

Agile Project Management

Este libro fue diseñado como guía para la gestión de proyectos en entornos ágiles. Se puede usar como base para la integración de prácticas descritas en marcos como Scrum, Kanban o Lean.

- Agilidad aplicada a proyectos
 - Introducción a la gestión de proyectos
- Consideraciones para la gestión de un proyecto ágil
 - 1. Contratación
 - 2. Finanzas
 - 3. Seguimiento y control
 - 4. Riesgos
 - 5. Cambios
 - 6. Diseño del producto
- Etapa 1: Inicio del proyecto
 - 7.1 - Etapa 1: Inicio del proyecto
- Etapa 2: Planificación de los incrementos
 - 7.2 - Etapa 2: Planificación del Sprint
- Etapa 3: Desarrollo de los incrementos
 - 7.3 - Etapa 3: Desarrollo del Sprint
- Etapa 4: Revisión de los incrementos
 - 7.4 - Etapa 4: Revisión del Sprint

- Etapa 5: Entrega de los incrementos
 - 7.5 - Etapa 5: Implementación
- Etapa 6: Cierre del proyecto
 - 7.6 - Etapa 6: Cierre del proyecto

Agilidad aplicada a proyectos

La agilidad está basado en la simplicidad, por lo que NO utilizamos el concepto “proceso” para describir las prácticas que permiten el desarrollo de un proyecto. Los procesos tal como se conocen es mejor dejarlos a las metodologías tradicionales. Se denominan prácticas ya que proponen recomendaciones flexibles que no son de obligatorio cumplimiento y además no cuentan con la estructura tradicional de un proceso (entradas, herramientas, procedimientos, salidas); por lo que resultan en más facilidad a la hora de ponerlas en marcha, esto sin dejar de lado los elementos clave en la gestión de proyectos. Un proyecto desarrollado en el contexto ágil tiene 6 etapas definidas en su ciclo de vida, cada etapa con un respectivo conjunto de prácticas (17 en total).

Introducción a la gestión de proyectos

Las consideraciones para la gestión de proyectos basados en Scrum describen 6 elementos que debe considerar el Equipo Scrum para garantizar que un proyecto entrega los resultados esperados.

Estas consideraciones se relacionan con las prácticas descritas en el Ciclo de vida de un proyecto basado en Scrum.

Las 6 consideraciones, son:

- **Contratación:** Esta consideración toma un enfoque en la contratación del personal, el contrato del proyecto, y la contratación de proveedores que son externos al Equipo Scrum
- **Finanzas:** Esta consideración aborda principalmente la definición del presupuesto del proyecto
- **Seguimiento y control:** Esta consideración aborda el concepto de la APMO (Oficina de Gestión de Proyectos Ágiles) y su rol frente al seguimiento, control y apoyo a los proyectos basados en Scrum
- **Gestión de riesgos:** Esta consideración comprende el tratamiento que se recomienda dar a los riesgos. El ciclo propuesto permite asegurar que durante toda la construcción del producto se ejecute la gestión de riesgos dada la naturaleza iterativa de Scrum.
- **Gestión de cambios:** En esta consideración se define el flujo para el tratamiento de los cambios que puedan surgir durante la construcción de un producto con Scrum. Cabe mencionar que según los postulados del manifiesto ágil, se busca la "Respuesta ante el cambio sobre seguir un plan", por lo que desde Scrum se asegura cumplir con esta premisa.
- **Diseño de producto:** En esta consideración se abordan recomendaciones para el diseño iterativo de producto, manteniendo el enfoque ágil de que no es necesario contar con el diseño detallado de producto para iniciar con su construcción.

Consideraciones para la gestión de un proyecto ágil

Las consideraciones para la gestión de proyectos basados en Scrum describen 6 elementos que debe considerar el Equipo Scrum para garantizar que un proyecto entregue los resultados esperados. Estas consideraciones se relacionan con las prácticas descritas en el Ciclo de vida de un proyecto basado en Scrum. Las 6 consideraciones, son:

1. Contratación: Esta consideración toma un enfoque en la contratación del personal, el contrato del proyecto, y la contratación de proveedores que son externos al Equipo Scrum
2. Finanzas: Esta consideración aborda principalmente la definición del presupuesto del proyecto
3. Seguimiento y control: Esta consideración aborda el concepto de la APMO (Oficina de Gestión de Proyectos Ágiles) y su rol frente al seguimiento, control y apoyo a los proyectos basados en Scrum
4. Gestión de riesgos: Esta consideración comprende el tratamiento que se recomienda dar a los riesgos. El ciclo propuesto permite asegurar que durante toda la construcción del producto se ejecute la gestión de riesgos dada la naturaleza iterativa de Scrum.
5. Gestión de cambios: En esta consideración se define el flujo para el tratamiento de los cambios que puedan surgir durante la construcción de un producto con Scrum. Cabe mencionar que según los postulados del manifiesto ágil, se busca la "Respuesta ante el cambio sobre seguir un plan", por lo que desde Scrum se asegura cumplir con esta premisa.
6. Diseño de producto: En esta consideración se abordan recomendaciones para el diseño iterativo de producto, manteniendo el enfoque ágil de que no es necesario contar con el diseño detallado de producto para iniciar con su construcción.

1. Contratación

1. Contratación

Esta consideración toma un enfoque en la contratación (selección) del personal del equipo, el contrato del proyecto, y la contratación de proveedores que son externos al Equipo Scrum

El primer enfoque que veremos es la contratación (selección) del personal del equipo:

1. Contratación (selección) del personal del equipo

Las personas ocupan un papel clave en la gestión de proyectos ágiles, su participación se da a través del conjunto de responsabilidades que les son asignadas, es decir, su rol en el proyecto. En un proyecto Scrum se consideran 2 categorías de clasificación para los roles:

Roles Comprometidos

Los comprometidos son los roles que **obligatoriamente** se requieren para **construir el producto del proyecto**, por ende, son los responsables del éxito de cada iteración del producto y del proyecto en sí.

Los roles comprometidos son:

- El Product Owner.
- El Scrum Master.
- El Equipo de Desarrollo.

Roles Involucrados

Los involucrados son los roles que **no son obligatoriamente necesarios** durante la ejecución del proyecto Scrum. Ellos pueden interactuar con el equipo Scrum, pero **NO** son responsables del éxito

del proyecto.

Los roles involucrados son:

- Clientes.
- Usuarios.
- Proveedores (Ej: Freelancers; Proveedores de servicios; etc).
- Oficina de Gestión de Proyectos Ágiles (APMO) (Más detalles de la APMO en la consideración 6.3 *Seguimiento y control*).

Criterios para la contratación/Selección del equipo Scrum

A continuación, se describen algunos de los elementos que se deben considerar para la asignación/contratación/selección de las personas que ocuparán los distintos roles en un proyecto Scrum.

Product Owner	Scrum Master	Developers
Experto en Scrum	Experto en Scrum	Conocimiento básico de Scrum
Amplia experiencia y conocimiento del negocio y el cliente	Capacidad para la resolución de problemas	Expertos técnicos
Buen negociador	Habilidades de coordinación	Multifuncionales
Organizado con la captura de información y redacción de Historias de usuario	Líder al servicio del equipo	Proactivos
Experiencia en gestión de proyectos (recomendable)		
Líder al servicio del equipo y el cliente (partes interesadas)		

Líderes al servicio del equipo

Los roles del Scrum Master y el Product Owner se consideran como líderes que están al servicio del equipo, para lo cual deben contar con varias características que les permiten apoyar a los Developers y el equipo en general:

- Capacidad de escucha: Escuchan con atención y son receptivos a lo que se dice y no se dice para comprender y reflexionar sobre la situación.
- Empatía: Aceptan y reconocen las diferentes necesidades de las personas, ya sea que hagan parte del equipo o sean las partes interesadas.

- **Persuasión:** Logran el consenso con el Equipo o con el cliente (según la situación) en la toma de decisiones.

Competencias del equipo: Matriz de competencias

La matriz de competencias permite identificar las competencias necesarias en los miembros que hacen o harán parte de los distintos equipos y de esta manera poder identificar los miembros que necesitarán capacitación adicional en un área o competencia específica, e incluso diseñar una estrategia de tutoría entre los miembros del equipo.

Nota: Esta matriz está basada en el Modelo de desarrollo de habilidades de Dreyfus explicado más adelante.

matriz competencias

Aunque generalmente para la contratación de los miembros del equipo se realiza con base en el conocimiento o habilidades técnicas, en Scrum se reconoce que también es necesario que los miembros del equipo cuenten con otro tipo de habilidades que le permitan desarrollar mejores interacciones, mejor comunicación y mayor cohesión. Las necesidades del equipo Scrum, se toman como criterios adicionales a tener en cuenta para la adecuada contratación de los miembros.

Los elementos necesarios para el buen desempeño de un equipo son:

- **Habilidades técnicas:** Son las habilidades específicas que requieren los miembros que desarrollarán los entregables del proyecto. (Ej, Diseño – Desarrollo de Software – Conocimiento de una herramienta específica, etc)
- **Habilidades esenciales:** También llamadas habilidades blandas, son las habilidades que permiten lograr la empatía y buenas relaciones entre todos los Stakeholders del proyecto.
- **Procesos y prácticas:** Las personas deben conocer los entregables, eventos, o herramientas a usar dentro del proyecto. Scrum, por ejemplo, describe 17 prácticas aplicables a los proyectos.
- **Herramientas:** Para desarrollar los entregables del proyecto siempre será indispensable el uso de herramientas, por ejemplo, herramientas de prototipado, herramientas colaborativas, herramientas de integración, etc.

necesidades equipo

Competencias de los Developers

Se suele pensar que en los equipos multifuncionales no pueden existir roles ni especialistas, siendo esta afirmación totalmente falsa, por lo general un equipo está compuesto por distintos especialistas tales como:

- Analistas.

- Desarrolladores.
- Probadores.
- Integradores.
- Diseñadores.
- Arquitectos.
- Etc.

¿Contratar Novatos o Expertos?

Si el objetivo es terminar el proyecto en poco tiempo y con muy alta calidad deberás contratar expertos técnicos, considerando que esto tendrá un impacto significativo en el costo; por otro lado, se podrían contratar solo novatos, pero el proyecto tardará bastante en desarrollarse debido a la curva de aprendizaje, además la calidad podría verse comprometida.

Realizar una combinación entre novatos y expertos, puede desmotivar a los expertos, aunque también puede darse la situación en la que los novatos crezcan rápido y se logre el equilibrio esperado.

2. Contrato del proyecto

Contrato de precio fijo

- Se define un precio total fijo para todo lo que se desarrolle en el proyecto.
- Mayor riesgo financiero si no se da cumplimiento a todo el proyecto en función del contrato.
- Los entregables son fijos y definidos al inicio del proyecto.
- **No son ideales para el uso de Scrum.**

Contrato de Unión Temporal

- Generalmente se utiliza cuando dos o más socios ejecutan un proyecto.
- El ROI (ingresos o beneficios) será compartido entre todos los socios.
- Es importante definir los % de participación y las responsabilidades en el proyecto.
- **Es ideal centralizar los equipos o dividir el proyecto en componentes.**

Contrato de desarrollo en fases

- Se considera el concepto de **Producto Mínimo Viable**.
- Se “fragmenta” el proyecto en varias fases, donde al final de cada fase se hacen pagos.
- Cada fase genera un conjunto importante de entregables.
- **Útil para proyectos de gran tamaño.**
- Se reduce el riesgo monetario, ya que los despliegues sin éxito no son financiados.

Contrato de entrega incremental

- El cliente/patrocinador puede tomar decisiones sobre el proyecto en cada inspección: Puede aceptar el entregable, Detener el desarrollo o Solicitar modificaciones.
- Cada Sprint debe generar un incremento por lo general es una épica o componente completo.
- El pago/facturación se hace con cada Iteración (se utiliza generalmente en proyectos internos).
- **También se le llama “Contrato por Sprints”.**

Contrato de tiempo y materiales

Este tipo de contrato cuenta con un componente fijo: el precio, el cual puede estar definido por (horas, unidades, metro cuadrado..., etc.) y, un componente variable, el cual se refiere a cantidades, es decir, (número de horas, número de unidades y/o metros cuadrados, etc.) invertidos finalmente en el desarrollo de un trabajo.

