

ToolKit vol 4 Metodologías Ágiles

Departamento de Desarrollo Institucional



Innova GRD

Subdirección de Gestión y Desarrollo
Estratégico



SENAPRED
Ministerio del Interior
y Seguridad Pública

Gobierno de Chile

Índice de Contenidos

Bienvenida	4
Metodologías Ágiles	5
Historia	5
Objetivo	7
¿Qué ventajas tienen las metodologías ágiles?	7
Metodologías ágiles en el sector público	8
Como aplicar metodologías ágiles	8
PRINCIPALES METODOLOGÍAS	10
SCRUM:	10
Pilares de Scrum	10
Los Valores del Scrum	10
Componentes del Scrum	10
EQUIPO	11
Dueño del producto (Product Owner):	11
Scrum Master:	11
Equipo de desarrollo (Development Team o DevTeam):	12
EVENTOS DEL SCRUM	13
Sprint:	13
Planificación del Sprint (Sprint Planning):	13
Scrum Diario (Daily stand-up):	13
Revisión del Sprint (Sprint review):	14
Retrospectiva de Sprint (Sprint retrospectiva):	14
ARTEFACTOS DEL SCRUM	15
Lista de Producto (Product Backlog):	15
¿Qué son las Historias de Usuario?	16
Componentes de una Historia de Usuario	16
Lista de pendientes del Sprint (Sprint Backlog):	17
Ejemplo:	17
KANBAN:	18
PRINCIPIOS Y PRÁCTICAS DE KANBAN	19
Principios Fundamentales de la metodología Kanban	19
Prácticas básicas del método Kanban	20

Ejemplo:	22
EXTREME PROGRAMMING O XP:	23
<i>Valores de XP</i>	23
<i>Principios de XP</i>	24
<i>Roles de XP</i>	24
<i>Prácticas de XP</i>	25
<i>Ejemplo:</i>	26
Listado tablas	27
Listado imágenes	27
Referencias	27

Bienvenida

Diversos factores han llevado a las organizaciones a buscar nuevas formas para desarrollar su quehacer, como el aumento de la conectividad, expansión global del uso de tecnología (móvil, principalmente), uso de las redes sociales y otros. Estos factores han permeado a la sociedad completa y han impulsado a una digitalización tanto del trabajo como los procesos que lo sustentan.

Hace ya unos 20 años, un grupo de desarrolladores de software se dio cuenta de que sus proyectos presentaban trabas de avance y que no estaban enfocados al resultado en sí, más bien a seguir metodologías que eran perfectibles. Se dieron la tarea de repensar en cómo realizar el trabajo de una manera más colaborativa, en menor tiempo y sobre todo enfocándose en la satisfacción del cliente en el producto final, surgió así el concepto de “agilidad” y con el correr del tiempo, se han refinado en distintas técnicas que están apuntando a un trabajo colaborativo, adaptativo y confiable para los contextos complejos que hemos vivido en la última década.

La Agilidad aporta resiliencia organizacional, usando como fuerza principal la mejora del desempeño en todo aspecto acompañado de la comunicación fluida. Por su parte la digitalización ha impulsado el desarrollo de procesos más eficientes e incorporar el trabajo remoto, vital este último, durante la pandemia que recientemente hemos estado viviendo.

Actualmente, se presenta un desafío para el sector público en la adaptación de estas metodologías ágiles para desarrollar una cultura de gestión (procesos, sistemas y personas) que integre tecnologías de información y comunicaciones, simplificando los procedimientos internos y acotando los tiempos de respuesta en la resolución de las solicitudes. Prestar servicios de calidad en respuesta a necesidades actuales y construir una relación cercana con los ciudadanos y empresas es vital para impulsar políticas públicas efectivas y con impacto en el desarrollo económico y social del país.

La finalidad del presente documento es dar el marco general de la cultura ágil y presentar las principales metodologías a fin de establecer las bases de futuros esfuerzos de incorporarlas como herramientas para la gestión de proyectos.

Metodologías Ágiles

Las metodologías ágiles son marcos de trabajo¹ para producir u obtener un resultado tangible, centrado en las personas, en sus relaciones de trabajo y colaboración en pos de la satisfacción del usuario final. Surge de la industria del desarrollo del software y se ha popularizado en varias disciplinas hasta el día de hoy.

Estas metodologías surgen de años de experiencia e incorporan la mejora continua como uno de sus pilares fundamentales. Cuenta entre sus puntos claves, el desarrollo y conformación de equipos de trabajo, flujos continuos de desarrollo, requerimientos acotados y adaptativos (en pequeños bloques) para dar como resultado una generación de valor apreciable por el usuario final.

Tabla 1 Comparativa Metodología / Marco de Trabajo

METODOLOGIA	MARCO DE TRABAJO
Objetivos claros.	Flexibilidad.
Alcance: muy cerrado y definido.	Adaptabilidad al cambio.
Planificación de tareas.	Resultados más rápidos y utilizables.
Terminar una fase para continuar con otra.	Satisfacción del cliente.
Lecciones aprendidas al término del proyecto.	Tiempo de respuesta más rápido.
Gestión de cambios y riesgos al término del proyecto.	Reducción del Time-to-Market².
Entregables por cada etapa o fase concluida.	Equipos auto organizados.
Contratos rigurosos.	Transparencia de la información.
Presupuestos definidos.	Agrupación de lecciones aprendidas en cada iteración al término del proyecto.
Equipos especializados.	Negociación del contrato del proyecto.
Satisfacción del producto	Estimación de costos por iteración.
	Equipos multidisciplinarios.

La **Tabla 1** refleja el contrapunto entre la metodología (métodos y técnicas aplicados sistemáticamente) con el marco de trabajo (cuenta con un sustento metodológico, pero se enfoca en resultados y se consiguen a través de la interacción de los miembros de un equipo de trabajo)

Historia

El concepto de metodología ágil nació en 2001 y surgió de un grupo de líderes de la informática, desarrolladores de software y administradores, que se reunieron para debatir y buscar alternativas a los procesos tradicionales de desarrollo de software (método de cascada), caracterizados por la rigidez de sus procesos y su gran dependencia de una planificación detallada previa al desarrollo.

De esta reunión se sociabilizaron sus experiencias y pudieron recoger sus **“mejores prácticas”**, dando origen al Manifiesto para el Desarrollo de Software Ágil³. En este marco de trabajo, se planifica, se crea, se comprueba el resultado y se mejora, algo que es constante y rápido, con plazos de entrega acotados y se centra toda la atención en una tarea encomendada

Los **“métodos ágiles”** no son una metodología, son una mentalidad y un comportamiento guiados por unos valores y unos principios comunes, recogidos en el “Manifiesto Ágil”.

Valores

1. **“Individuos e interacciones** sobre procesos y herramientas
2. **Software funcionando** sobre documentación extensiva
3. **Colaboración con el cliente** sobre negociación contractual

¹ Conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar. <https://www.linkedin.com/pulse/metodolog%C3%ADa-y-marco-de-trabajo-la-gran-duda-gustavo-mart%C3%ADnez-v%C3%A1zquez/?originalSubdomain=es>

² Tiempo que transcurre desde que proyectamos un producto y servicio y el tiempo que tarda en estar disponible para el consumidor final. <https://connectingvisionsgroup.com/ideas/crecer-fidelizar/time-to-market/>

³ <https://agilemanifesto.org/history.html>

4. Respuesta ante el cambio sobre seguir un plan⁴

En estos valores priman los resultados orientados hacia el cliente o usuario y cuidan del compromiso de todo el equipo que realiza las labores de un resultado tangible y que sea apreciado como de valor. Si bien no omite los procesos, planes o registros, estos son considerados como apoyo y el resultado de la constante adaptación y perfeccionamiento.

