https://www.springer.com/">https://www.springer.com/</a>

Tabla 1: Bases de artículos utilizadas para el estudio

El tercer paso del proceso de mapeo sistemático es la selección de los artículos, es decir, su inclusión y exclusión en la investigación. Como no ha sido posible definir un criterio de búsqueda en el apartado anterior, en este paso se han filtrado los artículos en base a la organización y su tamaño.

El número de empleados de las empresas seleccionadas oscila entre 40 y 15.000. La selección inicial abarca veinte artículos en total, de estos 20 se han seleccionado doce de ellos; seis son informes de experiencias y seis son Case study. En el caso de los informes de experiencias, los autores están afiliados a la empresa. Los Case study los realizó un investigador independiente. Dada la dificultad para encontrar artículos imparciales con alto valor científico que cumplieran los requisitos de inclusión, se ha decidido incluir los informes de experiencias en la investigación.

Grupos	Palabras Clave
Metodologías Ágiles	Agil, Lean, Scrum, Kanban, Backlog
	Agile and Sprint
Metodología de Escalado	Large Scale Scrum
	Scaled Agile Framework

Tabla 2: Grupos y palabras clave para la búsqueda



#### 4. RESULTADOS ESTUDIO SISTEMICO

- P1: ¿Qué métodos ágiles utilizan las grandes empresas?

Se han identificado 5 metodologías ágiles utilizados por las empresas de los artículos. Como se puede ver en la Tabla 3, el método ágil más utilizado ha sido Scrum.

Metodología Ágil	# de usos
Scrum	12
Lean	5
Kanban	2
LeSS	1
DAD	1

Tabla 3: Metodologías ágiles utilizadas por las compañías

El uso de Scrum se menciona en todos los artículos. El segundo método ágil más aplicado ha sido Lean, cinco artículos informaron del uso de prácticas Lean. Dos casos han adoptado Kanban y dos casos han utilizado algún marco para escalar el desarrollo ágil. Un caso adopta LeSS y otro utiliza de Disciplined Agile Delivery (DAD).

- P2: ¿Qué prácticas de escalado ágil utilizan las grandes empresas?

Las prácticas de escalado y el número de usos identificados se muestran en la Tabla 4. En este caso la práctica de escalado más utilizada ha sido *Scrum of Scrums*.

Practica de escalado	# aplicaciones
Scrum of Scrums	6
Comunidades de practica	5
Scaled Sprint Demo	3
Scaled Requirement Managemement	3
Scaled Sprint Planning	3
Feature Teams	2
Undone Department	2

Tabla 4: Uso de prácticas de escalado

*Scrum of Scrums* ha sido la práctica de escalado más utilizada [6]–[11]. Varios de los artículos utilizan *Scrum of Scrums* para la sincronización entre los equipos [8], [10], [11]. El orden del día de la reunión de *Scrum of Scrums* es informar a otros equipos sobre el estado actual del trabajo del equipo, los impedimentos que deben ser resueltos, y la discusión sobre las dependencias entre los equipos y la gestión de estas dependencias.

- P3: ¿Qué desafíos identifican las grandes empresas cuando se adoptan a las metodologías ágiles?

Se han clasificado los desafíos identificados en siete grupos; los grupos se muestran en la Tabla 5. Como se puede observar, la falta de conocimiento, coaching o formación ha sido el desafío más común.

Desafios	# de casos
Falta de conocimiento, Coaching o formación	10
Resistencia al cambio	8
Problemas de calidad	6
Presión elevada o sobrecarga de trabajo	5
Integración con partes no ágiles	4
Falta de trabajo en equipo y compromiso	4
Medición del progreso	3

Tabla 5: Retos identificados para el escalado de Agile

El reto más común entre los artículos analizados es la falta de formación, y entendimiento de los fundamentos de los procesos. Estos problemas se señalaron en 10 casos [6], [8]–[16].