3. Contratación de proveedores

Para garantizar que se trabaja con los mejores proveedores, la selección se realiza considerando las opiniones de todo el Equipo Scrum.

Algunos de los criterios que se pueden considerar para la selección de los proveedores son:

- Precio.
- Curva de Aprendizaje.
- Facilidad de uso.
- Velocidad de la Solución.
- Soporte.
- Seguridad.
- Complejidad para la migración de Datos.
- Tiempo de Implementación.
- Reputación.
- Facilidad de Pago.
- Referencias.

Los “Freelancers”

En algunos proyectos, podría ser necesario el apoyo de personal no disponible en el Equipo de Desarrollo, por ejemplo “Arquitectos”, “Diseñadores”, “Administradores de Bases de Datos”, etc.

Este tipo de personal es considerado como proveedor, y normalmente solo se contrata por días, y en momentos muy específicos del proyecto.

freelancer
Image found or type unknown

2. Finanzas

Las finanzas del proyecto son una de las consideraciones más importantes en la gestión de un proyecto (sobre todo en proyectos externos a la organización), ya que, de no realizarse una adecuada gestión financiera, el proyecto podría generar pérdidas, comportamiento no deseados, e incluso la cancelación total de un proyecto.

Las finanzas del proyecto se ven reflejadas principalmente en los siguientes momentos:

- **Inicio del proyecto:** En la definición del presupuesto inicial del proyecto, que permitirá evaluar la viabilidad de iniciar un proyecto, además de asegurar los recursos financieros del proyecto, e incluso identificar posibles riesgos financieros.
- **Ejecución del Proyecto:** Al final de cada Sprint, podría realizarse seguimiento y control financiero del proyecto, para así evitar desviaciones o la ocurrencia de posibles problemas, al ser realizada de forma iterativa, supone menor desgaste para la persona que asuma esta responsabilidad (normalmente es el Product Owner), además de garantizar que siempre se encuentra actualizada la información.
- **Cierre del Proyecto:** Por lo general, las organizaciones solicitan a los equipos de proyecto un reporte final que describa los costos del proyecto, en relación al presupuesto solicitado. En Scrum la función encargada de analizar dicha información se conoce como la APMO (Agile Project Management Office). Además esta información servirá para alimentar el portafolio de proyectos.

finanzas found or type unknown

Presupuesto inicial del proyecto

Una de los momentos clave en las prácticas Scrum es la definición del Presupuesto inicial del proyecto, para lo cual se deben tener en cuenta como mínimo los siguientes elementos:

- Personas que harán parte del proyecto.
- Materiales.
- Servicios.
- Infraestructura.
- Capacitación del equipo.
- Reservas.
- Otros gastos que afecten la ejecución del proyecto.

Nota: Es responsabilidad del Product Owner y el Patrocinador del proyecto (cliente), discutir, negociar y aceptar el presupuesto para asegurar que haya suficientes fondos disponibles para el

proyecto.

Algunas de las técnicas y herramientas que se pueden utilizar para gestionar las finanzas del proyecto son:

Retorno de la Inversión

El ROI (Return Of Investment) es el cálculo de las utilidades (o valor financiero) generado por un proyecto. Esto sirve a muchas organizaciones para determinar si los proyectos externos son exitosos o no.

 Image not found or type unknown

Nota: No todos los proyectos están diseñados para generar utilidades (ganancias), así que, cuando a un proyecto no se le puede calcular el ROI, lo que se mide es el valor entregado (es decir, los beneficios) utilizando VOI (Value of Investment).

Estimación basada en históricos

Los datos históricos del mismo proyecto o de otros proyectos serán de gran utilidad para lograr mejores estimaciones de los elementos que hacen parte del Product Backlog.

Dependiendo de la información disponible en la organización, se puede utilizar:

- **Históricos reales de proyectos anteriores:** Esta técnica funcionará siempre y cuando la organización mantenga el registro del esfuerzo y duración reales del proyecto (la técnica más recomendable en Scrum).
- **Históricos planeados:** En los que se cuenta con un estimado original, aunque no se haya tenido un seguimiento a los datos reales (la técnica más común). Esto es por ejemplo el histórico de cotizaciones de proyectos externos o casos de negocio de proyectos no ejecutados o no aprobados.

Coeficiente de conocimiento del negocio

Esta técnica está basada en la idea de que a mayor conocimiento o experiencia del negocio/proyecto, más precisa será la estimación del presupuesto, y, a menor conocimiento o experiencia del negocio/proyecto, habrá mayor riesgo y menor precisión en la estimación.

El objetivo de esta técnica es calcular en una escala porcentual el grado de conocimiento sobre el negocio y a partir de allí calcular la reserva financiera o presupuestal.

Algunos factores para el cálculo del coeficiente de conocimiento sobre negocio son:

- Experiencia en proyectos similares.

- Calidad de requerimientos iniciales del proyecto.
- Grado de riesgo al que está expuesto el proyecto.
- Desviación (%) en las estimaciones de proyecto.

El seguimiento y control del proyecto, es una actividad que debe hacerse de forma continua a lo largo del ciclo de vida del proyecto. Por lo general los momentos donde se realiza de forma intensiva son: Al inicio, a intervalos predefinidos durante el proyecto o en cualquier momento cuando surgen problemas o riesgos de viabilidad.

3. Seguimiento y control

6.3.1 - Oficina de Gestión de Proyectos Ágiles (APMO)

Una APMO (Agile Project Management Office) está conformada por un Grupo de expertos en Scrum u otros Marcos de Trabajo ágiles.

La Oficina de Gestión de Proyectos Ágiles (APMO) tiene como responsabilidades:

- Estudiar la viabilidad de las iniciativas de proyectos.
- Generar guías de apoyo.
- Brindar asesoría a los diferentes Equipos Scrum de la Organización.
- Dar seguimiento a los proyectos.
- Reportar a la Alta Dirección.
- Mantener actualizado el portafolio de proyectos.
- Definir los “Criterios de terminado” globales para la organización (Ver sección - Definición de terminado (DoD)).
- Por lo general se encarga de guiar las retrospectivas de proyecto.
- Garantizar la correcta gestión de las lecciones aprendidas de los proyectos.
- No toma decisiones sobre los proyectos, pero funciona como apoyo de consultoría, asesoramiento u orientación para todos los proyectos, programas y portafolios de proyectos de la organización.

dibujo

6.3.2 - Informes y reportes del proyecto

En cualquier momento es posible conocer el progreso del proyecto tan solo con sumar el trabajo total restante para alcanzar el objetivo. El Product Owner hace seguimiento de este trabajo restante total al menos en cada Revisión de Sprint. El Product Owner compara esta cantidad con el trabajo restante en Revisiones de Sprint previas, para evaluar el progreso hacia la finalización del

trabajo proyectado en el tiempo deseado para el objetivo. Esta información se muestra de forma transparente a todos los interesados.

Varias prácticas de proyección de tendencias se han utilizado para predecir el progreso, como trabajo pendiente (Burn Down), trabajo completado (Burn Up) y el flujo acumulado (Cumulative Flow). Estas han probado ser útiles, sin embargo, no reemplazan la importancia del empirismo. En entornos complejos se desconoce lo que ocurrirá. Solo lo que ya ha ocurrido puede utilizarse para la toma de decisiones con miras al futuro.

6.3.2.1 - Velocidad del Equipo Scrum

La velocidad del Equipo es la velocidad con la que el equipo puede completar el trabajo en un Sprint. Por lo general se expresa en las mismas unidades que las utilizadas para la estimación, normalmente puntos de historia.

- Todos los equipos tienen diferente velocidad, así hagan parte del mismo proyecto. A continuación, la fórmula para calcular la velocidad de un equipo Scrum:

dibujo

6.3.2.2 – Métricas en proyectos Scrum

Algunas de las métricas que son realmente útiles cuando se trabaja usando Scrum son:

Métrica	¿Qué se mide?	¿Para qué se utiliza?	Responsable	Responsable
Flujo acumulado	Historias de Usuario Completadas Historias de Usuario Pendientes Historias de Usuario En Progreso	<ul style="list-style-type: none">• Monitorear el progreso del equipo.• Evitar la deuda técnica del equipo en cada sprint.	Scrum Master	Sprint
Costos de Proyecto	Costos mensuales del Proyecto	Disminuir el riesgo de desvíos financieros.	Product Owner	Mes
Velocidad de Equipo	Velocidad de Equipo	Encontrar el punto de equilibrio del trabajo al que se puede comprometer el equipo en cada sprint.	Scrum Master	Sprint
Impedimentos	Cantidad de Impedimentos Resueltos	Evitar que el equipo tenga inconvenientes o atrasos en el próximo sprint.	Equipo de Desarrollo / Scrum Master	Sprint

Métrica	¿Qué se mide?	¿Para qué se utiliza?	Responsable	Responsable
Cambios	# Cambios aceptados # Cambios rechazados # Cambios pendientes por revisar	Garantizar la estabilidad de los proyectos y Garantizar que los cambios son atendidos a tiempo.	Product Owner	Semanal
Errores	Errores en pruebas unitarias Errores en pruebas de colegas Errores en producción Errores en Validación	Garantizar la calidad del producto.	Equipo de Desarrollo	Sprint
Riesgos	Riesgos Materializados Riesgos Mitigados Riesgos No tratados	Garantizar la calidad del producto y la estabilidad del proyecto	Product Owner	Sprint

6.3.2 - Seguimiento de Calidad

En Scrum, la calidad se define como la capacidad del producto para cumplir criterios de “terminado” y alcanzar el valor que espera el cliente. En los proyectos Scrum es sumamente importante realizar un seguimiento constante a la calidad del producto y así evitar inconvenientes en el futuro (esta es una práctica iterativa que se realiza en todos los Sprints).

6.3.3.1 - Control de calidad

El Control de Calidad se refiere a la ejecución de las actividades de calidad que se realizan a los incrementos de producto que están potencialmente listos para la entrega y posteriormente sobre el Producto.

Normalmente los controles de calidad son realizados por el Equipo de Desarrollo y el Scrum Master durante la práctica de Desarrollo del Sprint, y por el Product Owner en la Revisión del Sprint.

6.3.3.2 - Aseguramiento de calidad

El Aseguramiento de la Calidad se refiere a la evaluación de los procesos y normas que están definidos en el Ciclo de vida de un proyecto Scrum.

Normalmente el aseguramiento de calidad se hace por medio de auditorías de proceso, realizadas por la APMO (Sección 6.3.1).

Para realizar las actividades de Aseguramiento de calidad, puede usarse el Marco para la Evaluación y Mejora de Procesos de Tecnologías de la Información (MEMPTI).

4. Riesgos

La gestión de riesgos es una actividad que se realiza proactivamente y a través del ciclo de vida del proyecto.

- Un riesgo es definido como una situación inesperada que puede afectar los objetivos de un proyecto.
- Los riesgos pueden tener un impacto tanto positivo como negativo sobre el proyecto.

6.4.1 - Ejemplos de riesgos comunes:

Falta de capacidad del recurso humano actual: En algunas organizaciones, el personal disponible para la ejecución de los proyectos tiene asignadas otras responsabilidades que le impiden tener total disponibilidad para ejecutar las actividades del proyecto.

Algunas de las posibles acciones a tomar son:

- Planificar las actividades del personal actual.
- Evitar en lo posible los cambios urgentes.
- Considerar la contratación de personal o de proveedores externos que puedan aportar a la ejecución de los proyectos.

Falta de seguimiento y control sobre los proyectos: La falta de seguimiento y control del portafolio de proyectos puede provocar su obsolescencia y/o fracaso.

Algunas de las posibles acciones a tomar son:

- Usar una herramienta para el seguimiento y control de los proyectos.
- Realizar una actualización continua del portafolio de proyectos, para confirmar la entrega de valor.
- Monitorear el cronograma y el presupuesto de cada proyecto.