En el manifiesto, se declaran también 12 principios los que transcribiremos y haremos una pequeña explicación de cómo estos se traducen a una visión “agilista”

Principios

- I. **“Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor⁵”**. Significa realizar entregas en tiempos acotados de una solución que sirva las necesidades del usuario.
- II. **“Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente”**. Indica que los cambios nutren a la solución final y son necesarios en cualquier etapa del proyecto.
- III. **“Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible”**. Se refiere a que se garantiza un entregable de manera frecuente, y en menor tiempo posible.
- IV. **“Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto”**. De esto se desprende que la mejor manera de alinear al usuario y desarrolladores es trabajando juntos y entender el aporte de la solución a desarrollar para el negocio/interés del cliente/usuario.
- V. **“Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo”**. Se debe propiciar un marco para que se desarrollen todas las potencialidades de los involucrados, entregando confianza y propiciando su autonomía.
- VI. **“El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara”**. Implica que la manera más eficiente y efectiva, pudiendo además de usar las palabras, leer el lenguaje corporal y realizar aclaraciones in situ es cara a cara. Además, es una manera de generar empatía con los demás miembros del equipo dada la interacción que se genera.
- VII. **“El software funcionando es la medida principal de progreso”**. Se refiere a que poder ver una solución operativa, permite enfocarse en el uso que tendrá dicha solución y pone foco en los resultados que podemos esperar de ella.
- VIII. **“Los procesos Ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida”**. Se busca tener equipos de alto rendimiento que puedan ser consecuentes con el compromiso de entregas en tiempo acotados y que se potencien unos a otros.
- IX. **“La atención continua a la excelencia técnica y al buen diseño mejora la Agilidad”**. Se busca plantear soluciones con alta robustez, sencillas y con alto rendimiento. El equipo identifica que ha hecho bien, lo comparte y aplica y aquello que no ha hecho bien es su brecha de mejora.
- X. **“La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial”**. Se buscan soluciones evitando complejidades, procurando no abarcar demasiado y plantear un avance enfocado. Se apunta a desarrollar soluciones simples que aporten gran valor al cliente/usuario.
- XI. **“Las mejores arquitecturas, requisitos y diseños emergen de equipos auto organizados”**. Propone que, al conocer sus fortalezas y potencialidades, el trabajo se enfoca de manera especializada y se refuerzan aquellos puntos débiles del equipo.
- XII. **“A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia”**. El equipo es capaz de ajustar y perfeccionar sus métodos, compartir sus aciertos y hacer una “retrospectiva” de lo alcanzado.

⁴ <https://agilemanifesto.org/iso/es/manifesto.html>

⁵ <https://agilemanifesto.org/iso/es/principles.html>

De lo anterior, podemos resumir de manera simplificada los principios ágiles de la siguiente manera:

Imagen 1 Resumen simplificado Principios Ágiles



La **imagen 1** condensa en lenguaje más coloquial el trasfondo aplicado de los principios ágiles creados en 2001, incorporando el desarrollo, hasta la fecha, de conceptos como la experiencia de clientes y mejora continua.

Objetivo

Las metodologías ágiles son una forma de trabajar y organizar flujos, dividir los proyectos en partes y permiten adaptarse sobre la marcha, complementando y resolviendo etapas en periodos de tiempo reducidos y con plazos acotados.

Se utilizan mayormente en empresas de desarrollo de software, pero también se les ha utilizado como herramientas de estrategias de marketing, gestión de servicios o como metodologías para la transformación digital.

El objetivo principal es impulsar a las organizaciones a gestionar sus proyectos con rapidez y flexibilidad, ayudando en el desarrollo de proyectos o soluciones, que necesitan mayor enfoque para adecuarse a las necesidades del cliente.

¿Qué ventajas tienen las metodologías ágiles?

Más allá de ayudar en la organización del desarrollo de proyectos, las metodologías ágiles poseen otras ventajas, como:

- **Mayor satisfacción del cliente/usuario:** El cliente/usuario, está más satisfecho en la medida que puede verse involucrado y comprometido a lo largo de todo el proceso. Mediante varias demostraciones y entregas, el cliente/usuario, puede apreciar in situ las mejoras introducidas en el proceso y que surgen de su interacción.
- **Agilizar entregas rápidas y continuas:** una de las principales características y propulsor del cambio constante, es el acotar el tiempo de las entregas y de una manera continua. Esto hace que los equipos se enfoquen en avances pequeños, pero de gran impacto y que se centren en la operatividad de la solución.
- **Mejorar la calidad del producto:** a través de las metodologías ágiles, el foco es la búsqueda de la excelencia del producto/solución, a través de las interacciones con su usuario, lo que mejora el producto final.
- **Aumentar la motivación de los trabajadores:** al disponer de equipos de trabajo autogestionados, posibilitan el desarrollo de la capacidad creativa y de innovación entre sus miembros.
- **Estimular el trabajo colectivo:** dentro de las metodologías ágiles, se trabaja con distintos equipos y las reuniones de colaboración o socialización de avances son fundamentales. Esto permite una mejor organización del trabajo y una visualización de los avances de los equipos de distintas áreas de la empresa u organización.

- **Predecir resultados y minimizar los riesgos:** gracias a revisiones continuas y a la posibilidad de cambios, es posible tener una mirada predictiva sobre los resultados que se pueden obtener, lo que minimiza los riesgos de cometer grandes errores.
- **Reducir los costos:** la gestión continua del proyecto permite que los errores se vayan identificando a lo largo del desarrollo y pueden ser corregidos, lo que evita pérdidas.

Metodologías ágiles en el sector público

A nivel global, el sector público está experimentando grandes cambios y transformaciones tecnológicas como resultado del aumento de los costos, presiones presupuestarias, las necesidades cambiantes de los ciudadanos y, lo que es más importante, el surgimiento de nuevos avances e interrupciones tecnológicas⁶.

Dado el desarrollo y masificación de las metodologías ágiles en el sector privado con sus innegables resultados, hace pensar el adoptar sus beneficios, adaptándolos al quehacer público e incluso explorar oportunidades de asociaciones público-privadas dando un paso hacia el enfoque centrado en el "cliente público" o la experiencia del ciudadano al responder más rápido a sus necesidades cambiantes y brindar soluciones mejoradas y personalizadas.

Si bien implementar la "agilidad" a todo nivel es utópico para las organizaciones del sector público debido a los factores internos y externos, como los marcos normativos o la supervigilancia de otros organismos públicos. Pero al adoptar algunos rasgos ágiles, podrían preparar al sector para apreciar los cambios como oportunidades, no amenazas latentes.

Respecto al estado del arte, en términos de uso de metodologías ágiles en el sector público, la empresa multinacional Boston Consulting Group (BCG⁷) realizó una serie de entrevistas y encuestas a directivos de 23 instituciones del sector público global, federal y estatal en Australia, Alemania, Suecia, Suiza, el Reino Unido y los Estados Unidos⁸ y entre sus conclusiones destacaron:

- *Nueve de cada diez agencias del sector público están utilizando Agile, casi la mitad de ellas durante más de dos años.*
- *Tecnologías de la Información (TI), formulación de políticas y prestación de servicios son las áreas donde se han lugares populares para implementar Agile.*
- *Alrededor del 40% tiene al menos una división, utilizando Agile como su principal forma de trabajo, pero pocos han escalado Agile en toda la organización.*
- *Las agencias que han adoptado Agile están entregando servicios hasta un 50% más rápido y mejorando la satisfacción de los ciudadanos hasta en un 25%.*

De lo anterior, podemos apreciar una tendencia hacia el agilismo y podemos obtener el conocimiento de otras instituciones que han implementado estas metodologías con éxito.

