

Cartilla
Marco ágil en la gestión de proyectos de la
Dirección de Gestión de Innovación y
Tecnología

Proceso Información, Innovación y Tecnología
Subproceso Innovación y Tecnología
Versión 01
Código CT-IIT-0139
Año 2024

El contenido de este documento corresponde a Información Pública

Contenido

1. INTRODUCCIÓN	5
2. GLOSARIO	20
3. OBJETIVO	23
4. ALCANCE	23
5. ECOSISTEMA PARA LA TRANSFORMACIÓN ÁGIL DE LA ENTIDAD Y LA GESTIÓN DE PROYECTOS	24
6. RESPONSABILIDADES	28
6.1 Scrum Master o Maestro de Scrum	28
6.2 Product Owner o dueño de Producto.....	31
6.3 Ingeniero de Requerimientos.....	32
6.4 Developers o desarrolladores	34
6.5 Responsabilidades transversales de los Developers	35
6.5.1 Asegurar la integración funcional.....	36
6.5.2 Garantizar la integración técnica	36
6.5.3 Colaborador técnico.....	38
6.6 Stakeholders o Interesados	38
6.7 Agile Coach (Entrenador Ágil)	39
7. SCRUM ESTÁNDAR - GESTIÓN DE PROYECTOS ÁGILES	40
7.1 FASE I: ALISTAMIENTO	40
7.1.1 Activación de nuevos Scrum Teams (células).....	40
7.1.2 Conformación del Scrum Team	40
7.1.3 Apropiación del área funcional sobre la transformación ágil	41
7.2 FASE II: GESTIÓN DEL PROYECTO ÁGIL	41
7.2.1 Etapa de Inicio (Agile inception)	41
7.2.2 Etapa de ejecución de Sprints	53
7.2.3 Etapa de cierre.....	72
8. SCRUM@SCALE ESCALAMIENTO DE LA TRANSFORMACIÓN ÁGIL	73
8.1 Principios a cumplir para hacer escalamiento en la DIAN	73
8.2 Conceptos básicos para el escalamiento	75
8.3 Roles para el escalamiento de Scrum	76
8.3.1 Scrum of Scrums Master (El Scrum de Scrum Master)	76
8.3.2 Chief Product Owner (Jefe de Product Owners).....	77
8.4 Proceso de escalamiento.....	77
8.4.1 Ciclo de Scrum Master.....	78
8.4.2 Ciclo de Product Owner	83
8.4.3 Conectar los ciclos.....	86
9. PATRONES	87
10. CENTRO DE RECURSOS Y HERRAMIENTAS	89
11. REFERENCIAS	90
12. ANEXOS PARA EL SISTEMA DE GESTIÓN DE CALIDAD	90
13. No. CONTROL DE CAMBIOS	¡Error! Marcador no definido.

Índice de Ilustraciones

Ilustración 1 - Valores Scrum.....	24
Ilustración 2 - Ecosistema de transformación ágil	25
Ilustración 3 - Roles en una célula Scrum	40
Ilustración 4 - Impact Mapping.....	43
Ilustración 5 - Refinamiento del Product Backlog	45
Ilustración 6 - Ecuación de valor.....	47
Ilustración 7 - Scrum en modo gráfico	54
Ilustración 8 - Burndown chart	71
Ilustración 9 - Scrum Scale en la DIAN.....	75
Ilustración 10 - Scrum@Scale y sus ciclos	78
Ilustración 11 - Ciclo de Scrum Master	79
Ilustración 12 - Scrum Team o célula en la DIAN	79
Ilustración 13 - Scrum of Scrums en la DIAN	80
Ilustración 14 - Scrum of Scrum of Scrums en la DIAN	81
Ilustración 15 - Ciclo de Product Owners.....	83
Ilustración 16 - Chief Product Owner y equipo de Product Owners.....	84
Ilustración 17 - Conectar los ciclos	86

Índice de Tablas

Tabla 1 – Etapas de la Gestión Ágil.....	26
Tabla 2 – Scrum Master y sus Responsabilidades.....	28
Tabla 3 – Product Owner en las Etapas de la Gestión Ágil.....	31
Tabla 4 – Ingeniero de Requerimientos y sus Responsabilidades	33
Tabla 5 – Developers y sus responsabilidades	34
Tabla 6 – Integración funcional.....	36
Tabla 7 – Integración técnica.....	37
Tabla 8 – Ingeniero de requerimientos y sus responsabilidades.....	38
Tabla 9 – Agile Coach y sus responsabilidades	39
Tabla 10 – Refinamiento del Product Backlog.....	56
Tabla 11 – Daily Scrum.....	57
Tabla 12 – Review	57
Tabla 13 – Manejo de la deuda técnica	59
Tabla 14 – Manejo de los Bugs	59
Tabla 15 – Retrospective	60
Tabla 16 – Pilares empíricos y su relación con el monitoreo.....	61
Tabla 17 – Métricas de Scrum estándar y Scrum@Scale	63
Tabla 18 – Dashboard de monitoreo para un proyecto recién iniciado	70
Tabla 19 – Dashboard para proyecto que ha ejecutado al menos 3 sprints.....	70
Tabla 20 – Scrum of Scrums Master y sus responsabilidades	76
Tabla 21 – Chief Product Owner y sus responsabilidades	77
Tabla 22 – Executive Action Team y la Scaled Daily.....	82
Tabla 23 – EAT y sus responsabilidades.....	82

Tabla 24 – componentes del ciclo Scrum Master	83
Tabla 25 – Meta Scrum y sus reuniones	85
Tabla 26 – Componentes del ciclo del Product Owner.....	85
Tabla 27 – Retroalimentación y métricas.....	87
Tabla 28 – Patrones.....	87
Tabla 29 – Centro de recursos y herramientas.....	89

1. INTRODUCCIÓN

La Dirección de Gestión de Innovación y Tecnología – DGIT, como líder de las acciones necesarias para llevar a cabo la transformación digital de la DIAN, identifica la necesidad de contar con una gestión ágil de proyectos, la cual permita facilitar el acercamiento con los líderes funcionales, desarrollar los sistemas y servicios digitales oportunamente, con altos estándares de calidad, optimizando el uso de los recursos existentes y maximizando la oferta de valor a los clientes internos y externos.

Es así como ha optado por la transformación ágil, lo que implica un cambio fundamental en la forma en que se conciben, desarrollan y entregan los servicios públicos. A diferencia de los enfoques tradicionales que podrían llevar años para implementar cambios significativos, la metodología ágil aboga por una entrega iterativa e incremental. En este contexto, la transformación ágil busca generar valor de manera temprana al enfocarse en las necesidades y expectativas reales de los ciudadanos. Esto implica un proceso de colaboración estrecha entre los equipos de desarrollo y los usuarios finales, permitiendo la adaptación rápida a los cambios en los requisitos y prioridades. Al adoptar Scrum, la entidad busca acelerar la entrega de servicios, reducir el riesgo de proyectos a largo plazo y garantizar una mayor flexibilidad y capacidad de respuesta a las demandas emergentes de la sociedad.

En este contexto, la calidad se convierte en una piedra angular de la transformación ágil. La búsqueda de la mejor calidad no solo se refiere a la excelencia técnica en el desarrollo de software, sino también a la calidad percibida por los ciudadanos y por los clientes internos. La retroalimentación continua y la integración de los comentarios de los usuarios permiten realizar ajustes a medida que evoluciona el proyecto, asegurando que las soluciones sean verdaderamente útiles y eficaces. Además, la adopción de prácticas de desarrollo centradas en la calidad, como las pruebas automatizadas y la revisión constante de código, garantiza que los productos y servicios entregados sean confiables, seguros y cumplen con los estándares necesarios. En resumen, la transformación ágil en la DIAN no solo busca acelerar la entrega, sino también garantizar que cada una represente un paso significativo hacia la mejora de la calidad y la creación de valor tangible para los ciudadanos.

Este documento presenta y explica la manera como se aborda la transformación ágil por vía de la gestión de los proyectos de tecnología que se llevarán a cabo desde la DGIT, tomando como base lineamientos y buenas prácticas evidenciadas en la literatura sobre agilismo y referenciadas especialmente dentro del marco de trabajo SCRUM y su guía más reciente (Schwaber Ken, 2022) y Scrum@Scale (Sutherland, 2021).

El lector encontrará una lista de definiciones en el GLOSARIO. Luego, el OBJETIVO de este documento y su ALCANCE. Más adelante, en el apartado titulado: ECOSISTEMA PARA LA TRANSFORMACIÓN ÁGIL DE LA ENTIDAD Y LA GESTIÓN DE PROYECTOS se explican los pilares empíricos y valores Scrum, así como las etapas de la gestión de proyectos; para conectar con los R que define el marco y su adaptación, como roles propios que son importantes para la transformación según las necesidades de la Entidad.

El capítulo 7 denominado SCRUM ESTÁNDAR - GESTIÓN DE PROYECTOS ÁGILES desarrolla la manera procedimental como se abordará la transformación ágil desde la DIGIT. Para ello, organiza la información en dos fases:

FASE I: ALISTAMIENTO sobre la activación de equipos Scrum o células y FASE II: GESTIÓN DEL PROYECTO ÁGIL.

Por su parte, el capítulo 8 se enfoca en presentar la adaptación de SCRUM@SCALE para lograr el escalamiento de Scrum en la transformación ágil de la Entidad.

A manera de complemento se ofrece el apartado SCRUM@SCALE ESCALAMIENTO DE LA TRANSFORMACIÓN ÁGIL

Scrum@Scale es un marco de trabajo desarrollado para extender los principios ágiles de Scrum a empresas grandes y complejas. Este proporciona orientación sobre cómo coordinar y sincronizar múltiples equipos Scrum que trabajan en proyectos interdependientes dentro de una organización.

Scrum@Scale se centra en promover la flexibilidad y la agilidad en el escalado, permitiendo que los equipos ágiles mantengan su autonomía y eficacia mientras trabajan de manera conjunta para abordar desafíos a mayor escala.

Los beneficios de Scrum Scale incluyen:

- a) La capacidad de las organizaciones para adaptarse más rápidamente a los cambios del mercado.
- b) Mejorar la eficiencia operativa para lograr agilidad de negocio respondiendo a los cambios de manera rápida adoptando la configuración inicial de manera estable.
- c) Proporcionar un marco de trabajo que fomente la colaboración y la innovación en toda la entidad.
- d) Al facilitar la implementación ágil en entornos más amplios, Scrum@Scale busca mejorar la calidad del producto, acelerar la entrega y aumentar la satisfacción del cliente, contribuyendo así a una transformación organizativa exitosa hacia la agilidad.
- e) Entonces, Scrum@Scale procura el trabajo de red organizado, generando una saturación adecuada de la comunicación y vasos comunicantes que permitan hacer las definiciones estratégicas, técnicas y operativas sin aumentar en burocracia ni jerarquías. Los equipos base siguen siendo las células (Scrum Teams) sólo que ahora estarán acompañados de otros equipos que favorecerán la comunicación, coordinación, remoción de impedimentos y toma rápida de decisiones.

Con el escalamiento de Scrum en la Entidad se busca:

- a) Escalabilidad lineal: correspondencia entre aumento de porcentaje entrega de producto funcionando con el aumento de cantidad de equipo.
- b) Agilidad de negocio: la habilidad para responder a los cambios y retos adaptando la configuración estable inicial.

6.1 Principios a cumplir para hacer escalamiento en la DIAN

1. Respaldo de la alta dirección al escalamiento y todo lo que ello conlleva, por ejemplo, cambio cultural, ajuste en procesos, asignación de recursos y presupuesto.
2. Debe tenerse el respaldo de la dirección general y demás actores que participarán de los grupos líderes del escalamiento:
 - a) Scrum of Scrums (SoS) equipo coordinado de células.
 - b) Executive Action Team (EAT): enfocado en cómo hacerlo más rápido.
 - c) Executive Meta Scrum (EMS): enfocado en el valor de lo que produce.

Los últimos dos EAT y EMS deben respaldar al SoS para que operar efectivamente.

3. Contar con un sistema operacional estándar de Scrum, es decir, tener células o Scrum Teams activos desarrollando y entregando incrementos al final de cada Sprint. Deben estar funcionando bien, es decir, realizando un proceso Scrum completo, sin mezclas de otros marcos ágiles y mucho menos combinaciones con metodologías tradicionales como waterfall. La Entidad debe asegurarse de que esas células esas células están implementando bien Scrum, porque cualquier error que se esté presentando es potencialmente escalable.
4. Profesionales en los roles del marco Scrum listos para escalar.
5. Product Owners empoderados con el rol, con plena conciencia de la importancia y ejercicio de sus responsabilidades. Deben estar formados en Scrum y cumplir el perfil descrito en el apartado 6.6 de este documento.
6. Cambio en la cultura organizacional sensibilizada e impulsada por valores

Los pilares empíricos: transparencia, inspección y adaptación; así como los valores Scrum son columnas en la escalabilidad.

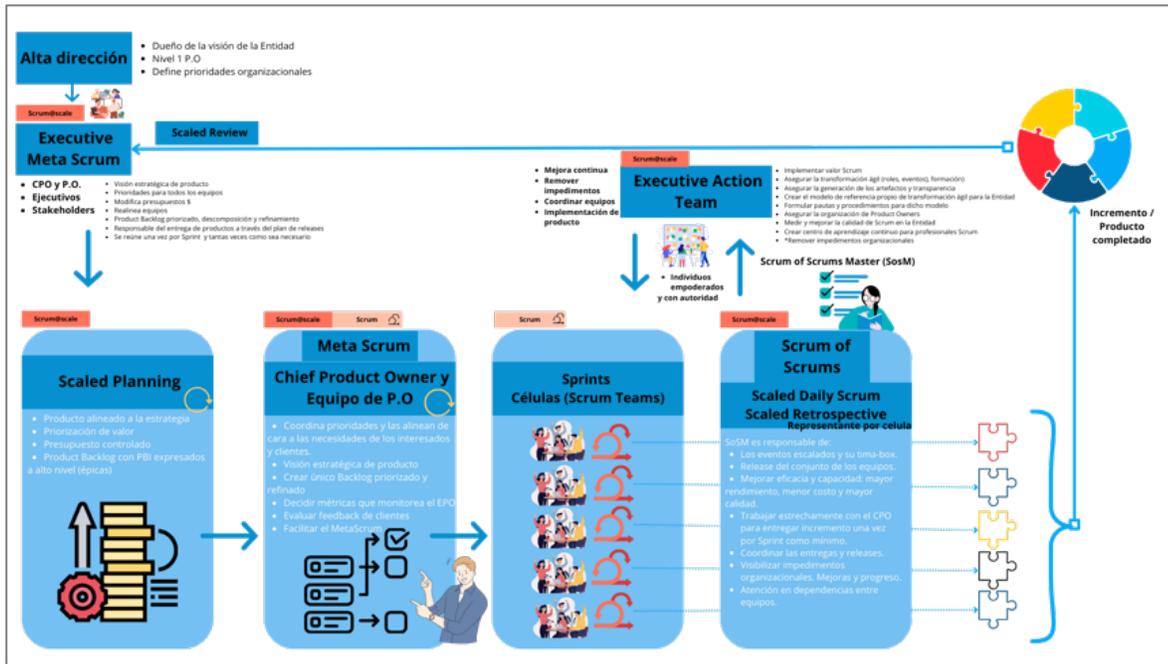
Es necesario contar con un sistema de aprendizaje enfocado en la transformación ágil para que la Entidad comprenda, apropie y colabore activamente en la gestión ágil y en la forma como se escala Scrum.

7. Modelo de referencia: el primer escalamiento debe ser un conjunto de equipos que se coordinan para entregar en cada sprint (prototipo para escalar en próximas redes de equipos). Este modelo marcará la pauta la creación de un sistema operativo ágil pues el resto de la entidad tendrá un ejemplo funcional y saludable de Scrum@Scale para replicar.

Es necesario corregir cualquier deficiencia antes de escalar más equipos. Escalar un sistema roto no resuelve nada, lo empeora todo. El Modelo de referencia debe funcionar bien.

8. Cada Scrum of Scrums debe cumplir con el proceso de equipo, tal como lo hace un Scrum Team o célula:
 - a. Maximizar el flujo de trabajo completado y que cumpla con la DoD
 - b. Aumentar el rendimiento del equipo a través del tiempo
 - c. Operar de manera sostenible y enriquecedora para el equipo
 - d. Acelerar el ciclo de retroalimentación del cliente
9. Sincronización de las células por SoS o más grupos de la red dependiendo de la complejidad y magnitud del proyecto. Esto implica que para un proyecto que tiene varios productos ejecutándose al mismo tiempo se requiere sincronizar el inicio del Sprint y también su duración. Esto se debe fundamentalmente a los eventos escalados. Así mismo, generar un solo Product Backlog priorizado para evitar la duplicación de trabajo y crear una mínima Definition of Done uniforme que aplique a todas las células. De igual forma aplica para su roadmap y plan de release. Todo se sincroniza para lograr el escalamiento lineal. Si una célula está por fuera de la sincronización no se puede escalar o si un artefacto o evento no se aborda de manera sincronizada, no se puede escalar.

Ilustración 9 - Scrum Scale en la DIAN



Fuente: Elaboración propia

6.2 Conceptos básicos para el escalamiento

Operar Scrum@Scale implica coordinar y sincronizar múltiples células Scrum que trabajan en conjunto para lograr objetivos organizacionales de gran envergadura. Los siguientes son los conceptos básicos que se ponen en marcha en el escalamiento de Scrum en la DIAN.

1. Equipos Ágiles Pequeños:

Scrum Scale parte del principio de que los equipos ágiles pequeños son la unidad fundamental de trabajo eficaz. Cada equipo sigue los principios de Scrum estándar con sus roles, eventos y artefactos. La guía de Scrum@Scale recomienda que los equipos sean de no más de 5 miembros.

2. Scrum of Scrums:

Para coordinar múltiples equipos, Scrum Scale utiliza el concepto de "Scrum de Scrums", donde representantes de cada equipo se reúnen periódicamente para compartir actualizaciones, coordinar dependencias y abordar problemas inter equipos.

3. Roles Adicionales:

Scrum Scale introduce roles adicionales para facilitar la coordinación a mayor escala, como el "Scrum Máster de Scrum Masters" y el "Product Owner de Product Owners". Estos roles supervisan y facilitan la colaboración entre los Scrum Masters y Product Owners de las células individuales.

4. Priorización Escalada:

La priorización de elementos de trabajo se realiza a diferentes niveles, desde equipos individuales hasta niveles más altos de coordinación. Scrum Scale proporciona una estructura para escalonar la planificación y la priorización a medida que se amplía a través de la organización.

5. Herramientas y Prácticas Específicas:

Scrum Scale sugiere herramientas y prácticas específicas para abordar desafíos a mayor escala, como el "Scrum of Scrums Backlog" para gestionar dependencias y el "MetaScrum" para la colaboración entre equipos.

6. Flexibilidad y Adaptabilidad:

Un principio clave de Scrum Scale es la flexibilidad y adaptabilidad. Las organizaciones pueden adaptar el marco de trabajo según sus necesidades específicas, manteniendo al mismo tiempo los principios fundamentales de Scrum.

7. Mejora Continua:

Scrum Scale fomenta la mejora continua a través de la retroalimentación y la adaptación constante. Los equipos y las estructuras pueden evolucionar para abordar los desafíos a medida que surgen.

6.3 Roles para el escalamiento de Scrum

6.3.1 Scrum of Scrums Master (El Scrum de Scrum Master)

Líder al servicio de los Scrum of Scrums. Es responsable de asegurar que se realicen los eventos escalados, que sean productivos, positivos y se mantengan dentro del time-box. También es responsable de las entregas de incrementos del conjunto de células (Scrum of Scrums) y de mejorar la capacidad: esto es: mayor rendimiento del equipo, menor costo y mayor calidad. Puede ser un Scrum Master del equipo o un profesional dedicado específicamente para este rol.

Si hay varios Scrum of Scrums agrupados en un Scrum of Scrums of Scrums se necesita un de un Scrum of Scrum of Scrums Mater para la coordinación más amplia.

Tabla 20 – Scrum of Scrums Máster y sus responsabilidades

Categoría	Función
Organización	Hacer que sean visibles los impedimentos y gestionar su eliminación. Mantener informado a la organización sobre el progreso del desarrollo del producto, estado de eliminación de impedimentos y otras métricas.
Scrum of Scrums	Coordinar la entrega de los equipos con los planes de release del equipo de Product Owners. Comprender las dependencias entre equipos (SoS y no-SoS) y facilitar su coordinación y comunicación. Facilitar la Scaled Daily la Scaled Retrospective.

Chief Product Owner	Trabajar estrechamente con el CPO para entregar incremento de producto potencialmente implementable como mínimo una vez por Sprint Mantener informado al CPO sobre el progreso del desarrollo del producto.
----------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Fuente: Elaboración propia con base en la guía Scrum@Scale

6.3.2 Chief Product Owner (Jefe de Product Owners)

Es responsable juntos con sus SoSM asociados, por la implementación eficiente de incrementos de producto de acuerdo con el Plan de Release. El CPO puede ser de un Scrum Team individual en donde ejerce el rol, o puede ser una persona específicamente dedicada que no está en ninguna célula. Al momento de escalar este rol, se agrega una “C” por cada nivel, ejemplo Chief of Chief Product Owner y así sucesivamente dependiendo de la red.

Tabla 21 – Chief Product Owner y sus responsabilidades

Categoría	Función
Equipo de Product Owners	Coordina las prioridades junto con el Equipo de Product Owners. Las alinean con las necesidades de los Stakeholders y los clientes. Definir qué métricas monitorea el Equipo de Product Owners. Facilitar la Review y refinamiento del Backlog siendo guiado por el CPO.
Visión estratégica	Definir la visión estratégica para todo el producto. Crear un único backlog priorizado para todos los equipos. Evaluar le feedback de los clientes sobre los productos y ajustar el backlog común. Facilitar el evento MetaScrum o Executive MetaScrum.
Plan de Release	Definirlo con el Equipo de Product Owners. Implementarlo con los SoSM asociados.

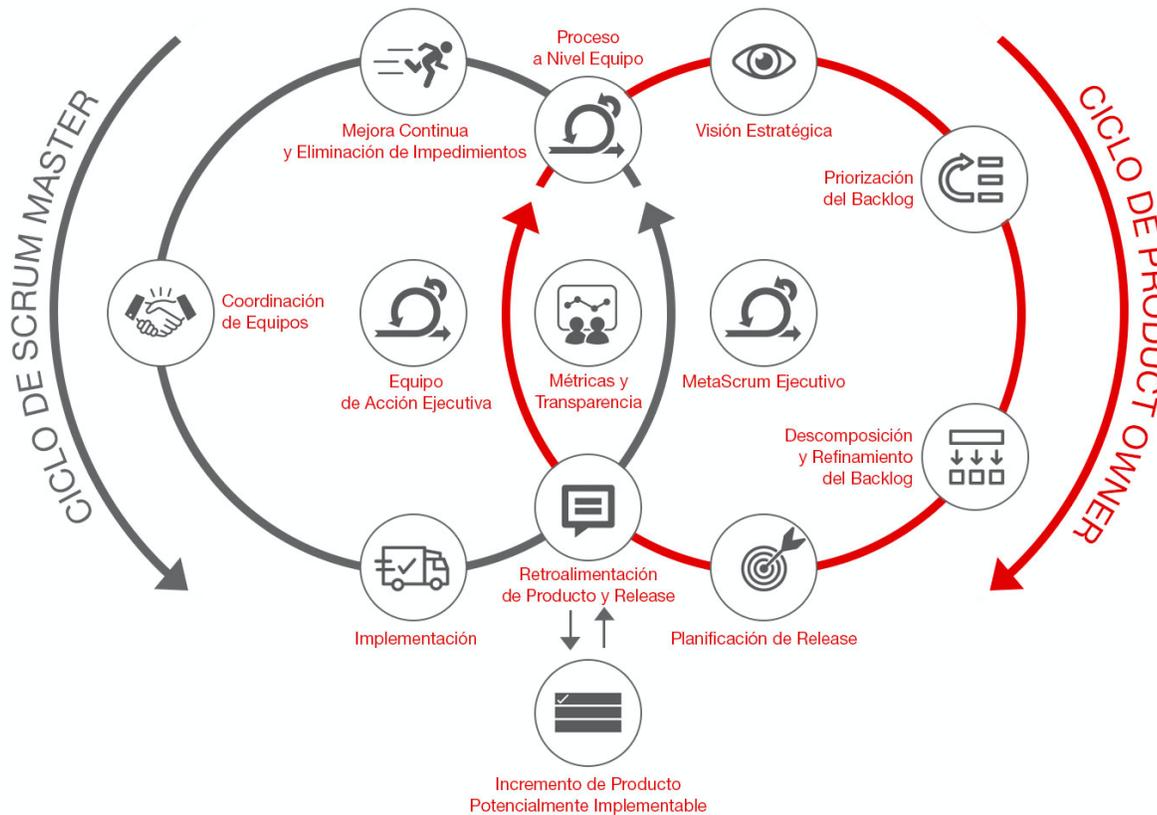
Fuente: Elaboración propia con base en la guía Scrum@Scale

6.4 Proceso de escalamiento

Lo primero que debe comprenderse es que Scrum@Scale tiene dos ciclos, el de Scrum Master enfocado en el cómo hacer las cosas y el de Product Owner centrado en el qué hacer.

Ilustración 10 - Scrum@Scale y sus ciclos





Fuente: Guía Scrum@Scale

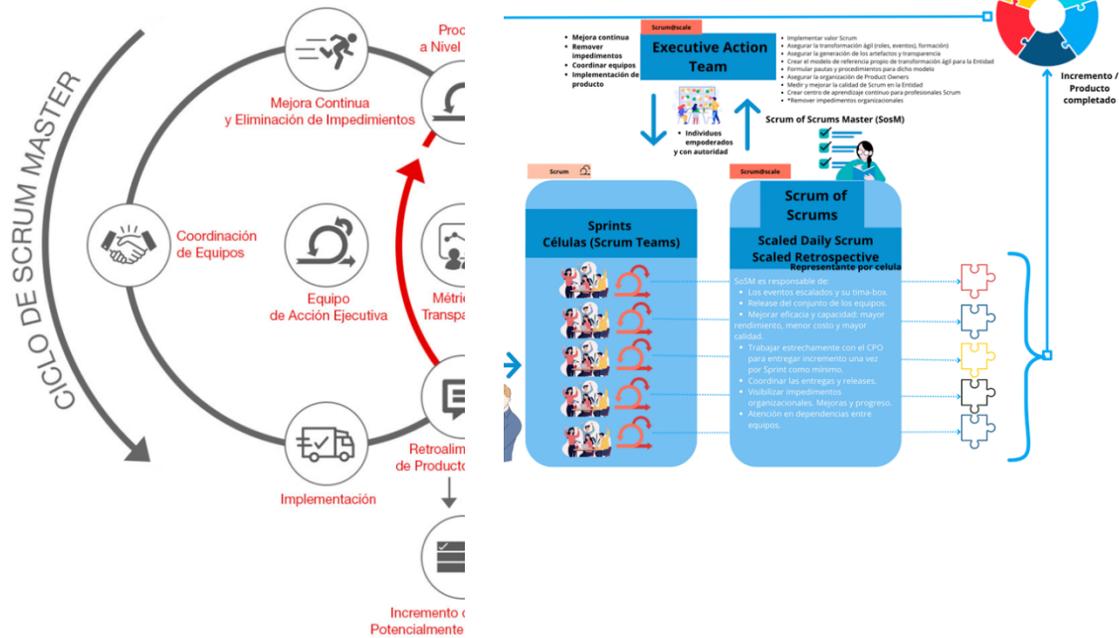
Los ciclos se explican en el siguiente apartado. Se sugiere que el lector vaya haciendo la conexión con la adaptación de Scrum@Scale para la DIAN según se presenta en la Ilustración 9 - Scrum Scale en la DIAN

6.4.1 Ciclo de Scrum Master

El ciclo de Scrum Master comprende todas las interacciones y componentes que deben darse para asegurar el CÓMO, responsabilidad del equipo de desarrollo y por tanto facilitado por el Scrum Master y todos los escalados en este rol. Acá se coordinan los esfuerzos para implementar Scrum con base en la inspección y adaptación.

Son responsables los Scrum Master, Scrum of Scrums Master y el Executive Action Team como centro de este ciclo.