Varios factores causan razones para la falta de entrenamiento y conocimiento, la mayoría de las veces se debe a la subestimación de la dificultad de la transformación ágil, las limitaciones financieras o una transición apresurado. Por



ejemplo, la organización [16] subestima la dificultad de la adopción de Agile. La organización no tiene suficientes conocimientos en sus filas, pero se niega a recurrir a un experto externo debido a las limitaciones financieras. En su lugar, los directivos que son nuevos en metodologías ágiles asumen el papel de expertos en Agile, lo cual puede ser perjudicial para todo el proceso.

- P4: ¿Qué factores de éxito identifican para adoptar y escalar el modelo ágil?

Se identifican los factores de éxito en siete grupos, todos los grupos, junto con el número de casos que se identificaron, se muestran en la Tabla 5.

Factores de éxito	# de casos
Formación	11
Trabajo en equipo	7
Visión única en valores y practicas	6
Transformación cautelosa	6
Herramientas e infraestructura	5
Patrocinio ejecutivo	5
Prácticas de ingeniería sólidas	2

Tabla 5: Factores de éxito identificados

Dado que el reto más señalado es la falta de conocimientos, no es de extrañar que el factor de éxito más señalado sea la obtención de los mismos. La adquisición y el intercambio de conocimientos de forma exhaustiva, profunda y sistemática se declaran como factores de éxito en 11 casos [6], [7], [9]–[17].

La forma más recomendable de adquirir conocimientos y experiencia es según los artículos contratar a un experto externo con una amplia y profunda familiaridad con desarrollo ágil [7], [9], [10], [12], [15], [16]. El experto externo comparte sus conocimientos con varios empleados internos, que luego los difundieron por toda la organización. Una de las razones por las que resulta más beneficioso traer a un experto que invertir en la adquisición de conocimientos de otras fuentes, es que las organizaciones se dan cuenta de que las sugerencias generales y teóricas no son fácilmente aplicables en su contexto particular, y que se necesita una comprensión más profunda de los métodos [7], [11].

## 5. CONCLUSIONES

El TFM ha estado enfocado al escalado de las metodologías ágiles en grandes organizaciones, para ello inicialmente se definieron los siguientes objetivos:

- Revisar y comparar 2 metodologías existentes para el escalado del modelo ágil en grandes organizaciones.
- Identificar desafíos y factores de éxito para el escalado del modelo *Agile*.

La primera parte del TFM se ha centrado en el análisis de las 2 metodologías de escalado del modelo *Agile*. Las dos metodologías seleccionadas han sido ‘*Scaled Agile Frameworks*’ SAFe y ‘*Large Scale Scrum*’. SAFe tienen una definición muy detallada de todo el proceso y organización. Define una organización con 3 niveles de jerarquía en los que se hace una alta definición de cargos. LeSS por lo contrario, presenta un planteamiento más sencillo, en el que no se hace una definición completa de la organización y procesos. Define unos valores a utilizar como base para la organización, pero permite la adaptación de todo el proceso a las necesidades de cada empresa.

El apartado 3 analiza los marcos de escalado utilizados por las grandes organizaciones y los desafíos y factores de éxito identificados durante el proceso de adopción. Se ha visto que el intentar implementar modelos de escalado demasiado complicados ha resultado en fracaso en las organizaciones que lo han intentado. Por otra parte, las organizaciones también resaltan que por lo general han necesitado hacer adopciones a los modelos de escalado seleccionados. Estos dos argumentos refuerzan el planteamiento que tiene LeSS sobre el escalado, haciéndolo una mejor opción frente SAFe. En cuanto a los retos más mencionados por las compañías son la falta de formación o coaching, la resistencia al cambio y los problemas de calidad. Los factores de éxito más repetidos han sido la formación, el trabajo en equipo y hacer las transiciones con cautela.