Como aplicar metodologías ágiles⁹

A continuación, trataremos de adaptar los consejos derivados de la aplicación de las metodologías ágiles en las empresas con un enfoque hacia el sector público.

Identificar objetivos

Antes de implementar esta metodología es fundamental identificar los objetivos del servicio, en específico en los que sustentaremos su aplicación, tener una visión clara de cómo se espera que la nueva metodología contribuya a alcanzar esos objetivos y como servirán de vaso comunicante con lo que los usuarios recibirán de nuestra parte.

⁶ https://www2.deloitte.com/content/dam/Deloitte/xs/Documents/About-Deloitte/mepovdocuments/mepov25/agility_mepov25.pdf

⁷ Firma global de estrategia empresarial fundada en 1963, dentro de sus principales aportes está la matriz que lleva sus siglas y describe el comportamiento de productos según tasa de crecimiento y participación de mercado. Mas información en <https://www.bcg.com/about/overview>

⁸ <https://www.bcg.com/publications/2020/getting-to-agile-at-scale-public-sector>

⁹ Adaptada a partir del artículo web disponible en <https://blog.hubspot.es/marketing/metodologia-agile>

Analiza la cultura

Dado que las metodologías ágiles requieren auto organización, comunicación cara a cara, confianza y apertura a los cambios, estos conceptos debieran imperar en los que participarán de ella. La cultura organizacional debe estar abierta a implementar este tipo de metodologías para que los esfuerzos no sean aislados, sino como parte integral de la institución.

Detectar el impacto potencial en beneficiarios

Es muy importante tener una idea acerca de cómo nuestra nueva forma de trabajar se reflejará en el usuario final, podemos estimar, por ejemplo: ¿cuál sería la mejoría real en la experiencia de los beneficiarios, (tiempos más acotados de respuestas, menos incertidumbre)? o ¿qué impacto tendrá en la calidad de nuestro servicio (mayor confianza del público/beneficiario, mayores habilidades de coordinación)? y ¿cómo mejorará la interacción Servicio/Usuario (mayores índices de confianza, menos resistencia a las instrucciones, mayor colaboración)? Tener estas respuestas nos ayudará a explicar como la implementación del Agilismo beneficiará a los usuarios y al servicio mismo.

Determinar los recursos disponibles

Para poder implementar las metodologías, es indispensable plantear si se cuenta con las personas y las tecnologías necesarias que permitan sacarle el mejor provecho a Agile. Algunas metodologías requieren equipos dedicados y una infraestructura acorde, lo que es fundamental para una implementación exitosa.

Apoyarse en los líderes

Implementar cualquier cambio tiene aparejada una cuota de resistencia a él, en este caso el enfoque de gestión de proyectos Agile representa un cambio de paradigma con efectos en procesos, personal y la manera en que se utilizará la tecnología. Por ello es importante que, desde el inicio de la implementación de esta nueva modalidad, se involucre a los líderes y expertos de la empresa. De esta forma se sentirá como un proceso validado y con el apoyo necesario para efectuar los cambios.

Implementa procesos con base en Agile

Se deben implementar procesos pensando en la agilidad (adaptabilidad, rapidez, mejora continua y con foco en el cliente/beneficiario) Para ello se deben considerar nuevos flujos de trabajos, se debe definir quiénes serán las personas que participarán, así como las acciones y tecnologías que usarán para cumplir con los objetivos.

PRINCIPALES METODOLOGÍAS

SCRUM:

Esta es una metodología que se lleva adelante en “Sprints”, o procesos cortos de trabajo, que deben ser lo más acotados posible. Al final de cada sprint, el equipo debe entregar una versión mejorada del proyecto (modelo, diseño, sistema o producto) para que sea analizado por todos los interesados; luego de eso, los evaluadores dan una retroalimentación y devolución, lo que da inicio al proceso de mejora.

Los integrantes del Scrum son un equipo pequeño de personas, altamente flexible y adaptativo. Su metodología de trabajo se basa en que el conocimiento se adquiere de la experiencia y de tomar decisiones basándose en lo que se conoce (proceso iterativo, con varias fases definidas en el tiempo (regulares) con el fin de notar esos avances y acostumbrar al equipo a su rutina de trabajo)

Al ser un equipo pequeño, la metodología Scrum tiene ciertas estructuras bien definidas las que son compartidas y respetadas por todo el equipo, a continuación

Pilares de Scrum

- **Transparencia:** socializar estado actual del proyecto para todos los involucrados, con la máxima claridad y franqueza, todos trabajan y colaboran colectivamente por el objetivo común y nadie tiene secretos respecto al proyecto
- **Inspección:** capacidad de cualquier persona, para revisar cada requerimiento, procesos, productos o prácticas para ver que se puede mejorar y si están bien definidos o no. El scrum busca la mejora continua.
- **Adaptación:** capacidad que tiene cualquier persona reconocer cambios e implementarlos lo antes posible, procurando redireccionar cualquier modificación lo antes posible y no se desvíe del objetivo del sprint.

Los Valores del Scrum

- **Compromiso:** cada miembro del equipo se va a comprometer en la iteración actual (en desarrollo) al 100%.
- **Coraje:** cada miembro del equipo debe sentirse seguro para decir sí, no, pedir ayuda, y/o trabajar en nuevos enfoques.
- **Enfoque:** todo el equipo está enfocado en resultados, lo que se comienza, se termina.
- **Apertura:** se escucha a cada uno de los que están involucrados en el proceso, buscando constantemente aprender y mejorar.
- **Respeto:** todas las relaciones de los integrantes del equipo se basan en el respeto, reconociendo cuáles son las fortalezas de cada uno.

Componentes del Scrum

Imagen 2 Componentes del Scrum



La **imagen 2** agrupa los componentes principales del Scrum y sus elementos.

EQUIPO

Dueño del producto (Product Owner):

Es el representante del cliente/usuario dentro del equipo y es el encargado de despejar dudas al equipo de desarrollo y debe estar en capacidad de explicar en qué consiste cada funcionalidad del producto.

El Product Owner:

- Es el dueño del producto (Product Backlog)
- Gestiona los requerimientos
- Atiende todas las peticiones del cliente
- Expresar claramente los elementos de la lista del producto y los objetivos de desarrollo.
- Dar prioridad a los elementos de la lista del producto y decidir en qué orden se van a realizar los diferentes desarrollos o funcionalidades.
- Asegurar que la lista de producto sea visible y clara para todos los involucrados en el proyecto.
- Asegurar que el equipo de desarrollo conozca los elementos de la lista de producto.

Imagen 3 Responsabilidades y características Product Owner

Product Owner	
Responsabilidades	Características
•Expresar claramente el backlog •Priorizar elementos del backlog •Optimizar el valor del Equipo de Desarrollo (Dev Team) •Backlog visible y claro •Que el Dev Team conozca el backlog	•Maximiza valor del producto •Gestiona backlog •Única persona autorizada para decir qué se hace (representante del cliente)

Scrum Master:

Es el encargado de asegurar la implementación de la metodología en el desarrollo de todo el proyecto. Es el responsable de promover y apoyar la ejecución de Scrum, tanto al interior del equipo, ayudando a todos a entender la teoría, la práctica, las reglas y los valores de este marco de trabajo, como al exterior del equipo coordinando y ayudando a entender al resto, cuando deben interactuar con el equipo scrum.