Ilustración 11 - Ciclo de Scrum Máster

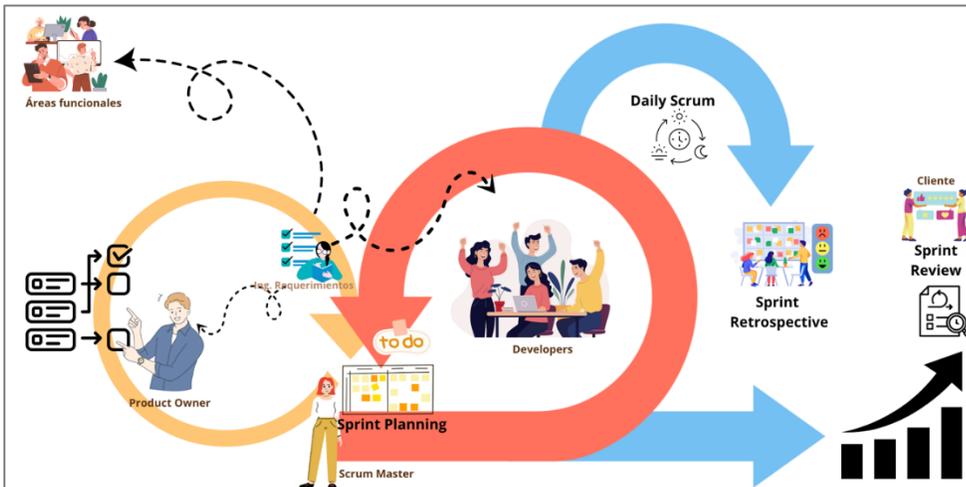


Fuente: Guía Scrum@Scale y elaboración propia

Scrum Team o célula

Como se ha dicho, la base fundamental para poder escalar es contar con equipos Scrum o células que estén operando eficientemente y entregando incremento al final de cada Sprint. De esto ya se ha hablado ampliamente a lo largo del capítulo SCRUM ESTÁNDAR - GESTIÓN DE PROYECTOS ÁGILES. Una célula de Scrum, con todos sus componentes, se grafica de la siguiente manera:

Ilustración 12 - Scrum Team o célula en la DIAN

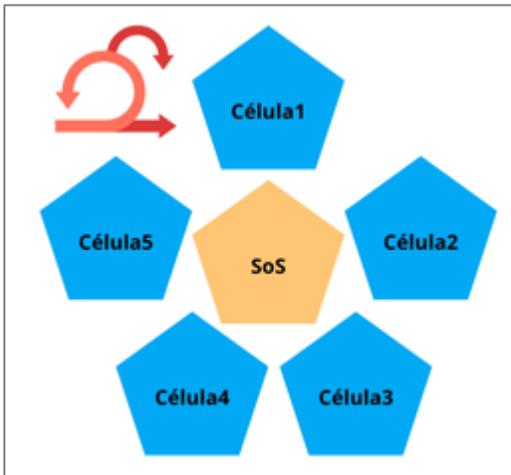


Fuente: Elaboración propia
Scrum of Scrums (SoS)

Un conjunto de equipos que necesitan coordinarse es un Scrum of Scrums, para entregar incrementos de manera coordinada que se integrarán para conformar un incremento mayor de valor para el cliente. Requiere contar con el rol SoSM Scrum of Scrum master líder al servicio de las células que trabajan en ese SoS.

Un SoS de la DIAN Se representa así:

Ilustración 13 - Scrum of Scrums en la DIAN



Fuente: Elaboración propia

Un SoS opera como si fuera un Scrum Team, siendo responsable de un conjunto completamente integrado de incrementos de producto potencialmente entregables al final de cada sprint.

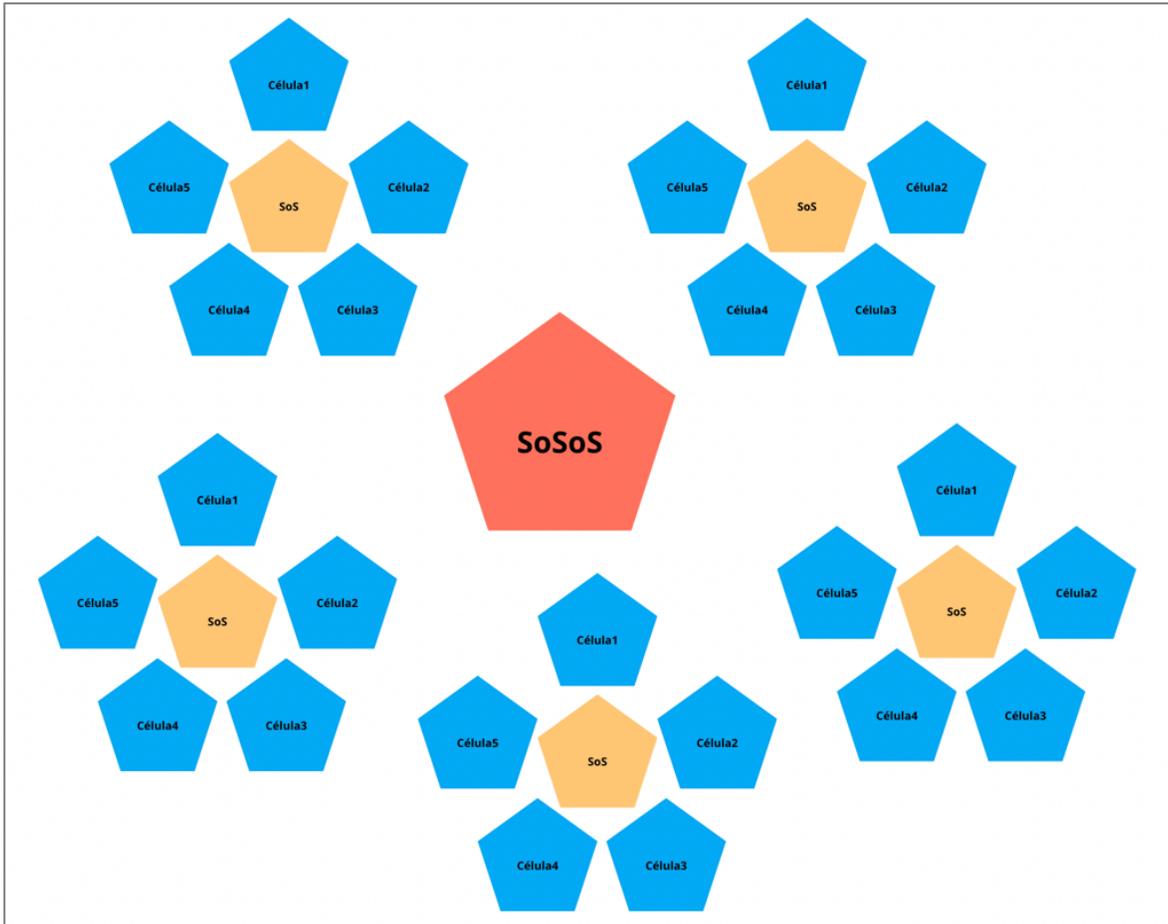
Scrum of Scrum of Scrums (SoSoS)

Se puede seguir escalando si la Entidad lo requiere, debido a la cantidad de células Scrum que activan para atender el mismo proyecto. Por lo cual, se agrupan cada 5 células como máximo en un SoS, y cada 5 SoS como máximo conforman un nuevo nivel de Scrum of Scrums of Scrums.

Cada SoSoS debe tener las versiones escaladas de los roles, artefactos y eventos Scrum of Scrums. Asimismo, interactúa con cada uno de sus SoS de la misma manera que un SoS actúa con una célula Scrum.

Un Scrum of Scrum of Scrum, se puede graficar para su comprensión, como se observa en la siguiente ilustración, en donde se sincronizan 25 equipos, de manera lineal.

Ilustración 14 - Scrum of Scrum of Scrums en la DIAN



Fuente: Elaboración propia

Siempre que la Entidad necesita, podrá seguir escalando los SoS con la misma lógica explicada y guardando todos los eventos, roles y artefactos del Scrum estándar.

Executive Action Team (EAT)

Es el eje del ciclo del Scrum Master. Es un equipo de liderazgo empoderado para tomar decisiones y remover impedimentos de orden organizacional. Por lo tanto, debe estar integrado por individuos que cuenten con total respaldo político y económico para poder eliminarlos.

Su función es coordinar varios Scrum of Scrums (o redes más grandes) e interactuar con la parte no-ágil de la Entidad para procurar su alineación y mitigar impactos negativos.

Debe contar con su propio Product Owner, Scrum Master y un Backlog de Acción ejecutiva. El EAT es responsable de la calidad de Scrum en la Entidad, por tanto, todos los Scrum Masters les reporta.

Tabla 22 – Executive Action Team y la Scaled Daily

Producto	Sistema operacional ágil
-----------------	--------------------------

	Modelo de referencia de Scrum@Scale Product Backlog con iniciativas concretas para el logro de sus responsabilidades
Integrantes	Profesionales con poder de decisión (delegados y respaldados con autoridad para remover impedimentos de manera inmediata) Coordinación CENIT Scrum of Scrum Masters
Convoca	Scrum Master del EAT
Participantes	Todos los miembros del EAT y los Scrum of Scrum Masters

Fuente: elaboración propia

Producto del EAT:

Un sistema operacional ágil para la Entidad, por lo cual deben crearlo y operarlo a través de iniciativas para esa transformación en un Backlog propio con ese propósito. Debe incluir las mejoras a los procesos que eliminan impedimentos y necesitan ser estandarizadas.

La Scaled Daily no es la única reunión que tiene el EAT, pueden organizar las sesiones que requieran a discreción para el desarrollo de sus responsabilidades, las cuales son:

Tabla 23 – EAT y sus responsabilidades

Categoría	Responsabilidades
Organización	Crear el sistema operacional ágil para la DIAN Aplicarlo al modelo de referencia (primer SoS del escalamiento) Definir reglas organizacionales, procedimientos y lineamientos para posibilitar la agilidad.
Roles	Asegurar la conformación de la una organización consistente de Product Owners, con fondo y respaldo institucional.
Métricas	Medir y mejor la calidad de Scrum en la DIAN
Capacidades	Desarrollar capacidades en la DIAN para conseguir la agilidad de negocio Crear un centro de aprendizaje continuo para profesionales Scrum y la experimentación en nuevas formas de trabajo.

Fuente: Elaboración propia

Componentes del ciclo de Scrum Masters

Los Scrum Masters, Scrum of Scrums Master y el Executive Action Team trabajan conjuntamente como un todo para implementar los siguientes componentes del ciclo:

Tabla 24 – componentes del ciclo Scrum Máster

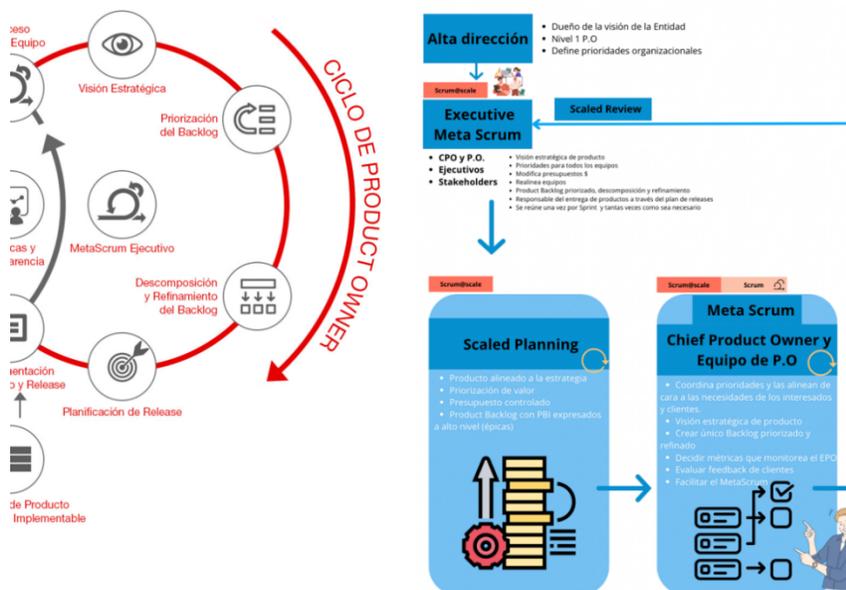
Mejora continua y eliminación de impedimentos	Coordinación entre equipos	Implementación de producto
<ul style="list-style-type: none"> • Identificar impedimentos organizacionales y removerlos • Garantizar la transparencia y visibilidad en la organización para efectuar cambios • Mantener un ambiente efectivo para priorizar • Verificar que las mejoras hayan impactado de manera positiva las métricas del equipo/producto. 	<ul style="list-style-type: none"> • Sincronizar procesos similares en las células relacionadas • Mitigar las dependencias entre células para evitar que se conviertan en impedimentos • Mantener alineadas normas y pautas de las células para lograr resultados consistentes 	<ul style="list-style-type: none"> • Lograr un único release coordinado (Equipo de P.O define contenido y momento óptimo de entrega a los clientes). • Para asegurarlo, los SoS deben: • Entregar un flujo constante de producto valioso terminado • Hacer la integración de los incrementos • Garantizar la alta calidad de la experiencia de cliente

Fuente: Elaboración propia

6.4.2 Ciclo de Product Owner

El ciclo de Product Owners se enfoca en el QUÉ se va a realizar, partiendo de la estrategia de la Entidad que se va atendiendo a través de los productos, todo se va engranando con la visión de los altos directivos y se garantiza con el buen ejercicio de los Chief Product Owners y el Equipo de Product Owners.

Ilustración 15 - Ciclo de Product Owners



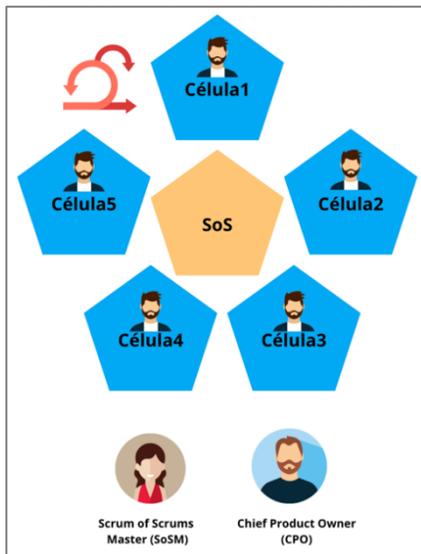
Fuente: Guía Scrum@Scale y elaboración propia
Equipo de Product Owners y Chief Product Owner

En este ciclo se requiere de la participación de los Product Owners de cada célula de trabajo y del rol Chief Product Owner que es el par del Scrum of Scrum Master por cada SoS.

El escalamiento del Equipo de Product Owners va en correspondencia con los Scrum of Scrums asociados.

Tal como se ve en la gráfica a continuación.

Ilustración 16 - Chief Product Owner y equipo de Product Owners



Fuente: Elaboración propia

El CPO es el responsable como Product Owner del conjunto de células. Para ello debe trabajar articuladamente con el Equipo de Product Owners y así enfocarse conjuntamente en que las prioridades estén comprendidas, alineadas con la Estrategia y necesidades de los clientes.

En Scrum@Scale, cada Product Owner de célula mantiene las mismas responsabilidades que en Scrum estándar, sólo que debe tomar las prioridades del Product Backlog compartido o generar elementos independientes según sea necesario para alcanzar los objetivos del negocio.

Este equipo de Product Owners debe trabajar conjuntamente con el CPO, todo esto sucede en un patrón de Scale llamado “Meta Scrum”, para:

1. Comunicar la visión de producto y hacerla visible en la DIAN.
2. Establecer un excelente relacionamiento con los Stakeholders para asegurar su apoyo en la implementación del Backlog.
3. Generar un Backlog único y priorizado que evite la duplicación de trabajo.
4. Trabajar con los Scrum of Scrums Máster para crear la “Definition of Done” que aplique a todas las células.
5. Eliminar dependencias creadas por los equipos.
6. Crear un roadmap y plan de release.
7. Monitorear las métricas sobre el producto y su impacto.

Tabla 25 – Meta Scrum y sus reuniones

Producto	Product Backlog Estratégico (Nivel features)
Convoca	Chief Product Owners con apoyo CenIT
Participantes	Chief Product Owners Equipo de Product Owners
Frecuencia	Al menos una vez por Sprint

Fuente: Elaboración propia

Executive MetaScrum (EMS)

Es un grupo de líderes que imprime dinamismo al escalamiento, establece la visión organizacional y prioridades estratégicas alineando a todas las células con los objetivos comunes.

Este grupo se reúne al menos una vez por Sprint, donde participan el CPO, Equipo de Product Owners, altos directivos y los Stakeholders clave. Estos últimos expresan preferencias y necesidades a los P.O y también se modifican presupuestos y alinean los Scrum Teams para maximizar la entrega de valor.

Componentes del ciclo de Product Owners

Los Product Owners, Chief Product Owners y el Executive MetaScrum trabajan conjuntamente para satisfacer los componentes del ciclo que se citan abajo.

Tabla 26 – Componentes del ciclo del Product Owner

Visión estratégica	Priorización del Backlog
<ul style="list-style-type: none"> Definir una visión estratégica convincente para clientes como para células. Alinear toda la Entidad en un camino futuro compartido. Claridad que permita la creación de Objetivos de Producto concretos. Describir qué hará la Entidad para aprovechar sus activos clave. Demostrar capacidad de responder rápidamente a las condiciones cambiantes del contexto. 	<ul style="list-style-type: none"> Eliminar la competencia entre prioridades para evitar la tracción de equipos en direcciones opuestas. Priorizar el Backlog identificando en orden claro los productos, capacidades y servicios que se entregarán. Reflejar en el orden del Backlog la creación de valor, mitigación de riesgos y dependencias internas. Priorizar las iniciativas del alto nivel antes del refinamiento.
Descomposición y refinamiento	Planificación de Release
<ul style="list-style-type: none"> El Backlog de un CPO contiene PBI de un alcance mayor, es posible que se deban descomponer y comprender mejor. Identificar productos y proyectos complejos y objetivo de producto asociados. Descomponer los productos y proyectos complejos en elementos independientes. 	<ul style="list-style-type: none"> Puede abarcar uno o más entregas al cliente- Pronosticar el plan de entregas de incrementos de productos y funcionalidades. Comunicar las expectativas de entregas a los Stakeholders Comunicar el impacto financiero del plan de entregas.

- Asegurarse que todos los PBI del Backlog pueden ser refinados por las células y se entregados en un Sprint.

Fuente: Elaboración propia

6.4.3 Conectar los ciclos

Scrum@Scale está diseñado para escalar la productividad y hacer que la Entidad entregue el doble del valor a la mitad del costo, pero para eso se necesita asegurar que los ciclos funcionen bien y se conecten mejor.

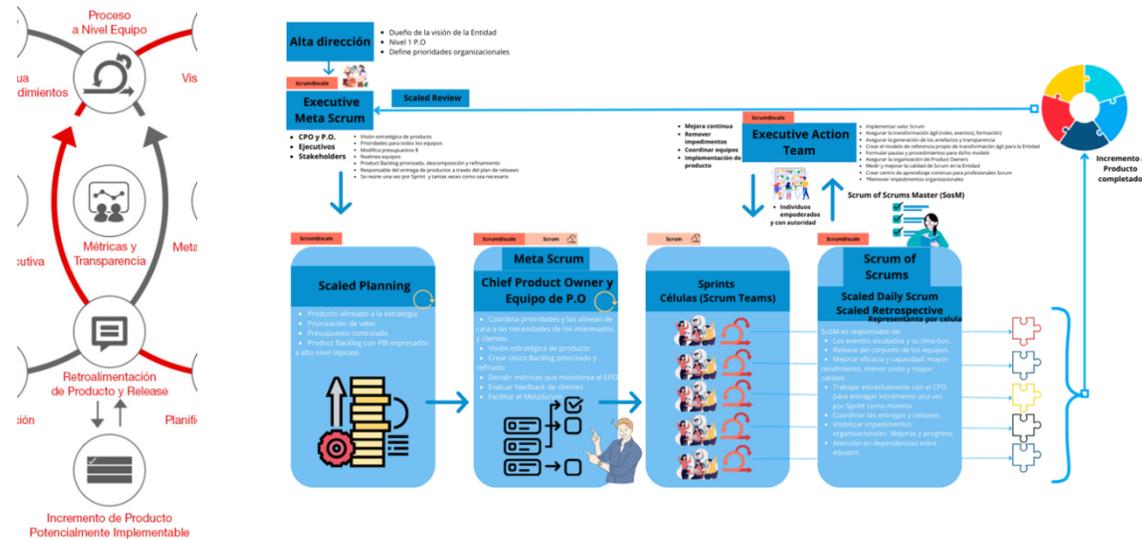
Los ciclos se conectan primero que todo con el “Proceso de equipo”, es decir, con la ejecución de los Sprints que llevan a cabo las células de cada SoS.

Tanto los Scrum o Scrums Masters como los Chief Product Owners asumen sus responsabilidades Enel proceso de equipo acerca del Cómo y del Qué.

Luego, los ciclos se vuelven a conectar en el componente “Retroalimentación del Producto y Release” donde se interpreta la respuesta que da el cliente sobre el producto. Su propósito es validar supuestos, comprender cómo los clientes usan e interactúan con el producto, y capturar nuevas ideas y requisitos emergentes para nuevas funcionalidades. (Es una Sprint Review escalada).

Para ello se usan “Métricas” que permitan tomar decisiones basadas en empirismo sobre la adaptación del siguiente ciclo de entrega.

Ilustración 17 - Conectar los ciclos



Fuente: Guía Scrum@Scale

Tabla 27 – Retroalimentación y métricas

Retroalimentación de producto y release	Métricas
La red de Product Owners interpreta la retroalimentación del producto hacer mejora continua que se debe reflejar en el Backlog.	Para la Entidad se ha desarrollado un apartado muy completo de Métricas que se puede ver con detalle en el capítulo Monitoreo y métricas .
La red de Scrum Masters interpretada la retroalimentación de release y asegura la mejora continua de los mecanismos de implementación.	La transparencia es fundamental y radical en el escalamiento, por tal razón las Métricas deben ser tomadas como objetividad para poder inspeccionar causas de posibles "fracasos" y hacer las adaptaciones para mejorar.

Fuente: Elaboración propia

1. PATRONES

en el cual se relacionen buenas prácticas ampliamente usadas en Scrum. Finalmente, el apartado CENTRO DE RECURSOS Y HERRAMIENTAS ofrece un compilado de materiales que se han venido construyendo para que los Scrum Masters y en general miembros de los Scrum Teams e interesados, profundicen en sus conocimientos.

2. GLOSARIO

- **Cadencia.** Se refiere al ritmo regular y predecible con el cual se realizan ciertos eventos dentro de un ciclo de desarrollo. Scrum tiene eventos definidos que estructuran el tiempo y la interacción en un proyecto. Estos eventos están diseñados para maximizar la transparencia, inspección y adaptación continua, y ocurren regularmente y de manera predecible. La idea detrás de la cadencia en Scrum es proporcionar un marco temporal claro y consistente para la planificación, la revisión y la mejora continua, permitiendo que el equipo se adapte rápidamente y entregue valor de manera regular. La cadencia también facilita la medición del progreso y la identificación temprana de posibles problemas.

Fuente. Definición de la DGIT con base en la guía Scrum disponible en <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-Latin-South-American.pdf>

- **Célula.** Se refiere al equipo de proyecto Scrum, conformado por el Scrum Máster, el Product Owner y los developers.

Fuente. Definición de la DGIT con base en la guía Scrum disponible en <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-Latin-South-American.pdf>

- **Deuda técnica.** En resumen, es un concepto que destaca la importancia de mantener la calidad y la sostenibilidad del software a lo largo del tiempo. Se refiere a la acumulación de decisiones de diseño o implementación que, aunque funcionales, no son óptimas y pueden causar problemas en el futuro. Es como un "préstamo" que se toma al elegir soluciones rápidas o no ideales para cumplir con plazos o requerimientos inmediatos, pero que con el tiempo puede requerir más trabajo para corregir o mejorar. Esta "deuda" puede tomar diversas formas, como código no optimizado, falta de

documentación, soluciones temporales, falta de pruebas adecuadas, entre otros. Similar a una deuda financiera, la deuda técnica debe "pagarse" eventualmente, ya que ignorarla puede afectar la calidad del software, la mantenibilidad y la capacidad de respuesta a los cambios futuros. La gestión de la deuda técnica implica evaluar continuamente el estado del código, identificar áreas de mejora y asignar recursos para abordar esas áreas.

Fuente. Definición de la DGIT con base en la guía Scrum disponible en <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-Latin-South-American.pdf>

- **Definition of Done (DoD).** El conjunto de criterios claros y específicos que deben cumplirse para que un elemento de trabajo (PBI) se considere "hecho" al final de un Sprint y por tanto, sea aceptado. Sirve para orientar al equipo hacia el objetivo del producto y así ayuda a garantizar que todos los elementos de trabajo entregados por el equipo de desarrollo cumplan con los estándares de calidad y estén listos para ser entregados al cliente o usuario final.
Fuente. Definición de la DGIT con base en la guía Scrum disponible en <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-Latin-South-American.pdf>
- **Definition of Ready (DoR).** El conjunto de criterios preestablecidos que deben cumplirse antes de que un elemento de trabajo pueda considerarse listo para ser tomado en un Sprint. Estos criterios aseguran que el PBI esté claramente definido, los requisitos estén comprendidos y que los developers tengan toda la información necesaria para comenzar el trabajo de manera efectiva. La Definition of Ready actúa como una herramienta que facilita una planificación más precisa y una ejecución más fluida, ya que garantiza que el equipo tenga la claridad y los recursos necesarios para abordar las tareas de manera eficiente cuando se comprometen a incluirlas en un Sprint.
Fuente. Definición de la DGIT con base en la guía Scrum disponible en <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-Latin-South-American.pdf>
- **Épica.** Consiste en un elemento o bloque de trabajo (o PBI - Product Backlog ítem) que por su gran extensión debe desagregarse en varias historias de usuario de menor tamaño, las cuales puedan gestionarse adecuadamente a lo largo de uno o varios Sprints del proyecto. Una épica debe ser flexible y debe estar basada en las necesidades de los clientes.
Fuente. Definición de la DGIT con base en la guía Scrum disponible en <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-Latin-South-American.pdf>
- **Historia de usuario.** Son conversaciones acerca de lo que el usuario quiere o necesita en el producto. Las historias de usuario están diseñadas para asegurar que los requisitos del cliente estén claramente representados y puedan ser plenamente comprendidos por el scrum Team.
Fuente. Definición de la DGIT con base en la guía Scrum disponible en <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-Latin-South-American.pdf>

- **Impedimentos organizacionales.** cualquier barrera, obstáculo o problema a nivel organizativo que afecta negativamente el desarrollo y entrega del producto. Estos impedimentos pueden provenir de políticas internas, estructuras organizativas, cultura empresarial, o cualquier otro aspecto dentro de la organización que obstaculice el rendimiento del equipo. Ejemplos: procesos burocráticos, falta de apoyo de la alta dirección, cultura organizacional resistente al cambio, problemas de comunicación, falta de colaboración de las Direcciones de Gestión, políticas inadecuadas de talento humano o restricciones presupuestarias.
Fuente. Definición de la DGIT con base en la guía Scrum disponible en <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-Latin-South-American.pdf>
- **Inception o inicio de proyecto.** Se refiere a la fase inicial del proyecto durante la cual se realizan actividades cruciales para establecer una comprensión común y sólida del alcance (producto), los objetivos y las restricciones del proyecto. Esta fase es fundamental para sentar las bases del trabajo que se llevará a cabo durante el desarrollo. También se definen los requisitos iniciales a nivel general, se establece el equipo y asignan roles y responsabilidades, se identifican riesgos, se hace una planeación inicial de alto nivel que proporciona una visión relativa de la línea de tiempo del proyecto.
Fuente. Definición de la DGIT con base en la guía Scrum disponible en <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-Latin-South-American.pdf>
- **Incremento de producto.** Es un peldaño hacia el objetivo del producto. Cada Incremento se acumula con los anteriores y se verifica su funcionamiento.
Fuente. Definición de la DGIT con base en la guía Scrum disponible en <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-Latin-South-American.pdf>
- **PBI.** "Product Backlog ítem" en el contexto de Scrum y desarrollo ágil de software. En español, se traduce como "Ítem del Backlog del Producto". Un PBI es una unidad de trabajo que representa una funcionalidad o característica específica que se debe desarrollar, probar y entregar como parte del producto final. Los PBI son elementos individuales que se encuentran en el backlog del producto y son priorizados por el Product Owner en función de su valor para el cliente y el objetivo del proyecto. Durante la etapa de sprint Planning, los PBI se seleccionan para ser abordados por el equipo de desarrollo en el siguiente sprint.
Fuente. Definición de la DGIT con base en la guía Scrum disponible en <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-Latin-South-American.pdf>
- **Portafolio.** Es un conjunto de programas relacionados, con el fin de facilitar su gestión en torno al cumplimiento de los objetivos y metas establecidas por el negocio dentro del plan estratégico de la Entidad.
Fuente. Definición de la DGIT con base en las definiciones de la PMI resumidas en: <https://asana.com/es/resources/what-is-project-portfolio-management>

- **Programa.** Es grupo de proyectos relacionados que se pueden clasificar a su vez en subprogramas, con el objetivo de entregar resultados basados en las metas propuestas dentro del portafolio de proyectos.
Fuente. Definición de la DGIT con base en las definiciones de la PMI resumidas en: <https://openaccess.uoc.edu/bitstream/10609/45590/7/lameijideTFC0116memoria.pdf>
- **Proyecto.** Es un conjunto de esfuerzos coordinados de forma temporal, para llevar a cabo la creación de un producto o servicio, el cual responderá a una necesidad que permita el alcance de las metas establecidas dentro del portafolio de proyectos. Los proyectos son únicos, están definidos dentro de un espacio de tiempo y cuentan con restricciones de recursos. Asimismo, están sujetos a incrementos continuos de producto/servicio, materializados en entregables y definidos previamente en el backlog de producto.
Fuente. <https://www.ealde.es/pmbok-7/#:~:text=Todas%20responden%20al%20est%C3%A1ndar%20o,el%20mejor%20de%20los%20casos.>
- **Refinamiento de backlog del producto.** Es un proceso continuo donde el equipo de desarrollo, el Scrum Máster y el Product Owner junto con el ingeniero de requerimientos colaboran para mejorar y clarificar los elementos de trabajo del Product Backlog. Durante estas sesiones de refinamiento, se revisan y ajustan las funcionalidades y PBI para asegurar que estén bien definidas, estimadas y priorizadas. Además, se pueden agregar detalles adicionales, se eliminan ambigüedades y se ajusta la prioridad según las necesidades cambiantes del proyecto. El refinamiento del Product Backlog contribuye a mantener una lista de PBI siempre lista para ser tomada en el siguiente Sprint Planning, facilitando así la planificación y ejecución efectiva del desarrollo.
Fuente. Definición de la DGIT con base en la guía Scrum disponible en <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-Latin-South-American.pdf>
- **Sprint Planning.** En el contexto de Scrum, es la reunión de planeación que hace el equipo Scrum al inicio de cada Sprint. Define cual es la meta del Sprint con lo cual se llega al Sprint Backlog.
Fuente. Definición de la DGIT con base en la guía Scrum disponible en <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-Latin-South-American.pdf>
- **Stakeholder.** Se refiere a todas las personas que están interesadas en el proyecto.
Fuente. [https://www.spentamexico.org/v12-n3/A15.12\(3\)230-255.pdf](https://www.spentamexico.org/v12-n3/A15.12(3)230-255.pdf)

3. OBJETIVO

Presentar el marco de gestión ágil de proyectos de la Dirección de Gestión de Innovación y Tecnología, el cual se encuentra articulado a través del Procedimiento PR-IIT-0153 Gestión de proyectos de tecnología.