El presupuesto se agota o no es suficiente: Se debe considerar que, para el desarrollo de la mayor parte de los proyectos internos, se necesitan considerables sumas de dinero, que deberán ser proyectadas dentro del presupuesto general de la organización.

Falta de compromiso por parte de los involucrados en los proyectos: Los proyectos Scrum requieren una colaboración entre los miembros del equipo y las partes interesadas, por lo que es importante contar con el compromiso de todos los involucrados.

Para este fin, se recomienda:

- Definir claras y correctas expectativas para todos los involucrados.
- Comunicar a tiempo y con el detalle suficiente los compromisos e importancia de la participación de los involucrados.
- Mantener siempre informados a los involucrados sobre los avances y/o cambios que puedan tener los proyectos (utilizar un lenguaje sencillo que todos los involucrados puedan entender).
- Evitar reuniones que no generen valor o reuniones demasiado extensas que interrumpen significativamente las actividades de los involucrados.
- Comunicar siempre los resultados y beneficios logrados con la ejecución de cada proyecto.
- Agradecer al equipo involucrado y funcionarios su participación y resaltar la importancia de sus ideas y revisiones.

Posible rotación del personal actualmente involucrado en el proyecto: Uno de los mayores riesgos a los que normalmente se encuentra expuesto un proyecto, es la posible renuncia de los miembros del equipo, por lo que se recomienda:

- Realizar transferencia de conocimiento entre los miembros del equipo.
- Constituir una base de conocimiento y lecciones aprendidas.
- Garantizar la documentación continua.

6.4.2 - Apetito de Riesgo

El Apetito de Riesgo es un modelo utilizado para medir la preferencia de las Partes Interesadas por el riesgo o su actitud hacia el riesgo. Esto define el nivel de las Partes Interesadas para aceptar riesgos.

dibujo

6.4.3 - Tolerancia al Riesgo

La tolerancia al riesgo es la cantidad máxima de riesgo que la organización está dispuesta a aceptar para lograr los objetivos del proyecto; la tolerancia al riesgo sirve como una alerta para evitar llegar a la capacidad de riesgo.

Capacidad de Riesgo

La capacidad de riesgo es el nivel de riesgo máximo que se puede permitir antes de que el proyecto se desvíe de tal forma que no entregue valor al cliente. En caso de superar la capacidad de riesgo, por lo general se da por terminado el proyecto (*Sección - Etapa 6: Cierre del proyecto*).

6.4.4 - Ciclo de vida de la gestión de riesgos

La gestión de riesgos se compone de cinco pasos que se enuncian a continuación:

dibujo

6.4.4.1 - Identificación del riesgo (1)

La identificación de riesgos permite conocer los posibles riesgos que pueden afectar el desarrollo del proyecto y de sus respectivos Sprints.

Existen 2 momentos importantes donde se realiza la identificación de riesgos:

- **Al inicio del proyecto:** Se identifican los riesgos globales del proyecto, por ejemplo, riesgos relacionados con el presupuesto, el personal, etc.
- **Los Sprints:** Se identifican los riesgos que pueden afectar el desarrollo de ese Sprint particular, por ejemplo, riesgos del producto. Esta actividad se realiza de forma iterativa durante todo el proyecto principalmente en las reuniones de Planificación de los Sprints.

Solo mirando el proyecto desde diferentes perspectivas y utilizando una variedad de técnicas, se puede hacer la identificación de los posibles riesgos. La técnica más utilizada es la lluvia de ideas.

6.4.4.2 - Evaluación del riesgo (2)

La evaluación de riesgos ayuda a entender el impacto potencial de un riesgo, ¿qué tan probable es que se produzca, y cuándo es posible que el riesgo se materialice? Con ello una decisión podrá ser tomada y determinar si sería buena idea continuar con el Sprint o incluso el Proyecto.

La evaluación de riesgos se hace considerando 3 factores:

- Proximidad = Cantidad de días o fecha estimada en la que se podría presentar el riesgo.
- Probabilidad = Medida porcentual que ayuda a determinar la posibilidad de que el riesgo ocurra.
- Impacto = Mide el daño que ocasiona el riesgo, suele clasificarse numéricamente según las siguientes categorías: Crítico (6), Muy Alto (5), Alto (4), Medio (3), Bajo (2), Muy Bajo

(1)

dibujo

6.4.4.2.1 – Valor Monetario Esperado

Esta técnica se utiliza para calcular el impacto monetario que podrá tener un riesgo, y con esta información realizar reservar monetarias para la prevención y/o mitigación de riesgos.

La técnica considera 2 factores: El impacto monetario del riesgo y la probabilidad de ocurrencia.

dibujo

Ejemplo: Una organización que implementa sistemas de energía solares detecta una posible tormenta tropical que le impediría continuar con la instalación de los paneles previstos para el próximo Sprint. Para calcular el impacto que esto tendría en el proyecto se identifican factores como posibles multas, salario de las personas, etc. Llegando a la conclusión de que para este riesgo se perderían 538 dólares. El siguiente paso es calcular la probabilidad de ocurrencia del riesgo, que según el instituto meteorológico es del 35%. Al aplicar la técnica como se explica, el Valor Monetario de ese riesgo es de 188.3 dólares.

dibujo

6.4.4.3 - Priorización del riesgo (3)

La priorización de riesgos permite establecer un orden para la mitigación de los riesgos. Para realizar la priorización de los riesgos se siguen los siguientes pasos:

- Identificar la fecha aproximada en la que se presentaría el riesgo. (Considerar que los riesgos más próximos deben ser atendidos primero).
- Calcular el factor de exposición (Relación entre probabilidad e impacto) y con este factor, prioriza entre los riesgos más próximos.

dibujo

6.4.4.4 - Mitigación del riesgo (4)

En la etapa de mitigación, el Equipo Scrum determina la acción a tomar con el riesgo. En Scrum existen 3 posibles respuestas a los riesgos:

dibujo

- La respuesta a cada riesgo dependerá de la probabilidad y el impacto del riesgo.

- La naturaleza iterativa de Scrum, con sus ciclos de tiempo de respuesta y retroalimentación rápida permite que las fallas se detecten de forma temprana; por lo tanto, hablando en términos prácticos, tiene una función de mitigación natural construida dentro del sistema.
- Los riesgos pueden ser mitigados mediante la implementación de una serie de respuestas que pueden ser Proactivas/preventivas o reactivas.

6.3.4.4.1 – Mitigación

- La mitigación se enfoca en reducir la probabilidad de ocurrencia o el impacto del riesgo.
- Comprende acciones que se toman por adelantado o acciones Proactivas.
- Se reduce la exposición al riesgo (probabilidad vs impacto) dentro de límites aceptables para el proyecto.

6.4.4.4.2 – Contingencia

- La contingencia se enfoca en definir una respuesta que se utiliza si el riesgo se materializa o se identifican señales de advertencia.
- Comprende acciones Reactivas y acciones de monitoreo en caso de que el riesgo sea inevitable.

6.4.4.5 - Comunicación del riesgo (5)

Las Partes Interesadas deben ser informadas continuamente acerca del estado de los riesgos, incluyendo el impacto potencial de estos riesgos y los planes para responder a cada riesgo.

Por lo general, la comunicación del Riesgo es llevada a cabo por el Scrum Master hacia el Product Owner y por el Product Owner hacia el cliente.

Esta comunicación siempre está en curso y debe ocurrir en paralelo durante los cuatro pasos secuenciales discutidos hasta ahora.

5. Cambios

Ágil implica la apertura al cambio, por lo que es común que todos los proyectos ágiles estén expuestos a cambios, y es de vital importancia que los miembros del equipo estén preparados para enfrentar los cambios en cualquier etapa del proyecto.

Es aún más importante que la organización sea consciente de la exposición a los cambios para crear sinergia con los equipos Scrum y buscar aprovechar los beneficios minimizando el impacto negativo que pudiese resultar del cambio.

Dada la naturaleza iterativa o incremental de Scrum, es posible manejar los cambios sobre el producto de una manera ordenada y cíclica, respondiendo continuamente a lo que espera el cliente; en correspondencia con el manifiesto ágil, *“Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente”*.

Aunque en Scrum hay una fuerte inclinación a no mantener grandes cantidades de información sin valor, una forma apropiada de solicitar los cambios sobre los productos Scrum, es a través del formato para Solicitud de Cambios – RFC (Request For Change), sin embargo esto no implica que necesariamente haya documentación de por medio.

Un buen RFC, debería tener como mínimo los siguientes componentes:

- ¿Quién solicita el cambio? ¿Por qué? ¿Para qué necesitamos hacerlo?
- ¿Para cuándo se necesita?
- ¿Qué tan importante es?
- ¿Qué pasa si no se realizara/aprobara el cambio?
- ¿Quién lo va a ejecutar? ¿Qué necesita? ¿Es un empleado o debemos contratar?

Es importante que detrás de cada cambio solicitado exista un “iniciador” para mantener la traza entre el cambio y saber de dónde proviene. Además dentro de la gestión de cambios en Scrum se reconoce la autoridad del Product Owner en la aprobación/rechazo de los cambios, por lo que es importante que durante todo el proyecto, el Product Owner esté enterado de lo que sucede con el producto.

A continuación, se describe el flujo de los cambios:

dibujo

- El Product Owner es el rol responsable de aprobar/rechazar los cambios, sin embargo, en algunas ocasiones cuando los cambios se salen de su conocimiento o están por fuera del alcance definido para el proyecto podrá escalar los cambios a la gerencia de la

organización.

- Los cambios pueden venir de distintas fuentes, las más comunes son:
 - Cambios solicitados por las partes interesadas.
 - Solicitudes realizadas por los miembros del Equipo.
 - Nuevas tecnologías emergen y es necesario realizar cambios al producto.
 - El producto definido comienza a perder vigencia y es necesario cambiar el alcance.

6. Diseño del producto

Scrum es usado comúnmente para la Gestión de Proyectos ágiles orientados a la construcción de productos/servicios, sin embargo, etapas previas al proyecto como la concepción, prototipado y diseño del producto no siempre se explican al detalle, siendo altamente importantes para el buen desarrollo del proyecto.

Algunos de los conceptos que se deben considerar en esta etapa son:

6.6.1 - Producto mínimo viable

El definir el Producto Mínimo Viable o también llamado Características Mínimas de Mercado es una actividad extremadamente importante, de modo que la primera versión del producto se construye tan pronto como sea posible, lo que lleva a un aumento de rendimiento de la inversión.

Normalmente, estos requerimientos se ubicarían como alta prioridad dentro del Product Backlog.

El Producto Mínimo Viable se define entre el Cliente y el Product Owner.

6.6.2 - Análisis de viabilidad del producto

El objetivo del modelo Lean Canvas (adaptado de Business Model Canvas) es el de identificar la viabilidad de un producto o servicio y así disminuir el riesgo y los posibles obstáculos.

En muchas ocasiones el Lean Canvas se utiliza como el artefacto sustituto del caso de negocio.

Las claves a la hora de construir un Lean Canvas son:

- Crear: Plantear la idea y probar con prototipos de bajo costo.
- Medir: Comprobar el interés de posibles usuarios y medir resultados.
- Aprender: Con los resultados se decide si se continua con la idea o se cambia algo.

El Lean Canvas está compuesto por 9 cuadrantes, tal como los que se describen a continuación:

6.6.3 – Principios para el diseño de un producto

Cuando se realiza el diseño de Productos Innovadores, se podrían considerar los siguientes principios:

1. División: Dividir el producto en partes independientes (permitirá construirlo iterativamente), crear productos modulares o fáciles de desarmar o dar mantenimiento o soporte. **Ej:** Arquitectura de microservicios.

2. Multifuncionalidad: Considera que el producto pueda ejecutar múltiples funciones, o que se elimine la dependencia de otros productos para funcionar.

Ej: Centralización de funcionalidades en una sola aplicación

3. Simplicidad: Reducir el número de acciones que debe realizar el usuario o consumidor del producto para poder operarlo, en función de la experiencia de usuario (UX) **Ej:** Reducir la cantidad de botones de un producto; Reducir la cantidad de clics necesarios para lograr una acción en una aplicación.