El Scrum Master:

- Implementa la metodología Scrum.
- Elimina obstáculos e impedimentos.
- Responsable de promover y apoyar el Scrum.
- Asegura que los objetivos y el alcance de los sprints sean entendidos por todo el equipo scrum, que todos entiendan en el planning qué se va a trabajar.
- Transmite los conocimientos sobre Scrum y prácticas ágiles.
- Facilita los eventos de Scrum según se requiera.
- Guía al equipo de desarrollo para que sea autoorganizado y multifuncional.
- Ayuda al equipo de desarrollo a crear productos de alto valor, asegurándose de que los objetivos durante el sprint sean entendidos. releva los objetivos al equipo durante el sprint.
- Lidera y guía a la organización en el proceso de adopción de Scrum.

- Trabaja en conjunto con otros Scrum Masters para incrementar la efectividad de la ejecución de Scrum en la organización.

Imagen 4 Inter relaciones del Scrum Master



La **Imagen 4** muestra como el Scrum Master se relaciona con el resto del equipo y apoya el avance del Scrum.

Equipo de desarrollo (Development Team o DevTeam):

Es el conjunto de profesionales que se encarga de crear y entregar el producto terminado, el cual se pueda poner en producción al final de cada sprint. De esta forma, el desarrollo es incremental.

El equipo de desarrollo en Scrum tiene las siguientes características:

- Autoorganizado. Sabe cómo y qué se va a desarrollar durante el sprint. Se autogestiona en términos de roles y actividades internas.
- Multifuncional. Es un grupo multidisciplinario capaz de realizar cualquier actividad, si cuenta con las personas indicadas para la ejecución de las diferentes tareas.
- No tiene títulos. Es decir, no hay jerarquía y todos se deben tratar por igual, con respeto y buscando el consenso para sacar lo mejor de las diferentes habilidades individuales.
- No hay sub-equipos. No existen grupos dentro del equipo de desarrollo. Las actividades y responsabilidades se asignan por igual.
- No se modifica el equipo de desarrollo hasta terminar el sprint. Esto con el fin de no afectar el desarrollo de las actividades planeadas.

Imagen 5 Características del Equipo de Desarrollo



La **imagen 5** describe de manera abreviada las principales características del equipo de desarrollo

EVENTOS DEL SCRUM

Ayudan a establecer una rutina en el desarrollo del proceso y minimizar las reuniones no programadas. A medida que se van realizando las iteraciones, se van realizando ajustes en el scrum.

Sprint:

Es el periodo de tiempo delimitado o iteración, donde el equipo trabaja en el desarrollo del producto.

Planificación del Sprint (Sprint Planning):

Es el comienzo del sprint. Es una reunión en la que participa todo el Equipo Scrum y es donde se definen las Historias de Usuario que se van a desarrollar durante el Sprint, el sprint goal. Se planifica qué ítems del Product Backlog pasarán a ser ejecutados, definiendo así el Sprint Backlog.

Características del Sprint Planning

Duración del Sprint Planning

El Sprint Planning no debe durar más de 8 horas para sprints de 4 semanas. Normalmente, esta ceremonia puede tomar 1 hora para un sprint de 2 semanas.

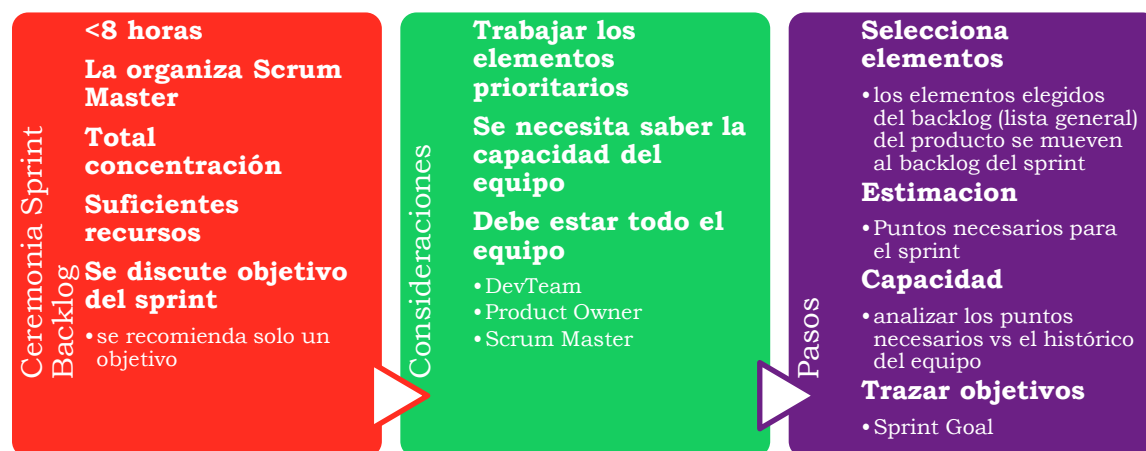
Funciones del Scrum Master en el Sprint Planning

El Scrum Master es el encargado de organizar esta ceremonia. Se debe llevar a cabo en un lugar adecuado y que sea lo suficientemente espacioso para el equipo, puede incluir un pizarrón o la tecnología necesaria para llevar a cabo una videoconferencia con el cliente.

Objetivos del Sprint Planning

Lo primordial es identificar que entregaremos al final del sprint, trabajando los elementos priorizados del Product Backlog identificando la capacidad del equipo y su velocidad de procesar las historias del usuario

Imagen 6 Sprint Planning, características y recomendaciones



Scrum Diario (Daily stand-up):

Es la reunión diaria en que participa todo el equipo de desarrollo para informar cómo va el proceso del desarrollo. Debe contener información importante respecto a ¿Qué se hizo el día anterior?; ¿Qué se hará el día del daily? y ¿Qué inconvenientes hay que no permiten seguir avanzado?

Scrum Diario

Características

- 15 minutos con el DevTeam
- Planeación de las próximas 24h
- Scrum master lo planea
- La información que se comparte debe estar enfocada solo a los objetivos del sprint
- puede haber personas externas pero no hablan

Proceso

- Esta reunión se realiza apuntando a ser muy concretos y no sobrepasar el tiempo estimado, trata de responder las siguientes preguntas:
 - ¿Qué hice ayer?
 - ¿Qué haré hoy?
 - ¿Tengo algún impedimento?
- Scrum Master ayuda a resolverlos

Beneficios del Daily Stand-Up

- Mejora la comunicación del equipo de Scrum, gracias a las 3 preguntas que se responden (¿qué hice ayer?, ¿qué hice hoy?, ¿tengo algún impedimento?).
- Hace posible conocer y medir el progreso para cumplir el objetivo del Sprint.
- Ayuda a identificar rápidamente problemas o inconvenientes que pueden frenar el avance del equipo de desarrollo.

Revisión del Sprint (Sprint review):

Corresponde a la reunión de revisión acabado el periodo del sprint. En ella se repasa el trabajo realizado y se inspecciona la incrementalidad del producto entregado

Dependiendo del feedback de esta reunión, pueden hacerse cambios en la lista de productos, añadiendo ítems, modificando los existentes o incluso aquellas que ya no sean necesarios.

Características de la Revisión del Sprint

- Esta reunión está enfocada en la comunicación con el cliente, es informal y abierta al diálogo, no es una sesión de seguimiento.
- Es facilitada por el Scrum Master y no debe durar demasiado tiempo.
- Al final de la revisión se tendrá una lista de producto actualizada.
- El objetivo principal es mostrar el producto desarrollado al cliente.