4. ALCANCE

Este marco para la transformación ágil en la gestión de proyectos está dirigido a todos los colaboradores y áreas de la entidad que desarrollen proyectos a partir del procedimiento “PR-IIT-0153 Gestión de proyectos de tecnología”.

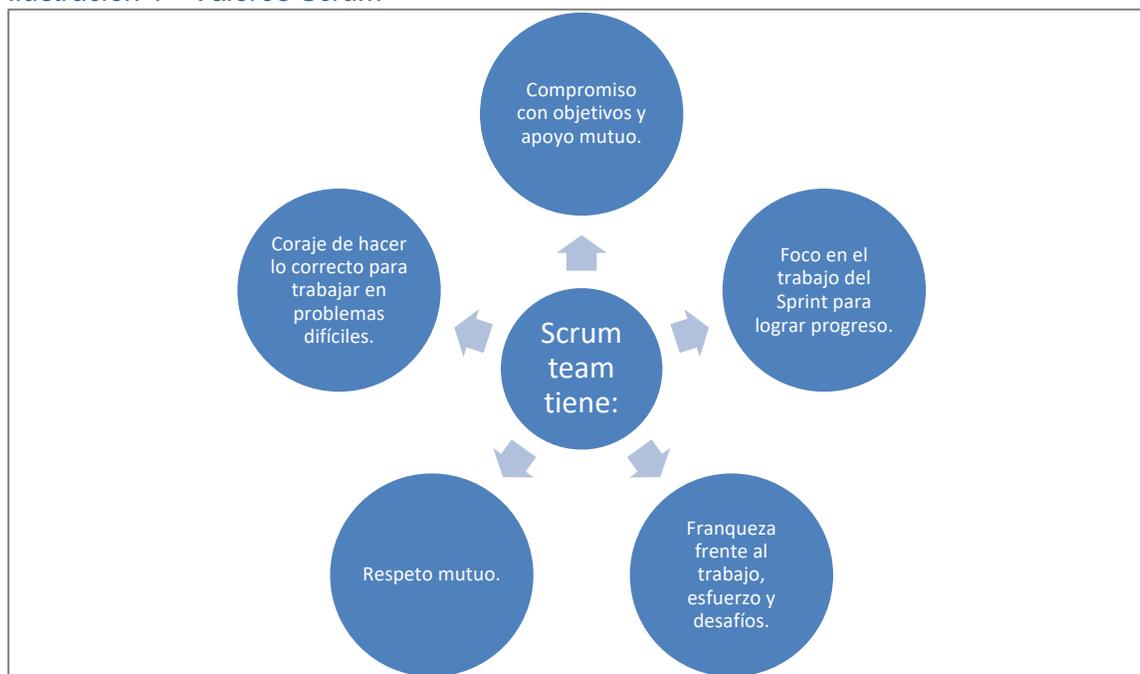
5. ECOSISTEMA PARA LA TRANSFORMACIÓN ÁGIL DE LA ENTIDAD Y LA GESTIÓN DE PROYECTOS

A continuación, se presenta el esquema general de la gestión ágil de proyectos que ha adoptado la Entidad basándose en la guía 2020 del marco Scrum. Lo primero es que todo proyecto ágil debe cuidar el cumplimiento de tres **pilares empíricos**:

- a) **Transparencia:** tanto el proceso realizado como todo lo que se espera obtener debe ser suficientemente claro para cada uno de los *interesados* y miembros del equipo Scrum.
- b) **Inspección:** todos los artefactos y el progreso hacia los objetivos se examinan con frecuencia para detectar variaciones y validar la calidad. Esto se hace dentro de cada Sprint.
- c) **Adaptación:** a través de la inspección se detecta aquello que hay que adaptar, y esto se hace de manera inmediata para evitar desviaciones en el progreso del trabajo o productos inaceptables. El equipo de trabajo debe autogestionar la adaptación con base en su capacidad de aprendizaje. La adaptación reduce la deuda técnica, la cual se debe evitar siempre o atender de manera inmediata cuando se detecte. La forma de atender la deuda técnica es en el momento de identificarla, parar lo que se está haciendo y solucionar el problema presentado.

El equipo Scrum debe cumplir con los siguientes **valores**:

Ilustración 1 - Valores Scrum



Fuente: Inspiración en la Guía Scrum 2020

Los **roles de la gestión ágil de proyectos** para la Entidad y con base en sus propias características, son los que se presentan a continuación:

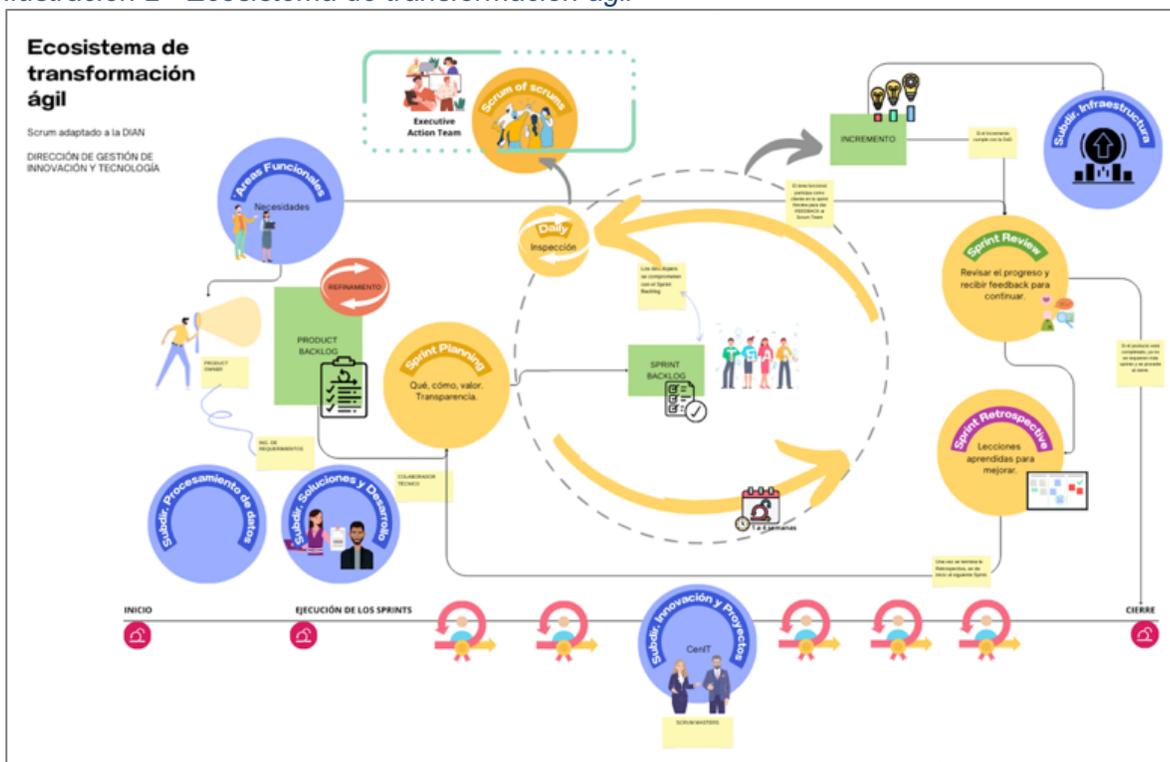
- a) Scrum Máster (Maestro de Scrum) y su escalado cuando aplique Scrum of Scrums Máster
- b) Product Owner (Dueño del producto) y su escalado cuando aplique, Chief Product Owner
- c) Developers (Desarrolladores)
- d) Interesados (Stakeholders)
- e) Agile Coach (Entrenador Ágil)

El ecosistema de transformación ágil de la entidad adopta Scrum como marco de trabajo para la gestión de proyectos y vincula de manera sinérgica las subdirecciones de la Dirección de Gestión de Innovación y Tecnología, a través de los roles expresados previamente, así como de las responsabilidades que comparte al disponer de los andamiajes necesarios a nivel operacional, técnico y de infraestructura para hacer posible la ejecución de los proyectos. Tanto los principios como los valores Scrum son materia intangible difíciles incluso de representar gráficamente porque se llevan a cabo a través de los roles y exclusivamente durante la práctica ágil.

En este ecosistema se integran las áreas funcionales como agentes colaboradores de la ejecución de los proyectos, fundamentando su acción en principios como que prevalecen las relaciones centradas en las personas y se favorece altamente el relacionamiento y buena comunicación entre los responsables del producto y quienes lo necesitan.

La siguiente ilustración pretende visualizar los roles, etapas y eventos que se dan en la transformación ágil a través de la gestión de proyectos. Todos estos aspectos, se detallan enriquecidamente a lo largo de este documento.

Ilustración 2 - Ecosistema de transformación ágil



Fuente: Elaboración propia

La siguiente tabla presenta un resumen de las etapas de la gestión ágil de proyectos, posteriormente se detallan en el apartado SCRUM ESTÁNDAR - GESTIÓN DE PROYECTOS ÁGILES.

Tabla 1 – Etapas de la Gestión Ágil

Etapa	Descripción	Formato
1. Inicio	Es la etapa de inicio del proyecto, en la cual se define qué se quiere obtener como producto. Allí se asigna el Product Owner y el Scrum Master. Se tiene un entendimiento general de las necesidades del área funcional, pero que sea suficiente para plantear el objetivo que se quiere alcanzar, las personas que se quieren impactar y cómo se va a lograr eso a través de grandes funcionalidades en el producto. De esta manera se obtiene una primera versión del Product Backlog que tendrá que ser refinado suficientemente antes de continuar con la etapa de ejecución de los Sprints.	<ul style="list-style-type: none"> Acta de constitución FT-IIT-2010 y Anexo 1. Matriz de riesgos proyectos tecnológicos

Etapa	Descripción	Artefactos
2. Ejecución de los Sprints	<p>Eventos del sprint</p> <p>Un Sprint es el centro de la planeación, ejecución y monitoreo de la gestión ágil de proyectos.</p> <p>Dependiendo del alcance y plazo de ejecución del proyecto, podrá tener más o menos Sprints. La longitud de los Sprints se define en el momento de inicio del proyecto.</p> <p>Cada Sprint ejecutado busca ir incrementando el valor del producto esperado hasta llegar a su total cumplimiento.</p> <p>Mientras no se haya cumplido la entrega completa del producto en el plazo de ejecución, se tendrá que realizar un nuevo Sprint.</p> <p>Los siguientes son los eventos que obligatoriamente se dan dentro de cada Sprint y en donde tienen lugar permanente la inspección y la adaptación:</p> <ul style="list-style-type: none"> Sprint Planning (Planificación): esta es una etapa transversal a la ejecución del proyecto, dado que consiste en la preparación del Sprint Backlog. El equipo Scrum se enfoca en qué hacer, cómo hacerlo y por qué. 	<p>Se obtiene el Product Backlog, un artefacto que se va depurando y detallando a lo largo de la ejecución del proyecto. Consiste en una lista ordenada de los Product Backlog Items (PBI) por hacer.</p> <p>Los PBI se refinan, quiebran o desglosan por parte del equipo de trabajo.</p> <p>El Product Backlog se registra en la herramienta de gestión ágil determinada por la entidad.</p> <p>El Product Owner ayuda al equipo a entender los PBI, lo que se quiere, pero no les dice cómo hacerlo.</p> <p>El Sprint Backlog es el artefacto de cada Sprint Planning. Consiste en la definición del objetivo del Sprint y su plan de trabajo detallado. Allí se contempla qué se hará (PBI detallados y</p>

Etapa	Descripción	Artefactos
	<ul style="list-style-type: none"> • Daily: Los developers o equipo de trabajo, cada día, a una hora definida se reúnen para observar su acercamiento a cumplir con el objetivo del Sprint, identificar si hay obstáculos y ver cómo removerlos. <p>El equipo de trabajo desarrolla el Sprint Backlog durante la duración del Sprint. Para ellos, se autoorganizan y autogestionan, es decir, definen quién trabaja en qué y las reuniones que tendrán para avanzar en las tareas técnicas. Toda esta dinámica es discrecional del equipo de trabajo.</p> <p>El monitoreo y control de los avances del Sprint Backlog se da durante cada Sprint.</p> <ul style="list-style-type: none"> • Sprint Review (Monitoreo y Control) este evento dirigido a los interesados y en el que participa todo el equipo Scrum, se enfoca en presentar y validar el incremento de producto. Permitiendo que el cliente realice claridades sobre lo que se espera del siguiente Sprint. Este evento se enfoca en qué se hizo, qué se espera obtener para el siguiente Sprint y que el cliente lo valide. <p>*Es importante aclarar que la Sprint Review no es para realizar pruebas detalladas, dado que estas se debieron realizar durante la ejecución del Sprint.</p> <ul style="list-style-type: none"> • Sprint Retrospective: Es un evento sólo para el equipo Scrum, se enfoca en identificar 1 o 2 puntos para mejorar en el siguiente Sprint, los cuales deben quedar incluidos en el Sprint Backlog. Allí se identifican las acciones a mantener, las que se deben modificar o eliminar, lo que se debe iniciar. <p>Dos momentos importantes:</p> <ul style="list-style-type: none"> • Refinamiento del Product Backlog: se realiza por parte del Product Owner y algunos miembros del equipo de trabajo, para detallar las grandes tareas que entrarán al siguiente Sprint. Esta acción se lleva a cabo en la etapa de planeación con un Product Backlog inicial que se refina de manera permanente durante 	<p>priorizados y estimados en esfuerzo) y cómo se van a hacer.</p> <p>El Sprint Backlog es un artefacto realizado por el equipo de trabajo y para ellos. Se registra en la herramienta de gestión ágil de la entidad.</p> <p>Se actualiza a lo largo del Sprint y se inspecciona en las reuniones Daily.</p> <p>Increment. El avance concreto y funcional hacia el objetivo del Product Backlog.</p> <p>Cada incremento se suma a los anteriores y se verifica que funcionen juntos.</p> <p>El producto debe tener una definición de hecho o (definition of done - DoD), que ha de cumplirse en cada incremento hasta lograr el producto completado.</p>

Etapa	Descripción	Artefactos
	<p>la ejecución del proyecto, hasta completar el producto.</p> <ul style="list-style-type: none"> Cierre del producto completado. Una vez se ha cumplido con el producto, se realiza el respectivo cierre en una reunión dentro de una Sprint Review. 	

Etapas	Descripción	Formatos
3. Cierre	<p>Cierre del proyecto:</p> <p>Cierre con el área solicitante.</p>	<p>Acta de finalización</p> <p>Acta de liquidación (sólo cuando se trata de un proveedor)</p>

6. RESPONSABILIDADES

Este apartado presenta el detalle de las responsabilidades de los roles de Scrum, con base en las buenas prácticas de agilismo. Se ha procurado armonizarlas con la realidad de la Entidad.

6.5 Scrum Master o Maestro de Scrum

Es el maestro de Scrum para la organización, su misión principal es facilitar la comprensión del marco Scrum y colaborar con los miembros del equipo Scrum para que lo implementen correctamente. Actúa como agente de cambio que lleva a la entidad hacia la gestión ágil de proyectos y responsable de implementar Scrum como lo señala la guía. Ayuda al equipo Scrum y a la organización a comprender la teoría y llevarla a la práctica.

Su actitud es la de un líder al servicio del scrum Team, del Product Owner y de la organización, para lograr la efectividad de la implementación de Scrum a través de la mejora consistente de buenas prácticas.

También es responsable de la efectividad del Equipo Scrum, lo hace apoyándolo y ofreciendo realimentación continua. Lo guía para ser autogestionado y multifuncional y a enfocarse en los incrementos de alto valor para alcanzar la Definición de Done (DoD). Apoya en la eliminación de impedimentos y se asegura de que todos los eventos sean positivos y productivos y se mantengan dentro de los límites de tiempo establecidos en la guía Scrum

También apoya al Product Owner ayudándolo a encontrar técnicas para la definición del Objetivo de Producto y la gestión del Product Backlog. Se enfoca también en llevar a la organización hacia una implementación clara y sistemática de Scrum, eliminando barreras y generando capacitaciones y entrenamientos para su adopción.

Tabla 2 – Scrum Máster y sus Responsabilidades

Categoría	Función
De cara general al proyecto	Es el responsable de la creación del acta de constitución del proyecto y el presupuesto (si aplica).
	Verificar que los recursos necesarios sean asignados para la ejecución del proyecto.

Categoría	Función
	<p>Facilitar las sesiones del equipo Scrum para llevar a cabo las diferentes reuniones de trabajo, de acuerdo con el marco Scrum</p> <p>Identificar los riesgos asociados a cada sprint del proyecto.</p> <p>Gestionar con los interesados los obstáculos, impedimentos, retrasos o desperdicios que se puedan presentar en la gestión del equipo Scrum, evitando las interferencias generadas dentro de la ejecución de las tareas.</p> <p>Mantener la comunicación fluida y periódica con el líder Scrum de Scrums, el Scrum máster del programa y el líder de integración (si aplica).</p> <p>Mantener actualizadas los instrumentos de seguimiento como la Release Burndown chart y la Sprint Burndown Chart.</p> <p>Firmar de conformidad los artefactos de cierre del proyecto definidos.</p>
De cara al Product Owner	<p>Ayudar a mejorar el trabajo del Product Owner: su disponibilidad hacia el equipo, la gestión del Product Backlog, el plan de releases.</p> <p>Ayudar a observar características de un buen Product Backlog:</p> <ul style="list-style-type: none"> - Priorizado - Emergente con base en los requisitos y deseos del cliente. - Manejable - PBI cumplen con INVEST (Independiente, Negociable, Valioso, Estimable, pequeño y testeable) - Deuda técnica y cómo evitarla <p>Asegurar que la herramienta automatizada para el Product Backlog es de valor para el proyecto y no una carga adicional.</p> <p>Ayudar a difundir información mediante gráficos o esquemas</p> <p>Facilitar al equipo Scrum el mantenimiento del Product Backlog y preparar la Sprint Planning</p> <p>Asegurar la comprensión de los PBI por parte del equipo y la estimación de su esfuerzo.</p> <p>Actualizar y dar a conocer el plan de release. Mantener gráficos Burndown Chart Release actualizados.</p> <p>Gestionar los cambios e impedimentos.</p>
De cara al equipo Scrum	<p>Fomentar la colaboración entre los miembros del equipo.</p> <p>Ayudar al equipo a mantenerse en estado de flujo, esto es:</p> <ul style="list-style-type: none"> - Objetivos claros y metas alcanzables alineadas a las capacidades del equipo. - Mantenerse dentro de los valores Scrum (Guía 2020) Compromiso, respeto, foco, concentración y coraje. - Feedback directo e inmediato. - Equilibrio entre la habilidad y el esfuerzo para conseguir logros. <p>La actividad es gratificante, no se necesita estar motivando al equipo.</p>

Categoría	Función
	<p>Observar, Orientar, Decidir y Actuar sobre los siguientes aspectos:</p> <ul style="list-style-type: none"> • El Sprint Backlog refleja lo que el equipo está haciendo, no hay tareas no reveladas ni más grandes que un día de trabajo. • El equipo está conformado por las personas con las habilidades suficientes para el incremento de producto. • Los artefactos y herramientas adicionales (tablero de tareas, lista de impedimentos) son visibles para el equipo. • El equipo se autogestiona, se ofrecen voluntariamente a realizar las tareas. • El equipo tiene claro el costo de la deuda técnica y aplica buenas prácticas para evitarla o resolverla.
De cara a las prácticas de ingeniería	<p>Observar, orientar decidir y actuar sobre los siguientes aspectos:</p> <ul style="list-style-type: none"> • El equipo detecta si hay fallo de regresión (se ha roto previamente alguna funcionalidad dada por cumplida). • Hay pruebas funcionales que validen el valor del incremento. • Se realizan pruebas del sistema (o de extremo a extremo E2E) y unitarias en el mismo lenguaje en el que se desarrolla y se descubre la zona gris. • Se cuenta con todas las condiciones técnicas para la integración de tests. • se cuenta con una herramienta de planificación ágil que todo el equipo conoce y donde pueden visualizar los PBI y avance del plan de release. • El equipo conoce y aplica las bondades de la refactorización. <p>Es deseable que para la verificación de la DOD se cuente con pruebas automatizadas y refactorización. (recomendable las técnicas de Test Driven Development -TDD).</p> <p>Se sugiere que los developers practiquen "pair programming" para asegurar el mantenimiento de código.</p>
De cara a la organización	<p>Observar, orientar decidir y actuar sobre los siguientes aspectos:</p> <p>Comunicación adecuada ente los miembros del equipo y los Scrum de Scrums que colaboran para lograr un mismo producto.</p> <p>Los diferentes Scrum Masters se reúnen para gestionar los impedimentos.</p> <p>Los impedimentos organizacionales se identifican y se hacen visibles en la Coordinación CenIT, Subdirección de Innovación y Proyectos, Subdirección de Soluciones y Desarrollo y DGIT o quiénes hagan sus veces.</p> <p>Se puede cuantificar el costo de retraso, pérdida de no salir a producción a tiempo, pérdida de calidad y de oportunidad para los clientes.</p>

Categoría	Función
	Generar capacitaciones sobre la gestión ágil de proyectos.
	Planear nuevas implementaciones de Scrum dentro de la organización.
	Eliminar barreras entre los interesados y los equipos Scrum.

Fuente: Elaboración propia

6.6 Product Owner o dueño de Producto

Rol ejercido por parte de un colaborador de la Dirección de Gestión dueña del producto que comprende lo que quiere y necesita. Este actor debe tener amplia experiencia sobre negocio y conocimientos técnicos, excelentes habilidades de comunicación, negociación, proactividad, pragmatismo y orientación al logro, y estar comprometido con el desarrollo del proyecto.

Es el responsable de maximizar el valor del producto resultante del trabajo del Scrum Team. Además:

- Desarrolla y comunica el objetivo del producto.
- Crea y comunica los elementos del Product Backlog.
- Ordena los elementos del Product Backlog.
- Se asegura de que el Product Backlog sea transparente y visible.

El Product Owner representa la voz del cliente, se encarga de articular los requisitos del cliente con los técnicos y es quien decide qué entra en el Product Backlog.

Se espera que este rol cuente con disponibilidad completa de tiempo para asumir sus responsabilidades y establecer la comunicación necesaria con los interesados y los developers.

Tabla 3 – Product Owner en las Etapas de la Gestión Ágil

Etapa de proyecto	Función
Inicio de proyecto	Participar en la creación del acta de constitución del proyecto y el presupuesto (si aplica), identificando los integrantes del área funcional que participarán de la Review del producto.
	Participar en la estructuración del equipo Scrum.
	Participar en la Inception del proyecto.
	Determinar riesgos relacionados con el negocio, el equipo, la tecnología, los costos y el plazo de ejecución.
Ejecución de los Sprints	Desarrollar y comunicar explícitamente el Objetivo de Producto.
	Crear y comunicar claramente los elementos (PBI) del Product Backlog
	Liderar la construcción y priorización del backlog de producto.
	Asegurar que los PBI de mayor valor para el cliente se resuelvan primero.
	Definir las épicas y sus features.
	Anticipar preguntas que el equipo pueda hacer sobre los PBI para agilizar su comprensión y priorización.
	Asegurar que el equipo comprende totalmente cada PBI.
Comprender la velocidad del equipo. Con la ayuda del scrum master, saber la velocidad del equipo y si hay margen para mejorarla.	

Etapa de proyecto	Función
	Ayudar en la estimación del esfuerzo a realizar en cada Sprint, cuidando no sobrecargar al equipo, para hacer que el proyecto sea sostenible.
	Definir junto con el equipo de trabajo los criterios de aceptación para cada uno de los ítems del Product Backlog y la Definition of Done (Definición de hecho) del incremento completo.
	Revisar todas las historias de los usuarios para asegurarse de que tengan toda la información: descripción, criterios de aceptación, enlaces al padre correcto (característica), puntos de la historia, enlaces a las historias de los usuarios conectados.
	Revisar la Definition de Ready (DoR) y que las historias de usuario que seleccionó para el próximo sprint están incluidas.
	Asegurarse de que el Product Backlog sea transparente, visible y se entienda.
	Gestionar el plan de lanzamiento de los incrementos del producto.
	Definir la agenda de la Sprint Review, apoyar la presentación y entrega del incremento y validar con los interesados lo que esperan del siguiente.
	Ordenar los elementos del Product Backlog y mantenerlo actualizado después de las Sprint Review.
	Estar disponible para interactuar con el equipo Scrum cada vez que se requiera.
	Participar de las reuniones Daily siempre que pueda, especialmente cuando hay alertas, impedimentos o riesgo de materialización de riesgos.
	Gestionar los cambios que se requieran en el Product Backlog con base en conocimiento y conversaciones con los interesados y equipo. Decir "no" a lo que no agregue valor.
	Tomar decisiones cuando se presentan problemas para avanzar en un Sprint.
	Gestionar de manera realista las expectativas de las entregas con base en la experiencia y pronósticos basados en la data. Para ello usa las gráficas de quemado de las historias por semana (puntos de historia).
	Para lograr una mejor priorización en los PBI del Product Backlog y garantizar el valor que el cliente desea.
	Mantenerse muy bien informado sobre la data y conocimiento empírico del proyecto.
	Mantener la comunicación fluida y periódica con los interesados y el equipo.
	Conocer muy bien las necesidades del cliente y saber identificar qué es prioritario con base en lo que le aporta más valor.
	Priorizar aquellos PBI que aportan más valor al cliente y dejar para el final aquellos que no.
	Apoyarse en el equipo para realizar la priorización de los PBI y su refinamiento.
Cierre	Informar el cierre del proyecto con la entrega del producto completado
	Firmar de conformidad los formatos de cierre del proyecto definidos.

Fuente: Elaboración propia

6.7 Ingeniero de Requerimientos

Toda vez que el Product Owner suele ser una persona en un cargo administrativo de nivel directivo o asesor y que no necesariamente cuenta con la formación específica para la comprensión formal de los requerimientos (algunos ejemplos: refinamiento del Product Backlog y documentación de historias de usuario), cuenta con el apoyo del ingeniero de Requerimientos. Este apoyo es una medida transitoria mientras la Entidad logra dejar la capacidad suficiente instalada en los P.O para el pleno ejercicio de

sus responsabilidades, entre ellas las que tienen que ver con la documentación de requerimientos y el refinamiento del Product Backlog.

Ahora bien, la guía del marco señala que el Product Owner puede realizar directamente sus responsabilidades o delegarlas, pero independientemente de eso, sigue siendo responsable del trabajo que se realice, es decir, no se trata de que el Ingeniero de Requerimientos asuma las responsabilidades del Product Owner, sino, más bien es un importante colaborador para garantizar que los requerimientos sean debidamente comprendidos y estén listos cada sprint para lograr el incremento. En ese sentido, elabora el levantamiento de historias de usuario, siguiendo las actividades y haciendo uso de los formatos señalados en el “PR-IIT-0498 Gestión ágil de proyectos de desarrollo” además de:

Tabla 4 – Ingeniero de Requerimientos y sus Responsabilidades

Categoría	Función
Comprender las Necesidades del Cliente	Colaborar con el Product Owner para comprender las necesidades y expectativas del cliente. Realizar análisis detallados de los requisitos para garantizar una comprensión completa.
Facilitar la Comunicación	Actuar como puente entre el equipo de desarrollo y el Product Owner. Asegurarse de que las comunicaciones sobre requisitos sean claras y efectivas. Llevar a cabo las conversaciones necesarias con el Product Owner y los colaboradores del área funcional para comprender sus necesidades y traducirlas en el formato Historias de Usuario que es el acuñado por la Entidad u otro si llega a haber algún cambio. El formato en este caso no es lo importante, sino, la comprensión de las necesidades del cliente y su registro como elementos del Product Backlog.
Descomposición de Historias de Usuario	Colaborar con el Product Owner en la descomposición de historias de usuario para detallarlas y dividir las de manera que sean más pequeñas y manejables. Ayudar en la elaboración de criterios de aceptación claros para las historias de usuario. Conversar con los developers para que al transparentar las historias de usuario ellos definan las tareas técnicas.
Documentación de Requisitos	Crear y mantener documentación detallada de los requisitos del producto. Asegurarse de que la documentación esté actualizada y disponible para todo el equipo.
Validación y Verificación	Participar en la validación y verificación de las historias de usuario para garantizar que cumplan con los criterios de aceptación. Colaborar en la identificación y resolución de discrepancias o cambios en los requisitos.
Priorización	Colaborar con el Product Owner en la priorización del backlog del producto basándose en el valor para el cliente y la complejidad técnica.
Participación en Sprint Planning	Asistir y contribuir en las reuniones de planificación de sprint para garantizar que los requisitos sean comprendidos por el equipo de desarrollo.