## 6. REFERENCIAS

- [1] «15th Annual State Of Agile Report | Digital.ai». <https://digital.ai/resource-center/analyst-reports/state-of-agile-report> (accedido 12 de mayo de 2022).
- [2] T. Dybå y T. Dingsoyr, «What Do We Know about Agile Software Development?», *Softw. IEEE*, vol. 26, pp. 6-9, nov. 2009, doi: 10.1109/MS.2009.145.
- [3] D. Mishra y A. Mishra, «Complex software project development: Agile methods adoption», *J. Softw. Maint. Evol. Res. Pract.*, vol. 23, pp. 549-564, dic. 2011, doi: 10.1002/smr.528.
- [4] B. Boehm y R. Turner, «Management challenges to implementing agile processes in traditional development organizations», *IEEE Softw.*, vol. 22, n.º 5, pp. 30-39, sep. 2005, doi: 10.1109/MS.2005.129.
- [5] K. Petersen, R. Feldt, S. Mujtaba, y M. Mattsson, «Systematic Mapping Studies in Software Engineering», presentado en 12th International Conference on Evaluation and Assessment in Software Engineering (EASE), jun. 2008. doi: 10.14236/ewic/EASE2008.8.
- [6] M. Paasivaara, C. Lassenius, V. T. Heikkilä, K. Dikert, y C. Engblom, «Integrating Global Sites into the Lean and Agile Transformation at Ericsson», en *2013 IEEE 8th International Conference on Global Software Engineering*, ago. 2013, pp. 134-143. doi: 10.1109/ICGSE.2013.25.
- [7] S. Hanly, L. Wai, L. Meadows, y R. Leaton, «Agile coaching in British Telecom: making strawberry jam», en *AGILE 2006 (AGILE'06)*, jul. 2006, p. 9 pp. - 202. doi: 10.1109/AGILE.2006.13.
- [8] M. Paasivaara y C. Lassenius, «Scaling Scrum in a Large Globally Distributed Organization: A Case Study», en *2016 IEEE 11th International Conference on Global Software Engineering (ICGSE)*, ago. 2016, pp. 74-83. doi: 10.1109/ICGSE.2016.34.
- [9] K. Long y D. Starr, «Agile Supports Improved Culture and Quality for Healthwise», en *Agile 2008 Conference*, ago. 2008, pp. 160-165. doi: 10.1109/Agile.2008.61.
- [10] J. Schnitter y O. Mackert, «Large-Scale Agile Software Development at SAP AG», en *Evaluation of Novel Approaches to Software Engineering*, Berlin, Heidelberg, 2011, pp. 209-220. doi: 10.1007/978-3-642-23391-3\_15.
- [11] «Inter-organizational Co-development with Scrum: Experiences and Lessons Learned from a Distributed Corporate Development Environment | Semantic Scholar». <https://www.semanticscholar.org/paper/Inter-organizational-Co-development-with-Scrum%3A-and-Vallon-Strobl/ae9f5e36dfdf38b90fe08c20f0f4707df9fe4b7> (accedido 12 de mayo de 2022).
- [12] G. Benefield, «Rolling Out Agile in a Large Enterprise», en *Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008)*, ene. 2008, pp. 461-461. doi: 10.1109/HICSS.2008.382.
- [13] M. Paasivaara, B. Behm, C. Lassenius, y M. Hallikainen, «Towards Rapid Releases in Large-Scale XaaS Development at Ericsson: A Case Study», en *2014 IEEE 9th International Conference on Global Software Engineering*, ago. 2014, pp. 16-25. doi: 10.1109/ICGSE.2014.22.
- [14] N. Moe, «Key Challenges of Improving Agile Teamwork», jun. 2013. doi: 10.1007/978-3-642-38314-4\_6.
- [15] A. Atlas, «Accidental Adoption: The Story of Scrum at Amazon.com», sep. 2009, pp. 135-140. doi: 10.1109/AGILE.2009.10.
- [16] H. Hajjidiab, A. Taleb, y J. Ali, «An Industrial Case Study for Scrum Adoption», *JSW*, vol. 7, pp. 237-242, ene. 2012, doi: 10.4304/jsw.7.1.237-242.
- [17] B. Schatz y I. Abdelshafi, «Primavera Gets Agile: A Successful Transition to Agile Development», *Softw. IEEE*, vol. 22, pp. 36-42, jun. 2005, doi: 10.1109/MS.2005.74.