4. Personalización: Garantizar que los consumidores o usuarios de un producto, pueden personalizarlo de tal forma que se ajuste a sus necesidades **Ej:** Personalización de colores, tamaños, cantidades, tipografías, imágenes, productos on-demand, etc

5. Curvatura y flexibilidad: En lugar de usar componentes, superficies o formas cuadradas, rectangulares, cúbicas, planas o rígidas, usar las curvas redondeadas, trabajar con estructuras o materiales flexibles esto permitirá un mejor uso del producto, facilidad en las interacciones e incluso aumenta la resistencia del mismo **Ej:** Teléfonos plegables, autos de chasis flexible, tendencia hacia el uso de diseños con formas redondeadas

6. Diseño o apariencia: Construir productos con diseños atractivos, simples y que reflejen la calidad de los materiales o componentes. Además, realizar un cambio de imagen a los productos, refleja a los clientes que existe una innovación constante.

7. Retroalimentación: Permitir la recolección de datos, estadísticas o comentarios del producto para realizar mejora continua. **Ej:** Aplicaciones que recolectan datos de uso o consumo; Productos que permiten enviar comentarios o recibir calificaciones.

8. Autoservicio: Se refiere a que el producto sea útil por sí mismo y no dependa de acciones de terceros para ser utilizado (diagnóstico de fallas, configuración, implementación) **Ej:** Autos modernos con sistema de ayuda para la identificación y prevención de fallas; Aplicaciones en la

nube que no requieren de capacitación previa para ser usadas.

9. Homogeneidad: Si tu organización diseña varios productos o servicios, debe crearse un ecosistema consistente de elementos como interoperabilidad, diseños similares, o productos secundarios. **Ej:** Suites ofimáticas “Google Apps” – “Microsoft Office”; Ecosistema de funciones de Apple, etc.

Estos principios son genéricos y aplicables a cualquier producto o servicio, y harán parte de la práctica **“Definición de la Arquitectura de Producto”**.

Etapa 1: Inicio del proyecto

7.1 - Etapa 1: Inicio del proyecto

Esta etapa del ciclo de vida, está compuesta por 6 prácticas. El objetivo principal de esta etapa es establecer las condiciones básicas de conocimiento y planificación del producto, que permitirán su correcto desarrollo durante la ejecución de los distintos Sprints.

Nota: Las prácticas de esta etapa no suelen ser prácticas iterativas

dibujo

ID	Práctica	Rol vs Nivel de Involucramiento
	Etapa 1: Inicio del proyecto	
1	7.1.1 - Definir la visión del proyecto (1)	Product Owner: Alto - Scrum Master: Nulo - Equipo de desarrollo: Nulo
2	7.1.2 - Formar el Equipo Scrum (2)	Product Owner: Alto - Scrum Master: Alto - Equipo de desarrollo: N/A
3	7.1.3 - Construir el Product Backlog (3)	Product Owner: Alto - Scrum Master: Medio - Equipo de desarrollo: Medio
4	7.1.4 - Priorizar el Backlog (4)	Product Owner: Alto - Scrum Master: Medio - Equipo de desarrollo: Medio
5	7.1.5 - Definir el cronograma de entregas (5)	Product Owner: Alto - Scrum Master: Alto - Equipo de desarrollo: Alto
6	7.1.6 - Definir la Arquitectura de Producto (6)	Product Owner: Medio - Scrum Master: Medio - Equipo de desarrollo: Alto

7.1.1 - Definir la visión del proyecto (1)

En la práctica de “Definir la visión del proyecto” se estructura la visión del proyecto, explicando las necesidades empresariales que el proyecto busca satisfacer, considerando el alcance, el tiempo, el presupuesto y la calidad esperada por el patrocinador del proyecto.

Algunas de las actividades clave dentro de esta práctica son:

7.1.1.1 - Restricciones de un proyecto

Los proyectos siempre tienen limitaciones, también llamadas “restricciones”. Por lo general se contemplan 4 restricciones principales (tiempo, presupuesto, alcance y calidad), sin embargo, según la naturaleza del proyecto y/o el estilo de la organización, podrían existir otras restricciones.

dibujo

7.1.1.2 - Identificar las partes interesadas

Al inicio del proyecto, el Product Owner, se encarga de garantizar que todas las Partes Interesadas son identificadas. Para ello, se recomienda que todas a su vez sean registradas, en lo que comúnmente se llama “Matriz de Partes Interesadas”

Esta matriz contiene la siguiente información:

- Empleados que trabajan para el proyecto, su respectivo rol y el tiempo en horas que dedican al proyecto.
- Proveedores del proyecto.
- Clientes / Usuarios que serán entrevistados, revisarán el proyecto o brindarán información útil para la construcción del producto.

7.1.1.3 - Definición de terminado (DoD)

Cuando un elemento del Product Backlog o del Incremento se describe como “Terminado”, todo el mundo debe entender lo que significa “Terminado” (Done); aunque esto puede variar significativamente para cada Equipo Scrum.

Los miembros del equipo deben tener un entendimiento compartido de lo que significa que el trabajo esté completado para asegurar la transparencia. Esta es la definición de “Terminado” (“Done”) para el Equipo Scrum, y se utiliza para evaluar cuándo se ha completado el trabajo sobre el Incremento de producto.

Esta misma definición guía al Equipo de Desarrollo en saber cuántos elementos del Product Backlog puede seleccionar durante la Planificación del Sprint. El propósito de cada Sprint es entregar Incrementos de funcionalidad que potencialmente se puedan poner en producción y que se ajusten a la Definición de “Terminado” (Done) actual del Equipo Scrum.

Los Equipos de Desarrollo (Development Team) entregan un Incremento de funcionalidad de producto en cada Sprint. Este Incremento es utilizable, de modo que el Product Owner podría elegir liberarlo inmediatamente. Si la definición de “Terminado” (Done) para un incremento es parte de

las convenciones, estándares o guías de la organización de desarrollo, al menos todos los Equipos Scrum (Scrum Team) deben seguirla. Si “Terminado” (“Done”) para un incremento no es una convención de la organización, el Equipo de Desarrollo del Equipo Scrum debe especificar una definición de “Terminado” (Done) apropiada para el producto. Si hay múltiples Equipos Scrum (Scrum Teams) trabajando en la entrega del sistema o producto, los equipos de desarrolladores en todos los Equipos Scrum (Scrum Teams) deben definir en conjunto la definición de “Terminado” (Done).

Cada Incremento se integra con todos los Incrementos anteriores y es probado de manera exhaustiva, asegurando que todos los Incrementos funcionan en conjunto.

A medida que los Equipos Scrum maduran, se espera que su definición de “Terminado” (Done) se amplíe para incluir criterios más rigurosos para una mayor calidad. El uso de los nuevos criterios puede descubrir trabajo por hacer en los incrementos previamente “Terminados” (Done). Cualquier producto o sistema debería tener una definición de “Terminado” (Done) que es un estándar para cualquier trabajo realizado sobre él.

La Definición de Terminado puede ser definida para:

- Un sólo Sprint.
- Un Conjunto de Sprints.
- Un Proyecto.
- Un Programa.
- Toda la Organización.

Un ejemplo genérico de criterios de terminado puede ser:

- Los elementos fueron revisados por otros miembros del equipo (se realizaron las pruebas de colegas).
- Todos los defectos están arreglados.
- Se completaron las pruebas unitarias de la Historia de Usuario.
- Se finalizó toda la documentación técnica y de usuario final.
- Todos los archivos y documentos del proyecto están en el repositorio de la organización.
- Se hizo la demostración exitosa a los Socios y/o representantes de la empresa.
- El cliente dio su visto bueno a los entregables.

7.1.2 - Formar el Equipo Scrum (2)

Durante esta práctica se eligen los miembros que harán parte de los distintos Equipos Scrum.

7.1.2.1 - Plan de colaboración

Este artefacto define cómo las distintas partes interesadas y miembros del equipo de proyecto participan, se comunican y colaboran durante todo el proyecto. También se pueden definir las herramientas o técnicas específicas que se utilizarán para este fin.

Por ejemplo, cuándo y cómo se llevarán a cabo las reuniones, qué tipo de herramientas de comunicación se utilizarán, y quién debe estar involucrado en las diversas reuniones.

En proyectos grandes y complejos, especialmente con equipos distribuidos, este artefacto tiene mucha más formalidad. En otros proyectos pequeños, a veces puede ser simplemente un entendimiento verbal.

Por lo general es un artefacto que se construye en una reunión breve (30 minutos máximo), y se tiene en cuenta la opinión del todo el equipo.

7.1.2.2 - Reunión de inicio de proyecto o Kickoff

En esta reunión participan los responsables de ejecutar el proyecto y el cliente para formalizar el nuevo proyecto y en qué momento inicia. Esta reunión también resulta una herramienta que facilita la comunicación en el proyecto. (Esta reunión la dirige el Product Owner).

Los principales objetivos de la reunión de Kickoff son:

- Presentar los objetivos, beneficios y alcance del proyecto.
- Presentar el equipo de proyecto.
- Obtener el compromiso del equipo y partes interesadas frente al proyecto.
- Presentar el plan de trabajo inicial.
- Presentar la identificación inicial de riesgos.
- Definir la comunicación durante el proyecto.

7.1.2.3 - Modelo de desarrollo de equipos – Dr. Bruce Tuckman

El Dr. Bruce Tuckman definió el modelo para describir el curso que la mayoría de los equipos siguen en su camino hacia un alto rendimiento.

dibujo

- 1. Formación (Dirección) Esta es la etapa en que se conforma el equipo y los miembros del equipo están empezando a conocerse, por lo que inicialmente su comportamiento tiende a ser individualista. Durante esta fase de Formación se

establecen las primeras interacciones entre ellos, por lo que es común que no se presenten conflictos o discusiones.

En esta etapa también se definen las reglas del equipo, su propósito y su identidad (un nombre para el equipo), lo cual hace parte de un buen entorno de trabajo para el equipo.

En esta fase el Scrum Master toma una posición de “director” dado que los miembros del equipo dependen de él para la definición del rumbo del equipo y su orientación.

- 2. Enfrentamiento/Conflicto En esta fase ya se ha generado confianza entre los miembros del equipo, por lo que se presentan diferencias en el equipo y los miembros compiten entre sí, estableciendo o rompiendo relaciones entre ellos. Dado que cada miembro tiene su propia personalidad y estilo de trabajo, será necesario que se alcance un acuerdo de convivencia para evitar la disminución de la motivación.

Aunque la fase hace alusión al enfrentamiento, también es necesario que esos enfrentamientos se resuelvan para asegurar el buen rendimiento del equipo.

En esta fase el Scrum Master toma una posición de “coach”, brindando acompañamiento al equipo y guiando a los miembros para mantener los conflictos bajo control. También debe mantener la calma y dar ejemplo del comportamiento deseado para que los demás miembros del equipo lo imiten y adopten sanamente. El Scrum Master también es responsable de promover actividades que aumenten la motivación del equipo e identificar cuáles son las interacciones que mejoran o deterioran la convivencia en el equipo.

- 3. Normalización En esta fase los conflictos se reducen y se generan acuerdos que mejoran las interacciones entre los miembros del equipo. Durante esta fase los miembros comprenden mejor cuáles son sus responsabilidades, fortalezas y debilidades, lo cual propicia un entorno de mayor confianza y colaboración.

Cada miembro ya es consciente de su lugar en el equipo y es capaz de entender y aceptar el estilo de trabajo de los demás miembros, por lo que los intereses personales pasan a un segundo plano para dar prioridad a los intereses del equipo como un todo.

En esta fase el Scrum Master es un “facilitador”, ya que toma un enfoque hacia la mejora y acondicionamiento del entorno para facilitar el rendimiento del equipo. Dado que en esta fase el Scrum Master conoce más al equipo, puede identificar de qué manera encajan mejor sus miembros y cómo puede aumentar su rendimiento y motivación.