Categoría	Función
Retroalimentación Continua	Proporcionar retroalimentación continua al Product Owner sobre la claridad y la viabilidad de los requisitos.
Refinamiento del Product Backlog	<p>Participar activamente en sesiones de refinamiento para mejorar la calidad de las historias de usuario antes de que se incluyan en el sprint.</p> <p>Mantener el proceso de refinamiento de los PBI antes y durante cada sprint. Adelantar el refinamiento con los developers “transparentando” los requerimientos de manera que se logre llegar a las tareas técnicas para comprender el esfuerzo, complejidad, riesgos e incertidumbre y así los PBI sean lo suficientemente detallados, pequeños y estimables como para poder desarrollarlos en un Sprint.</p> <p>Asegurar que a cada Sprint Planning se llegue con el Product Backlog refinado. En la práctica debe ir como mínimo 1 Sprint adelantado en refinamiento. Un proceso continuo que debe tener apropiado.</p> <p>Garantizar la creación adecuada de los elementos del Product Backlog en la herramienta de gestión.</p>
Desarrollo de Relaciones de Trabajo	Trabajar en estrecha colaboración con el Scrum Master, el Product Owner y el equipo para construir relaciones efectivas y promover un ambiente de trabajo colaborativo.

Fuente: Elaboración propia

6.8 Developers o desarrolladores

Son las personas del Equipo Scrum que se comprometen a desarrollar los incrementos funcionales del producto en cada Sprint. Se usa la palabra “Developers” a manera de integrar cualquier profesional independientemente de su campo de acción que participa interdisciplinariamente en el desarrollo del incremento. Para que puedan desarrollar los valores Scrum, es altamente deseable que tengan disponibilidad óptima del 100% para el desarrollo de cada producto.

Los Developers se encargan de:

- Crear un plan para el Sprint, el Sprint Backlog.
- Inculcar calidad al adherirse a la DoD.
- Adaptar su plan cada día hacia el Objetivo del sprint.
- Son responsables de realizar el trabajo bien hecho, eliminando la deuda técnica.

Tabla 5 – Developers y sus responsabilidades

Categoría	Función
Sprint Planning	Crear el plan de desarrollo para el Sprint, es decir, el Sprint Backlog.
	Mantener comunicación fluida con el Product Owner para la comprensión completa de los PBI del Product Backlog.
	Definir el objetivo del Sprint.
	Detallar los PBI a través de tareas técnicas que se desarrollan en un Sprint específico.
	Estimar el esfuerzo (Story points o puntos de historia a alcanzar en cada Sprint) lo cual contribuye a medir la velocidad y que el equipo logre un ritmo.

Categoría	Función
	Apoyar al Product Owner en el refinamiento del Product Backlog.
Ejecución del Sprint	Ejecutar el desarrollo de las historias de usuario y tareas técnicas del Sprint.
	Asegurarse de la ejecución de las pruebas de sistema, unitarias y todas aquellas requeridas de acuerdo con la naturaleza y exigencia del proyecto.
	Enfocar su trabajo en hacer las cosas bien, eliminando siempre la deuda técnica.
	Apoyarse, acudir y confiar en el Scrum Master para la implementación de buenas prácticas Scrum.
	Autogestionarse y autoorganizarse para el desarrollo del trabajo técnico.
	Prepararse para las reuniones Daily, comunicando los avances concretos y los obstáculos e impedimentos, y adaptando el plan de trabajo hacia el objetivo del Sprint.
	Mantener actualizados los artefactos de acuerdo con la fase de proyecto y Sprint programados.
	Mantener actualizada la lista de tareas programadas dentro del sprint.
	Crear los entregables correspondientes a los PBI terminados. (Incremento + documentación)
	Participar activamente y prepararse para los eventos Scrum: Sprint Planning, Daily, Sprint Review y Sprint Retrospective.
	Presentar los entregables finalizados al Product Owner para su aprobación.
Gestión de riesgos y control de cambios	Identificar y comunicar los riesgos y cambios asociados a cada sprint del proyecto y procurar su solución en las reuniones técnicas o comunicarlo al Scrum Master cuando no es posible mitigarlos.
	Ejecutar acciones de mitigación de riesgos y de gestión de control de cambios con apoyo del Scrum Master y del Product Owner cuando sea requerido.
Medida de felicidad	Trabajar en equipo con cada uno de los miembros del equipo, el Product Owner y el Scrum Master.
	Interiorizar y practicar los valores Scrum en el trabajo diario.
	Identificar oportunidades de mejora durante los sprint en ejecución para ser tenidos en cuenta durante los siguientes Sprint programados.
	Incluir en cada Sprint las acciones de mejora del proceso del equipo de trabajo.

Fuente: Elaboración propia

6.9 Responsabilidades transversales de los Developers

Para la entidad, la integración funcional y técnica es responsabilidad del equipo de Developers, dado que es a través de esto que se logra asegurar buena parte de la calidad y estabilización anticipada del producto funcionando. Por tanto, los developers en cada Sprint, para cada Incremento deben asegurar y garantizar que hay una adecuada integración funcional y técnica de acuerdo con el compromiso que se ha adquirido. La manera como se aseguran de realizar las tareas necesarias es dejarlas explícitamente como PBI del Product Backlog y reflejadas en el Sprint Backlog.

A continuación, se explican a modo de funciones las responsabilidades mencionadas con el propósito de que queden claras y el equipo de manera autogestionada vele por su implementación:

6.9.1 Asegurar la integración funcional

La integración funcional juega un papel crucial en garantizar que las diversas funciones de un sistema trabajen de manera conjunta y cumplan con los objetivos del negocio. Esto implica una combinación de habilidades técnicas y habilidades de gestión de proyectos, así como la capacidad de colaborar efectivamente con equipos multidisciplinarios.

Tabla 6 – Integración funcional

Categoría	Función
Diseño Funcional	Asegurar la coordinación y unificación de diferentes funciones o componentes dentro de un sistema para garantizar que trabajen de manera coherente y cumplan con los requisitos del negocio. Desarrollar y documentar soluciones que satisfacen los requisitos del negocio. Asegurar la coherencia y la integración de las diferentes funciones dentro del diseño general del sistema.
Gestión de Interfaces	Trabajar en la definición y gestión de interfaces entre los diferentes componentes del sistema. Asegurar que las interfaces sean claras y que la comunicación entre las funciones sea eficiente.
Pruebas de Integración Funcional	Diseñar y ejecutar casos de prueba que evalúan la integración efectiva de las diversas funciones del sistema. Identificar y resolver problemas de integración funcional.
Gestión de Cambios	Participar en la gestión de cambios, asegurándose de que cualquier modificación en los requisitos se refleje adecuadamente en el diseño funcional y en la implementación. Evaluar el impacto de los cambios propuestos en las funciones y en la integración global del sistema.
Documentación	Crear y mantener documentación detallada sobre los procesos de integración y los requisitos funcionales. Asegurar que la documentación esté actualizada y disponible para todas las partes interesadas.
Colaboración Interfuncional	Trabajar estrechamente con equipos funcionales diversos, como desarrollo, calidad, operaciones, etc., para asegurar una comprensión mutua y una colaboración efectiva. Facilitar la resolución de conflictos o desafíos que puedan surgir entre diferentes funciones.
Soporte a Usuarios y Capacitación	Proporcionar soporte a los usuarios finales en relación con el uso del sistema. Colaborar en la creación de material de capacitación y realiza sesiones de capacitación según sea necesario.
Optimización Continua	Colaborar en iniciativas de mejora continua para optimizar la eficiencia y la eficacia de las funciones y la integración global del sistema.

Fuente: Elaboración propia

6.9.2 Garantizar la integración técnica

Se refiere a las responsabilidades de los developers para lograr la integración de componentes y sistemas desde una perspectiva técnica. Es fundamental para garantizar una integración exitosa y un funcionamiento eficiente del sistema o proyecto.

Tabla 7 – Integración técnica

Categoría	Funcional
Diseño de la Arquitectura de Integración	Definición de la Estructura Técnica: Diseñar la arquitectura técnica que permitirá la integración eficiente de los diferentes componentes o sistemas. Esto puede incluir la selección de tecnologías, protocolos de comunicación y enfoques de integración. Esto debe hacerse en completa armonización con la Arquitectura de TI de la Entidad.
Desarrollo de Interfaces	Desarrollo de Interfaces Técnicas: Crear interfaces técnicas que permitan la comunicación y la interoperabilidad entre los diversos componentes del sistema. Esto puede incluir el diseño de APIs, la definición de formatos de datos y otros aspectos técnicos relacionados.
Pruebas de Integración Técnica	Desarrollo de Escenarios de Prueba Técnica: Diseñar y ejecutar pruebas técnicas para evaluar la interoperabilidad y el funcionamiento correcto de las interfaces entre los componentes. Esto puede incluir pruebas unitarias, pruebas de integración y pruebas de rendimiento. Identificación y Resolución de Problemas Técnicos: Identificar y resolver problemas técnicos que surjan durante el proceso de integración, como problemas de compatibilidad, errores de código o problemas de rendimiento.
Coordinación con Equipos de Desarrollo	Colaboración con Desarrolladores: Trabajar estrechamente con los equipos de desarrollo responsables de los diferentes componentes, facilitando la comunicación y asegurando una comprensión clara de los requisitos de integración. Gestión de Dependencias Técnicas: Gestionar y coordinar las dependencias técnicas entre los diversos módulos o sistemas para evitar conflictos y asegurar una integración fluida.
Documentación Técnica	Creación y Mantenimiento de Documentación Técnica: Documentar los detalles técnicos del proceso de integración, incluidas las interfaces, los protocolos y cualquier consideración técnica relevante.
Gestión de Configuración	Control de Versiones y Configuración: Participar en la gestión de la configuración, asegurándose de que las versiones de los componentes estén controladas y que los cambios se realicen de manera controlada y documentada.
Resolución de Problemas Técnicos	Investigación y Resolución de Problemas Técnicos: Abordar y resolver problemas técnicos que puedan surgir durante la integración, utilizando habilidades de diagnóstico y resolución de problemas.
Optimización del Rendimiento	Monitoreo y Mejora del Rendimiento: Supervisar el rendimiento del sistema integrado y colaborar en la optimización para garantizar la eficiencia y la capacidad de respuesta.
Gestión de Cambios Técnicos	Evaluación de Impacto de Cambios Técnicos: Evaluar el impacto de los cambios propuestos en la arquitectura técnica e interfaces, y colaborar con los equipos relevantes para implementar cambios de manera efectiva.

Fuente: Elaboración propia

6.9.3 Colaborador técnico

El colaborador técnico acompaña a los developers en el desarrollo del producto, ofreciendo su apoyo y asesoramiento, para ello colabora activamente removiendo impedimentos técnicos y contribuye con ideas eficientes para que los desarrolladores logren avanzar más ágilmente en el incremento. Esta responsabilidad principal se puede organizar en las siguientes categorías:

Tabla 8 – Ingeniero de requerimientos y sus responsabilidades

Categoría	Función
Capacitación y Desarrollo	Proporcionar entrenamiento y orientación técnica a los miembros del equipo. Ayudar en el desarrollo de habilidades técnicas y la adopción de mejores prácticas. Facilita el acceso al conocimiento digital de la Entidad para que los desarrolladores no dependan de terceros, sino que desarrollen sus propias capacidades.
Resolución de Problemas Técnicos	Colaborar con el equipo en la resolución de problemas técnicos y desafíos. Actuar como recurso técnico para abordar preguntas y problemas específicos.
Revisión de Código	Participar en revisiones de código para garantizar la calidad y la coherencia del código, y ayudar a eliminar la deuda técnica. Compartir conocimientos sobre buenas prácticas de codificación.
Colaboración en el Diseño	Trabajar junto con el equipo en el diseño técnico de soluciones. Facilitar sesiones de diseño técnico y aportar ideas.
Automatización de Pruebas	Ayudar en la creación y mantenimiento de pruebas automatizadas. Colaborar en la implementación de estrategias de pruebas efectivas.
Optimización del Rendimiento	Identificar oportunidades para mejorar el rendimiento del código y colaborar en su implementación. Realizar análisis de rendimiento y proporcionar recomendaciones.
Facilitación de Sesiones Técnicas	Facilitar sesiones técnicas, como reuniones de planificación técnica y sesiones de retrospectiva técnica. Fomentar la comunicación y la colaboración en cuestiones técnicas.
Documentación Técnica	Contribuir a la creación y mantenimiento de documentación técnica. Asegurarse de que la documentación sea clara y accesible para todo el equipo.
Gestión de Dependencias Técnicas	Coordinar con otros equipos y áreas para gestionar dependencias técnicas. Minimizar los bloqueos relacionados con dependencias técnicas.
Mentoría y Apoyo Personal	Ofrecer mentoría y apoyo personal a los desarrolladores, especialmente a aquellos que están en etapas iniciales de sus carreras. Fomentar un ambiente de trabajo positivo y colaborativo.

Fuente: Elaboración propia

6.10 Stakeholders o Interesados

Grupo de funcionarios de las distintas áreas de la DIAN y/o externos, que influyen en el proyecto a lo largo de su desarrollo. Son *Interesados*:

- a) Director de Gestión de Innovación y Tecnología.
- b) Usuario experto funcional.
- c) El patrocinador es la persona o la organización que provee recursos y apoyo estratégico para la ejecución del proyecto. El patrocinador es a quien todos le deben rendir cuentas al final. (Director o subdirector de área funcional).
- d) Proveedores.

Las siguientes son las responsabilidades de los interesados:

- Participar en la inceptions del proyecto/producto.
- Participar en la creación del acta de constitución del proyecto y el presupuesto (si aplica).
- Participar en la elaboración de las épicas e historias de usuario del producto.
- Mantener una comunicación fluida con el Product Owner e ingeniero de requerimientos.
- Validar el direccionamiento de avance sobre los proyectos frente a la visión definida y la alineación estratégica.
- Comprometerse activamente a participar en las reuniones que sea convocado.
- Participar y dar retroalimentación del producto en los sprint Review.
- Firmar de conformidad los incrementos del producto y el producto completado.

6.11 Agile Coach (Entrenador Ágil)

Desempeña un papel clave en guiar a la organización a través de la transformación ágil, asegurándose de que los equipos adopten las prácticas ágiles de manera efectiva y se alcancen los objetivos estratégicos deseados.

Tabla 9 – Agile Coach y sus responsabilidades

Categoría	Función
Facilitador del Cambio	Ayuda a la organización a adoptar y adaptarse a los principios ágiles. Facilita la transición de procesos tradicionales a ágiles.
Formación y Capacitación	Imparte formación a equipos y líderes sobre marcos ágiles. Ayuda a construir conocimientos y habilidades en la organización.
Asesoramiento a Equipos	Trabaja directamente con equipos ágiles para mejorar su rendimiento. Proporciona orientación sobre buenas prácticas ágiles.
Mentoría	Ofrece orientación y apoyo individual a Scrum Masters, Product Owners y otros roles ágiles. Fomenta el desarrollo profesional de los miembros del equipo.
Identificación y Resolución de Problemas	Identifica obstáculos y problemas en la implementación ágil y trabaja en soluciones. Facilita la resolución de conflictos y promueve la mejora continua.
Fomenta la Colaboración	Promueve la colaboración efectiva entre equipos, así como entre los equipos y otras partes interesadas. Ayuda a construir una cultura de transparencia y comunicación abierta.

Categoría	Función
Medición y Evaluación	Define métricas y KPI (Key Performance Indicators) para evaluar el progreso y el rendimiento ágil. Proporciona retroalimentación basada en datos para la mejora continua.
Liderazgo Transformacional	Actúa como líder de cambio y promotor de la cultura ágil en toda la organización. Trabaja con la alta dirección para alinear la estrategia empresarial con los principios ágiles.

Fuente: Elaboración propia

7. SCRUM ESTÁNDAR - GESTIÓN DE PROYECTOS ÁGILES

7.1 FASE I: ALISTAMIENTO

7.1.1 Activación de nuevos Scrum Teams (células)

Desde la dirección de gestión de Innovación y Tecnología o desde la subdirección de Innovación y Proyectos o desde la Coordinación Centro de Gestión de Proyectos de Innovación y Tecnología (CENIT) se validan las células que se pueden activar de acuerdo con las necesidades estratégicas de la Entidad.

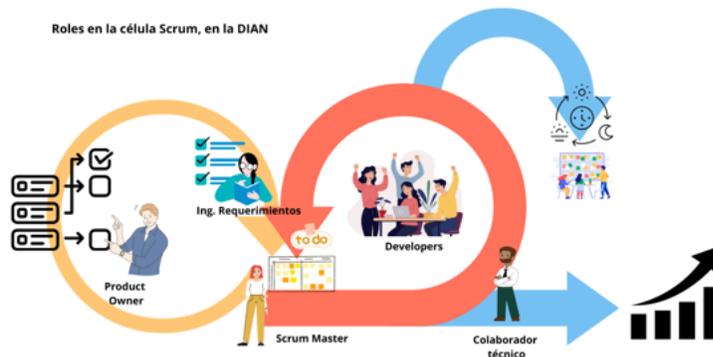
Una vez identificadas y agrupadas estas necesidades, se realizan las siguientes actividades:

1. Conformar el Scrum Team, lo cual implica: por parte de la DIGIT y específicamente por el CenIT, la definición del Scrum Máster y asignación a los developers.
2. Adelantar una primera reunión de alto nivel con la dirección de gestión o el área funcional con el fin de entender sus necesidades prioritarias, así como explicar el marco Scrum y solicitar la asignación del Product Owner.
3. El área funcional se compromete con fechas para realizar su proceso de apropiación de transformación ágil.
4. El Scrum Máster sensibiliza al Product Owner sobre su rol, responsabilidades e impacto en el éxito del proyecto y este entra a completar el Scrum Team.
5. El Product Owner define en su dirección de gestión el equipo funcional que acompañará el proyecto.
6. El Scrum Máster realiza ejercicios de formación y buenas prácticas sobre Scrum al equipo asignado, de acuerdo con las necesidades que se identifican.

7.1.2 Conformación del Scrum Team

Para Scrum, los roles básicos son Scrum Master, Product Owner y Developers. Para la Entidad, dado su propio contexto, se requiere, además de otros roles, tal como se describió en el apartado R. En ese sentido, las células se conforman con los roles requeridos dependiendo del producto a desarrollar.

Ilustración 3 - Roles en una célula Scrum



Fuente: Elaboración propia

7.1.3 Apropiación del área funcional sobre la transformación ágil

La gestión ágil de proyectos incluye varios detalles que se explican adelante. Para lograr que las áreas funcionales sean elementos propositivos en la transformación ágil requieren pasar por un cambio cultural que los lleve a participar de una manera diferente y permanente.

En ese sentido, la DIGIT prepara la siguiente estrategia de apropiación que estará renovándose como mínimo cada año o antes si la guía Scrum tiene actualizaciones o las buenas prácticas de agilidad así lo proponen:

1. Ver los videos sobre transformación ágil.
2. Presentar los quizzes de comprensión de conocimiento.
3. El Scrum Team revisará los resultados de la sensibilización, en caso de que no hayan sido aprobados programará una sesión de coaching para aclarar dudas al área funcional.
4. El área deberá volver a ver los videos y presentar los quizzes hasta lograr la apropiación del conocimiento.

Este ejercicio de apropiación frecuente servirá para que los interesados se vuelvan expertos y reflexionen sobre cosas que están haciendo de manera correcta y aquellas que no. Algo que puede pasar también es que el P.O. o su equipo de trabajo no estén haciendo algo correcto y es el momento de reflexionar y de mejorar.

La Apropiación entonces va más allá del momento en el que el área se aproxima a la transformación ágil por primera vez, sino que se trata de un proceso de mejora continua.

7.2 FASE II: GESTIÓN DEL PROYECTO ÁGIL

Los siguientes apartados presentan las etapas en las que se desarrolla un proyecto Scrum en la DIAN:

7.2.1 Etapa de Inicio (Agile inception)

La inception o etapa de inicio es fundamental porque permite definir el rumbo del proyecto, en aspectos claves que si no están claros sería muy complejo avanzar hacia el primer Sprint.

La Agile Inception es un proceso de varias sesiones de trabajo que dependiendo de la complejidad del proyecto puede tomar 5 o 6 sesiones de 2 horas, el equipo no debe limitar las sesiones a un determinado número de horas, sino procurar ser lo más inteligente posible obteniendo el mayor valor en cada encuentro para reducir el número de sesiones, no obstante, si requieren usar las 5 o 6 sesiones deberán hacerlo, porque prima la calidad del trabajo y el entendimiento de las necesidades antes de ser rápidos.

Por ello, se establecen los encuentros entre el área funcional y el Scrum Team para determinar las necesidades que han de ser satisfechas a través de tecnología. En estos encuentros se espera obtener el objetivo, las personas a las que se quiere impactar y cómo se va a lograr esto a través del producto, la Definition of Done y la cantidad de Sprints.

La activación de la célula Scrum es un paso fundamental para llegar al área funcional, avanzar en el proceso de inicio (inception), tener un entendimiento conjunto y más cercano de las necesidades del área, por ello hace parte del alistamiento.

Los siguientes son los pasos que debe garantizarse a nivel general en la etapa de inicio de cualquier proyecto:

1. Ejecutar la agile inception del proyecto para determinar el objetivo del producto y obtener la primera versión del mapa de PBI. El Scrum Máster realiza las sesiones de inception necesarias para entender claramente el objetivo del proyecto. Estas sesiones se hacen con todo el Scrum Team y el área funcional. La primera reunión de inception es la que marca el inicio del proyecto y por tanto, debe tenerse en cuenta para la firma del acta de inicio.
2. Constituir oficialmente el proyecto. El Scrum Master elabora el acta de inicio del proyecto (FT-IIT-2010 "Acta Constitución Proyecto) y junto con el Product Owner documentan la matriz de riesgos en el Anexo 1. Matriz de riesgos proyectos tecnológicos.
3. Creación y administración del Product Backlog. Se registra en la herramienta de gestión el Product Backlog y sus PBI. Se da inicio al proceso continuo de refinamiento del Product Backlog.
4. Posteriormente, se da inicio al primer Sprint y sucesivos con el Sprint Planning, es decir, se da inicio a la Etapa de

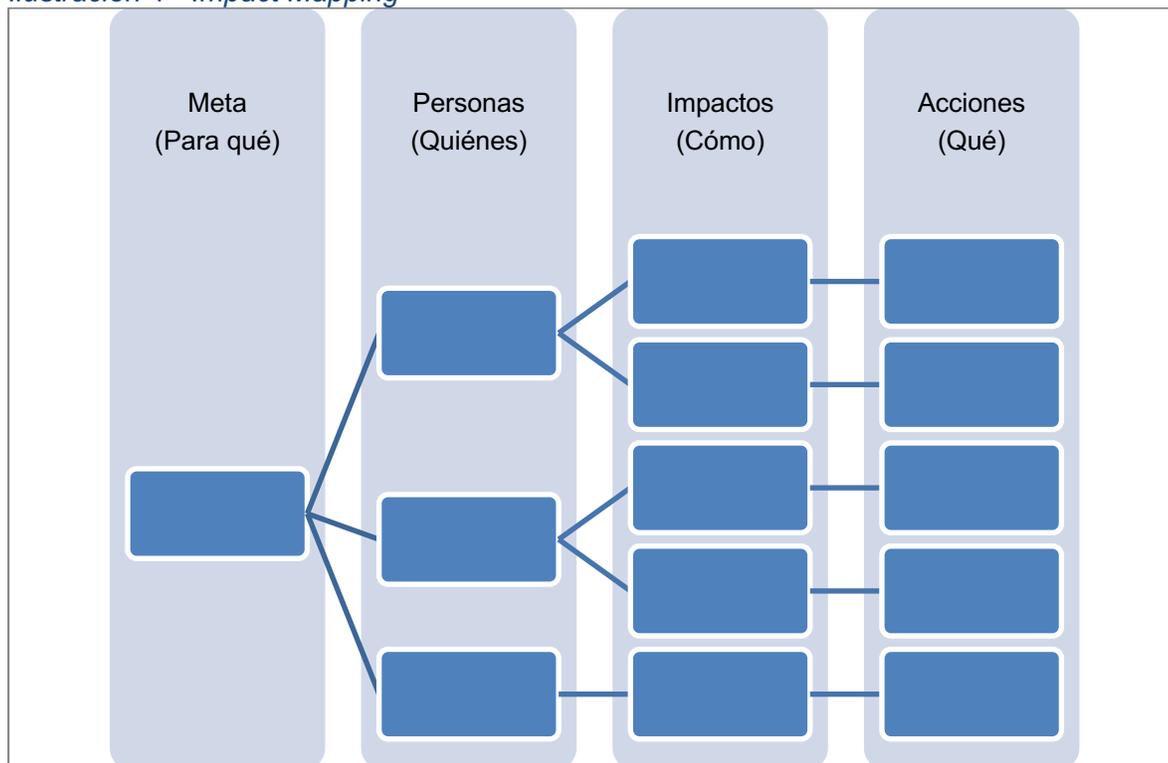
7.2.1.1 Ejecutar la inception usando un impact mapping

El Impact Mapping es una herramienta estratégica que se utiliza para visualizar y comunicar cómo un proyecto o iniciativa puede contribuir a los objetivos estratégicos de una organización. Sirve como un medio para alinear a todas las partes interesadas, desde los interesados en el producto hasta los developers, identificando los resultados deseados y mapeando las actividades necesarias para lograrlos.

A través de este proceso, se fomenta la colaboración y se mejora la comprensión de los impactos potenciales, permitiendo tomar decisiones más informadas sobre el alcance y las prioridades del proyecto.

El Impact Mapping no solo ayuda a evitar el desarrollo de características innecesarias, sino que también facilita la adaptabilidad a medida que evolucionan los requisitos y las circunstancias, contribuyendo así a la entrega de soluciones más efectivas, alineadas con los objetivos estratégicos de la organización y que dan mayor valor al área funcional.

Ilustración 4 - Impact Mapping



Fuente: Elaboración propia

7.2.1.2 Constituir oficialmente el proyecto

Durante esta fase se deben diligenciar dos formatos: Acta de constitución FT-IIT-2010 y Anexo 1. Matriz de riesgos proyectos tecnológicos.

Cuando se tiene el primer mapa de PBI, se ha completado la fase inicio y se procede a diligenciar el “Formato FT-IIT-2010 “Acta Constitución Proyecto”, el cual contiene los siguientes campos:

- **Datos del proyecto:**

- Número de solicitud
- Id de proyecto
- Plazo de ejecución del proyecto: Fecha de inicio y finalización
- Datos del Scrum Máster y del Product Owner

- **Alineación estratégica:**

- Programa
- Línea estratégica
- Objetivo estratégico.

- **Visión del proyecto**

- Objetivo general. (Beneficio que se logrará al desarrollar el producto)
- Objetivo de Producto. (Alcance) Se sugiere que responda a:

Para (roles específicos)

Quiénes (necesidad/deseo/interés)

El (producto)

Es un (solución concreta)

Que (funcionalidades)

Diferente (la diferencia con otras soluciones) *no obligatorio

- **Backlog del proyecto**

- Product backlog ítems (PBI) Enlistar las funcionalidades a nivel general que se espera se cumplan en el producto completado. (Vienen del mapa de PBI)
- Plazo de ejecución

- **Supuestos y restricciones del proyecto (si aplica).**

- **Riesgos del proyecto**

Se registran los riesgos generales identificados por los Interesados, Product Owner y Scrum Master. Se complementan y detallan en el formato Anexo 1. Matriz de riesgos proyectos tecnológicos.

- **Recursos físicos (si aplica).**

- **Recursos financieros (si aplica).**

Una vez listo el mapa de PBI se crea el Product Backlog en la herramienta de gestión definida por la Entidad

7.2.1.3 Creación y administración del Product Backlog

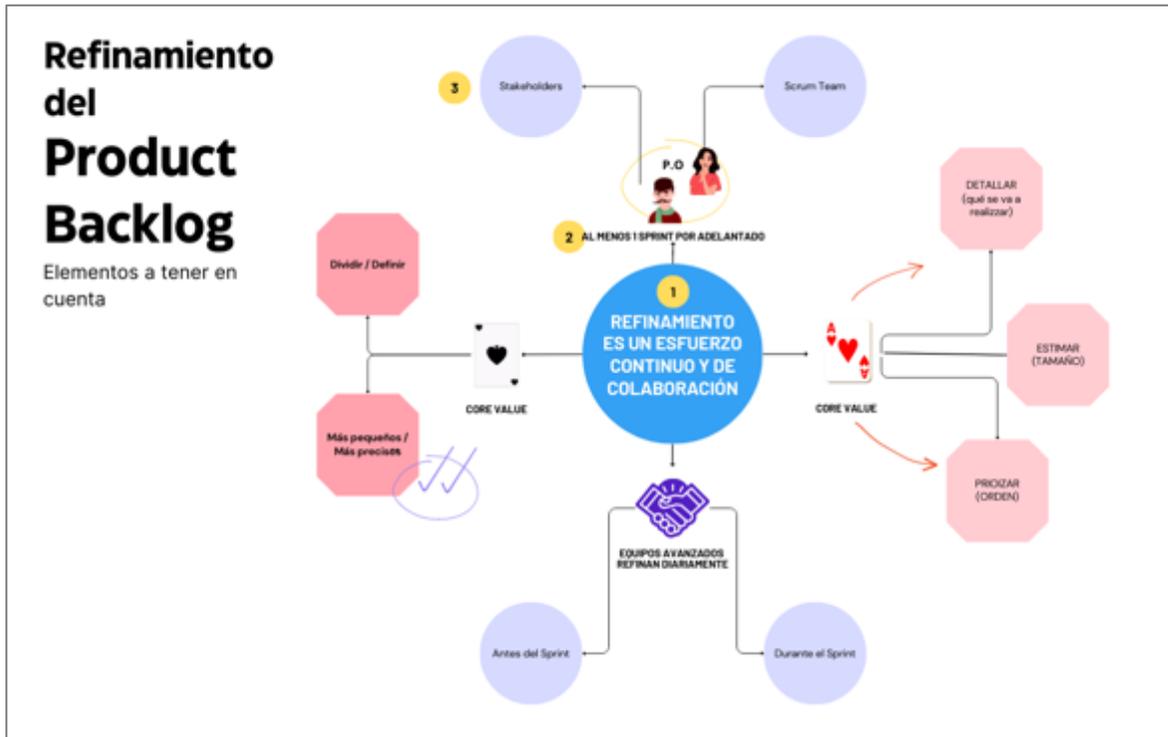
El Product Backlog es el artefacto que reúne los bloques de trabajo que podrán entrar a desarrollarse dentro de un Sprint, además:

- El Backlog consta de Elementos de Product Backlog (PBI).
- Debe refinarse y ordenarse por prioridad, aquello que más valor agrega al cliente.
- Su compromiso es lograr el objetivo de producto, definido en el acta de constitución del proyecto.
- Sólo hay un Product Backlog que se comparte entre los equipos que trabajan en el mismo producto.
- Cualquiera puede agregar PBI en el Product Backlog, pero, sólo el Product Owner decide si se desarrollan y cuándo.
- Con frecuencia los Scrum Team usan las “historias de usuario” para definir/detallar los PBI.