Retrospectiva de Sprint (Sprint retrospective):

Corresponde a la reunión interna en la que solo participa el equipo de Scrum y donde se evalúa lo que se realizó y las mejoras que pueden aplicar (mejora continua)

Características de la Retrospectiva

- Es una reunión positiva y productiva.
- No es una sesión para buscar culpas.
- Debe ser una reunión ejecutiva y enfocada a las oportunidades de mejora.

Elementos que se discuten en la Retrospectiva:

- **Herramientas:** Evaluar si se tiene todo lo necesario para cumplir el objetivo.
- **Relaciones:** Identificar si la comunicación es fluida o hay problemas.
- **Personas:** Analizar cómo es la interacción entre los integrantes del equipo.
- **Procesos:** Determinar si los procesos se ajustan al desarrollo óptimo que se espera dentro del equipo.

Imagen 8 Ceremonias de Revisión y Retrospectiva del Sprint. principales recomendaciones, interacciones y puntos relevantes.

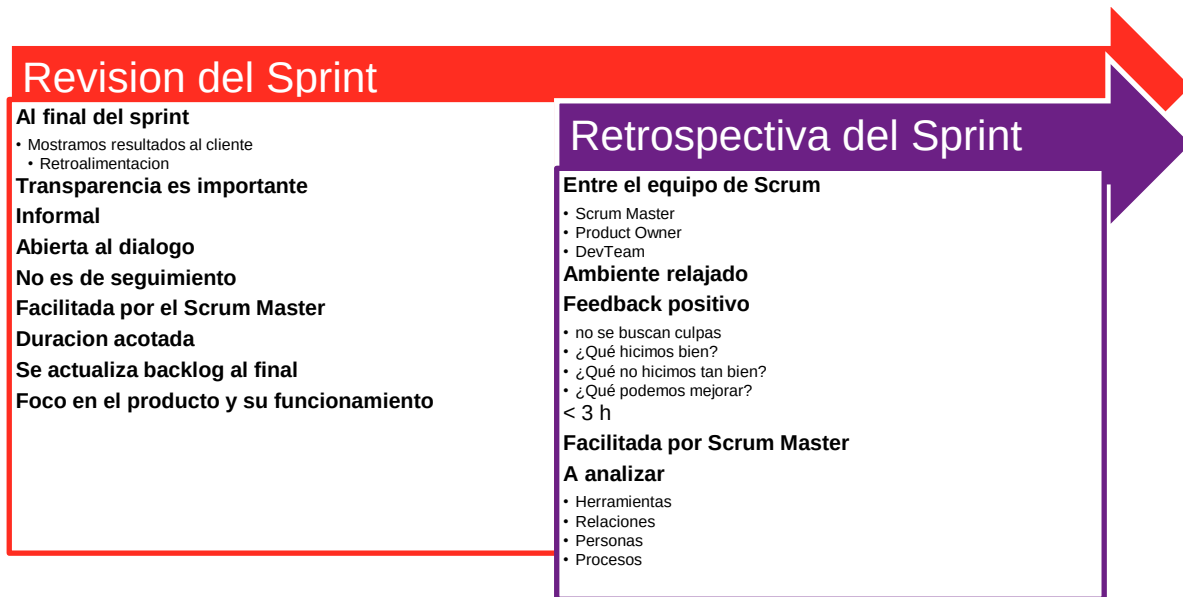
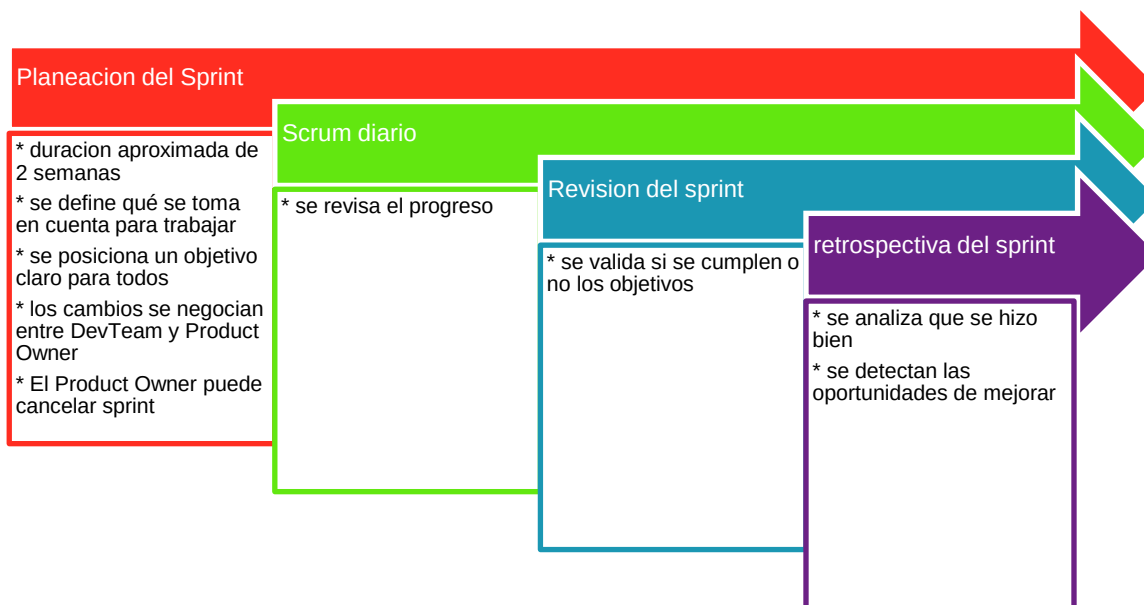


Imagen 9 Eventos del Scrum, sus principales tareas y características



ARTEFACTOS DEL SCRUM

Son los elementos que definen en que consiste el producto a entregar (requerimientos o características)

Lista de Producto (Product Backlog):

Es una lista ordenada de todo lo que se conoce y que es necesario para que el producto funcione. Si durante la revisión del producto el cliente sugiere un cambio, ese cambio debe agregarse a la lista del producto, si una historia de usuario no está en la lista del producto no existe y nadie debe trabajar en ello.

Característica de la lista de Producto o Backlog:

- Es dinámico, por tanto, va evolucionando a medida que avanza el proyecto, según los nuevos requerimientos.
- Los elementos en la lista que tienen mayor prioridad tienen descripciones más completas.

- La Lista del Producto es un artefacto vivo, está cambiando constantemente y en ella residen todos los elementos necesarios para completar el producto.

¿Qué son las Historias de Usuario?

Los elementos del Backlog (Lista del Producto) se denominan Historias de Usuario. Estas historias de Usuario son los elementos más específicos de la lista de producto y contienen la visión del usuario sobre la funcionalidad esperada del producto. No se deben confundir las Historias de Usuario con requerimientos, en ellas se referencia lo que el usuario quiere o espera del producto terminado.