Es común que en esta fase se puedan incorporar nuevos miembros al equipo, por lo que el Scrum Master tiene que velar por mantener la estabilidad del equipo asegurando que los nuevos miembros se integren fácilmente al equipo sin disminuir su rendimiento ni su motivación.

- 4. Desempeño En esta fase se cuenta con un equipo maduro con suficiente confianza, motivación y autonomía por lo que los miembros están en la capacidad de tomar ciertas decisiones sin la presencia de un “jefe”, y se le pueden delegar tareas que

antes eran exclusivas del Scrum Master (como llevar a cabo reuniones diarias, identificar impedimentos, etc.). Además, los miembros del equipo ya son capaces de resolver los desacuerdos o diferencias positivamente y en poco tiempo.

Se debe reconocer que no todos los equipos llegan a esta etapa pues se logra después de enfrentar conflictos y grandes dificultades, convivir y compartir experiencias, lo que implica que muchos equipos no logren completamente la etapa de Normalización y se queden estancados en la etapa de Enfrentamiento.

En esta etapa no se requiere de mucho esfuerzo por parte del Scrum Master, lo que le permite enfocarse mucho más en mantener y mejorar el entorno de trabajo del equipo favoreciendo el alto rendimiento de los miembros.

- 5. Finalización/Disolución Esta etapa comprende dos situaciones: la disolución del equipo o la finalización del equipo. La disolución se presenta cuando eventualmente durante el proyecto, uno o más integrantes se van del equipo, lo que provoca un atraso en el avance que ya se había logrado, conllevando a una disminución de la motivación y del rendimiento, pues el equipo queda resentido tras la pérdida de su estabilidad.

La finalización del equipo se presenta cuando acaba el proyecto y los miembros del equipo pasan a ser parte de otro proyecto o de otra empresa, lo cual implica que, aunque se haya alcanzado exitosamente el objetivo del proyecto, entre los miembros se genera una sensación de “pérdida” pues las interacciones que ya estaban establecidas y que ya funcionaban adecuadamente se terminarán junto con el proyecto. En algunos casos el mismo equipo puede hacer parte de un próximo proyecto, sin embargo, al tratarse de un nuevo proyecto no necesariamente significa que las interacciones permanecerán de la misma manera.

Es necesario que el Scrum Master brinde acompañamiento al equipo, pues ya sea si se trata de la disolución o la finalización, representa un cambio significativo para los miembros del equipo.

7.1.2.4 - Equipos de alto rendimiento

Un equipo de alto rendimiento es aquel que tiene un desempeño notablemente superior al promedio. En Scrum normalmente el rendimiento se asocia a la velocidad con la que el equipo consigue entregar valor. Para lograr equipos de alto rendimiento, se necesitan considerar como mínimo los siguientes factores:

Se debe medir el rendimiento continuamente.

- Se deberían utilizar herramientas para automatizar la mayor cantidad de tareas posibles.
- La visión y reglas deben estar claras desde el inicio del proyecto.
- Se debe garantizar la presencia de elementos que fomenten la creatividad.
- Los objetivos definidos deben suponer un reto para el equipo.
- El equipo debe mantenerse motivado.

7.1.2.5 - Modelo de desarrollo de habilidades de Dreyfus

Cuando se consideran temas como la contratación de personal para un proyecto Scrum, y se cuenta con el tiempo apropiado, con los recursos suficientes y una complejidad aceptable para desarrollar una curva de aprendizaje, o cuando se quiere formar un equipo multidisciplinar con una visión de largo plazo, entonces un modelo de desarrollo de habilidades y competencias es aplicable y rendirá sus frutos en términos de calidad del trabajo, gestión del conocimiento, cumplimiento de tiempos y reducción de costos.

Este modelo permite a las organizaciones identificar jóvenes talento con potencialidad para desarrollar sus habilidades desde un nivel novato a un nivel experto a través de una combinación de aprendizaje y práctica, así como determinar los perfiles necesarios para el desarrollo de un proyecto. El modelo Dreyfus se divide en cinco (5) etapas:

1. Novato: Carece de experiencia, utiliza razonamiento analítico y reglas para determinar una acción (causa-efecto), le es difícil sintetizar o alcanzar un nivel de generalización o priorización de información. Necesita monitoreo y retroalimentación.
2. Principiante: Identifica puntos recurrentes o patrones significativos que componen la situación, o unidades de comprensión, independientes del contexto. Utiliza el razonamiento analítico y el reconocimiento de patrones para resolver problemas, además puede abstraer la información más general de un problema.
3. Competente: Tiene un equilibrio entre el razonamiento clínico metódico y analítico que le permite reconocer e identificar con mayor facilidad patrones y aspectos comunes de un problema. Depende del razonamiento analítico para el abordaje de problemas complejos o poco frecuentes.
4. Profesional: Evalúa situaciones conocidas y las extrapola a otras desconocidas, haciendo frente a la ambigüedad, con visión de conjunto y un reconocimiento aparentemente intuitivo obtenido a través de la experiencia, es capaz de construir perspectivas particulares con base en el conocimiento y la experiencia.
5. Experto: Dicta intuitivamente una acción adecuada, descubre conscientemente las normas o reglas presentes en una situación, está abierto a situaciones inesperadas, va más allá de la imagen global, considerando aspectos culturales y de contextos más amplios a cada situación.

dibujo

7.1.3 - Construir el Product Backlog (3)

Durante esta práctica se construye el Product Backlog (Ver Sección 4.1 - Product Backlog).

Los elementos que hacen parte del Product Backlog son:

- Épicas.
- Historias de Usuario.
- Errores.
- Tareas.
- Pruebas de Concepto.

Algunas de sus características más relevantes son:

- Sus elementos están ordenados según la prioridad.
- Su contenido está directamente relacionado con lo acordado con el patrocinador del proyecto.
- Cada elemento que hace parte del Product Backlog se llama PBI (Product Backlog Ítem).

7.1.3.1 - Clasificando el Product Backlog

Durante la construcción del Product Backlog es importante clasificar los elementos del Product Backlog, para lo cual se podría utilizar el concepto de “épicas”.

Las épicas permiten agrupar los elementos del Product Backlog, permitiendo una mejor navegación por el mismo. Algunas ideas de las que se podrían definir las épicas son:

- Módulos.
- Componentes.
- Hitos.
- Entregables.
- Funcionalidades.

7.1.4 - Priorizar el Backlog (4)

Priorizar los elementos del Product Backlog es una práctica clave para garantizar la entrega de valor al cliente. Algunos de los factores que se deben considerar para la priorización de las Historias de usuario dentro del Product Backlog son:

- Tiempo.
- Esfuerzo.
- Valor Funcional.
- Dependencias.
- Riesgos.
- Opinión de los Stakeholders.

- Opinión del equipo de desarrollo.
- Valor de Mercado (\$).

Nota: La prioridad de los elementos puede cambiar durante la ejecución del proyecto.

7.1.4.1 – Priorización por urgencia

Una de las técnicas de priorización del Product Backlog está determinada por la urgencia y el valor que representan los elementos del Product Backlog para las Partes Interesadas.

Los elementos que aporten más valor al negocio y tengan una mayor urgencia tendrán una mayor prioridad dentro del Product Backlog:

dibujo

7.1.4.2 - Visual Story Mapping

Esta es una técnica para proporcionar un esquema visual del producto y sus componentes claves. El Visual Story Mapping, fue formulado por Jeff Patton en 2005, y es comúnmente utilizado para ilustrar la trayectoria del producto.

Este representa la secuencia de iteraciones de desarrollo de los productos (de izquierda a derecha se organizan los componentes según su prioridad de desarrollo, tal como se ve en la imagen):

dibujo

7.1.5 - Definir el cronograma de entregas (5)

En esta práctica, el Product Owner en compañía de los demás miembros del Equipo Scrum, definen el cronograma de alto nivel del proyecto.

En esta práctica, podrían determinarse la duración estimada de los diferentes Sprints (dado que se ya tienen los elementos del Product Backlog priorizados).

Este cronograma incluye la descripción de alto nivel de las actividades, compromisos, tareas asignados específicamente a los miembros del equipo de desarrollo, los cuales se irán refinando a lo largo del proyecto, en las reuniones de Planificación de cada sprint.

dibujo

Nota: Este cronograma se irá actualizando de manera iterativa durante los Sprints.

7.1.5.1 - Calendario del equipo

El calendario del equipo contiene información sobre la disponibilidad de los miembros del equipo, considerando la información correspondiente a:

- Las vacaciones de los miembros del Equipo Scrum.
- Fechas de licencia.
- Acontecimientos importantes en la organización.
- Días festivos.
- Eventos de la oficina.
- Proyectos en curso.
- Etc.

7.1.6 - Definir la Arquitectura de Producto (6)

El objetivo de esta práctica es establecer o refinar el diseño técnico del producto o del componente de producto a desarrollar. Aunque esta actividad es obligatoria siempre al inicio del proyecto, podría hacerse también de forma iterativa antes de desarrollar componentes en los distintos Sprint.

Para garantizar que la arquitectura de producto que se defina cumple exactamente con los requerimientos del cliente se debe hacer en conjunto entre el equipo de desarrollo y el Product Owner.

Antes de iniciar la construcción del producto es importante identificar la relación entre todos los componentes que harán parte del mismo.

Ejemplo: Una empresa de desarrollo de software planea construir una aplicación para realizar cursos virtuales. Al ser un proyecto de software, en la definición de arquitectura se deberán tener en cuenta los siguientes elementos:

- Bases de datos.
- Servidores de almacenamiento.
- Servidores de procesamiento.
- Nombres de dominio.
- Servidores de procesamiento de correo electrónico.
- Firewall.
- Web services.

- Entre otros.

7.1.6.1 - Sprint Cero

Esta iteración es la primera en realizarse. El objetivo del Sprint Cero es preparar el equipo de proyecto desde una perspectiva tecnológica, metodológica y organizativa, buscando conformar un equipo y no simplemente un grupo de personas.

En esta actividad se analizan y evalúan las posibles soluciones que se pueden establecer para el desarrollo del proyecto.

Estas posibles soluciones pueden ser:

- Herramientas.
- Soluciones Tecnológicas.
- Soluciones técnicas.

Por cada solución escogida se deberá realizar una prueba de concepto, en donde se deben tener en cuenta los siguientes criterios:

- Curva de aprendizaje.
- Escalabilidad (Solo si es solución técnica o tecnológica).
- Soporte.
- Costo. **Nota:** Este sprint solo se utiliza para los casos en que el equipo de proyecto tiene poca o nula experiencia en la tecnología en la cual se va a construir el producto o incluso si cuenta con poca o nula experiencia en el tipo de producto que se va a desarrollar.

7.1.6.2 - Evaluación y Selección de Proveedores

El propósito de esta actividad es garantizar la selección objetiva de los proveedores, utilizando los criterios definidos por la organización. (Ver más información en la sección - Contratación de Proveedores).

Etapa 2: Planificación de los incrementos

7.2 - Etapa 2: Planificación del Sprint

Durante esta etapa del ciclo de vida, se realiza la planificación de cada uno de los Sprints, está compuesta por 3 prácticas, 2 de las cuales (7 y 8) pueden ser ejecutadas bajo el concepto “Planificación en paralelo”

Nota: Las 3 prácticas de esta fase son iterativas

dibujo

ID	Práctica	Rol vs Nivel de Involucramiento
	Etapa 2: planificación	
7	7.2.1 - Escribir las historias de usuario y tareas (7)	Product Owner: Alto - Scrum Master: Medio • Equipo de desarrollo: Alto
8	7.2.2 - Priorizar las Historias de Usuario (8)	Product Owner: Alto - Scrum Master: Medio - Equipo de desarrollo: Alto
9	7.2.3 - Reunión de Planificación del Sprint (9)	Product Owner: Alto - Scrum Master: Alto - Equipo de desarrollo: Alto

7.2.1 - Escribir las historias de usuario y tareas (7)

7.2.1.1 - ¿Qué son las Historias de Usuario?

Las historias de usuario son una forma más entendible y precisa para la escritura de los requerimientos a desarrollar en el proyecto.