El Product Owner debe tener el criterio para decir lo que NO se va a desarrollar y lo que se desarrolla primero. Por lo tanto, su autoridad debe ser respaldada por la entidad, la DGIT y el área funcional. Es así que, el Product Owner sabe que no todas las funcionalidades son iguales, por eso se enfoca en obtener un MPV (Mínimo Producto Viable) para ir mejorándolo en los siguientes Sprints. Para que pueda priorizar adecuadamente los PBI, en la lógica ágil, debe ser un conocedor experto de lo que el cliente necesita, de aquello que le genera valor y de los datos que contribuyen a tomar la decisión de priorizar.

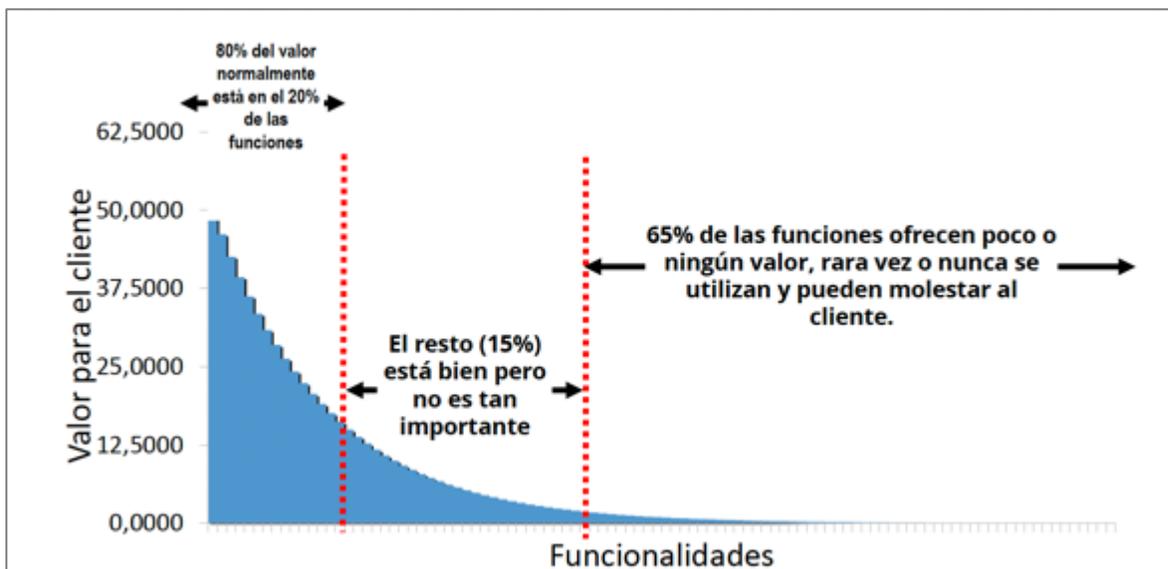
El refinamiento del Product Backlog está a cargo (al menos provisionalmente) del ingeniero de requerimientos, tal como se explicó en el apartado 6.7. Es un esfuerzo continuo que implica la colaboración de este rol con el Product Owner, el área funcional y los developers. Debe refinarse antes del primer Sprint y durante cada Sprint, para obtener un Product Backlog refinado que sirva de insumo para la Sprint Planning del siguiente sprint. Como se muestra en la gráfica a continuación:

Ilustración 5 - Refinamiento del Product Backlog



Fuente: elaboración propia

La ley de Pareto señala que el 80% de lo que realmente tiene valor para el cliente, está en el 20% de las funcionalidades. Tal como se observa en la siguiente gráfica. Es por eso que el Product Owner es un conocedor de las necesidades del cliente y del mercado en el que se encuentra.



Fuente: ScrumInc - ScrumNetwork

Este artefacto que en realidad es el plan de desarrollo del producto, debe ser:

- Detallado apropiadamente para cumplir con la transparencia
- Estimado para que el esfuerzo en cada Sprint sea el real
- Emergente porque pueden entrar nuevos PBI
- Priorizado dado que no todo lo que entra al Product Backlog se desarrollará

Es por eso, que la lista de los PBI ha de ordenarse en la parte superior con los de mayor valor conocido y por tanto esos serán los que deben tener mayor detalle para pasar a un Sprint. Todos los PBI priorizados han de orientarse a cumplir con el objetivo de producto y su Definition of Done.

Para que los PBI estén bien refinados, deben cumplir los siguientes criterios:

I	Independiente (es funcional en sí mismo)
N	Negociable (puede ser conversado y aclarado)
V	Valioso (para el usuario/cliente, que se sienta satisfecho con el resultado)
E	Estimable (el esfuerzo para desarrollarlo está claro para los developers)
S	Small (Pequeño) no puede ser un grupo de funcionalidades
T	Testeable que se puede evaluar, probar que realmente funcione y lo hace bien.

7.2.1.4 Priorización y ecuación de valor

La priorización se enfoca en aquellos PBI que genera más y mejor valor al negocio a través del producto o servicio que se está desarrollando. Por lo tanto, para un Product Owner es de gran importancia que los PBI priorizados correspondan a aquello que más valor genera al cliente, tal como se explica en la Ley de Pareto, donde el 20% de las funcionalidades es lo que se usa el 80% de las veces.

Así las cosas, se entiende como valor, la siguiente ecuación:

Ilustración 6 - Ecuación de valor



Fuente: Inspirado en ScrumInc

Entonces, el Product Owner como máxima autoridad en la administración del Product Backlog debe tener en cuenta el balance de dicha ecuación de valor, procurando:

2. Ordenar los PBI de manera que ofrezcan mayor valor a los clientes. Cuidando para el caso de la entidad la relación costo-beneficio, es decir, los beneficios que van a servir como retorno a la DIAN en relación con el costo de desarrollo. Si el beneficio se generará de manera permanente durante largo plazo, es altamente probable que ese sea un PBI de prioridad.
3. Al ordenar los PBI debe también contabilizar los cambios de valor resultantes en los incrementos de producto y explicarlo al cliente.
4. Si bien, él tiene la autoridad para tomar decisiones en nombre de las partes interesadas, debe sopesar las recomendaciones e ideas para equilibrar la ecuación de valor.
5. No debe ordenar los PBI únicamente con la percepción de los Developers sobre las dependencias, dado que eso puede afectar el incremento posible a través de otros PBI cuyo orden no está restringido.
6. Debe cuidar el costo del retraso. Lo cual implica observar el beneficio que se puede perder al demorar la entrega del producto. Usualmente esto se mide con la relación dinero / tiempo.
7. El Product Owner debe escribir lo que el equipo necesita hacer y por qué. El “cómo” hacerlo lo definen los Developers. En otras palabras, el Product Owner llega hasta la funcionalidad y los developers a las tareas técnicas necesarias para lograr dicha funcionalidad.
8. Es interesante que cada vez que se transparente cualquier PBI se motive a los Developers con el impacto que van a causar, y favorecer su entusiasmo.

7.2.1.5 ABC de la priorización

Ahora bien, habiendo diferentes métodos de priorización y para procurar un estándar a la hora de ordenar los PBI de cualquier Product Backlog, se recomienda que el Product Owner sea un conocedor del negocio, de cómo opera, sus métricas, proceso y cómo esto se conecta con la Estrategia de la Entidad. Toda esa información le ayudará de manera suficiente a realizar una priorización adecuada, y minimizar el riesgo de la presión Top-down como a favorecer el desarrollo sostenible de cualquier producto. Para lo cual se recomienda tenga en cuenta los siguientes métodos:

A. Análisis de contexto para encontrar lo que mayor valor aporta (usar siempre para cualquier producto o servicio).

De acuerdo con el tipo de productos o servicios que se van a desarrollar, muchas veces no es fácil agruparlo en una propuesta de valor asociado a un retorno monetario en el recaudo. Para ello se pueden usar algunas de estas métricas u otras que el Product Owner conozca en el contexto del negocio y que deberá buscar con apoyo de las áreas de gestión:

- Indicadores estratégicos y alineación del objetivo del producto con estos
- Reducción del tiempo de los contribuyentes o usuarios aduaneros en los procesos.
- Reducción del tiempo en la realización del proceso por parte del funcionario DIAN.
- Mejora en la facilidad de uso del servicio (Facilitación)
- Mejora en el control y seguimiento
- Aumento del recaudo o cierre de la brecha tributaria
- Reducción en la evasión o cierre de la brecha tributaria

B. Relación costo-beneficio (Aplicable en todo producto y combinable con el costo de retraso)

La relación costo-beneficio es esencial para tomar decisiones informadas sobre qué funcionalidades o mejoras (PBI) implementar primero. Al adoptar un enfoque centrado en el valor, los equipos pueden asegurarse de que están asignando recursos de manera eficiente y entregando continuamente un valor significativo al cliente y al negocio. Para esto se recomienda que el Product Owner tenga a la mano la información de los costos asociados al desarrollo del producto o servicio y el impacto que, en términos de beneficios, normalmente medidos a través de indicadores, se podrán recibir.

1. **Estimación de Costos:** Se evalúan los costos asociados con la implementación de cada elemento del backlog. Esto puede incluir el tiempo y los recursos necesarios, los costos directos e indirectos, y cualquier inversión requerida.
2. **Estimación de Beneficios:** Se evalúan los beneficios que se derivarán de la implementación de cada elemento del backlog. Estos beneficios pueden incluir mejoras en la funcionalidad del producto, la satisfacción del cliente, la captación de nuevos usuarios, etc.
3. **Comparación de Relación Costo-Beneficio:** Los elementos del backlog se priorizan en función de su relación costo-beneficio. Aquellos que proporcionan el mayor valor con los recursos disponibles se colocan en una posición más alta en la lista de prioridades.
4. **Iteración y ajuste Continuo:** La priorización basada en la relación costo-beneficio no es estática. A medida que se obtiene más información, se realizan ajustes en las prioridades. Esto puede ocurrir durante la planificación de sprint, reviews, o cuando hay cambios en los requisitos del negocio.

C. Análisis de equilibrio (Usarlo durante la Sprint Planning cuando se tiene dudas sobre qué PBI priorizar, cuando los Developers consideran otras variables técnicas)

El análisis de equilibrio en la priorización de un backlog es un enfoque que busca encontrar un equilibrio entre diferentes consideraciones, como la urgencia, la importancia estratégica y la complejidad técnica. Es decir, que este enfoque ayuda a evitar la sobre ponderación de un solo aspecto (como la urgencia inmediata) a expensas de otros factores importantes. Proporciona una visión más holística de los elementos del backlog y ayuda a tomar decisiones informadas que consideran múltiples dimensiones.

Los siguientes son algunos aspectos clave de cómo funciona este análisis y que los debería considerar un Product Owner con apoyo de los Developers:

1. **Identificación de Criterios de Equilibrio:** El Product Owner con ayuda de los Developers identifican los criterios que son esenciales para el equilibrio adecuado en el contexto específico del proyecto. Estos criterios pueden incluir factores como el valor del negocio, la complejidad técnica, la dependencia de otros elementos, la urgencia, entre otros.
2. **Asignación de Puntajes o Pesos:** Se asignan puntajes o pesos a cada criterio para reflejar su importancia relativa. Por ejemplo, un elemento del backlog puede recibir un puntaje más alto si es crítico desde el punto de vista estratégico, aunque no sea tan urgente. Esto sólo lo sabe y lo hace el Product Owner, no los Developers.
3. **Evaluación de Elementos del Backlog:** Cada elemento del backlog se evalúa en función de los criterios identificados. Esto puede implicar discusiones y análisis detallados para comprender

cómo cada elemento contribuye al objetivo de generar valor más rápidamente, pero de manera responsable de tal forma que no se ponga en riesgo el incremento del Sprint o el desarrollo sostenible del producto.

4. **Priorización Basada en el Análisis de Equilibrio:** Los elementos del backlog se priorizan en función de la combinación de los puntajes o pesos asignados a los diferentes criterios. Los elementos que equilibran de manera efectiva las diferentes consideraciones se colocan en posiciones más altas del backlog.
5. **Revisión Continua:** La priorización basada en el análisis de equilibrio no es estática. A medida que cambian las circunstancias, se obtiene más información o se reciben nuevos PBI de parte de las áreas funcionales, los criterios y los puntajes pueden actualizarse, cada Sprint señalará si ha de usarse o no este análisis.

D. Costo de retraso

Cuando los anteriores métodos no son suficientes para determinar la priorización de los PBI o alguno de ellos, se sugiere que el Product Owner use la técnica el "costo de retraso". Básicamente este es un concepto que se utiliza en la priorización para cuantificar el impacto financiero de posponer la entrega de un determinado trabajo o característica, materializado en un Incremento de producto.

Este enfoque reconoce que el tiempo es un recurso valioso y que el aplazamiento de ciertos elementos (PBI) puede tener consecuencias económicas significativas. La aplicación del costo de retraso permite al Product Owner tomar decisiones informadas que equilibran la entrega rápida con el valor económico asociado a la puntualidad. Este enfoque contribuye a maximizar el retorno de inversión al centrarse en las tareas que tienen el mayor impacto financiero positivo o evitan pérdidas significativas debido a demoras.

1. **Identificación de la Importancia del Tiempo:** el Product Owner trabaja en estrecha colaboración con los interesados y las partes relevantes para comprender la importancia del tiempo en el contexto del proyecto. ¿Existen fechas límite críticas? ¿La velocidad de entrega impacta directamente en la ventaja competitiva o en la satisfacción del cliente?
2. **Estimación del Costo de Retraso:** se realiza una estimación del costo financiero asociado con el retraso de cada elemento del backlog. Este costo puede incluir pérdida de recaudo, oportunidades de normatividad, o incluso penalizaciones contractuales en algunos casos.
3. **Priorización Basada en el Costo de Retraso:** los elementos del backlog se priorizan considerando su costo de retraso. Aquellos con un mayor impacto financiero debido al tiempo se sitúan en posiciones más altas del backlog. Esta priorización asegura que se aborden primero las tareas que pueden generar mayores beneficios o evitar pérdidas significativas.
4. **Comunicación Transparente:** es crucial que el Product Owner comunique de manera transparente las decisiones de priorización basadas en el costo de retraso. Esto ayuda a que el equipo y los interesados comprendan la lógica detrás de las decisiones y la importancia de cumplir con ciertos plazos.
5. **Revisión Continua:** a medida que evolucionan las circunstancias, se obtiene nueva información o se ajustan los objetivos incluso de la Entidad, el costo de retraso puede cambiar. Por lo tanto, la revisión y ajuste continuo de la priorización basada en este concepto es esencial.

7.2.1.6 Mapa de historias y plan de lanzamientos

El mapa de historias es un instrumento sugerido para la priorización y refinamiento de los PBI del Backlog, que permite llegar a un nivel necesario de detalle e incluso permite definir los lanzamientos

con base en lo que va alcanzando. Es esencialmente un instrumento para tener la visión de la estructura de desarrollo del producto. Este mapa de historias y lanzamientos es una buena práctica para crear el Product Backlog y continuar su administración.

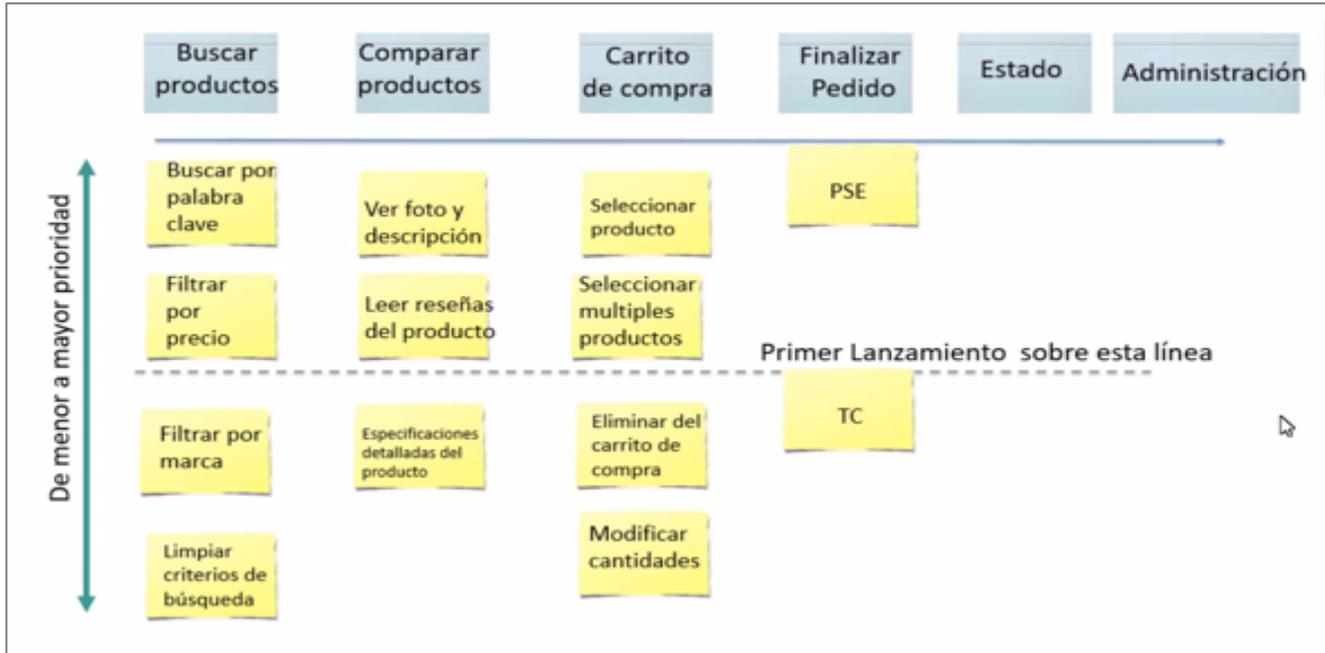
Este instrumento es versátil y pueden incluirse nuevas columnas con base en las necesidades del equipo Scrum cuando están refinando. No es un instrumento estático. Hay elementos fundamentales que son los que se detallan a continuación, pero la creatividad del equipo lo puede volver todavía más útil.

Se construye con base en esta estructura de desarrollo y alcance de producto:

Visión del proyecto	Se trae de la constitución	
Objetivo de producto	Se trae de la constitución, es el COMPROMISO del Product Backlog.	
Funcionalidades	Se trae de los PBI más generales que se definieron en la Agile Inception	
Épicas	Paquetes grandes de funcionalidades. Se asemeja a los capítulos de un libro.	
Features	Detalle mayor de funcionalidades que hacen parte de la épica.	
Product Backlog Items que hacen parte de la feature. (Historias de usuario)	<p>Elementos micro más detallados</p> <p>¿Cuál es la mínima funcionalidad que podría lanzar para cumplir esa funcionalidad? ¿Qué mejora le puedo hacer a esa funcionalidad luego de cumplirla?</p> <p>No se detalla el proceso, sino, cuál es la funcionalidad mínima a implementar y que después se va a ir mejorando.</p> <p>Esto se va incluyendo en el Sprint Backlog con su respectiva estimación de puntos.</p>	<p>Lanzamientos</p> <p>Se van definiendo de acuerdo a lo que se va priorizando.</p> <p>Se hacen marcaciones con línea punteada para definir los lanzamientos.</p>
Tareas	Las tareas técnicas que los developers tendrán que hacer para resolver con la debida calidad cada historia de usuario.	

Fuente: Elaboración propia

Ejemplo:



Fuente: ScrumInc – ScrumNetwork

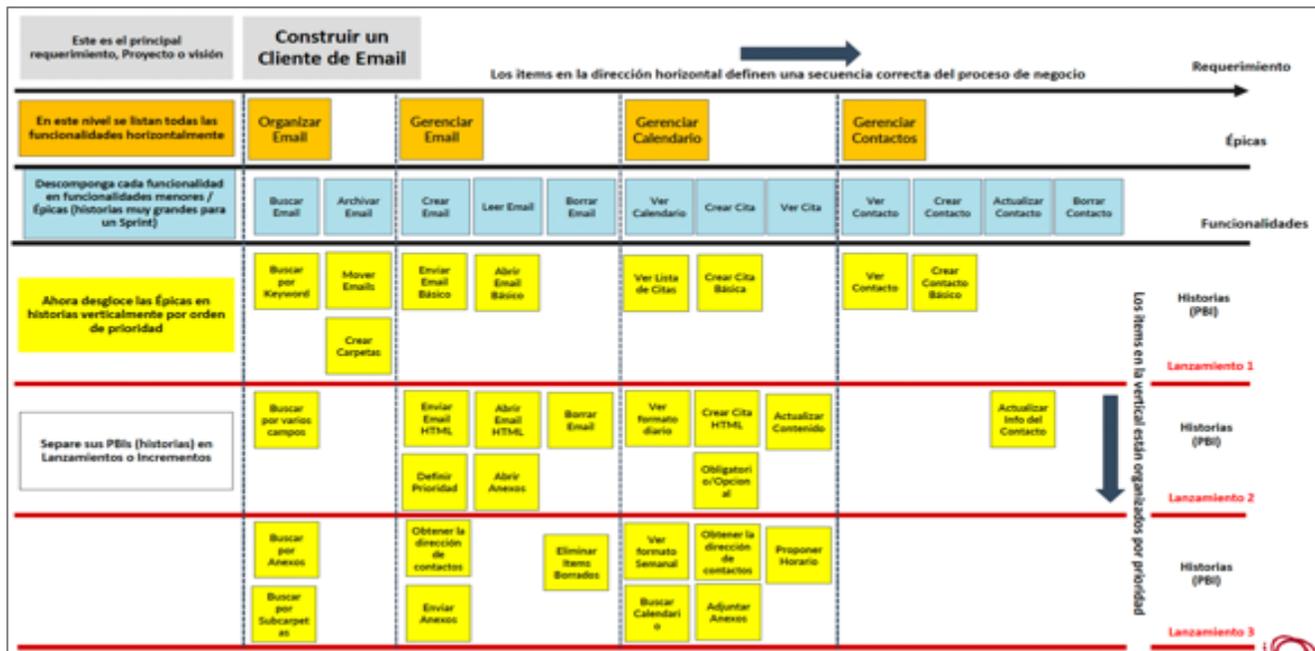
El Product Owner define lo que entra a cada lanzamiento en cuanto a producto, no en cuanto a proceso de desarrollo técnico o lo que los Developers deben tener en cuenta. A manera de buena práctica cada equipo puede agregar las columnas y filas que desee en el Mapa de PBI, dado que es una herramienta de trabajo que ayuda al refinamiento y a preparar el principal insumo para cada Sprint Planning.

Ahora, un lanzamiento es un grupo de funcionalidades categorizadas que se entregan para ambiente Beta o de pruebas o idealmente para el cliente en una determinada fecha.

Los lanzamientos no cumplen con fecha y alcance fijo al mismo tiempo. Puede ser que para la fecha definida no se tiene “el todo” estimado, por eso no se puede definir alcance fijo en los lanzamientos. Si el alcance se fija, el tiempo debe ser flexible.

El plan de lanzamiento es un estimado de lo que el Product Owner y Scrum Master estiman que va a estar listo a nivel interno o para el cliente final. Es responsabilidad de estos roles definir dicho plan y realizar las acciones necesarias para llevarlo a cabo junto con los demás actores de la Entidad.

Ejemplo:



Fuente: ScrumInc - ScrumNetwork

7.2.2 Etapa de ejecución de Sprints

Esta etapa de la gestión ágil de proyectos incluye elementos que se consideran propicios para la implementación de buenas prácticas. Estas se logran con la puesta en escena de eventos, dentro de ellos la construcción de artefactos y un equipo Scrum que adopta patrones que ya han sido probados con éxito en marcos ágiles de trabajo como Lean, XP, Kanban u otras. Aunque los patrones no son prescriptivos por parte de Scrum, la guía 2020 de este marco sí recomienda conocerlos y adoptarlos para alcanzar una dinámica de trabajo todavía más ágil.

7.2.2.1 Eventos para la ejecución ágil de proyectos

Los eventos son todos los momentos en los que el Scrum Team se reúne para definir un compromiso o para inspeccionarlo o adaptarlo. El Sprint es el evento que compila otros 4 eventos y momentos de encuentro, es el latido del marco Scrum, en donde se lleva a cabo la planeación, desarrollo, monitoreo y control (inspección y adaptación) del producto.

Un proyecto ágil puede tener tantos Sprints como necesite, ese dependerá de la complejidad del producto, la madurez del equipo en Scrum, su expertise técnica, la disponibilidad de recursos para los despliegues, si se ha fijado o no fecha de entrega y el presupuesto. La experiencia de los equipos y el respaldo de la DIGIT lograrán configurar proyectos con la cadencia más adecuada para cada contexto.

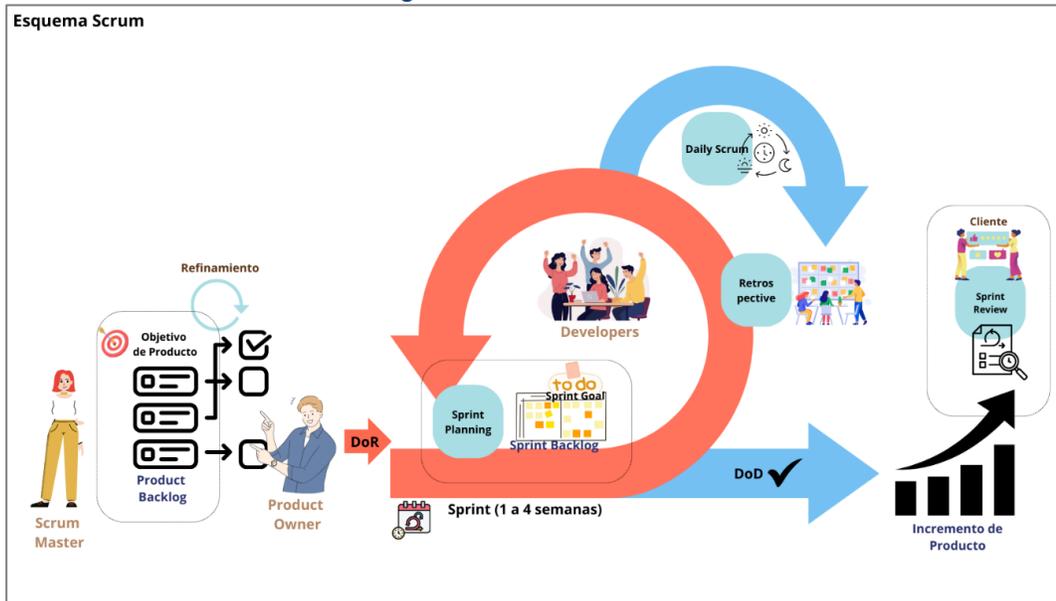
Las siguientes son cualidades de un Sprint:

- Cada Sprint tiene la misma lógica a nivel general, pero cuenta con un objetivo diferente y con un incremento de producto que va sumando al producto completado.
- Un Sprint tiene una duración mínima de una semana y máximo de 4.

- c. Cada Sprint cuenta con al menos 4 eventos obligatorios: Sprint Planning, Daily, Sprint Review y Sprint Retrospective.
- d. Durante un Sprint el equipo debe autogestionarse y llevar a cabo las reuniones técnicas necesarias para el desarrollo del incremento del producto comprometido.

La siguiente gráfica ilustra los eventos, artefactos y otros elementos de un Sprint que se van a ir mencionando en el desarrollo de este apartado:

Ilustración 7 - Scrum en modo gráfico



Fuente: Elaboración propia con base en guía Scrum 2020

1. PLANNING - Ejecutar la planeación detallada del Sprint

Tabla 12 - Planning

Artefacto a obtener:	Sprint Backlog
Compromiso:	Objetivo del Sprint
Convoca:	Scrum Master
Participantes:	Todo el Scrum Team (aunque en una parte el Product Owner se retira para dejar a los developers refinar en tareas técnicas).
Otros:	El Product Owner actualiza el plan de lanzamientos.
	El equipo define los criterios de aceptación del incremento.
	El equipo verifica la definición de listo (Definición of Ready), criterios a tener chequeados antes de que un PBI entre a un Sprint.
Duración:	Este evento es de máximo 8 horas cuando el Sprint es de 4 semanas, el tiempo total de la Sprint Planning corresponderá a la duración del Sprint con base en esta métrica. Si el Sprint es de 2 semanas, el tiempo de planeación es de 2 horas.

Fuente: elaboración propia

Es el primer evento del Sprint. Allí El Scrum Team establece el trabajo que se realizará en el Sprint. El Product Owner y el Scrum Master se reúnen con el equipo de trabajo para obtener el Sprint Backlog

del Sprint. Se puede invitar a otras personas en esta reunión si se consideran clave para la claridad y la planeación.