Tabla 2 Ejemplo Tarjeta de Historia de Usuario

Historia de Usuario		
Número:	Nombre Historia	Criterios de Aceptación ☐ Funcionalidad ☐ Código Actualizado ☐ Pruebas realizadas ☐ Documentación
Usuario:	iteración asignada:	
Prioridad en negocio: (Alta / Media / Baja)	Puntos estimados:	
Riesgo en desarrollo: (Alto / Medio / Bajo)	Puntos reales:	
Descripción		
Observaciones:		

Componentes de una Historia de Usuario

Son los elementos esenciales que se deben identificar en la descripción de funcionalidades de un producto y que aportan a generar valor, facilitar la comprensión por parte del equipo, y simplificar el proceso de desarrollo. Estos componentes son:

- **Título o Nombre:** Permite conocer rápidamente de qué se trata la historia y por ello debe ser concreta y clara.
- **Descripción:** Contiene información sobre cómo se deben ejecutar las tareas, puede incluir componentes técnicos, especificar el tipo de diseño, el lugar donde se van a almacenar las tareas o el flujo de arquitectura que se debe seguir.
- **Puntos:** Los puntos (escala de puntaje) de una Historia de Usuario representan el esfuerzo que le va a tomar al equipo de desarrollo completar las actividades de la historia.
- **Criterio de aceptación:** A través de él se definen los requisitos que la historia debe cumplir para que esté completa (checklist).

Para estimar Historias de Usuario en un proyecto Scrum se deben tener en cuenta 4 elementos:

- **Complejidad de la historia:** Junto con el equipo se debe definir qué tan difícil es implementar la historia.
- **Cantidad de trabajo requerido:** Representa el esfuerzo que debe invertir el equipo para llevar a cabo la historia.
- **Conocimientos necesarios:** Habilidades técnicas, de diseño y de negocio, entre otras, para completar la historia.
- **Incertidumbre:** Acciones que se requieren para ejecutar la Historia de Usuario, aunque no se tenga claro el tiempo o esfuerzo real.

El proceso de estimación o priorización de las Historias de Usuario se hace a través de puntos. Estos puntos no están relacionados con una escala de medición, no representan horas o días de trabajo, sino una apreciación con base en la experiencia del equipo.

Lista de pendientes del Sprint (Sprint Backlog):

Es un subconjunto de la lista del producto y contiene todos los elementos que serán desarrollados durante el Sprint.

Características del Sprint Backlog

- Contemplar un plan lo suficientemente detallado para que todo el equipo esté en capacidad de comprenderlo en el Scrum diario.
- Asegurar que todas las personas que participan en el Scrum diario tengan conocimiento del Spring Backlog.
- El dueño del Backlog del Sprint es el equipo de desarrollo, por lo tanto, tiene la autoridad sobre la lista, para aceptar o no que se agreguen elementos al Sprint Backlog.
- Si un elemento se vuelve innecesario a mitad de un sprint se puede sacar de la lista de pendientes.
- El Product Owner podrá dialogar con el equipo de desarrollo para bajar la prioridad de una historia o incluso eliminarla.

Imagen 10 Claves del Backlog



Ejemplo:

Supongamos el siguiente escenario:

- **Dueño del Producto (Product Owner):** Se comunica y representa al cliente y sus necesidades
- **Equipo de Desarrollo (Development Team):** Son los desarrolladores, programadores y diseñadores que entregan el producto
- **Scrum master:** Es el encargado de que las prácticas scrum se sigan y que el equipo este centrado en el objetivo

¿Cómo funciona la metodología Scrum?

A manera de simplificación la **planificación del Sprint (Sprint Planning)** solo se hará por el Product Owner, así la metodología seguiría de la siguiente forma:

- El Dueño del producto elabora una lista de deseos con diferentes prioridades (**Lista de Producto o Product backlog**)
- El Equipo de Desarrollo toma una pequeña parte de la lista de deseos (**Lista de pendientes del Sprint o Sprint backlog**) y planea su implementación
- El equipo completa sus tareas del sprint backlog en un **Sprint**. Se evalúa el progreso en una reunión diaria (**Scrum diario o Daily Stand-up, también conocido como Daily Scrum**)
- Cuando acaba el Sprint, se envía o revisa el trabajo completado (**Revisión del Sprint o Sprint Review**) y se cierra este sprint, para comenzar con el siguiente.

Cuando el sprint termina o se cierra, se puede realizar una **Retrospectiva del Sprint o Sprint Backlog**, para incorporar todos aquellos hallazgos, aprendizajes y socialización de errores para incorporarlos en sucesivos desarrollos.

KANBAN:

Kanban deriva de una palabra japonesa que en español significa “tarjeta visual”. Esta metodología sugiere una comunicación en tiempo real y controla el trabajo a través de un esquema visual y consecutivo. Por lo general se establecen tres columnas de organización de trabajo: **pendientes o por hacer (To Do)**, **en proceso (Doing)** y **terminadas (Done)**. De esa forma, es posible clasificar las tareas y visualizar fácilmente sus avances.



Imagen 11 Tablero KANBAN¹⁰

Kanban es un sistema de gestión del cambio evolutivo no disruptivo. Lo que significa que el proceso existente se mejora en pequeños pasos, aplicando cambios menores (en lugar de uno grande), reduciendo el riesgo para el sistema general. El enfoque evolutivo de Kanban hace que la resistencia del equipo y de las partes interesadas sea escasa o nula.

Existen dos modelos básicos de Kanban que se utilizan ampliamente en las industrias, pero que pueden aplicarse exitosamente en el día a día empresarial: el de **producción** y **movimiento**.

En el **Kanban de producción**, los Paneles (en el caso de un Kanban manual) o el software están disponibles para todas las personas responsables de realizar las tareas en la fase de producción.

En esta metodología se elaboran tarjetas que contienen información de la parte o tarea que forma parte del proceso productivo, con una breve explicación de la tarea o producto, la fecha límite para comenzarla y finalizarla y el nombre de la persona responsable.

¹⁰ Obtenido de <https://www.forecast.app/blog/scrum-board-vs-kanban-board#What-is-a-kanban-board>

Tabla 3 Ejemplo de tarjeta de KANBAN Productivo

KANBAN PRODUCTIVO			
Responsable: Ernesto Ruiz F. Inicio: 20 de diciembre F. Fin: 21 de diciembre			
 Apariencia	Parte N°	1234-567-89	
	Nombre del Producto	Vino Añejo	
	Tipo de envase	Botella	N° Procesos
	Capacidad del envase	100 CC	2/3
		Embotellado EM – 01313 Relleno de los envases desde mosto fermentado FM-316	

Las tarjetas se colocan en cada una de las etapas y a medida que se van completando el proceso de producción avanza

El **Kanban de movimiento** se utiliza para comunicar a los sectores de producción sobre el momento ideal para realizar una tarea o cuando se debe esperar el comando para iniciar la siguiente fase de la cadena de producción.

En esta metodología las tarjetas tienen un mayor enfoque a los procesos, desde que paso vienen y a que proceso aportan, también pueden hacer referencia a un producto terminado, sus características físicas o de dimensiones de empaque para poder aportar a su destinación final.

Tabla 4 Ejemplo de tarjeta de KANBAN de Movimiento o Transporte

KANBAN DE MOVIMIENTO			
Responsable: Juan Casey F. Inicio: 21 de diciembre F. Fin: 21 de diciembre			
Proceso anterior	Embotellado EM – 01313	 Guarda GU - 011	Proceso siguiente
Código	EM – 01313	Nombre del Artículo	Vino Añejo Caja
Dimensiones		Capacidad del contenedor	12 botellas
Tamaño del lote	12 cajas	N° de Contenedor	120

PRINCIPIOS Y PRÁCTICAS DE KANBAN

Principios Fundamentales de la metodología Kanban

1. Empezar con lo que se está haciendo ahora
 2. Aceptar realizar un cambio incremental y evolutivo
 3. Respetar inicialmente las funciones, las responsabilidades y los cargos actuales
 4. Fomentar los actos de liderazgo a todos los niveles
- **Comenzar con lo que está haciendo ahora:** El Método Kanban hace hincapié en no hacer ningún cambio en su configuración/proceso actual de manera inmediato. La metodología debe aplicarse directamente al flujo de trabajo actual. Cualquier cambio se producirá

gradualmente durante un período de tiempo a un ritmo con el que el equipo se sienta cómodo.