Están compuestas por 3 elementos principalmente:

- **Como:** Describe el Rol de la persona o grupo que solicita (o usaría) la funcionalidad o requerimiento.
- **Quiero:** Describe la necesidad o requerimiento del usuario, por lo general, es una frase corta.
- **Para:** Describe el beneficio esperado por el usuario una vez se desarrolle el requerimiento.

Por lo general, las historias de usuario se complementan de otros aspectos que ayudan al equipo a entenderlas mejor, algunos de estos pueden ser:

- Criterios de aceptación
- Criterios de prueba
- Prototipos

Desarrollando mejores historias de usuario

Bill Wake inventó el acrónimo INVEST para describir las características de una buena historia de usuario:

- **Independiente:** Las historias pueden completarse en cualquier orden.
- **Negociable:** Los detalles de la historia son co-creados por los desarrolladores y los clientes durante el desarrollo.
- **Valiosa:** La funcionalidad es valiosa para los clientes o los usuarios del producto.
- **Estimable:** Los desarrolladores pueden encontrar una estimación razonable para construir la historia.
- **Pequeña (Small):** Las historias deberían construirse en poco tiempo, generalmente alrededor de "días/persona". Se tiene que poder construir muchas historias en una iteración.
- **Probable (Testeable):** Se debe poder escribir pruebas que verifiquen que el producto de la historia funcione adecuadamente.

Criterios de aceptación para las historias de usuario

Los criterios de aceptación describen con mayor detalle el trabajo que debe ser completado en cada historia de usuario. En la mayoría de ocasiones, estos criterios describen a nivel técnico el trabajo a desarrollar. Algunas características de estos criterios son:

- Los criterios de aceptación son únicos para cada historia de usuario o tarea a completar
- Estos criterios suelen ser definidos por el Product Owner
- Estos criterios permiten saber si esa historia de usuario es funcional y cumple las necesidades, expectativas y/o requerimientos de los usuarios y/o clientes.

7.2.1.2 - Mockups y prototipos iniciales de proyecto

Los Mockups representan el diseño del producto antes de su desarrollo, consideran el flujo que debe seguir el producto, y sirven para mostrar las posibles funcionalidades al cliente y así confirmar que estas entregarán el valor esperado.

Los mockups normalmente son un conjunto de bocetos, diseños, diagramas y/o representaciones; y es especialmente necesario contar con un mockup o prototipo previo al Sprint, pues éstos servirán de guía a los miembros del equipo.

dibujo

7.2.1.3 - Planificación en paralelo

Una actividad que puede representar una gran diferencia es la planificación en paralelo. Esta actividad consiste en formar un pequeño equipo de personas que se encargan de realizar las actividades de planificación paralelamente al desarrollo de los elementos realizados. Con esto se logrará que las reuniones de planificación del Sprint sean muchísimo más eficientes. Algunas de las cosas que se puedan hacer en paralelo al desarrollo son:

- Realizar las entrevistas o investigaciones necesarias para escribir las historias de usuario.
- Realizar los prototipos y confirmar el valor con el usuario.

7.2.1.4 - Desglosar las Historias de Usuario en tareas

Esta actividad sirve para identificar las posibles dependencias entre las historias de usuario, para ello, se elabora un listado con todas las tareas que se deben llevar a cabo para completar la Historia de Usuario.

Normalmente se distinguen 4 tipos de dependencias entre historias de usuario:

- Dependencias obligatorias.
- Dependencias discrecionales.
- Dependencias externas.
- Dependencias internas. Algunas técnicas que pueden ser usadas para desglosar las Historias de Usuario en tareas son:
 - Los Diagramas de Flujo.
 - Los Diagramas de Gantt.

7.2.2 - Priorizar las Historias de Usuario (8)

Esta práctica está relacionada con la planificación en paralelo, debido a que mientras el equipo termina de desarrollar lo establecido para el último Sprint, el Product Owner realiza una priorización de las historias de usuario que pueden considerarse para el siguiente Sprint y que se pueden tomar como base para guiar el rumbo del equipo.

Es importante que esta priorización se realice antes de que se lleve a cabo la Reunión de Planificación del Sprint con el fin de optimizar el tiempo.

Los elementos que se deben considerar para la priorización de las historias de usuario son:

- Valor para el negocio.
- Riesgo.
- Dependencia.

7.2.3 - Reunión de Planificación del Sprint (9)

7.2.3.1 - Seleccionar el trabajo a desarrollar

El Product Owner expone el objetivo que el Sprint debería lograr y los elementos del Product Backlog que, si se completan en el Sprint, lograrían el objetivo del Sprint, con esto el Equipo de Desarrollo trabaja para proyectar la funcionalidad que se desarrollará durante el Sprint.

La entrada a esta reunión está constituida por el Product Backlog y tomando la priorización de historias de usuario realizada por el Product Owner, incluyendo además:

- El último incremento de producto.
- La capacidad proyectada del Equipo de Desarrollo para el Sprint.
- La velocidad del Equipo de Desarrollo.
- Fallos encontrados al producto o incrementos de producto (registro de errores).

El número de elementos del Product Backlog seleccionados para el Sprint depende únicamente del Equipo de Desarrollo. Solo el Equipo de Desarrollo puede evaluar qué es capaz de lograr durante el Sprint que comienza.

Durante la Planificación del Sprint el Equipo Scrum define el objetivo del Sprint. El objetivo del Sprint debería lograrse durante el Sprint a través de la implementación del Product Backlog y proporciona una guía al Equipo de Desarrollo del por qué se está construyendo el incremento.

7.2.3.2 - Estimación del trabajo seleccionado

Es muy importante realizar la estimación del trabajo seleccionado (comúnmente escrito en forma de historias de usuario) para poder hacer planificaciones más precisas.

Para garantizar una estimación más precisa, se deberían considerar criterios como:

- Tamaño.
- Complejidad.
- Duración.
- Cantidad de recursos.
- Riesgos.
- Limitaciones.

Normalmente las técnicas de estimación usadas por Scrum están basadas en el juicio de expertos, sin embargo existen otras técnicas que se pueden utilizar y combinar:

Planning Poker

El póker de planificación consiste en un conjunto de tarjetas numéricas que sirven para realizar la estimación de tareas o historias de usuario.

- El póker puede ir numerado de 1 a 10 o utilizando la sucesión de Fibonacci.
- El uso del póker promueve una mayor interacción y una mejor comunicación entre los participantes, además de hacer más dinámica la reunión de planificación del Sprint.

Fist of Five

Es una técnica que permite lograr el consenso en los Equipos Scrum. Se hace usando los dedos de la mano, donde el número de dedos que se utiliza para la votación indica el nivel de acuerdo y el deseo para el debate:

- Un dedo: no estoy de acuerdo con la conclusión del grupo y tienen grandes preocupaciones.

- Dos dedos: no estoy de acuerdo con la conclusión del grupo y me gustaría hablar de algunos problemas menores.
- Tres dedos: no estoy seguro y me gustaría asumir la conclusión de consenso del grupo.
- Cuatro dedos: Estoy de acuerdo con la conclusión del grupo, pero me gustaría discutir algunos problemas menores.
- Cinco dedos: Estoy totalmente de acuerdo con la conclusión del grupo.

7.2.3.3 - Compromiso del equipo

Una de las actividades clave durante la Planificación de un Sprint es la apropiación de las Historias de Usuario y tareas por parte del equipo de desarrollo, esto se logra cuando los miembros del equipo se auto-asignan el trabajo a desarrollar, a su vez que adquieren el compromiso público de terminar dicho trabajo.

Es sumamente importante garantizar que existe equilibrio en la cantidad de trabajo seleccionado por cada uno de los integrantes del equipo.

Suele suceder que en equipos Scrum donde se encuentran combinados miembros expertos y miembros novatos existirá un “desequilibrio” en etapas tempranas del proyecto, pues los miembros expertos contarán con más capacidad de trabajo mientras los miembros novatos se nivelan en su curva de aprendizaje.

7.2.3.4 - ¿Cómo se conseguirá “terminar” el trabajo seleccionado?

Una vez que se ha establecido el objetivo y se han seleccionado los elementos del Product Backlog para el Sprint, el Equipo de Desarrollo decide cómo construirá esta funcionalidad para formar un Incremento de producto “Terminado” durante el Sprint. Los elementos del Product Backlog seleccionados para este Sprint, junto con el plan para terminarlos, recibe el nombre de Sprint Backlog.

Durante la Planificación del Sprint se debe asegurar que se planifica el suficiente trabajo como para que el Equipo de Desarrollo pueda hacer una proyección de lo que cree que puede completar en el Sprint que comienza.

Para el final de esta reunión, el trabajo planificado por el Equipo de Desarrollo para los primeros días del Sprint es descompuesto en unidades de un día o menos. El Equipo de Desarrollo se autoorganiza para asumir el trabajo del Sprint Backlog, tanto durante la Planificación del Sprint como a lo largo del Sprint.

El Product Owner puede ayudar a clarificar los elementos del Product Backlog seleccionados y hacer concesiones. Si el Equipo de Desarrollo determina que tiene demasiado trabajo o que no tiene suficiente trabajo, podría renegociar los elementos del Product Backlog seleccionados por el

Product Owner. El Equipo de Desarrollo podría también invitar a otras personas a que asistan para proporcionar asesoría técnica o relacionada con el dominio.

7.2.3.4.1 - Objetivo del Sprint

El objetivo del Sprint es una meta que se establece para cada Sprint durante la Reunión de Planificación del Sprint. Dicha meta describe el alcance de cada Sprint en alineación con el Product Backlog, a su vez proporciona una guía al Equipo de Desarrollo acerca de por qué está construyendo el incremento.

Definir el objetivo del Sprint brinda al Equipo de Desarrollo cierta flexibilidad con respecto a la funcionalidad que ha de ser construida en el Sprint.

- Los elementos seleccionados del Product Backlog ofrecen una orientación sobre lo que puede ser el objetivo del Sprint.
- El objetivo del Sprint garantiza que el Equipo de Desarrollo trabaje en conjunto y no en iniciativas separadas.
- Por lo general el objetivo del Sprint suele ser una frase corta que explica de forma simple el incremento esperado al final del Sprint. A medida que el Equipo de Desarrollo trabaja mantiene el objetivo del Sprint en mente. Si el trabajo resulta ser diferente de lo esperado, los miembros del equipo escalan la situación con el Scrum Master para contactar al Product Owner y se pueda negociar el alcance del Sprint.

7.2.3.4.2 - Spikes

Un Spike sirve para incluir en un sprint tareas que NO implican desarrollar una historia de usuario y por tanto NO aportan directamente al incremento de producto que se está desarrollando.

Algunos ejemplos de Spikes son:

- Capacitación del Equipo.
- Documentación del proyecto y/o producto (ejemplo: Documentar el código fuente).
- Despliegues / Implementaciones.
- Adopción de nuevas herramientas.

Etapa 3: Desarrollo de los incrementos

7.3 - Etapa 3: Desarrollo del Sprint

Esta etapa contiene 2 prácticas que se realizan durante el desarrollo de los incrementos del producto. Ambas prácticas son iterativas y se realizan en todos los Sprints del proyecto.

Como se aprecia en el gráfico a continuación, el Scrum Diario es una práctica que se realiza en forma paralela al desarrollo de los entregables del Sprint.

dibujo

ID	Práctica	Rol vs Nivel de Involucramiento
	Etapa 3: desarrollo del sprint	
10	7.3.1 - Desarrollar los entregables (10)	Product Owner: Bajo - Scrum Master: Medio - Equipo de desarrollo: Alto
11	7.3.2 - Scrum diario (11)	Product Owner: Nulo - Scrum Master: Alto - Equipo de desarrollo: Alto

7.3.1 - Desarrollar los entregables (10)

El objetivo de esta actividad es el desarrollo de los entregables del Sprint, los manuales o documentación relacionada considerando siempre la definición de “terminado”.

Nota: En conjunto con esta práctica, se ejecutan los Scrums Diarios, y Las pruebas de los entregables.