Aspectos a tener en cuenta para el desarrollo de la reunión:

- El Product Owner presenta la visión del producto y monitorea la comprensión del equipo. Es decir, se asegura de que los developers entienden lo que se espera obtener.
- Asimismo, presenta los PBI definidos hasta el momento y sugiere con cuál comenzar dada su visión como Product Owner.
- Señala lo que se espera "lanzar" al cierre del Sprint, lo cual, contribuirá con la construcción del "Plan de releases" o plan de lanzamientos, durante esta reunión se obtendrá un cronograma aproximado que el Product Owner debe ir perfilando en comunicación con los developers.
- El equipo de trabajo realiza preguntas para comprender tanto la visión del producto como de los PBI sugeridos para el Sprint.
- Los developers deben estar seguros de su comprensión para poder elaborar el Sprint Backlog y definir su objetivo.
- Los developers deciden "cómo desarrollar el trabajo", el Product Owner define qué va en cada Sprint aunque puede y se recomienda tener en cuenta los aportes del Scrum Team e interesados.

Durante la Sprint Planning el equipo Scrum se asegura de resolver las siguientes preguntas:

2. ¿Por qué es valioso este Sprint?

Allí el Product Owner señala cómo el producto que se espera obtener de este Sprint resulta en valor y utilidad para el cliente y el problema a solucionar.

Todo el Scrum Team define un objetivo del Sprint en donde se explica por qué este Sprint es valioso para los interesados.

Este objetivo del Sprint debe estar listo antes de que termine la Sprint Planning.

3. ¿Qué se puede hacer en este Sprint?

El equipo selecciona los PBI a incluir en el Sprint a través de conversar con el Product Owner. Es importante que el Scrum Team refine en ese momento los PBI seleccionados para ganar comprensión y confianza.

El equipo debe seleccionar los PBI que pueden completar dentro del Sprint con base en su desempeño pasado (en otros proyectos o el actual) y la definición de terminado (DoD).

4. ¿Cómo se realizará el trabajo?

El equipo desagrega cada PBI en elementos más pequeños de un día o menos de duración para lograr el increment con el que se dará alcance al objetivo del Sprint. Aquí se comprometen con el increment a través de desarrollar determinado número de PBI.

Si un PBI no se terminó en el Sprint, vuelve al Product Backlog y se deberá analizar nuevamente su priorización y si es necesario volver a estimar.

2. PRODUCT BACKLOG REFINEMENT - Actualizar y refinar el Product Backlog

Tabla 10 – Refinamiento del Product Backlog

Artefacto a obtener:	Product Backlog refinado Mapa de PBI para la siguiente Sprint Planning
Convoca:	Product Owner con apoyo del CenIT
Participantes:	Product Owner con apoyo del CenIT Ingeniero de requerimientos Profesionales delegados del área funcional o cliente Developers
Otros:	El Product Owner es responsable del refinamiento. Levantar las historias de usuario de cara a las áreas funcionales o clientes. Es importante llegar hasta tareas técnicas con los Developers para que el mapa de PBI esté suficientemente refinado para la Sprint Planning.
Duración:	Este es un proceso continuo que debe realizarse antes de cada sprint para llegar a la Planning con el insumo suficiente. Una buena práctica es que se tome el tiempo necesario con el área funcional y preferiblemente no más del 10% del tiempo de los developers durante el Sprint.

Fuente: Elaboración propia

Aunque el refinamiento no es propiamente un “evento” Scrum, sí es un proceso continuo, permanente y de colaboración, que se cita acá dada la importancia que tiene en cada Sprint. Durante la ejecución del proyecto y previendo lo que se va a desarrollar en cada Sprint, el Product Owner y el ingeniero de requerimientos tienen como responsabilidad primaria actualizar y refinar el Product Backlog. Este procedimiento se explicó con detalle en el apartado **Creación y administración del Product Backlog**.

Cuando el Sprint ha avanzado lo suficiente, el Product Owner ajusta/adapta el Product Backlog dependiendo de los resultados de la Sprint Review.

Proceso de refinamiento:

También debe realizar el refinamiento de los PBI del Product Backlog, para lo cual puede solicitar el apoyo del resto del Scrum Team o de algunos de sus miembros, pero no debe superar el 10% del total de horas hombre dedicada por todo el Scrum Team en el Sprint. Ejemplo, si el equipo suma 96 horas, no se destinarían más de 9,6 horas para refinar con otros miembros del equipo.

En estas reuniones de refinamiento se ha de lograr:

- Tener los PBI listos para ser desarrollados en el siguiente Sprint.
- Esté todo listo para el siguiente Sprint Planning.
- Esté bien entendido, dividido y detallado.
- Si hay bloqueos, eliminarlos o solucionarlos.
- Lograr la DoR (Definition of Ready) criterios a tener chequeados antes de que un PBI entre a un Sprint. Se valida en la Sprint Planning por cada PBI que entre al Sprint.
- Contar preferiblemente con las tareas técnicas para cada PBI.

Todo lo anterior va a garantizar un Sprint Planning más fluido, eficiente y ágil.

3. DAILY SCRUM - Inspeccionar y adaptar el trabajo diariamente (Daily)

Tabla 11 – Daily Scrum

Artefacto:	Sprint Backlog
Convoca:	Scrum Master
Participantes:	Developers Scrum Master
Duración:	15 minutos cada día, a la misma hora y lugar, para generar el hábito, pero puede ser modificada de acuerdo con la dinámica de los developers. (No tiene que ser presencial, se adapta el proceso a la virtualidad).
Otros:	Si el Product Owner o el Scrum Master participan en el desarrollo de increment, en la Daily lo hacen como developers. Es loable que el Product Owner participe, pero no es obligatorio. Es fundamental que el Scrum Master participe y oriente las Daily, especialmente cuando el equipo es novato. Cuando el equipo es maduro seguramente van a querer ellos mismos orientar su reunión de diario.

Fuente: Elaboración propia

Esta reunión es organizada por los developers o equipo de trabajo. Busca inspeccionar el progreso hacia el objetivo del Sprint y adaptar el Sprint Backlog (ajustarlo) con base en lo que se va planeando. Es una reunión autogestionada. Aunque es muy seguro que el Scrum Master deba marcar la pauta de organización y dinamismo durante las primeras sesiones, especialmente cuando el equipo es novato.

Los developers deben lograr lo siguiente en cada Daily:

- a. Obtener un plan de progreso hacia el objetivo del Sprint a cumplir en el siguiente día.
- b. Tener visible el tablero de estados de tareas (Todo, doing, done). Ese plan es el Sprint Backlog que se adapta luego de cada Daily.
- c. Mejorar la comunicación entre los miembros del equipo.
- d. Identificar desafíos, problemas, impedimentos, riesgos y/o cambios.
- e. Comunicar al Product Owner para gestionar cambios o riesgos.
- f. Tomar las decisiones rápidamente para reorganizar su estrategia de trabajo.
- g. Eliminar la necesidad de más reuniones de inspección (seguimiento, monitoreo y control).

Los developers deben ser conscientes de que la Daily tiene un propósito claro centrado en tanto se va acercando al incremento, pero eso no impide tener tantas reuniones técnicas como necesiten para ajustar su plan y volver a planificar su Sprint.

4. REVIEW - Entregar el incremento funcional, inspeccionarlo e identificar el PBI de valor para el siguiente Sprint (Sprint Review)

Tabla 12 – Review

Artefacto a obtener:	Incremento del producto
Compromiso:	Incremento
Convoca:	Product Owner con apoyo del CenIT
Participantes:	Todo el Scrum Team

Otros:	Se inspecciona el incremento, para corroborar que efectivamente se desarrolló lo que se comprometió en el Sprint Backlog. Los interesados deben dar feedback a lo que el Scrum Team está planeando agregar al nuevo Sprint El Product Owner tiene en cuenta el feedback para definir qué se agrega y qué se prioriza en el Product Backlog.
Duración:	Tiene una duración de máximo 4 horas para un Sprint de un mes, se ajusta su duración en su correspondencia con la duración efectiva de cada sprint. Si el Sprint es de 2 semanas, la Sprint Review tendrá una duración de 2 horas.

Fuente: Elaboración propia

El Scrum Team presenta los resultados a los interesados y se revisa el progreso hacia el objetivo del Producto, es en concreto una reunión de trabajo para inspeccionar y adaptar, o en otras palabras para monitorear y controlar el avance hacia el objetivo.

Antes de entregar el increment a los interesados, el Equipo Scrum ha cotejado sus criterios de aceptación y Definition of Done contra el incremento, asegurándose de que lo cumplen. Si algún PBI no se desarrolló, éste vuelve al Product Backlog.

No se trata de una presentación de avances, sino que es una sesión de trabajo en donde todos los actores participan. Previamente, durante el Sprint, los developers deben garantizar las pruebas técnicas y funcionales.

En otras palabras, el Scrum Team se asegura de:

- Entregar qué se hizo en el Sprint. El producto (increment) funcionando.
- Qué quiere el cliente en el siguiente Sprint.
- El Scrum Team puede presentar sus ideas y validarlas con el cliente.
- El cliente valida lo que se va a realizar y enseguida pasa a ser adaptado el Product Backlog.

Manejo de la deuda técnica y gestión de Bugs

Antes de concertar la reunión de Review y durante todo el Sprint, el Scrum Team, y principalmente los Developers deben estar atentos al manejo de la deuda técnica y de bugs. Ambos están relacionados con la calidad y gestión del código.

La deuda técnica se refiere a la acumulación de diseño o implementación, que, en el corto plazo, permiten un desarrollo más rápido, pero, en el largo plazo, aumentan la complejidad del sistema y afectan su mantenimiento. Surge por decisiones conscientes de compromisos en el diseño o en la calidad del código para cumplir con plazos ajustados o requisitos específicos.

La administración de la deuda técnica es un proceso continuo, que debe configurarse como una práctica sostenible del Scrum Team. Una premisa para el Scrum Team es que, si se detecta deuda técnica, se debe parar de hacer lo que se está haciendo y atender y eliminar la deuda técnica, porque de no atenderse inmediatamente se acumularía y esto llevaría a un increment/producto de mala calidad y a un fracaso en la entrega de valor.

Tabla 13 – Manejo de la deuda técnica

Acción	Recomendación para el Scrum Team
Análisis y priorización	Realizar un análisis exhaustivo del código para identificar áreas de deuda técnica. Priorizar estas áreas según su impacto en la calidad y mantenibilidad del software.
Refactorización	Planificar y realizar refactorizaciones en áreas identificadas. Esto implica mejorar la estructura del código sin cambiar su comportamiento externo. Dividir las refactorizaciones en PBI manejables.
Automatización de pruebas	Implementa y mejora las suites de pruebas automatizadas. Esto ayuda a garantizar que las refactorizaciones y modificaciones no introduzcan nuevos errores.
Establecimiento de estándares de codificación	Establece revisiones de código regulares. Durante estas revisiones, busca y aborda patrones de código que contribuyan a la deuda técnica.
Gestión de dependencias	Mantén las bibliotecas y dependencias actualizadas para aprovechar mejoras de rendimiento y correcciones de errores. Actualizar regularmente puede evitar problemas de compatibilidad y mantener la seguridad del software.
Participación del equipo	Involucra a todo el equipo en la identificación y abordaje de la deuda técnica. Fomenta una cultura donde todos sean responsables de la calidad del código.
Revisión posterior a la implementación	Después de cada implementación, realiza una revisión para identificar y corregir cualquier nueva deuda técnica introducida durante el desarrollo. Hasta lograr la estabilización.

Fuente: Elaboración propia

Ahora bien, también se puede establecer un proceso de capacitación continua donde se proporcionen oportunidades para que el equipo esté al tanto de las mejores prácticas y tecnologías actuales.

En cuento a la administración o manejo de los Bugs:

Un Bug, también conocido como error o defecto, es un comportamiento no deseado o incorrecto en el software y, por tanto, afecta su funcionamiento. Pueden surgir por errores de programación, problemas de diseño o ambos, o incluso cuando los requisitos no se manejaron adecuadamente.

El manejo de los Bugs se puede llevar a cabo con las siguientes acciones, no todas se deben implementar al mismo tiempo. Cada Scrum Team puede tener un protocolo de gestión de los Bugs para el producto.

Tabla 14 – Manejo de los Bugs

Acción	Recomendación para el Scrum Team
Sistema de Seguimiento de Problemas	Utilizar un sistema de seguimiento de problemas para registrar, priorizar y dar seguimiento a los bugs. Asegúrate de que todos los miembros del equipo tengan acceso y sepan cómo utilizarlo.
Claridad en la descripción del Bug	Al reportar un bug, proporciona una descripción clara y concisa del problema. Incluye pasos para reproducirlo, información del entorno

Acción	Recomendación para el Scrum Team
	y cualquier dato relevante que ayude a los desarrolladores a entender y abordar el bug.
Priorización de Bugs	Establecer un sistema de priorización para los bugs. Clasificar los bugs según su gravedad y su impacto en el usuario y en el sistema. Esto ayuda a enfocar los esfuerzos en corregir los problemas más críticos primero.
Asignación de responsabilidades	Asignar cada bug a un miembro específico del equipo para su corrección. Clarificar quién es el responsable de abordar cada problema, y establecer plazos razonables.
Comunicación Transparente	Mantener a todos los miembros del equipo y a los stakeholders informados sobre el estado de los bugs. Utilizar comentarios en el sistema de seguimiento de problemas para proporcionar actualizaciones y resolver dudas.
Pruebas de regresión	Desarrollar y ejecutar pruebas de regresión después de corregir un bug para asegurar que la solución no haya introducido nuevos problemas. La automatización de pruebas puede ser valiosa en este contexto.
Establecimiento de políticas de cierre	Definir políticas claras sobre cuándo un bug puede considerarse cerrado. Esto puede incluir la verificación del equipo de control de calidad, la aprobación del stakeholder, o la validación del usuario final.
Análisis de tendencias	Analizar las tendencias de los bugs para identificar patrones o áreas del software que requieren una atención más detallada. Esto puede ayudar a prevenir problemas similares en el futuro.
Automatización de reportes	Configurar informes automatizados para rastrear la cantidad de bugs, su estado y el tiempo de resolución. Esto proporciona visibilidad sobre la calidad del software a lo largo del tiempo.
Revisión posterior al Sprint	Después de cada sprint, realiza una revisión detallada de los bugs registrados y resueltos. Esto ayuda a identificar áreas de mejora en los procesos de desarrollo y pruebas.
Gestión de Versiones y Ramas	Utilizar un sistema de control de versiones eficaz y gestiona las ramas de desarrollo. Esto facilita la corrección de bugs en versiones específicas del software y evita la propagación de problemas a otras partes del código.

Fuente: Elaboración propia

RETROSPECTIVE - Inspeccionar el progreso e identificar los aspectos a mejorar

Tabla 15 – Retrospective

Producto a obtener:	Items de mejoramiento para accionar en el siguiente Sprint
Convoca:	Scrum Master
Participantes:	Este evento es para todo el Scrum Team sin invitados externos.
Duración:	Tiene una duración de máximo 3 horas para un Sprint de un mes y su correspondencia según la duración real de cada Sprint.

Fuente: Elaboración propia

Se enfoca en observar cómo avanzó el Sprint, en inspeccionar la forma como se desarrolló: las interacciones de las personas, el proceso, el desarrollo de las Daily y su efectividad, las herramientas usadas, el alcance a la Definition of Done (definición de terminado) y todo lo que el Scrum Team considere hace parte del camino del progreso. Se identifican los problemas que tuvieron y cómo (o no) los solucionaron.

Deben encontrar al menos 1 o 2 instancias de mejoramiento y agregarlas al siguiente Sprint con las acciones constitutivas para llevarlas a cabo. El Scrum Team decide qué es prioritario mejorar para atender primero y de qué manera. Por ejemplo: (1) Que el Product Owner asista a las Daily que se le convoca, (2) Que el tester haga las pruebas y siempre avise cuando termine (3) Que el equipo se comunique mejor cada día. Es decir, cualquiera sea lo que se seleccione a mejorar, ingresa automáticamente al siguiente Sprint para ejecutar acciones concretas de mejora y que se mantienen a lo largo de la línea de tiempo de los demás Sprints.

7.2.2.2 Monitoreo y métricas

En una metodología tradicional de gestión de proyectos se habla de seguimiento, monitoreo, control de cambios. En la entidad, el sistema de gestión de calidad a través de sus procedimientos también cuenta con este lenguaje. En el marco Scrum, se habla de tres pilares empíricos. La siguiente tabla procura la armonización de estas miradas, asegurando que el monitoreo y seguimiento se cumplan ampliamente, pero a través de una lógica diferente dentro de cada Sprint y su escalamiento:

Tabla 16 – Pilares empíricos y su relación con el monitoreo

Pilares empíricos Scrum	Acciones de gestión	Fases/Eventos de la gestión ágil en donde se lleva a cabo
Transparencia: tanto el proceso realizado como todo lo que se espera obtener debe ser suficientemente claro para cada uno de los <i>interesados</i> y miembros del equipo Scrum.	Planeación Comunicación Entendimiento Aclaración Monitoreo	Planeación general Ejecución-Sprint Planning Ejecución-Daily Ejecución-Refinamiento de Product Backlog
Inspección: todos los artefactos y el progreso hacia los objetivos se examinan con frecuencia para detectar variaciones y validar la calidad. Esto se hace dentro de cada Sprint.	Seguimiento Monitoreo de avance y riesgos Control de cambios Gestión de alertas	<u>Ejecución:</u> <u>Daily:</u> se comunican todos los impedimentos que son o no capaces de gestionar los developers y se pasan al Product Owner y al Scrum Master aquellas que les competen para su gestión con los interesados o demás áreas de la entidad. La comunicación cobra un papel esencial entre los miembros del equipo, para informar a tiempo los impedimentos y gestionarlos ágilmente. *Durante la ejecución de cada Sprint el Scrum Master ayuda a remover impedimentos (riesgos, cambios, alertas).

Pilares empíricos Scrum	Acciones de gestión	Fases/Eventos de la gestión ágil en donde se lleva a cabo
		<p>Tanto el Scrum Master como el Product Owner cuentan con responsabilidades de gestión complementaria del proyecto, no sólo lo técnico. Ellos ayudarán a identificar también impedimentos y a gestionarlos.</p> <p><u>Sprint Review</u> Durante esta reunión todos los miembros del equipo Scrum participan y debe estar el cliente, juntos inspeccionan el incremento, verificando los criterios de aceptación y allí se puede también identificar riesgos o cambios que deben atenderse. Es un evento altamente aprovechable dado que se encuentran todos los roles reunidos.</p> <p><i>Sprint Retrospective:</i> Se inspecciona el progreso (la manera, el modo y el momento) relacionamiento del equipo y calidad - hacia el logro del incremento. Si se identifican impedimentos se deben pasar a acciones concretas para su gestión.</p>
<p>Adaptación: a través de la inspección se detecta aquello que hay que adaptar, y esto se hace de manera inmediata para evitar desviaciones en el progreso del trabajo o productos inaceptables. El equipo de trabajo debe autogestionar la adaptación con base en su capacidad de aprendizaje. La adaptación reduce la deuda técnica, la cual se debe evitar siempre o atender de manera inmediata cuando se detecte.</p>	<p>Ajustar Adaptar Cambiar Reiniciar Asegurar Garantizar Actualización Documentación</p>	<p>Ejecución: Daily: se adapta el Sprint Backlog Sprint Review: Se actualiza el Product Backlog Sprint Retrospective: se ajusta el progreso.</p> <p>*Durante todo el Sprint se realiza la adaptación de aquello que se considera impedimento, ya sea a través de los patrones y la implementación de buenas prácticas y con el acompañamiento del Product Owner y del Scrum Master.</p>

Fuente: elaboración propia

Tablero para las métricas

En Scrum, hay varias métricas que se utilizan para evaluar el rendimiento del equipo y medir el progreso del proyecto. Es importante destacar que, aunque las métricas pueden proporcionar información valiosa, deben interpretarse con cuidado y no utilizarse de manera aislada. Estas métricas proporcionan

una visión integral del rendimiento del equipo y del progreso del proyecto. Sin embargo, es esencial utilizarlas de manera equilibrada y considerar el contexto específico de cada equipo y proyecto. Además, Scrum enfatiza la adaptabilidad, por lo que las métricas deben ajustarse según las necesidades cambiantes del equipo y del proyecto.

Los objetivos de tener Métricas y Transparencia son:

- a. Proveer el contexto apropiado para tomar decisiones basadas en datos (practicar el pilar de empirismo).
- b. Reducir la latencia de decisiones (Cuanto mayor sea la latencia de decisiones, más tiempo lleva resolver problemas, adaptarse a cambios o implementar nuevas estrategias).
- c. Optimizar el trabajo requerido por las células, Stakeholders o la alta dirección.

Tabla 17 – Métricas de Scrum estándar y Scrum@Scale

Métrica	Descripción	Fórmula	Utilidad/recomendación
Velocidad normalizada del Equipo (Team Velocity)	Mide la cantidad de trabajo que el equipo es capaz de completar en un sprint. Se expresa en unidades relativas (historias de usuario, puntos de historia, etc.).	Promedio de puntos de historia de los últimos 3 Sprints	Útil para establecer la velocidad de un equipo en contexto y extrapolar a otros equipos de la Entidad en condiciones similares.
Burn Down Chart	Representa visualmente la cantidad de trabajo pendiente en un sprint a lo largo del tiempo.	Ver apartado 7.2.2.3	Ayuda a visualizar si el equipo está en camino de completar todas las tareas planificadas.
Historias de Usuario Completadas (Completed User Stories)	Mide la cantidad de historias de usuario que el equipo ha completado en cada sprint.	Historias de usuario completadas / total de historias planeadas * 100	Indica la capacidad del equipo para cumplir con los compromisos de sprint, se asocia a los puntos de historia que son capaces de desarrollar.
Satisfacción del Cliente Medida de Calidad Scrum@Scale	Mide la satisfacción del cliente con el producto entregado.	Puede evaluarse mediante la Sprint Review, aplicación de encuestas, retroalimentación directa u otras métricas de experiencia del usuario.	Altamente recomendable.
Capacidad del equipo en teoría	Mide la cantidad de developers disponibles durante el Sprint	Número de developers * días del Sprint	Útil para establecer la base de la capacidad del equipo

Métrica	Descripción	Fórmula	Utilidad/recomendación
Capacidad real del equipo	Mide la disponibilidad de los developers en el sprint	Se calcula usando el Tool-DIAN Yesterday's weather	Contribuye en tener la realidad de la capacidad del equipo para la planeación del Sprint
Búfer del Sprint	Mide la cantidad de horas de interrupción que tiene un equipo durante un sprint y proyecta en porcentaje de tiempo el búfer para la planning.	$\text{Búfer} = \frac{\text{Horas búfer promedio}}{\text{Horas de todo el equipo en el sprint}} * 100$	Ayuda a planear mejor los puntos de historia a desarrollar en el Sprint
Promedio de quemado diario	Sirven para rastrear y comunicar el progreso diario del equipo durante un sprint.	Cada día durante el sprint, el equipo registra cuántos puntos de historia se han completado hasta ese momento. Esto se basa en el progreso real del trabajo realizado.	La tendencia del gráfico de quemado (burndown chart) (si los puntos están aumentando o disminuyendo) proporciona una visión rápida del progreso diario del equipo hacia la finalización del sprint
Ciclo de Desarrollo (Development Cycle Time)	Mide el tiempo que transcurre desde que se inicia una tarea hasta que se completa. Proporciona información sobre la eficiencia del equipo en la entrega de funcionalidades.	$\text{Ciclo de desarrollo} = \text{tiempo de finalización} - \text{tiempo de inicio}$	Sólo cuando el equipo se está observando lento, o hay indicios de una estimación por debajo de la capacidad real del equipo.
Cobertura de Pruebas (Test Coverage) Medida de Calidad Scrum@Scale	Mide el porcentaje de código que está cubierto por pruebas automatizadas. Proporciona información sobre la solidez de las pruebas del equipo.	$\frac{\text{Número de líneas de código cubiertas por pruebas automatizadas}}{\text{Total de líneas de código}} * 100\%$	Cuando se hacen pruebas automatizadas o se usan Pipelines.
Adopción de usuarios finales	Puede medirse de diversas maneras dependiendo del contexto y los objetivos específicos de un producto o servicio. No hay una fórmula única y universal para medir la adopción de	Porcentaje de adopción de usuarios finales $\text{UAM (Usuarios Activos Mensuales)} = \frac{\# \text{ usuarios que interactúan con el producto / servicio en un periodo específico}}{\text{Total de usuarios}} * 100\%$	La clave es seleccionar las métricas que mejor reflejen el éxito en términos de adopción de usuarios finales para el contexto particular en el que se esté trabajando.

Métrica	Descripción	Fórmula	Utilidad/recomendación
	usuarios finales, ya que puede variar según la naturaleza del producto o servicio. Sin embargo, hay algunas métricas comunes que se utilizan para evaluar la adopción de usuarios finales.	<p>UAD (usuarios Activos Diarios) = # de usuarios que interactúan con el producto / servicio en un día específico</p> <p>Tiempo promedio de sesión = suma del tiempo total de sesiones de usuarios / número total de sesiones</p> <p>Frecuencia de uso = número total de sesiones / número total de usuarios activos</p>	
<p>Costo del Sprint Y Costo del Sprint Historia Usuario</p>	<p>Determinar el costo del sprint implica considerar diversos factores, como los salarios del equipo, los costos de infraestructura, herramientas, entre otros. Una métrica comúnmente utilizada para medir el costo de un sprint es el Costo del Sprint por Historia de Usuario (User Story Cost).</p>	<p>Costo del Sprint por H.U = Costo total del sprint / Número total de Historias de Usuario completadas</p> <p>Costo total del Sprint: Suma de todos los costos asociados con la ejecución del sprint: Costo de salarios del Scrum Team. Costo de licencias. Costo de infraestructura. Costo de adquisiciones.</p> <p>Número total de Historias de Usuario completadas: Cantidad de historias de usuario que fueron implementadas, probadas y aceptadas durante el sprint.</p>	<p>Esta métrica proporciona una idea del costo promedio asociado con la implementación de cada historia de usuario en el sprint. Es importante tener en cuenta que esta métrica es solo una parte del análisis del rendimiento del equipo y del valor entregado, y no debe considerarse de manera aislada.</p> <p>Al utilizar esta métrica, es crucial también considerar otros factores cualitativos, como la complejidad de las historias de usuario, la calidad del código, la satisfacción del cliente y la capacidad del equipo para cumplir con los plazos y los objetivos del sprint.</p> <p>Además, se debe tener en cuenta que el valor y el éxito del sprint no solo se miden por el costo, sino también por la entrega efectiva de valor al cliente.</p>

Métrica	Descripción	Fórmula	Utilidad/recomendación
Costo del producto en producción	<p>La métrica del costo del producto en producción puede ser calculada utilizando la métrica conocida como Costo Total de Propiedad (Total Cost of Ownership - TCO).</p> <p>La fórmula general para el Costo Total de Propiedad puede variar según los componentes específicos</p>	<p>Costo total de propiedad (TCO) = Costos iniciales + costos operativos + costos de mantenimiento</p> <p>Opcional: sumar costos de soporte y capacitación.</p> <p>Costos Iniciales: Incluye los costos asociados con la adquisición inicial del producto o sistema. Esto puede incluir costos de desarrollo, licencias de software, hardware, y otros costos asociados con la implementación inicial.</p> <p>Costos Operativos: Incluye los costos recurrentes asociados con el funcionamiento continuo del producto en producción. Pueden ser costos de hosting, energía, personal operativo, y otros gastos continuos.</p> <p>Costos de Mantenimiento: Incluye los costos asociados con las actividades de mantenimiento del producto. Esto puede abarcar actualizaciones de software, corrección de errores, parches de seguridad y cualquier otra actividad que mantenga el producto funcionando eficientemente.</p> <p>Opcional: Costos de Soporte y Capacitación: También incluir los costos asociados con el soporte</p>	<p>Es importante ajustar la fórmula según las características específicas de tu producto y los factores que consideres más relevantes. Además, al calcular el Costo Total de Propiedad, puedes obtener una visión más completa de los gastos asociados con el producto a lo largo de su ciclo de vida.</p> <p>Es relevante destacar que el TCO no solo se trata de reducir costos, sino de optimizar la eficiencia y maximizar el valor del producto.</p> <p>La calidad del producto, su capacidad para cumplir con los objetivos del negocio y la satisfacción del usuario también deben ser consideradas junto con los costos.</p>

Métrica	Descripción	Fórmula	Utilidad/recomendación
		técnico y la capacitación del personal relacionados con el uso y mantenimiento del producto.	
Productividad Scrum@Scale	<p>Esta métrica busca cuantificar los cambios en la cantidad de funcionalidades o características que se entregan al final de cada Sprint.</p> <p>Es una herramienta valiosa para evaluar y mejorar la eficiencia y la consistencia del equipo en la entrega de software funcional.</p>	Número total de cambios en el producto funcionando entregado por Sprint	<p>Si la métrica muestra un aumento en la cantidad de producto funcional entregado por Sprint, indica que el equipo está mejorando su capacidad de entrega y está siendo más efectivo en la generación de valor con el tiempo.</p> <p>Por otro lado, si la métrica muestra fluctuaciones o disminuciones, puede ser una señal para que el equipo y los interesados investiguen las posibles razones detrás de estos cambios. Podría implicar la identificación de obstáculos, ajustes en la planificación o mejoras continuas en las prácticas de desarrollo para garantizar una entrega constante y confiable de productos funcionales en cada iteración</p>
Entrega de valor Scrum@Scale	Es una medida que busca evaluar la eficiencia y efectividad de un equipo en términos de la relación entre el valor entregado al negocio y el esfuerzo invertido para producir ese valor.	<p>Valor del negocio por unidad de esfuerzo = Valor del negocio / unidades de esfuerzo del equipo</p> <p>Para calcular esta métrica, se divide el valor del negocio obtenido (medido en términos de beneficios, ingresos o cualquier otra unidad de valor) por la cantidad de esfuerzo o trabajo del equipo (medido en unidades de tiempo, como horas de trabajo o</p>	Un resultado más alto indica que el equipo está generando más valor en relación con el esfuerzo invertido, lo cual es un indicador positivo de eficiencia. Por otro lado, un valor más bajo podría sugerir que el equipo está enfrentando desafíos en la generación de valor de manera eficiente y podría requerir ajustes en sus prácticas, procesos o enfoques de trabajo.