- **Aceptar realizar un cambio incremental y evolutivo:** Esta metodología incentiva a realizar pequeños cambios de manera aditiva en lugar de hacer modificaciones radicales que podrían producir resistencia dentro del equipo. Una vez introducidos y vistos sus beneficios, estos pasarían a validarse y aceptarse de mejor forma.
- **Inicialmente, respetar los roles, responsabilidades y títulos de trabajo actuales:** A diferencia de otros métodos, Kanban no impone ningún cambio organizacional o estructural. El equipo identificará e implementará en un ambiente colaborativo los cambios necesarios.
- **Fomentar los actos de liderazgo en todos los niveles:** esta metodología incentiva la mejora continua en todos los niveles de la organización y promueve que los actos de liderazgo no tienen que provenir únicamente de los roles directivos. Las personas de todos los niveles pueden aportar ideas y son libres para dirigir e implementar cambios que mejoren la forma en que se trabaja.

Prácticas básicas del método Kanban

1. Visualizar el flujo de trabajo
 2. Limitar el WIP (trabajo en curso)
 3. Gestionar el flujo
 4. Hacer explícitas las políticas de los procesos
 5. Implementar bucles de retroalimentación
 6. Mejorar en colaboración, evolucionar experimentalmente
- **Visualizar el flujo de trabajo:** Es necesario tener a la vista -ya sea en un tablero físico o en un tablero Kanban electrónico- los pasos del proceso que actualmente se utilizan para entregar un trabajo o servicios. Una vez que visualicen los elementos de trabajo y las interacciones que involucra, se podrá determinar problemas o riesgos que pudieran aparecer.

Suele utilizarse tarjetas con diferentes colores para identificar las diferentes clases de servicio o tipos de elementos de trabajo, pero también pueden utilizarse dentro de cada tarjeta un distintivo para indicar cuán crítico es la realización de una tarea.

También pueden utilizarse más de una “columna” además de las mencionadas anteriormente: pendientes o por hacer (To Do), en proceso (Doing) y terminadas (Done) para cada clase de servicio o para cada tipo de elemento de trabajo. Sin embargo, se recomienda iniciar la metodología de la manera más sencilla posible y hacer cualquier rediseño del tablero más tarde.

- **Limitar el WIP (Work in Progress o Trabajo en curso):** Al limitar el trabajo en curso, se anima al equipo a completar el trabajo que tiene a mano antes de emprender uno nuevo. Así se crea capacidad en el sistema, de modo que el equipo puede retomar nuevos trabajos y conocer su “carga” en la medida que realiza sus labores.

Limitar el WIP y poner los límites del WIP en cada columna del tablero (cantidad de actividades en desarrollo al mismo tiempo) no sólo ayuda a que los miembros del equipo terminen primero lo que están haciendo antes de retomar nuevas responsabilidades, sino que también comunica al cliente y a otras partes interesadas que la capacidad de trabajo de cualquier equipo es limitada, y que deben planificar cuidadosamente el trabajo que piden al equipo.

- **Gestionar el flujo:** la metodología Kanban ayuda a gestionar el flujo destacando la carga de cada equipo y el estado del avance en cada una de sus etapas. Pudiendo observar la rapidez con la que el trabajo se desplaza desde el principio hasta el final del flujo y puede ayudar a los equipos a analizar su funcionamiento y a realizar ajustes para mejorar el flujo con el fin de reducir el tiempo que se tarda en completar cada etapa del trabajo. Al tener una perspectiva visual del flujo de trabajo, se puede observar y resolver/eliminar de los cuellos de botella e identificar las etapas de espera intermedias (el paso de una columna a otra dentro de nuestro tablero Kanban) y ver cuánto tiempo permanecen los elementos de trabajo en estas «etapas de traspaso». Si podemos reducir el tiempo que se pasa en estas etapas de espera podremos reducir el Tiempo de Ciclo completo.

A medida que mejora el flujo por el uso y el aprendizaje, la entrega de trabajo de los equipos se vuelve dinámica y predecible. Lo que permitirá realizar compromisos más ciertos y determinar cuándo va a terminar cualquier trabajo que se les encomiende.

- **Hacer explícitas las políticas del proceso:** Al poner de manifiesto las reglas del proceso, se establece una base común para que todos los participantes entiendan cómo realizar cualquier tipo de trabajo dentro de esta metodología. Estas directrices pueden ser a nivel de tablero, a nivel de carril (separaciones horizontales) y para cada columna, pudiendo ser: un check list de los pasos a realizar para cada tipo de elemento de trabajo; criterios de entrada-salida para cada columna; o cualquier indicación que ayude a los miembros del equipo a gestionar bien el flujo de trabajo en el tablero.
- **Implementar bucles de retroalimentación:** establecer etapas de revisión en el flujo de trabajo en el tablero Kanban, apoyadas en métricas, informes y una serie de señales visuales, proporcionan una base de retroalimentación continua sobre el progreso del trabajo -o la falta de él- en el sistema. Es posible, para etapas tempranas una revisión al término de un ciclo (inicio de un proceso hasta el fin de este) y con lo aprendido poder establecer correcciones sobre el progreso e identificar puntos a mejorar.
- **Mejorar en colaboración,** El método Kanban es un proceso de mejora evolutivo. que ayuda a adoptar pequeños cambios y a mejorar gradualmente a un ritmo y tamaño que los equipos puedan manejar fácilmente. Su tarea principal es evaluar su proceso constantemente y mejorar continuamente según sea necesario y posible.

El impacto de cada cambio que se realiza puede ser observado por todos y medido utilizando las distintas señales que le proporciona su sistema Kanban (señales visuales ajustes en los tiempos de avance). Esta metodología ayuda a recopilar muchos de los datos referente al rendimiento de los equipos. Utilizando estos datos, y las métricas que le ayudan a generar, se puede evaluar fácilmente si el rendimiento está mejorando o cayendo y ajustarlo en la medida que sea necesario.

Imagen 12 Beneficios de la Metodología Kanban

Beneficios de la Metodología KANBAN

- Proceso evolutivo, simple y fácil de aplicar.
- Estimula el rendimiento.
- Reduce tiempos inactivos y elimina actividades que no generan valor al equipo.
- Potencia la colaboración y organización.
- Permite distribuir mejor el trabajo.
- Reduce costos y en casos productivos, limita el desperdicio.

En la actualidad existen herramientas digitales, online y gratuitas que usan la metodología Kanban, entre ellas están: Planner (en el paquete office 365), Trello¹¹ o Monday¹².

¹¹ <https://trello.com/>

¹² <https://monday.com/lang/es/>

Ejemplo:

Supongamos que tenemos el proceso de elaboración de vino.

En primera instancia identificamos los procesos involucrados y asignamos responsables, tiempos estimados para realizarlos, definiendo 4 etapas para este ejemplo, donde:

- **FM** se refiere al proceso de Fermentación
- **EN** se refiere al proceso de Embotellado
- **GU** se refiere al proceso de Guarda o añejamiento
- **DE** se refiere al proceso de Despacho

En la medida que se termina en el proceso de embotellado puede realizar el proceso de guarda (resaltado para este ejemplo con una flecha roja).