7.3.1.1 – El ciclo de desarrollo para proyectos de software

Para el desarrollo de proyectos de software, normalmente se sigue el flujo que se muestra en el gráfico a continuación:

dibujo

7.3.1.1.1 – Las pruebas

Las Pruebas son una forma de comprobar el correcto funcionamiento del producto. Por lo general existen varios tipos de pruebas, siendo los principales:

- Pruebas unitarias: Son escritas por el mismo desarrollador siguiendo los criterios de aceptación de la historia de usuario. Estas pruebas suelen realizarse de manera automatizada.
- Pruebas de colegas: Suelen ser manuales, en las cuales el colega revisan no solo el funcionamiento, sino la calidad del código, y compara el resultado vs prototipos.
- Pruebas de integración: Suelen ser automatizadas. Éstas se ejecutan en cada integración y antes de cada entrega.

También se distinguen 2 tipos de ejecución de las pruebas:

- Pruebas manuales.
- Pruebas automatizadas.

Las 2 posibles salidas de una prueba son:

- Aprobada: Se cumple con todas las expectativas y/o criterios de aceptación de las historias de usuario que se están probando.
- Rechazada: NO se cumple con todas las expectativas y/o criterios de aceptación de las historias de usuario que se están probando. Cuando se rechaza una historia de usuario durante las pruebas, se debe reportar el error en un “Log de Errores”, normalmente dentro del Product Backlog.

Reporte de errores Es importante tener en cuenta que los errores deben pasar por un proceso de estimación igual que las Historias de Usuario (se planifica su resolución en los diferentes Sprints del proyecto).

Cuando se reporten los errores, se debe tener en cuenta:

- Evitar escribir declaraciones vagas o suposiciones no probadas.
- Poner títulos genéricos a los errores.
- Se deberían categorizar los errores, ej: Errores en producción, Errores de colegas, Errores en la revisión, etc.
- Deberían describirse como pasos de reproducción sencillos y repetibles para que la persona que corregirá el error pueda seguir la secuencia.

7.3.1.1.2 – Integración continua

La Integración Continua es una práctica de desarrollo de software mediante la cual los desarrolladores combinan los cambios en el código en un repositorio central de forma periódica, tras lo cual se ejecutan versiones y pruebas automáticas para asegurar la integridad del código.

Para facilitar la integración continua:

- Se debe tener un repositorio único para el código fuente.
- Se debe garantizar que el código se actualiza constantemente tanto en el repositorio, como en el ambiente de trabajo de cada desarrollador.
- Siempre ejecutar pruebas automatizadas para garantizar que cuando se mezcla el código, no se pierden o dañan las funcionalidades existentes.

7.3.1.1.3 – Documentación

La documentación es un aspecto vital en los proyectos de software, esto permitirá el posterior entendimiento del código fuente y así garantizar la propiedad colectiva del producto. Para que la documentación sea efectiva, se debe garantizar que:

- Sea fácil de redactar y entender.
- Debe estar compartida con todos.
- Se actualiza diariamente.
- Debe tener un índice o búsqueda.
- Debe estar organizada por módulos o componentes.
- Debe ser alimentada por todo el equipo.

7.3.1.1.4 – Refactorización

El objetivo de esta técnica es mejorar el mantenimiento del código existente y hacerlo más simple, más conciso y más flexible. Refactorizar significa mejorar el diseño del código actual, sin cambiar el comportamiento del código. Algunas de las ventajas de la refactorización son:

- Mejorar la facilidad de comprensión del código.
- Mejorar su estructura y diseño.
- Eliminar código muerto.
- Facilitar el mantenimiento en el futuro.

Algunos de los momentos clave para realizar la refactorización, son:

- Después de pasar una prueba automatizada.
- Cuando agregar una nueva característica es bastante complicado.
- Cuando mejora el conocimiento o experiencia del equipo de desarrollo.
- Cuando el equipo dedica bastante tiempo a entender el detalle del código.

Algunos ejemplos de refactorización son:

- Organizar el repositorio de documentos/archivos.

- Separación de funciones.
- Renombrar variables.
- Simplificación de las interfaces.

7.3.2 - Scrum diario (11)

Para más detalles ver la Sección 3.3 - Daily Scrum (Scrum Diario)

7.3.2.1 - Seguimiento del Progreso del Sprint

En cualquier momento durante un Sprint es posible conocer el progreso del Sprint sumando el trabajo restante total en los elementos del Sprint Backlog. El Equipo de Desarrollo hace seguimiento de este trabajo restante total al menos en cada Scrum Diario para proyectar la posibilidad de conseguir el objetivo del Sprint. Haciendo seguimiento del trabajo restante a lo largo del Sprint el Equipo de Desarrollo puede gestionar su progreso.

Etapa 4: Revisión de los incrementos

7.4 - Etapa 4: Revisión del Sprint

El objetivo de esta etapa es recolectar retroalimentación de cada uno de los Sprints, y así, permitir la adaptación y mejora continua tanto del producto como de las prácticas ejecutadas por el Equipo Scrum. Está compuesta por 2 prácticas que son iterativas y deberán ejecutarse en cada Sprint.

Durante esta etapa del ciclo de vida, también se realiza el monitoreo y control del proyecto, dando paso a: Gestión de Cambios, Gestión de Riesgos y la Gestión Financiera.

dibujo

ID	Práctica	Rol vs Nivel de Involucramiento
	Etapa 4: Revisión del Sprint	
12	7.4.1 - Reunión de Revisión del Sprint (12)	Product Owner: Alto - Scrum Master: Alto - Equipo de desarrollo: Alto
13	7.4.2 - Reunión de Retrospectiva del Sprint (13)	Product Owner: Nulo - Scrum Master: Alto - Equipo de desarrollo: Alto

7.4.1 - Reunión de Revisión del Sprint (12)

En esta reunión de revisión se realiza la demostración del incremento de producto al Product Owner y partes interesadas invitadas, con el fin de obtener aprobación sobre lo que se esperaba y lo que fue desarrollado por el equipo.

A continuación, se listan las actividades que se realizan durante la Reunión de Revisión del Sprint:

- Presentar el incremento de producto: El equipo de desarrollo presenta al Product Owner y al Scrum Master el incremento del producto desarrollado.
- Probar el incremento de producto (Validación): El Product Owner y las partes interesadas invitadas (solo si las hay) prueban el incremento de producto con base en la comparación del desarrollo frente a cada historia de usuario y sus criterios de aceptación. (Es válido

que se cuente con una lista de chequeo para llevar el registro de lo que se cumple y lo que no se cumple).

- Aprobación del incremento de producto: Con base en el incremento presentado y la validación el Product Owner da la aprobación del incremento de producto. En caso que el incremento no cumpla con lo esperado puede darse el Rechazo del incremento.
- Monitoreo, control y seguimiento del proyecto: Ver más en Sección 6.2 Seguimiento y control del proyecto.
- Ritmo/velocidad del equipo: Se identifica el ritmo/velocidad del Equipo de Desarrollo.
- Compromisos del equipo: Se identifican compromisos por parte del equipo en caso del rechazo del incremento de producto o de historias de usuario puntuales.
 - En caso de rechazo se agregan las mejoras al Product Backlog con la prioridad más alta posible, con el fin de corregirlas en el próximo sprint de ser posible.
- Actualizar el Product Backlog: Se debe actualizar el Product Backlog y dar por terminado el sprint actual.

¿Cómo se realizan las validaciones? En esta reunión se presenta el incremento de producto ante los interesados, los cuales se cerciorarán que cada una de las funcionalidades establecidas y solicitadas en el Sprint estén terminadas (bajo los lineamientos de los criterios de aceptación y los criterios de “terminado”).

7.4.1.1 - Deuda técnica

La deuda técnica se refiere al trabajo que los equipos omiten o no se completa durante uno o varios Sprints.

- Si la deuda técnica se acumula, puede conllevar a un mantenimiento, integración y costos elevados en el despliegue del producto.
- La priorización frecuente de Historias de Usuario contribuye a disminuir la deuda técnica.
- Cualquier deuda técnica no debería llevarse más allá de un sprint.

Algunas de las causas de la deuda técnica son:

- Evaluación inadecuada o incompleta de Historias de Usuario
- Falta de coordinación entre los miembros del equipo
- Intercambio deficiente del conocimiento empresarial

7.4.1.2 - Refinamiento del Product Backlog

El refinamiento del Product Backlog es el acto de añadir detalle, estimaciones y orden a los elementos del Product Backlog; se trata de una actividad continua en la cual el Product Owner y el Equipo de Desarrollo examinan, revisan y detallan los elementos del Product Backlog.

- Los elementos del Product Backlog pueden actualizarse en cualquier momento por El Product Owner o a criterio suyo.
- El refinamiento usualmente consume no más del 10% del tiempo de la Reunión de Revisión del Sprint.

7.4.2 - Reunión de Retrospectiva del Sprint (13)

La Retrospectiva del Sprint es una oportunidad para el Equipo Scrum de inspeccionarse a sí mismo y de crear un plan de mejoras que sean abordadas durante el siguiente Sprint. La Retrospectiva debería realizarse en un ambiente liberador que permita al equipo fluir todo tipo de ideas.

La Retrospectiva del Sprint tiene lugar después de la Revisión de Sprint (Sprint Review) y antes de la siguiente Planificación de Sprint. Se trata de una reunión restringida a lo máximo de 4 horas para Sprints de un mes.

El Scrum Master enseña a todos a mantener el evento positivo y productivo; y participa de esta reunión como un miembro del equipo ya que parte de la responsabilidad de los incrementos recae sobre él.

El Scrum Master motiva al equipo para que mejore su proceso de desarrollo y sus prácticas para hacerlos más efectivos y amenos para el siguiente Sprint. Durante cada Retrospectiva del Sprint el Equipo Scrum identifica y planifica formas de mejorar la calidad del producto mediante el mejoramiento de la calidad de las prácticas, que pueden ser implementadas en el próximo Sprint.

El hecho de implementar estas mejoras en el siguiente Sprint constituye la adaptación subsecuente a la inspección del Equipo de Desarrollo mismo. Aunque las mejoras pueden implementarse en cualquier momento, la Retrospectiva del Sprint ofrece un evento dedicado para este fin, enfocado en la inspección y la adaptación.

En resumen, durante la Retrospectiva:

- El equipo de trabajo se enfoca en responder a las siguientes preguntas:
 - Qué cosas han funcionado bien.
 - Cuáles hay que mejorar.
 - Qué cosas quiere probar el equipo en la siguiente iteración.
 - Qué ha aprendido el equipo.
- El equipo identifica las mejoras accionables para mantener o mejorar el trabajo del equipo, y mantener la motivación de los miembros.
- El equipo documenta y actualiza el registro de lecciones aprendidas.

7.4.2.1 - Técnicas para la realización de la Retrospectiva del Sprint

La retrospectiva es una de las ceremonias más importantes dentro del desarrollo de un proyecto usando Scrum, ya que es una de las principales fuentes de aprendizaje continuo que tiene el equipo, además sirve para mantener la motivación y el compromiso con el proceso de desarrollo.

Uno de los elementos que se deben tener en cuenta, es que el realizarla siempre de la misma manera (monotonía), podría disminuir el interés de los equipos en participar. Así que muchos de los Scrum Máster, suelen utilizar la reunión de retrospectiva del Sprint como un evento en el cual experimentar e introducir nuevas técnicas.

Por lo general una retrospectiva suele tener 5 momentos importantes:

1. Preparación de la sesión
2. Recolección/Análisis de los datos
3. Generar ideas de mejora/cambio
4. Definir los planes de acción (experimentos a aplicar en los próximos Sprint)
5. Recolección de retroalimentación de la retrospectiva

A continuación, se mencionan algunas de las tantas técnicas disponibles que se pueden realizar como parte de una Retrospectiva del Sprint.

Índice de felicidad del Equipo El índice de felicidad del equipo es una técnica utilizada para medir el nivel de felicidad alrededor del tiempo de cada uno de los miembros de un equipo. Se caracteriza por su facilidad de aplicación, además de tener el anonimato como uno de sus pilares centrales.