Métrica	Descripción	Fórmula	Utilidad/recomendación
		puntos de historia de usuario)	Esta métrica es valiosa para la toma de decisiones y la mejora continua, ya que proporciona una medida cuantitativa de la eficacia del equipo en la entrega de valor. Sin embargo, es importante considerar que evaluar el valor del negocio puede ser subjetivo y puede requerir una comprensión clara de los objetivos y las expectativas del negocio para interpretar adecuadamente la métrica.
<p>Sostenibilidad Scrum@Scale</p>	<p>Se refiere a la capacidad del equipo y del proceso de desarrollo para mantener un ritmo de trabajo constante y saludable a lo largo del tiempo. Esta métrica busca evaluar si el equipo puede mantener su productividad y calidad a largo plazo sin experimentar agotamiento o deterioro en su rendimiento.</p>	<p>Velocidad del Equipo: La velocidad del equipo, medida en la cantidad de trabajo completado en cada Sprint, puede indicar la capacidad sostenible del equipo para cumplir con sus compromisos. Desarrollada previamente en esta misma tabla.</p> <p>Tiempo de Resolución de Problemas: La rapidez con la que el equipo aborda y resuelve problemas puede ser un indicador de la capacidad de la célula para manejar desafíos de manera efectiva y sostenible.</p> <p>Rotación del Equipo: Un alto índice de rotación de miembros de la célula puede indicar problemas de sostenibilidad, ya que la estabilidad del equipo es crucial para un rendimiento consistente.</p> <p>Retroalimentación del Equipo: La percepción y</p>	<p>Evaluar la sostenibilidad en un proyecto Scrum es clave para evitar el agotamiento del equipo y garantizar que la calidad del trabajo no se vea comprometida a medida que el proyecto avanza. Un enfoque sostenible es esencial para el éxito a largo plazo, promoviendo la eficacia continua y el bienestar del equipo.</p>

Métrica	Descripción	Fórmula	Utilidad/recomendación
		<p>retroalimentación del equipo sobre su carga de trabajo y su capacidad para mantener un ritmo sostenible es esencial. Las retroalimentaciones de las retrospectivas pueden proporcionar información valiosa.</p> <p>Entregas Consistentes: La capacidad del equipo para realizar entregas consistentes de incrementos de producto funcional en cada Sprint es fundamental para evaluar su sostenibilidad a largo plazo. Su capacidad de mantenerse en un buen ritmo de cadencia en uno u otro producto.</p> <p>Equilibrio Trabajo-Vida: La capacidad del equipo para mantener un equilibrio saludable entre el trabajo y la vida personal puede ser indicativo de una sostenibilidad adecuada.</p>	

Fuente: Elaboración propia

A continuación, se presenta un tablero de monitoreo en el cual se incluyen algunas de estas métricas. No obstante, el Scrum Master siempre será libre de modificar y enriquecer su reporte con estas o nuevas métricas. Además, usando la herramienta de gestión ágil se podrán mejorar ampliamente o reemplazar estos tableros.

Tabla 18 – Dashboard de monitoreo para un proyecto recién iniciado



Fuente: Elaboración propia

Tabla 19 – Dashboard para proyecto que ha ejecutado al menos 3 sprints



Fuente: Elaboración propia

7.2.2.3 Gráficas Burndown:

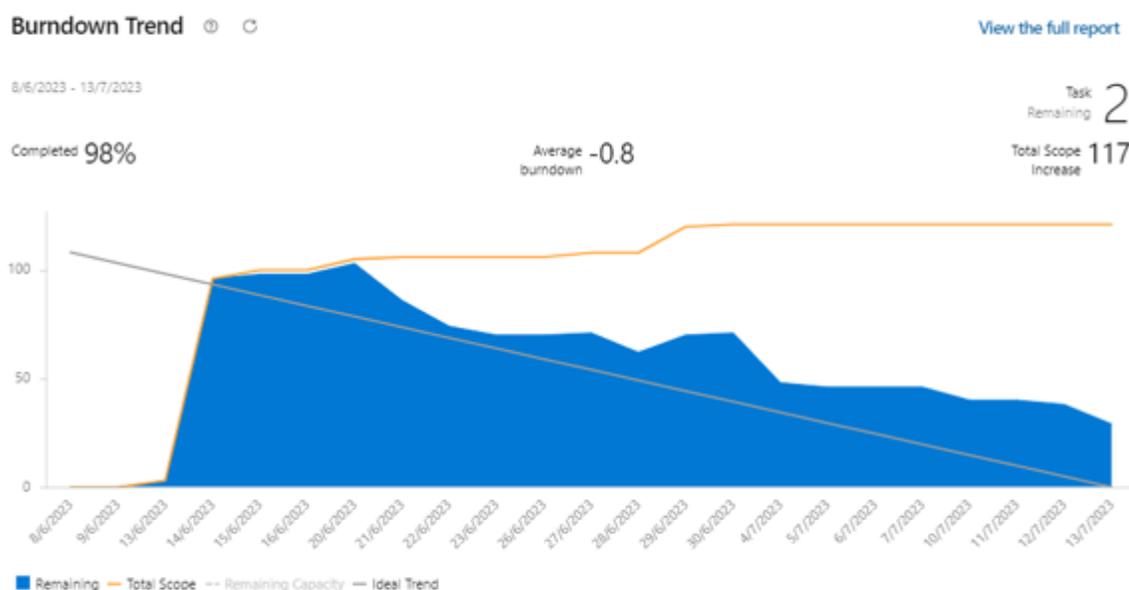
La representación gráfica conocida como Burndown Chart puede utilizarse en diversas situaciones para conocer el progreso de un proyecto. En el contexto de un Sprint, es común emplear esta herramienta para visualizar el estado de avance del trabajo en dicho periodo. Estas gráficas se pueden generar en la herramienta de gestión ágil de la entidad.

Por otro lado, también es posible utilizar **Burndown Chart para monitorear el progreso** de una entrega o release específica. Por ejemplo, si deben entregar las funcionalidades A, B y C en un plazo determinado; en este caso, un burndown chart alimentado con los datos correspondientes permitiría conocer **el estado de la entrega** sin depender del número de Sprints involucrados.

Finalmente, **product burndown** chart puede ser útil en el contexto de un proyecto cerrado, en el que se han definido previamente el alcance, el tiempo y el costo del mismo. En este escenario, un burndown chart alimentado con dicha información permitiría anticiparse a posibles situaciones problemáticas y asegurar el cumplimiento de los objetivos del proyecto en cuanto a tiempo y costo.

Elementos para construir las Burndown Charts:

Ilustración 8 - Burndown chart



Fuente: Elaboración propia DevOps

Una Burndown chart se construye con los siguientes elementos:

- Puntos de datos: Progreso

Los puntos de datos en la gráfica burndown representan el progreso real del trabajo en función del tiempo. En otras palabras, estos puntos indican la cantidad de trabajo que se ha completado hasta el momento en que se toma la medición. Idealmente, estos puntos deberían estar por debajo de la línea guía, lo que indicaría que el equipo está completando el trabajo a un ritmo más rápido de lo esperado.

- Tendencia: Predicción

La tendencia en la gráfica burndown se refiere a la dirección en la que se está moviendo la curva de progreso. Si la tendencia es descendente, significa que el equipo está avanzando a un ritmo adecuado para completar todo el trabajo comprometido antes del plazo establecido. Por otro lado, si la tendencia es plana o ascendente, puede indicar que el equipo está atrasado o que se están agregando más tareas al Sprint.

- Métricas de desempeño: Análisis

Las gráficas burndown también permiten calcular una serie de métricas de desempeño, que pueden ser utilizadas para analizar el progreso del proyecto. Algunas de las métricas más comunes son la velocidad del equipo, que mide la cantidad de trabajo que se completa en un Sprint; y el índice de burnup, que compara la cantidad de trabajo completado con la cantidad de trabajo comprometido para el proyecto en su totalidad. Estas métricas pueden ayudar a identificar problemas y oportunidades de mejora en el proceso de desarrollo.

- Línea de progreso

Es importante recordar que la gráfica burndown es una herramienta valiosa para Product Owner y los developers, ya que permite monitorear el progreso del proyecto en tiempo real y tomar decisiones basadas en datos objetivos. Sin embargo, para que la gráfica burndown sea efectiva, es necesario que el equipo estime las tareas de manera precisa y realista, y que actualice la gráfica regularmente para reflejar el progreso real del proyecto. Además, es importante tener en cuenta que la gráfica burndown es solo una herramienta más en el proceso de desarrollo ágil, y no debe ser vista como la única medida de éxito del proyecto.

7.2.3 Etapa de cierre

La etapa de cierre consiste en revisar la última entrega de producto para poder determinar que ha sido “completado”. Esto hace referencia al estado de entregado y recibido a satisfacción por parte del usuario.

Durante esta fase se lleva a cabo la última Sprint Review para la entrega del producto completado y verificar su funcionamiento. Luego, durante la misma reunión se diligencia el formato FT-IIT-2722 Acta de entrega de proyecto, la cual contiene los siguientes campos:

- Datos generales del proyecto
- Entregables del backlog del producto
- Razones que motivan el cierre
- Pendientes / observaciones / Comentarios
- Lecciones aprendidas
- Cierre financiero
- Entrega del proyecto
- Si el proyecto ha tenido un operador o tercero que lo ha ejecutado, debe diligenciarse el acta de liquidación de acuerdo con el procedimiento de gestión contractual.

9. SCRUM@SCALE ESCALAMIENTO DE LA TRANSFORMACIÓN ÁGIL

Scrum@Scale es un marco de trabajo desarrollado para extender los principios ágiles de Scrum a empresas grandes y complejas. Este proporciona orientación sobre cómo coordinar y sincronizar múltiples equipos Scrum que trabajan en proyectos interdependientes dentro de una organización.

Scrum@Scale se centra en promover la flexibilidad y la agilidad en el escalado, permitiendo que los equipos ágiles mantengan su autonomía y eficacia mientras trabajan de manera conjunta para abordar desafíos a mayor escala.

Los beneficios de Scrum Scale incluyen:

- f) La capacidad de las organizaciones para adaptarse más rápidamente a los cambios del mercado.
- g) Mejorar la eficiencia operativa para lograr agilidad de negocio respondiendo a los cambios de manera rápida adoptando la configuración inicial de manera estable.
- h) Proporcionar un marco de trabajo que fomente la colaboración y la innovación en toda la entidad.
- i) Al facilitar la implementación ágil en entornos más amplios, Scrum@Scale busca mejorar la calidad del producto, acelerar la entrega y aumentar la satisfacción del cliente, contribuyendo así a una transformación organizativa exitosa hacia la agilidad.
- j) Entonces, Scrum@Scale procura el trabajo de red organizado, generando una saturación adecuada de la comunicación y vasos comunicantes que permitan hacer las definiciones estratégicas, técnicas y operativas sin aumentar en burocracia ni jerarquías. Los equipos base siguen siendo las células (Scrum Teams) sólo que ahora estarán acompañados de otros equipos que favorecerán la comunicación, coordinación, remoción de impedimentos y toma rápida de decisiones.

Con el escalamiento de Scrum en la Entidad se busca:

- c) Escalabilidad lineal: correspondencia entre aumento de porcentaje entrega de producto funcionando con el aumento de cantidad de equipo.
- d) Agilidad de negocio: la habilidad para responder a los cambios y retos adaptando la configuración estable inicial.

8.1 Principios a cumplir para hacer escalamiento en la DIAN

- 10. Respaldo de la alta dirección al escalamiento y todo lo que ello conlleva, por ejemplo, cambio cultural, ajuste en procesos, asignación de recursos y presupuesto.
- 11. Debe tenerse el respaldo de la dirección general y demás actores que participarán de los grupos líderes del escalamiento:
 - d) Scrum of Scrums (SoS) equipo coordinado de células.
 - e) Executive Action Team (EAT): enfocado en cómo hacerlo más rápido.
 - f) Executive Meta Scrum (EMS): enfocado en el valor de lo que produce.

Los últimos dos EAT y EMS deben respaldar al SoS para que operen efectivamente.

12. Contar con un sistema operacional estándar de Scrum, es decir, tener células o Scrum Teams activos desarrollando y entregando incrementos al final de cada Sprint. Deben estar funcionando bien, es decir, realizando un proceso Scrum completo, sin mezclas de otros marcos ágiles y mucho menos combinaciones con metodologías tradicionales como waterfall. La Entidad debe asegurarse de que esas células esas células están implementando bien Scrum, porque cualquier error que se esté presentando es potencialmente escalable.

13. Profesionales en los roles del marco Scrum listos para escalar.

14. Product Owners empoderados con el rol, con plena conciencia de la importancia y ejercicio de sus responsabilidades. Deben estar formados en Scrum y cumplir el perfil descrito en el apartado 6.6 de este documento.

15. Cambio en la cultura organizacional sensibilizada e impulsada por valores

Los pilares empíricos: transparencia, inspección y adaptación; así como los valores Scrum son columnas en la escalabilidad.

Es necesario contar con un sistema de aprendizaje enfocado en la transformación ágil para que la Entidad comprenda, apropie y colabore activamente en la gestión ágil y en la forma como se escala Scrum.

16. Modelo de referencia: el primer escalamiento debe ser un conjunto de equipos que se coordinan para entregar en cada sprint (prototipo para escalar en próximas redes de equipos). Este modelo marcará la pauta la creación de un sistema operativo ágil pues el resto de la entidad tendrá un ejemplo funcional y saludable de Scrum@Scale para replicar.

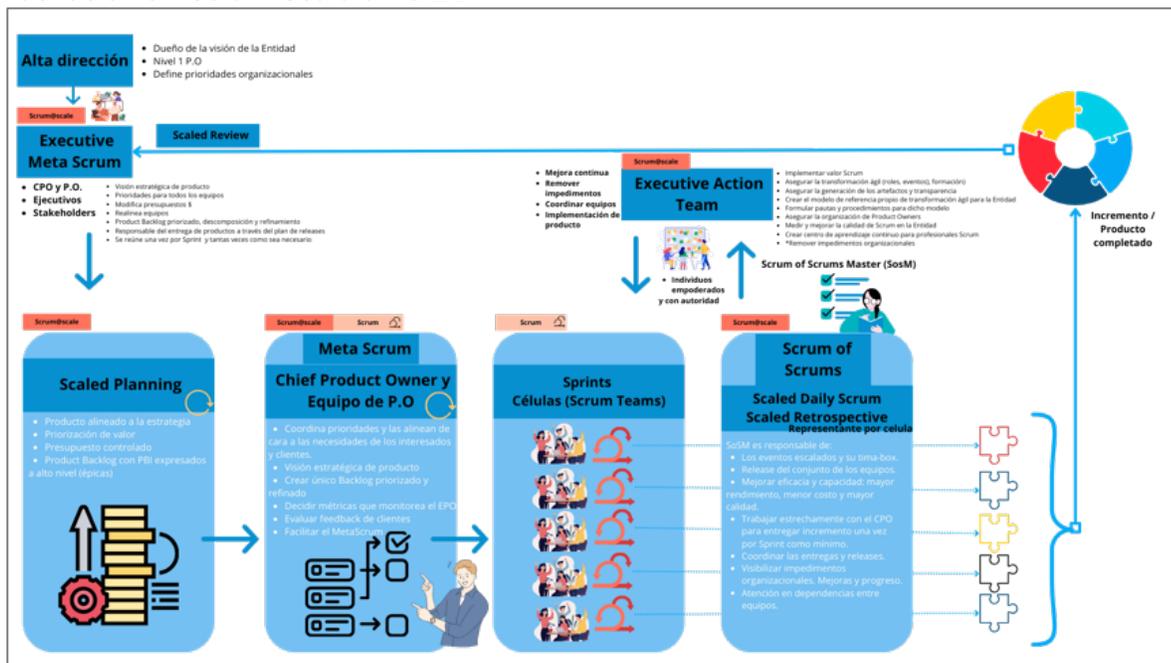
Es necesario corregir cualquier deficiencia antes de escalar más equipos. Escalar un sistema roto no resuelve nada, lo empeora todo. El Modelo de referencia debe funcionar bien.

17. Cada Scrum of Scrums debe cumplir con el proceso de equipo, tal como lo hace un Scrum Team o célula:

- a. Maximizar el flujo de trabajo completado y que cumpla con la DoD
- b. Aumentar el rendimiento del equipo a través del tiempo
- c. Operar de manera sostenible y enriquecedora para el equipo
- d. Acelerar el ciclo de retroalimentación del cliente

18. Sincronización de las células por SoS o más grupos de la red dependiendo de la complejidad y magnitud del proyecto. Esto implica que para un proyecto que tiene varios productos ejecutándose al mismo tiempo se requiere sincronizar el inicio del Sprint y también su duración. Esto se debe fundamentalmente a los eventos escalados. Así mismo, generar un solo Product Backlog priorizado para evitar la duplicación de trabajo y crear una mínima Definition of Done uniforme que aplique a todas las células. De igual forma aplica para su roadmap y plan de release. Todo se sincroniza para lograr el escalamiento lineal. Si una célula está por fuera de la sincronización no se puede escalar o si un artefacto o evento no se aborda de manera sincronizada, no se puede escalar.

Ilustración 9 - Scrum Scale en la DIAN



Fuente: Elaboración propia

8.2 Conceptos básicos para el escalamiento

Operar Scrum@Scale implica coordinar y sincronizar múltiples células Scrum que trabajan en conjunto para lograr objetivos organizacionales de gran envergadura. Los siguientes son los conceptos básicos que se ponen en marcha en el escalamiento de Scrum en la DIAN.

8. Equipos Ágiles Pequeños:

Scrum Scale parte del principio de que los equipos ágiles pequeños son la unidad fundamental de trabajo eficaz. Cada equipo sigue los principios de Scrum estándar con sus roles, eventos y artefactos. La guía de Scrum@Scale recomienda que los equipos sean de no más de 5 miembros.

9. Scrum of Scrums:

Para coordinar múltiples equipos, Scrum Scale utiliza el concepto de "Scrum de Scrums", donde representantes de cada equipo se reúnen periódicamente para compartir actualizaciones, coordinar dependencias y abordar problemas inter equipos.

10. Roles Adicionales:

Scrum Scale introduce roles adicionales para facilitar la coordinación a mayor escala, como el "Scrum Máster de Scrum Masters" y el "Product Owner de Product Owners". Estos roles supervisan y facilitan la colaboración entre los Scrum Masters y Product Owners de las células individuales.

11. Priorización Escalada:

La priorización de elementos de trabajo se realiza a diferentes niveles, desde equipos individuales hasta niveles más altos de coordinación. Scrum Scale proporciona una estructura para escalonar la planificación y la priorización a medida que se amplía a través de la organización.

12. Herramientas y Prácticas Específicas:

Scrum Scale sugiere herramientas y prácticas específicas para abordar desafíos a mayor escala, como el "Scrum of Scrums Backlog" para gestionar dependencias y el "MetaScrum" para la colaboración entre equipos.

13. Flexibilidad y Adaptabilidad:

Un principio clave de Scrum Scale es la flexibilidad y adaptabilidad. Las organizaciones pueden adaptar el marco de trabajo según sus necesidades específicas, manteniendo al mismo tiempo los principios fundamentales de Scrum.

14. Mejora Continua:

Scrum Scale fomenta la mejora continua a través de la retroalimentación y la adaptación constante. Los equipos y las estructuras pueden evolucionar para abordar los desafíos a medida que surgen.

8.3 Roles para el escalamiento de Scrum

8.3.1 Scrum of Scrums Master (El Scrum de Scrum Master)

Líder al servicio de los Scrum of Scrums. Es responsable de asegurar que se realicen los eventos escalados, que sean productivos, positivos y se mantengan dentro del time-box. También es responsable de las entregas de incrementos del conjunto de células (Scrum of Scrums) y de mejorar la capacidad: esto es: mayor rendimiento del equipo, menor costo y mayor calidad. Puede ser un Scrum Master del equipo o un profesional dedicado específicamente para este rol.

Si hay varios Scrum of Scrums agrupados en un Scrum of Scrums of Scrums se necesita un de un Scrum of Scrum of Scrums Mater para la coordinación más amplia.

Tabla 20 – Scrum of Scrums Máster y sus responsabilidades

Categoría	Función
Organización	Hacer que sean visibles los impedimentos y gestionar su eliminación. Mantener informado a la organización sobre el progreso del desarrollo del producto, estado de eliminación de impedimentos y otras métricas.
Scrum of Scrums	Coordinar la entrega de los equipos con los planes de release del equipo de Product Owners. Comprender las dependencias entre equipos (SoS y no-SoS) y facilitar su coordinación y comunicación. Facilitar la Scaled Daily la Scaled Retrospective.

Chief Product Owner	Trabajar estrechamente con el CPO para entregar incremento de producto potencialmente implementable como mínimo una vez por Sprint Mantener informado al CPO sobre el progreso del desarrollo del producto.
----------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Fuente: Elaboración propia con base en la guía Scrum@Scale

8.3.2 Chief Product Owner (Jefe de Product Owners)

Es responsable juntos con sus SoSM asociados, por la implementación eficiente de incrementos de producto de acuerdo con el Plan de Release. El CPO puede ser de un Scrum Team individual en donde ejerce el rol, o puede ser una persona específicamente dedicada que no está en ninguna célula. Al momento de escalar este rol, se agrega una “C” por cada nivel, ejemplo Chief of Chief Product Owner y así sucesivamente dependiendo de la red.

Tabla 21 – Chief Product Owner y sus responsabilidades

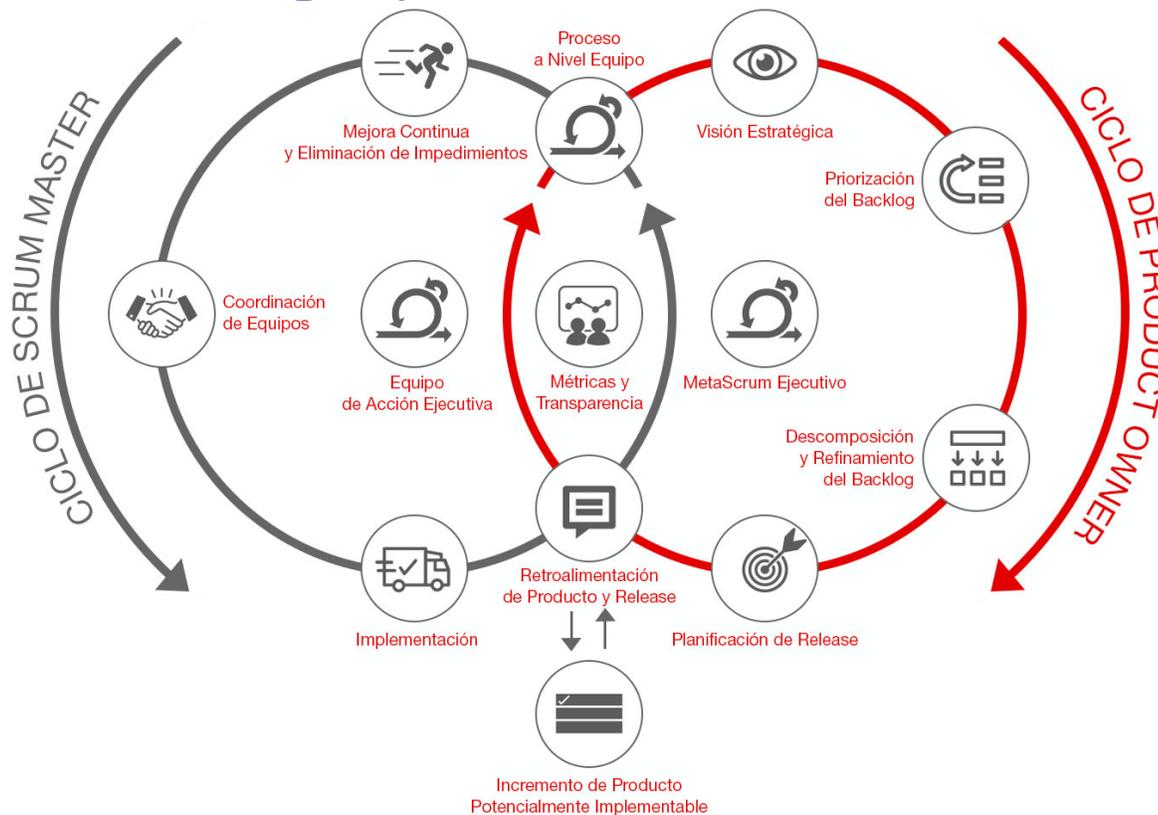
Categoría	Función
Equipo de Product Owners	Coordina las prioridades junto con el Equipo de Product Owners. Las alinean con las necesidades de los Stakeholders y los clientes. Definir qué métricas monitorea el Equipo de Product Owners. Facilitar la Review y refinamiento del Backlog siendo guiado por el CPO.
Visión estratégica	Definir la visión estratégica para todo el producto. Crear un único backlog priorizado para todos los equipos. Evaluar le feedback de los clientes sobre los productos y ajustar el backlog común. Facilitar el evento MetaScrum o Executive MetaScrum.
Plan de Release	Definirlo con el Equipo de Product Owners. Implementarlo con los SoSM asociados.

Fuente: Elaboración propia con base en la guía Scrum@Scale

8.4 Proceso de escalamiento

Lo primero que debe comprenderse es que Scrum@Scale tiene dos ciclos, el de Scrum Master enfocado en el cómo hacer las cosas y el de Product Owner centrado en el qué hacer.

Ilustración 10 - Scrum@Scale y sus ciclos



Fuente: Guía Scrum@Scale

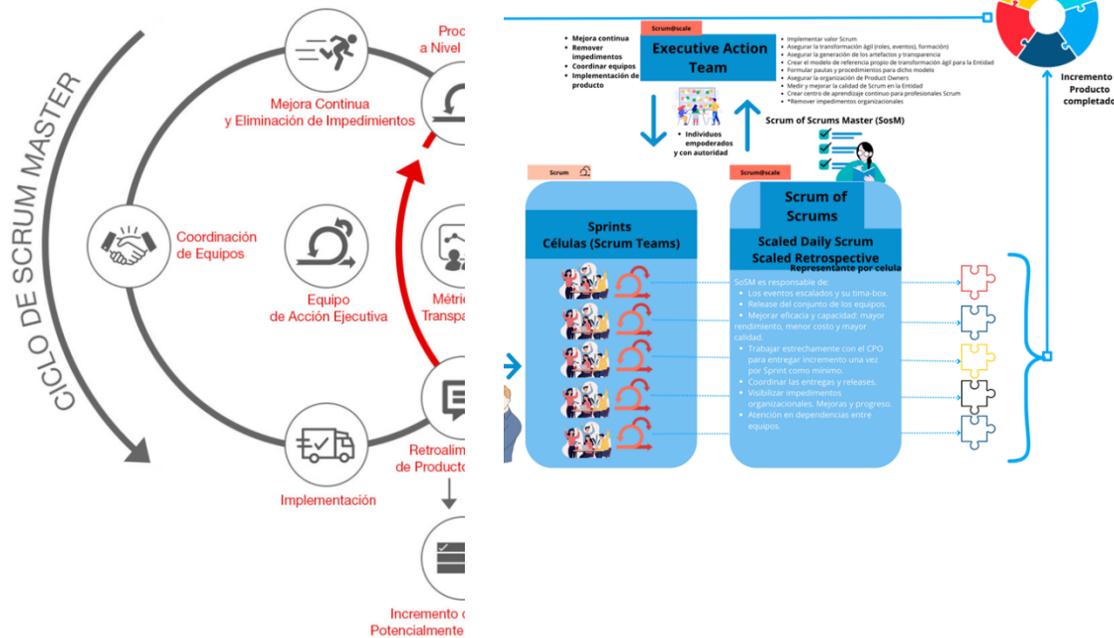
Los ciclos se explican en el siguiente apartado. Se sugiere que el lector vaya haciendo la conexión con la adaptación de Scrum@Scale para la DIAN según se presenta en la Ilustración 9 - Scrum Scale en la DIAN

8.4.1 Ciclo de Scrum Master

El ciclo de Scrum Master comprende todas las interacciones y componentes que deben darse para asegurar el CÓMO, responsabilidad del equipo de desarrollo y por tanto facilitado por el Scrum Master y todos los escalados en este rol. Acá se coordinan los esfuerzos para implementar Scrum con base en la inspección y adaptación.

Son responsables los Scrum Master, Scrum of Scrums Master y el Executive Action Team como centro de este ciclo.