Para aplicar la técnica, el Scrum Master suele construir un tablero (canvas) como el que se muestra a continuación:

dibujo

- Suele estar en un lugar visible de las oficinas de trabajo del equipo Scrum
- Si está físico (recomendable), se utilizan stickers de colores para completar el tablero y así mapear el estado de ánimo
- Las variaciones en la felicidad del equipo, pueden ser el reflejo de las situaciones vividas en la organización.
- Es una medición subjetiva, por lo que los resultados deben manejarse con precaución
- Le aportan bastante valor a las labores realizadas por el Scrum Master, ya que se puede enfocar en hacer la vida más feliz a todos los miembros de los equipos que lidera.
- Siempre se debe garantizar el anonimato, ya que, de lo contrario, los resultados mostrados en el tablero no serán precisos o acordes a la realidad

- Aunque los datos se recolectan a lo largo del desarrollo de un Sprint, su análisis se puede hacer en conjunto con todo el equipo como parte de una retrospectiva.

4L's

Esta técnica descrita ampliamente en el libro “Agile Retrospectives” se utiliza como ejercicio en las retrospectivas. Consiste en un tablero (canvas) dividido en 4 cuadrantes:

1. Liked: Aquí cada uno de los miembros puede poner todas las cosas que les gustaron durante el desarrollo del Sprint (Ej: La planificación organizada, la nueva herramienta con la que se trabajó, una nueva técnica implementada por el Scrum Master, etc)
2. Learned: En esta sección, se listan todos aquellos elementos que el equipo siente que aprendió y que le servirán para próximos Sprints (Ej: La importancia de comentar el código, nuevas técnicas para revisar los entregables, nuevas técnicas de retrospectiva, etc)
3. Lacked: Aquí se colocan todas las cosas que faltaron durante el Sprint, así como las necesidades que el equipo tiene. (Por lo general suelen estar asociadas a impedimentos que haya tenido el equipo)
4. Longed For: Aquí los integrantes del equipo colocan todas aquellas cosas que quieren que sucedan en el futuro próximo (Ej: Mayor participación del cliente, mejor redacción en las historias de usuario, una nueva herramienta, etc)

dibujo

- Suele tenerse el canvas impreso y disponible en la oficina, para ser completado con sticky-notes
- Todos los datos son analizados por el Scrum Master, en conjunto con los miembros del equipo para así definir planes de acción para los próximos Sprint.

Feedback

Con el fin de que los equipos de trabajo tengan oportunidades de mejora en sus enfoques y desarrollo del trabajo, es necesario que el líder del equipo o incluso entre colegas, una vez finalizado un Sprint (o en cualquier momento que se considere necesario), se brinden retroalimentación.

El objetivo de la retroalimentación es hablar de situaciones buenas, para alentar dichos comportamientos, o de situaciones malas, para que la otra persona pueda convertirlas en oportunidades de cambio. Las sesiones de retroalimentación están orientadas a mejorar y fortalecer al equipo de trabajo, ayudando al desarrollo de competencias y crecimiento personal, individual y grupal.

Las sesiones de retroalimentación suelen tener las siguientes características:

1. Son reuniones en privado por lo general
2. No solo deberían usarse para hablar de cosas negativas

3. No sólo deberían realizarse por los líderes, también debería alentarse la realización de retroalimentación entre compañeros de equipo, así que por lo general la retroalimentación hace parte de una cultura de mejora y aprendizaje continuo.
4. La retroalimentación debe realizarse de forma continua
5. Siempre debería realizarse de forma inmediata al hecho o hechos que se vayan a tratar durante la sesión.

Feedforward El feedforward es una variación de la retroalimentación, esta técnica se centra más en las posibilidades futuras que en errores pasados, no solo se trata de comunicar, sino de hacerlo pensando en las oportunidades de mejoramiento de las prácticas a futuro, todo bajo un ambiente de motivación, optimismo y pensando en todo aquello que se pueda lograr para transformar las oportunidades en casos de éxito.

Son características que se deben tener en cuenta:

- Dar sugerencias para el futuro y aprender tanto como se pueda.
- No se deberían tratar hechos pasados.
- Es más productivo que el feedback, puesto que ayuda a las personas a hacer lo “correcto” más que probar que estaban “equivocados”.

Etapa 5: Entrega de los incrementos

7.5 - Etapa 5:

Implementación

Durante la etapa de implementación se realiza la entrega y/o puesta en marcha de todos los entregables desarrollados por el Equipo Scrum, durante esta etapa, se realiza el valor del producto. Está compuesta por 2 prácticas que se realizaran en forma iterativa (hasta donde sea posible y así lo permita la naturaleza del producto).

dibujo

ID	Práctica	Rol vs Nivel de Involucramiento
	Etapa 5: Implementación	
14	7.5.1 - Planificación de la implementación (14)	Product Owner: Medio - Scrum Master: Bajo - Equipo de desarrollo: Alto
15	7.5.2 - Implementación de entregables (15)	Product Owner: Medio - Scrum Master: Bajo - Equipo de desarrollo: Alto

El objetivo de esta etapa es la puesta en producción de los incrementos aprobados en la Revisión del Sprint, para su inmediata utilización y aprovechamiento por parte del cliente y/o interesados. En los proyectos Scrum, suele ser una etapa iterativa, para así aprovechar todos los incrementos utilizables desde etapas tempranas del proyecto.

Según la naturaleza de la organización, es común que éstas prácticas sean ejecutadas por grupos distintos al Equipo Scrum, sin embargo es altamente recomendable que sea el mismo Equipo el que se haga responsable de este trabajo, así se garantiza un real compromiso por la calidad de los entregables, a la vez que se evita la aparición de una cultura de la culpa.

7.5.1 - Planificación de la implementación (14)

El objetivo de esta práctica es preparar todos los elementos necesarios para realizar una implementación exitosa de los incrementos generados en los Sprint, disminuyendo significativamente los riesgos que puedan impactar negativamente las operaciones de los usuarios/clientes.

La cantidad de entregas (despliegues) dependerá de lo acordado con el cliente o patrocinador del proyecto, y por lo general, dependen del valor para la organización.

7.5.1.1 - Coordinación con operaciones

Una actividad clave es la coordinación con los distintos equipos de operaciones del cliente para preparar la implementación.

Algunos elementos clave en esta actividad son:

- ¿Cuándo se hará la implementación?
- ¿Qué usuarios recibirán el nuevo producto?
- ¿La implementación del producto afectará las operaciones?
- ¿Cómo se notificará a los usuarios?
- ¿Los usuarios están capacitados para usar el nuevo producto?
- ¿Qué hará el Equipo de Desarrollo en caso de que la implementación tenga fallos?

Nota: Es altamente recomendable que los equipos de operaciones hayan participado de las reuniones de Revisión de los Sprint.

7.5.2 - Implementación de entregables (15)

Esta práctica busca poner a disposición de los usuarios los incrementos finalizados y utilizables previamente desarrollados por el Equipo Scrum.

Cabe aclarar que esta práctica no es posible aplicarla en todos los tipos de proyectos, ni es obligatorio ejecutarla al finalizar cada Sprint, solo será aplicable cuando el incremento de producto genere valor para los usuarios o para la organización.

La implementación de entregables puedes realizarse de varias formas, las 2 más comunes son:

- Big bang: Este tipo de implementación, busca poner el incremento a disposición de toda la comunidad de usuarios al mismo tiempo. También se le llama “Implementación masiva”.
- Por fases: Este tipo de implementación se usa en caso de que se quiera hacer una segmentación de los usuarios para implementar el incremento (ej: entregarlos solamente a usuarios específicos).

7.5.2.1 - Confirmación de implementación exitosa

Una vez el incremento de producto está en ambientes productivos, es importante confirmar que no hubo afectaciones a la operación, por lo que podrían ejecutarse revisiones o pruebas post-implementación.

En muchas ocasiones luego de la confirmación de implementación exitosa, se realiza la notificación formal al cliente (incluso se firma algún artefacto como constancia), y así disminuir la probabilidad de que surjan nuevas solicitudes de cambios sobre los elementos ya entregados.

7.5.2.2 - Despliegues Fallidos

En caso de que los resultados de la implementación sean negativos se evaluará de manera rápida si se pueden solucionar durante una ventana de mantenimiento, en caso contrario se deberá realizar un “rollback” y planear un nuevo Sprint en el que se solucionen los problemas encontrados, dando paso a un nuevo despliegue.

Etapa 6: Cierre del proyecto

7.6 - Etapa 6: Cierre del proyecto

Esta es la etapa final de un proyecto Scrum, tiene como objetivos formalizar el cierre del proyecto de cara a las partes interesadas y recolectar lecciones aprendidas que permitan mejorar el desarrollo de futuros proyectos. Esta etapa es crucial en el trabajo de una APMO (Agile Project Management Office), ya que en esta etapa es donde se elaboran comúnmente informes que permiten a la organización determinar el valor realizado por un proyecto.

dibujo

ID	Práctica	Rol vs Nivel de Involucramiento
	Etapa 6: Cierre del proyecto	
16	7.6.1 - Cierre del proyecto (16)	Product Owner: Alto - Scrum Master: Bajo - Equipo de desarrollo: Bajo
17	7.6.2 - Reunión de Retrospectiva del proyecto (17)	Product Owner: Alto - Scrum Master: Alto - Equipo de desarrollo: Medio

7.6.1 - Cierre del proyecto (16)

En la reunión de cierre de proyecto se dejará registro del cierre y motivo del cierre del proyecto, para este fin podría construirse un acta de Cierre del proyecto. Los siguientes pueden ser motivos para el cierre del proyecto:

- Cierre exitoso del proyecto: Se cumplió con todos los entregables programados dentro del presupuesto asignado y de acuerdo con los criterios de aceptación (satisfacción total del cliente).
- Cierre parcial del proyecto o aplazamiento: Se cumplió parcialmente con los entregables programados, se está superando el presupuesto asignado o no se ha cumplido completamente con todos los criterios de aceptación (satisfacción parcial del cliente).
- Cancelación del proyecto: No se cumplió con ninguno de los entregables programados, se agotó el presupuesto asignado antes de tiempo, no se cumplió con ningún criterio de aceptación (insatisfacción total del cliente).

¿Qué hacer durante esta etapa?

- Entregar el producto terminado.
- Obtener la aprobación de los entregables del proyecto por parte del cliente.
- Cerrar formalmente el proyecto.
- Verificar la satisfacción por parte del cliente y las partes interesadas.

¿Qué se necesita para la reunión?

- Producto o incremento de producto desarrollado por el equipo: Para registrar el estado del producto al momento de cierre del proyecto.
- Acta de Inicio del Proyecto: Para validar el alcance, objetivos y aspectos iniciales del proyecto.
- Resultados de las pruebas de integración: Incluye un informe de las pruebas que se realizaron tras la entrega de cada incremento del producto para garantizar la correcta integración entre componentes.
- Criterios de aceptación del producto: Para verificación y validación de lo esperado en comparación con lo que se entregó.
- Documentación/Capacitación al cliente: Para garantizar que el cliente cuenta con el conocimiento sobre el uso, instalación, configuración o soporte básico del producto.
- Informes de proyecto: Que muestran el estado actual del progreso del proyecto (Se mostrarán informes como: diagrama de flujo acumulado y diagrama de presupuesto).

Es importante que quede un artefacto como evidencia del cierre del proyecto, con lo cual se garantiza la terminación oficial de las actividades del proyecto, o el cumplimiento de los compromisos posteriores (en caso que existan).

7.6.2 - Reunión de Retrospectiva del proyecto (17)

Dentro de los objetivos de esta práctica encontramos:

- Analizar y aprender de los aciertos y errores que se cometieron a lo largo de todo el proyecto.
- Analizar los informes generados por Aseguramiento de Calidad durante todo el proyecto.
- Crear un Informe de mejoras y lecciones aprendidas a lo largo del proyecto.

¿Quiénes participan de la reunión?

- Scrum Master.
- Product Owner.
- Equipo de Desarrollo.
- En algunas ocasiones incluso podría participar el cliente.