Ilustración 11 - Ciclo de Scrum Máster

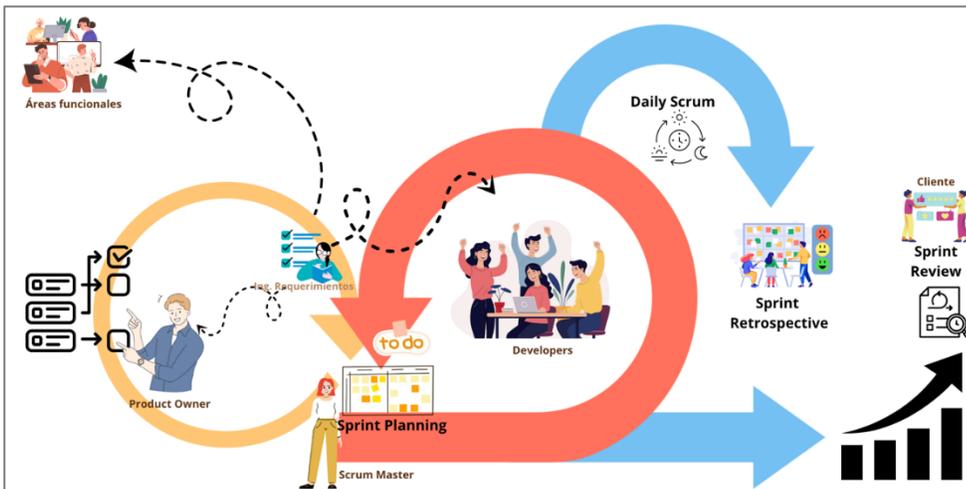


Fuente: Guía Scrum@Scale y elaboración propia

Scrum Team o célula

Como se ha dicho, la base fundamental para poder escalar en contar con equipos Scrum o células que estén operando eficientemente y entregando incremento al final de cada Sprint. De esto ya se ha hablado amplia y suficientemente a lo largo del capítulo SCRUM ESTÁNDAR - GESTIÓN DE PROYECTOS ÁGILES. Una célula de Scrum, con todos sus componentes, se grafica de la siguiente manera:

Ilustración 12 - Scrum Team o célula en la DIANI



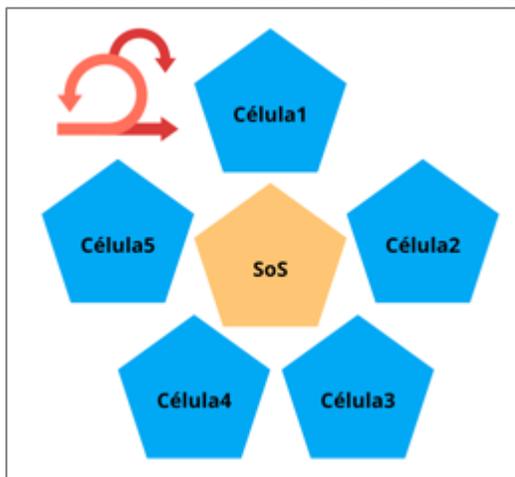
Fuente: Elaboración propia

Scrum of Scrums (SoS)

Un conjunto de equipos que necesitan coordinarse es un Scrum of Scrums, para entregar incrementos de manera coordinada que se integrarán para conformar un incremento mayor de valor para el cliente. Requiere contar con el rol SoSM Scrum of Scrum master líder al servicio de las células que trabajan en ese SoS.

Un SoS de la DIAN Se representa así:

Ilustración 13 - Scrum of Scrums en la DIAN



Fuente: Elaboración propia

Un SoS opera como si fuera un Scrum Team, siendo responsable de un conjunto completamente integrado de incrementos de producto potencialmente entregables al final de cada sprint.

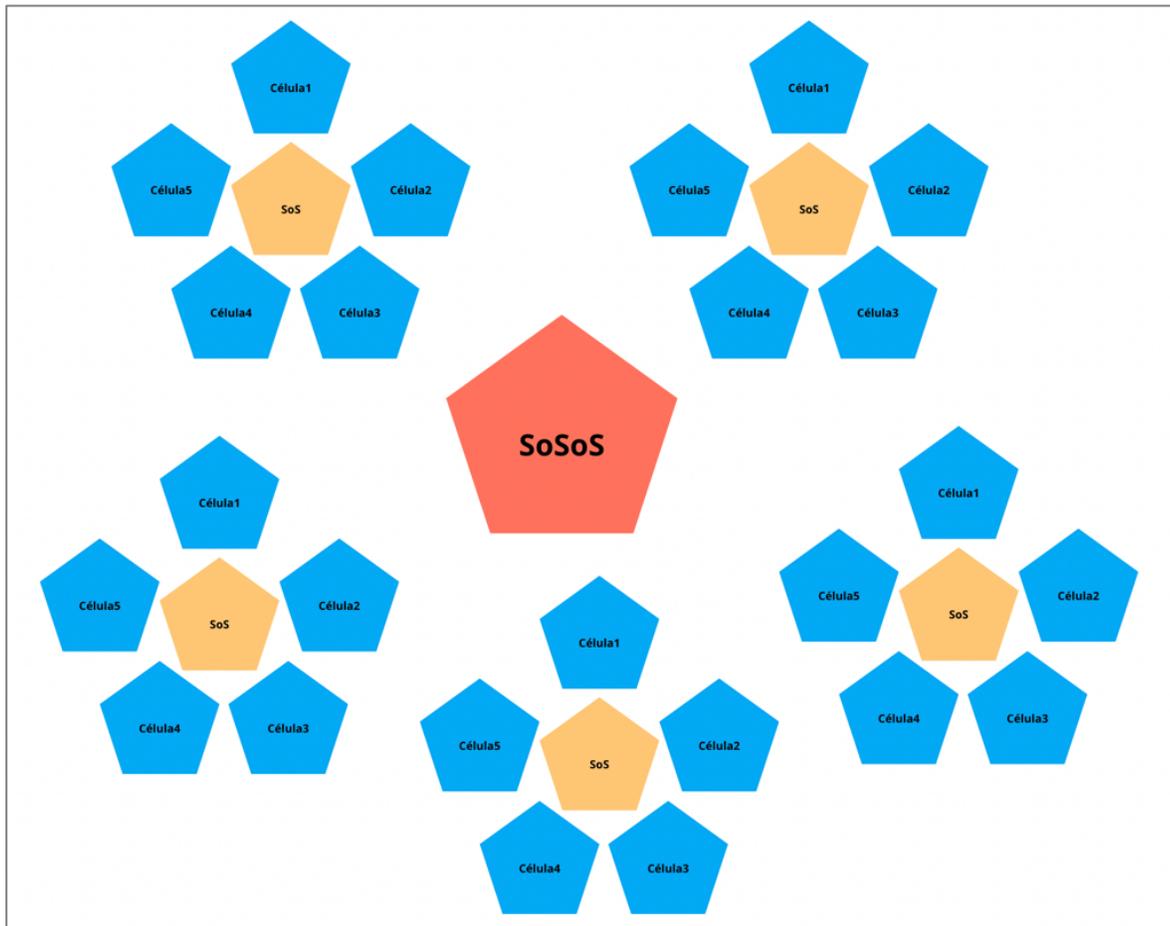
Scrum of Scrum of Scrums (SoSoS)

Se puede seguir escalando si la Entidad lo requiere, debido a la cantidad de células Scrum que activan para atender el mismo proyecto. Por lo cual, se agrupan cada 5 células como máximo en un SoS, y cada 5 SoS como máximo conforman un nuevo nivel de Scrum of Scrums of Scrums.

Cada SoSoS debe tener las versiones escaladas de los roles, artefactos y eventos Scrum of Scrums. Asimismo, interactúa con cada uno de sus SoS de la misma manera que un SoS actúa con una célula Scrum.

Un Scrum of Scrum of Scrum, se puede graficar para su comprensión, como se observa en la siguiente ilustración, en donde se sincronizan 25 equipos, de manera lineal.

Ilustración 14 - Scrum of Scrum of Scrums en la DIAN



Fuente: Elaboración propia

Siempre que la Entidad necesita, podrá seguir escalando los SoS con la misma lógica explicada y guardando todos los eventos, roles y artefactos del Scrum estándar.

Executive Action Team (EAT)

Es el eje del ciclo del Scrum Master. Es un equipo de liderazgo empoderado para tomar decisiones y remover impedimentos de orden organizacional. Por lo tanto, debe estar integrado por individuos que cuenten con total respaldo político y económico para poder eliminarlos.

Su función es coordinar varios Scrum of Scrums (o redes más grandes) e interactuar con la parte no-ágil de la Entidad para procurar su alineación y mitigar impactos negativos.

Debe contar con su propio Product Owner, Scrum Master y un Backlog de Acción ejecutiva. El EAT es responsable de la calidad de Scrum en la Entidad, por tanto, todos los Scrum Masters les reporta.

Tabla 22 – Executive Action Team y la Scaled Daily

Producto	Sistema operacional ágil Modelo de referencia de Scrum@Scale Product Backlog con iniciativas concretas para el logro de sus responsabilidades
Integrantes	Profesionales con poder de decisión (delegados y respaldados con autoridad para remover impedimentos de manera inmediata) Coordinación CENIT Scrum of Scrum Masters
Convoca	Scrum Master del EAT
Participantes	Todos los miembros del EAT y los Scrum of Scrum Masters

Fuente: elaboración propia

Producto del EAT:

Un sistema operacional ágil para la Entidad, por lo cual deben crearlo y operarlo a través de iniciativas para esa transformación en un Backlog propio con ese propósito. Debe incluir las mejoras a los procesos que eliminan impedimentos y necesitan ser estandarizadas.

La Scaled Daily no es la única reunión que tiene el EAT, pueden organizar las sesiones que requieran a discreción para el desarrollo de sus responsabilidades, las cuales son:

Tabla 23 – EAT y sus responsabilidades

Categoría	Responsabilidades
Organización	Crear el sistema operacional ágil para la DIAN Aplicarlo al modelo de referencia (primer SoS del escalamiento) Definir reglas organizacionales, procedimientos y lineamientos para posibilitar la agilidad.
Roles	Asegurar la conformación de la una organización consistente de Product Owners, con fondo y respaldo institucional.
Métricas	Medir y mejor la calidad de Scrum en la DIAN
Capacidades	Desarrollar capacidades en la DIAN para conseguir la agilidad de negocio Crear un centro de aprendizaje continuo para profesionales Scrum y la experimentación en nuevas formas de trabajo.

Fuente: Elaboración propia

Componentes del ciclo de Scrum Masters

Los Scrum Masters, Scrum of Scrums Master y el Executive Action Team trabajan conjuntamente como un todo para implementar los siguientes componentes del ciclo:

Tabla 24 – componentes del ciclo Scrum Máster

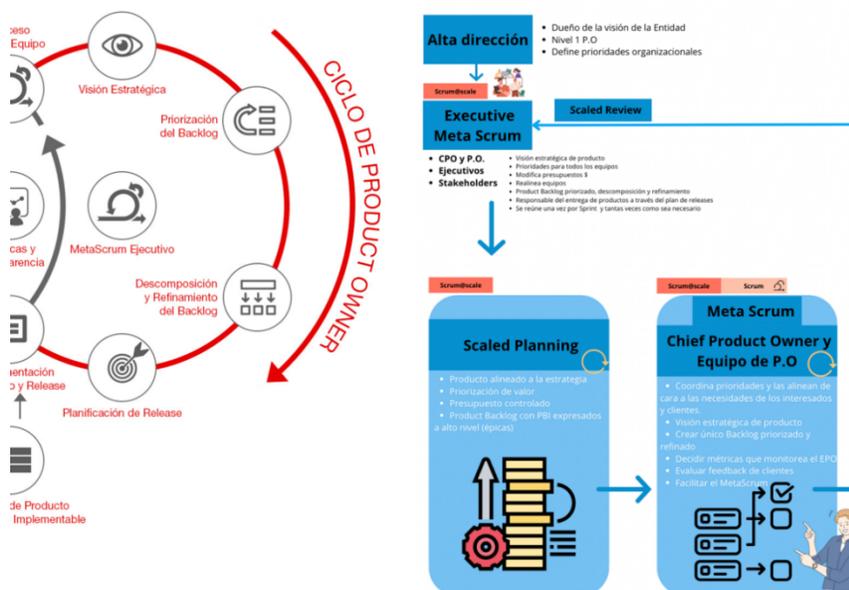
Mejora continua y eliminación de impedimentos	Coordinación entre equipos	Implementación de producto
<ul style="list-style-type: none"> • Identificar impedimentos organizacionales y removerlos • Garantizar la transparencia y visibilidad en la organización para efectuar cambios • Mantener un ambiente efectivo para priorizar • Verificar que las mejoras hayan impactado de manera positiva las métricas del equipo/producto. 	<ul style="list-style-type: none"> • Sincronizar procesos similares en las células relacionadas • Mitigar las dependencias entre células para evitar que se conviertan en impedimentos • Mantener alineadas normas y pautas de las células para lograr resultados consistentes 	<ul style="list-style-type: none"> • Lograr un único release coordinado (Equipo de P.O define contenido y momento óptimo de entrega a los clientes). • Para asegurarlo, los SoS deben: • Entregar un flujo constante de producto valioso terminado • Hacer la integración de los incrementos • Garantizar la alta calidad de la experiencia de cliente

Fuente: Elaboración propia

8.4.2 Ciclo de Product Owner

El ciclo de Product Owners se enfoca en el QUÉ se va a realizar, partiendo de la estrategia de la Entidad que se va atendiendo a través de los productos, todo se va engranando con la visión de los altos directivos y se garantiza con el buen ejercicio de los Chief Product Owners y el Equipo de Product Owners.

Ilustración 15 - Ciclo de Product Owners



Fuente: Guía Scrum@Scale y elaboración propia

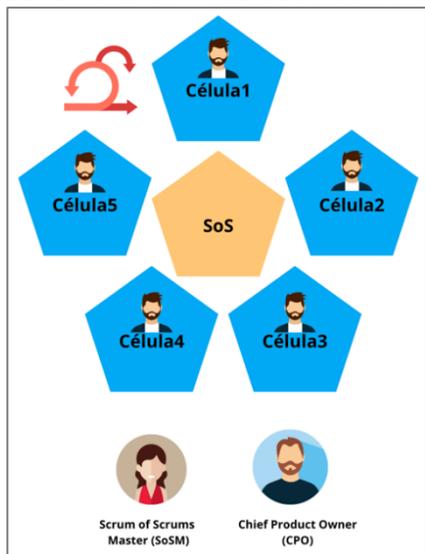
Equipo de Product Owners y Chief Product Owner

En este ciclo se requiere de la participación de los Product Owners de cada célula de trabajo y del rol Chief Product Owner que es el par del Scrum of Scrum Master por cada SoS.

El escalamiento del Equipo de Product Owners va en correspondencia con los Scrum of Scrums asociados.

Tal como se ve en la gráfica a continuación.

Ilustración 16 - Chief Product Owner y equipo de Product Owners



Fuente: Elaboración propia

El CPO es el responsable como Product Owner del conjunto de células. Para ello debe trabajar articuladamente con el Equipo de Product Owners y así enfocarse conjuntamente en que las prioridades estén comprendidas, alineadas con la Estrategia y necesidades de los clientes.

En Scrum@Scale, cada Product Owner de célula mantiene las mismas responsabilidades que en Scrum estándar, sólo que debe tomar las prioridades del Product Backlog compartido o generar elementos independientes según sea necesario para alcanzar los objetivos del negocio.

Este equipo de Product Owners debe trabajar conjuntamente con el CPO, todo esto sucede en un patrón de Scale llamado “Meta Scrum”, para:

8. Comunicar la visión de producto y hacerla visible en la DIAN.
9. Establecer un excelente relacionamiento con los Stakeholders para asegurar su apoyo en la implementación del Backlog.
10. Generar un Backlog único y priorizado que evite la duplicación de trabajo.
11. Trabajar con los Scrum of Scrums Máster para crear la “Definition of Done” que aplique a todas las células.
12. Eliminar dependencias creadas por los equipos.
13. Crear un roadmap y plan de release.

14. Monitorear las métricas sobre el producto y su impacto.

Tabla 25 – Meta Scrum y sus reuniones

Producto	Product Backlog Estratégico (Nivel features)
Convoca	Chief Product Owners con apoyo CenIT
Participantes	Chief Product Owners Equipo de Product Owners
Frecuencia	Al menos una vez por Sprint

Fuente: Elaboración propia

Executive MetaScrum (EMS)

Es un grupo de líderes que imprime dinamismo al escalamiento, establece la visión organizacional y prioridades estratégicas alineando a todas las células con los objetivos comunes.

Este grupo se reúne al menos una vez por Sprint, donde participan el CPO, Equipo de Product Owners, altos directivos y los Stakeholders clave. Estos últimos expresan preferencias y necesidades a los P.O y también se modifican presupuestos y alinean los Scrum Teams para maximizar la entrega de valor.

Componentes del ciclo de Product Owners

Los Product Owners, Chief Product Owners y el Executive MetaScrum trabajan conjuntamente para satisfacer los componentes del ciclo que se citan abajo.

Tabla 26 – Componentes del ciclo del Product Owner

Visión estratégica	Priorización del Backlog
<ul style="list-style-type: none"> Definir una visión estratégica convincente para clientes como para células. Alinear toda la Entidad en un camino futuro compartido. Claridad que permita la creación de Objetivos de Producto concretos. Describir qué hará la Entidad para aprovechar sus activos clave. Demostrar capacidad de responder rápidamente a las condiciones cambiantes del contexto. 	<ul style="list-style-type: none"> Eliminar la competencia entre prioridades para evitar la tracción de equipos en direcciones opuestas. Priorizar el Backlog identificando en orden claro los productos, capacidades y servicios que se entregarán. Reflejar en el orden del Backlog la creación de valor, mitigación de riesgos y dependencias internas. Priorizar las iniciativas del alto nivel antes del refinamiento.
Descomposición y refinamiento	Planificación de Release
<ul style="list-style-type: none"> El Backlog de un CPO contiene PBI de un alcance mayor, es posible que se deban descomponer y comprender mejor. Identificar productos y proyectos complejos y objetivo de producto asociados. 	<ul style="list-style-type: none"> Puede abarcar uno o más entregas al cliente- Pronosticar el plan de entregas de incrementos de productos y funcionalidades. Comunicar las expectativas de entregas a los Stakeholders Comunicar el impacto financiero del plan de entregas.

- Descomponer los productos y proyectos complejos en elementos independientes.
- Asegurarse que todos los PBI del Backlog pueden ser refinados por las células y se entregados en un Sprint.

Fuente: Elaboración propia

8.4.3 Conectar los ciclos

Scrum@Scale está diseñado para escalar la productividad y hacer que la Entidad entregue el doble del valor a la mitad del costo, pero para eso se necesita asegurar que los ciclos funcionen bien y se conecten mejor.

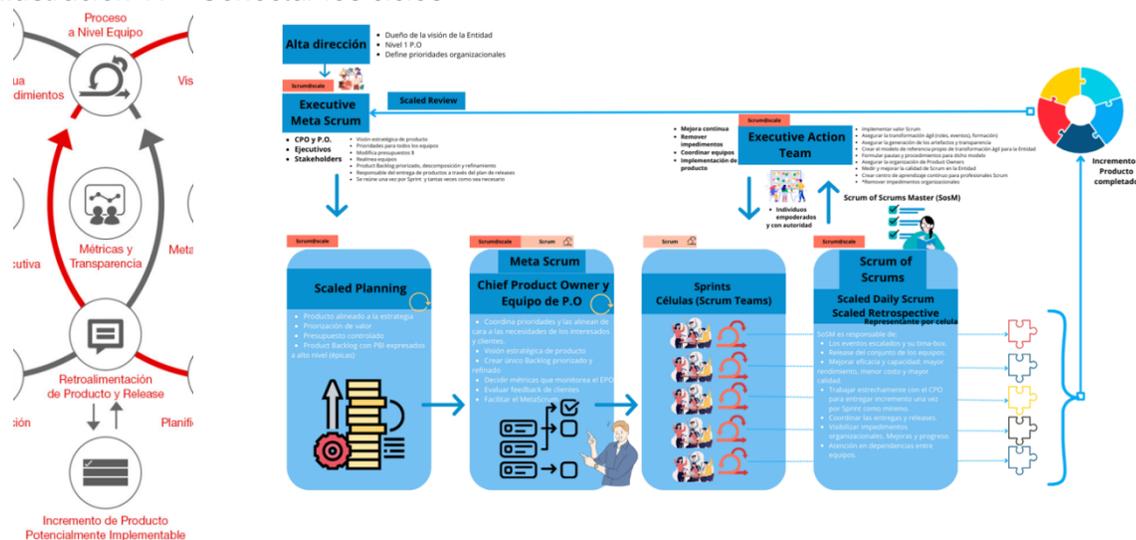
Los ciclos se conectan primero que todo con el “Proceso de equipo”, es decir, con la ejecución de los Sprints que llevan a cabo las células de cada SoS.

Tanto los Scrum o Scrums Masters como los Chief Product Owners asumen sus responsabilidades En el proceso de equipo acerca del Cómo y del Qué.

Luego, los ciclos se vuelven a conectar en el componente “Retroalimentación del Producto y Release” donde se interpreta la respuesta que da el cliente sobre el producto. Su propósito es validar supuestos, comprender cómo los clientes usan e interactúan con el producto, y capturar nuevas ideas y requisitos emergentes para nuevas funcionalidades. (Es una Sprint Review escalada).

Para ello se usan “Métricas” que permitan tomar decisiones basadas en empirismo sobre la adaptación del siguiente ciclo de entrega.

Ilustración 17 - Conectar los ciclos



Fuente: Guía Scrum@Scale

Tabla 27 – Retroalimentación y métricas

Retroalimentación de producto y release	Métricas
<p>La red de Product Owners interpreta la retroalimentación del producto hacer mejora continua que se debe reflejar en el Backlog.</p> <p>La red de Scrum Masters interpretada la retroalimentación de release y asegura la mejora continua de los mecanismos de implementación.</p>	<p>Para la Entidad se ha desarrollado un apartado muy completo de Métricas que se puede ver con detalle en el capítulo Monitoreo y métricas.</p> <p>La transparencia es fundamental y radical en el escalamiento, por tal razón las Métricas deben ser tomadas como objetividad para poder inspeccionar causas de posibles “fracasos” y hacer las adaptaciones para mejorar.</p>

Fuente: Elaboración propia

10. PATRONES

Los patrones se refieren a prácticas, estrategias o soluciones que han demostrado ser efectivas en situaciones específicas dentro del marco de trabajo Scrum. Estos patrones son enseñanzas acumulativas basadas en experiencias pasadas y pueden ayudar a los equipos Scrum a abordar problemas comunes, mejorar la colaboración y optimizar su desempeño.

Tabla 28 – Patrones

Patrón	Descripción
<p>Yesterday's Weather</p>	<p>Es una práctica en Scrum que se basa en la premisa de que la mejor estimación para la duración de un PBI en el futuro es la que se ha observado en el pasado reciente.</p> <p>Este enfoque reconoce que, en muchos casos, las tareas tienden a seguir patrones predecibles basados en experiencias anteriores.</p> <p>Para aplicar el patrón "Yesterday's Weather", el equipo considera las estimaciones y el tiempo real dedicado a PBI similares en sprints anteriores para prever el esfuerzo requerido en el futuro.</p> <p>Al mirar "el clima de ayer", es decir, cómo se realizaron las tareas en sprints previos, el equipo puede ajustar sus expectativas y realizar estimaciones más realistas para mejorar la precisión en la planificación de sprints futuros.</p> <p>Este patrón promueve un enfoque empírico y basado en datos históricos para la estimación y planificación en lugar de depender exclusivamente de predicciones teóricas.</p> <p>Usar la herramienta “Tool-Yesterday’s Wheather.xlsx” Altamente recomendable después del tercer Sprint, para la Planning.</p>

Patrón	Descripción
Interrupt Buffer	<p>Se guarda una parte de la capacidad del equipo para atender los imprevistos. Esto sirve para atender riesgos y cambios. Incluso se puede usar para atender solicitudes de los Interesados.</p> <p>Se lleva el registro de horas de interrupción de los tres sprints anteriores para obtener un promedio y así calcular la cantidad de horas máximas para la "posible interrupción" del siguiente Sprint. Para la entidad se calcula el 20% en los tres primeros Sprints de cada proyecto y luego se continúa con el cálculo de lo sucedido en los últimos tres Sprint.</p> <p>Si los developers detectan la emergencia, informa al Product Owner quien es el único que autoriza usar el Buffer y lo debe incluir en el Product Backlog.</p> <p>Si es obvio que el búfer de interrupción se va a desbordar, el Product Owner debe activar inmediatamente el procedimiento de emergencia de Scrum (Emergency Procedure), abortando el Sprint e incluyendo el asunto a resolver en el siguiente Sprint.</p> <p>Las alertas se van identificando durante la inspección en las Daily.</p>
Emergency Procedure	<p>El equipo analiza su desempeño para arreglar o corregir inmediatamente.</p> <ol style="list-style-type: none"> 1. Si no se llega a cumplir el objetivo del sprint a través del incremento terminado, se debe hacer algo diferente de manera inmediata. 2. Los developers deben hablar con el Product Owner para que se revise, es posible que se reduzca el alcance. 3. Si no se logra se termina anormalmente el Sprint.
Happiness Metric	<p>Es una herramienta utilizada en diversos contextos, incluido el desarrollo ágil y la gestión de equipos, para medir el grado de satisfacción y bienestar de los miembros del equipo.</p> <p>Esta métrica se centra en evaluar el nivel de satisfacción y motivación de los individuos, contribuyendo a la evaluación del entorno de trabajo y la eficacia de las prácticas adoptadas.</p> <p>La Happiness Metric puede abordar aspectos como el equilibrio entre vida laboral y personal, la colaboración efectiva, el reconocimiento y el apoyo entre compañeros de equipo.</p> <p>Al monitorear y analizar esta métrica, las organizaciones pueden obtener una comprensión más profunda del clima laboral y tomar medidas para mejorar la satisfacción, la productividad y la retención del talento.</p>

Patrón	Descripción
	<p>Es fundamental recordar que la felicidad en el trabajo no solo contribuye al bienestar individual, sino que también puede tener un impacto positivo en los resultados y el rendimiento del equipo en su conjunto.</p> <p>Recomendable usar en cada Retrospectiva.</p> <p>Archivo: "Tool Happiness metric.xlsx"</p>
Daily Registro	<p>Para los equipos novatos en Scrum puede resultar un poco difícil llevar a cabo las reuniones Daily, una manera de ayudarlos a entrar en la dinámica de enfocarse en lo más importante es que usen un registro breve que provea una estructura sencilla que guíe a los developers en la preparación de la Daily y la adaptación de su plan.</p> <p>Este registro se puede eliminar luego del primer sprint cuando ya el equipo ha apropiado la dinámica. "Tool Daily Scrum Registro.xlsx"</p>
Swarming	<p>Es una práctica ágil que se refiere a la colaboración intensiva y la concentración del equipo en un único PBI para completarlo de manera eficiente y rápida. En lugar de dividir el trabajo en partes y asignarlo a diferentes miembros del equipo, el enfoque de swarming implica que todos los miembros del equipo se concentren en la misma tarea al mismo tiempo.</p> <p>Esto promueve la comunicación en tiempo real, la resolución rápida de problemas y la maximización de los esfuerzos colectivos. Swarming es particularmente útil cuando se enfrentan desafíos inesperados, se requiere una rápida toma de decisiones o cuando la colaboración intensiva puede acelerar significativamente el progreso.</p> <p>Este enfoque fomenta la flexibilidad y el trabajo en equipo, permitiendo que el equipo se adapte rápidamente a las demandas cambiantes y optimice la eficiencia en la entrega de trabajo.</p>

Fuente: Elaboración propia

11. CENTRO DE RECURSOS Y HERRAMIENTAS

Los Scrum Masters o cualquier interesado en la transformación ágil a través de la gestión de proyectos de tecnología, puede usar el centro de recursos y herramientas para su autoformación y/o para usarlos como material base en su desempeño diario.

Tabla 29 – Centro de recursos y herramientas

ID	Elemento del ecosistema ágil	Material
1	Principios y valores ágiles	Scrum framework.pptx
2	Scrum marco según la guía 2020 Roles Eventos	
3	Product Owner Historias de usuario	Product Owner adaptada.pptx

ID	Elemento del ecosistema ágil	Material
	Product Backlog Refinamiento	
4	Inception - Impact mapping - User story map (Product Backlog y Sprint Backlog)	Scrum Inception.pptx
5	Estimación en tallas de camiseta y puntos de historia.	Scrum Estimación.pptx Estimación camisetas.pptx
6	Eventos	Scrum eventos.pptx
	Planning	Yesterday's Wheather
	Daily	DIAN Daily Scrum registro
	Review	N/A
	Retrospective	"Retrospective" carpeta con varias plantillas Happiness Metric
7	Patrones (buenas prácticas)	Patrones.pptx
8	Dark Scrum (malas prácticas)	Malas prácticas.pptx
9	Herramientas para las buenas prácticas (Patrones)	Tool- DIAN Mapa de PBI Tool- DIAN Yesterday's weather Tool- DIAN Happiness metric Tool- DIAN Daily Scrum Registro
10	SCRUM SCALE	Scrum Scale para la DIAN

12. REFERENCIAS

Schwaber Ken, S. J. (2022). La guía definitiva de Scrum: Las reglas del juego.

Sutherland, J. &. (2021). *Guía Scrum@Scale. La guía definitiva de Scrum@Scale: Escalamiento que funciona.*

13. ANEXOS PARA EL SISTEMA DE GESTIÓN DE CALIDAD

Los siguientes formatos complementan el procedimiento de gestión ágil de proyectos y hacen parte del mismo en el sistema de gestión de calidad de la entidad. Se encuentran en la carpeta anexa a este documento enumerados como se ve a continuación:

FT-IIT-2010 Acta de constitución

FT-IIT-2722 Acta de entrega de proyecto

Anexo 1. Matriz de riesgos proyectos tecnológicos

14. CONTROL DE CAMBIOS

Versión	Vigencia		Descripción de los cambios	Tipo de información
	Desde	Hasta		
1	01/03/2024		Versión inicial.	Esta versión corresponde a Información Pública

Elaboró:	Nancy Vianeth Vera Rodriguez	Jefe de Coordinación	CENIT
	Elaboración técnica		
	Tito Alejandro Menjura	Gestor II	Coordinación de Procesos y Riesgos Operacionales
Elaboración Metodológica	Alfredo A. Ahumada A.	Gestor II	Coordinación de Procesos y Riesgos Operacionales
	Elaboración Metodológica		
Revisó:	Tony Samir Peña Guzmán	Subdirector	Subdirección de Innovación y Proyectos
Aprobó:	Julián David Medina Herrera	Director	Dirección de Gestión de Innovación y Tecnología