Una vez terminados los procesos de guarda se puede procesar el despacho (resaltado en este ejemplo con unos puntos rojos)

De esa manera Kanban nos proporciona la información de manera visual de que procesos están en desarrollo y cuales faltan para entregar un proceso terminado.

Imagen 13 Ejemplo tablero Kanban

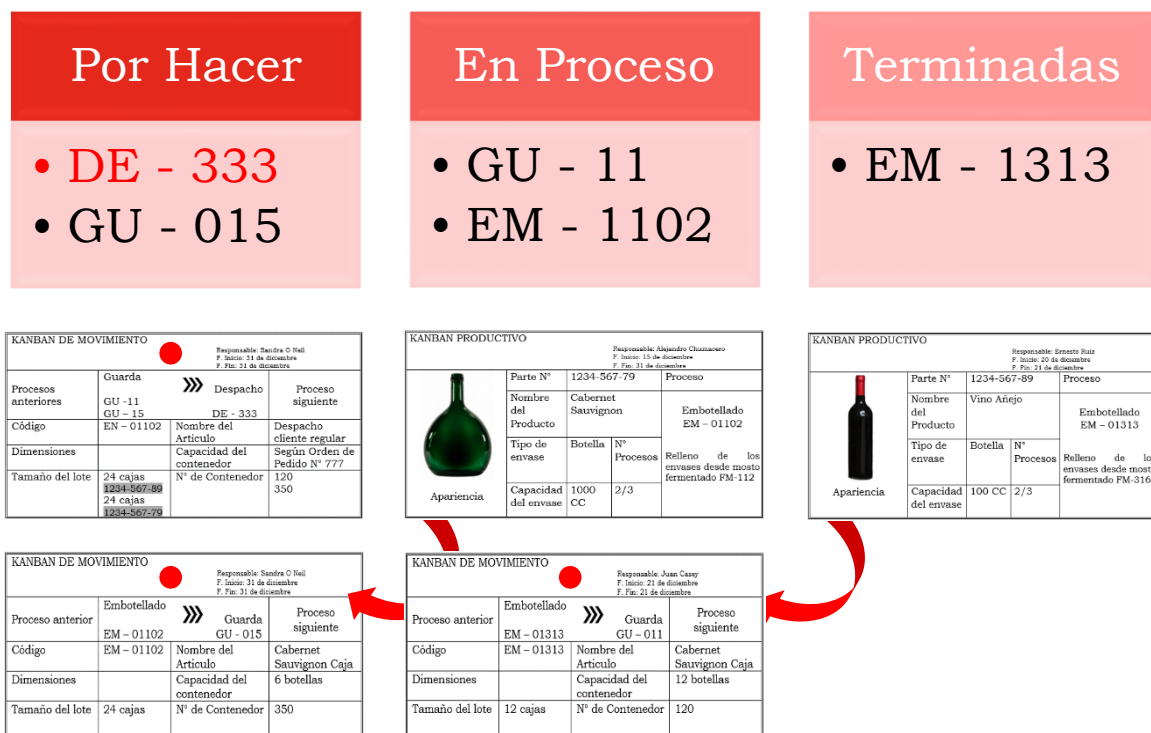
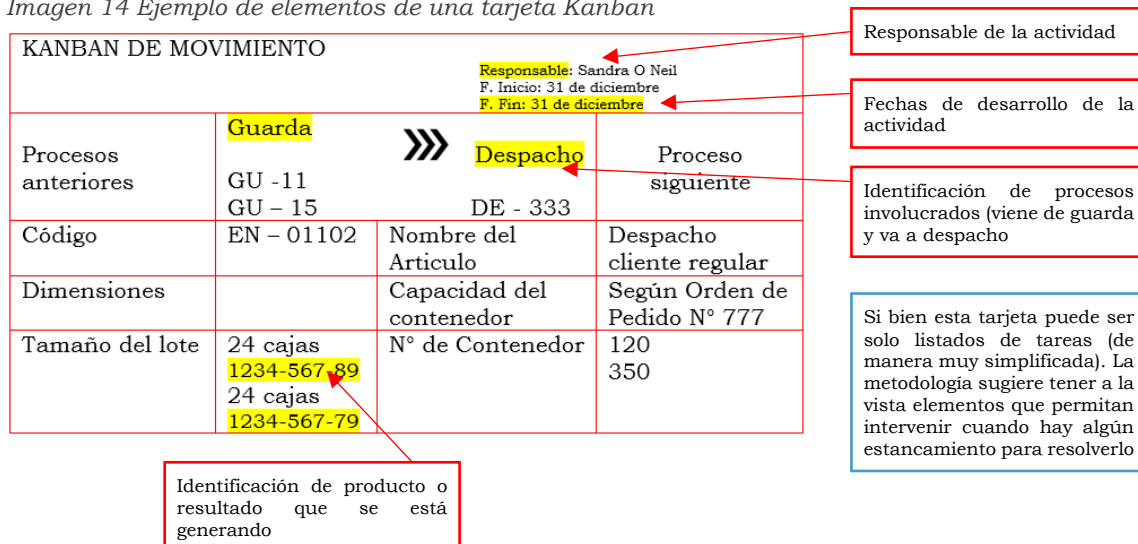


Imagen 14 Ejemplo de elementos de una tarjeta Kanban



EXTREME PROGRAMMING O XP:

Esta es una metodología que surge de la gestión de proyectos de desarrollo de software, creada para responder a ambientes muy cambiantes donde se necesita una retroalimentación permanente. Se centra en la velocidad y la simplicidad, con ciclos de desarrollo cortos y con menos documentación que los enfoques tradicionales. Si bien XP está diseñado sobre un enfoque de desarrollo de Software, es aplicable a cualquier equipo de trabajo, sea o no de software, ya que su objetivo principal es promover buenas prácticas y herramientas para que puedan convertirse en un equipo de excelencia y alto rendimiento.

Al igual que otros métodos ágiles, el diseño de la programación extrema permite a los desarrolladores responder a las solicitudes de los clientes, adaptarse y realizar cambios en tiempo real. Sin embargo, la programación extrema es mucho más disciplinada; realiza revisiones de código frecuentes y pruebas unitarias para realizar cambios rápidamente. Además, es muy creativa y colaborativa, ya que promueve el trabajo en equipo durante todas las etapas de desarrollo.

La estructura del proceso está determinada por 5 valores fundamentales, 5 Principios centrales y 12 prácticas de XP

Valores de XP

1. **Comunicación:** Para trabajar de manera efectiva, el equipo debe ser abierto y honesto entre sí. Cuando surgen problemas, se espera que todos aporten sus comentarios e ideas, ya que probablemente alguno de ellos ya tenga una solución adecuada. Y si no la tiene, podrán resolver el problema más rápidamente como grupo de lo que podrían hacerlo solos.
2. **Simplicidad:** Este valor se centra en reducir la complejidad y características adicionales. El fundamento es evitar el desperdicio y hacer solo cosas absolutamente necesarias, como realizar el trabajo lo más simple posible para que sea más fácil de mantener, respaldar y revisar.
3. **Retroalimentación:** En la programación extrema, se lanzan varias versiones con regularidad para obtener información nueva desde el primer momento y con frecuencia. La retroalimentación continua permite un diseño simple, construyendo algo, recopilando comentarios sobre su diseño e implementación, y ajustando la versión. Si un comentario u observación ayuda a reducir el tiempo de ejecución de una etapa de un proyecto, el equipo puede ajustar el proceso para que los desarrolladores trabajen para mejorar el tiempo de retraso en lugar de ajustar el proyecto completo.
4. **Valor:** cada miembro del equipo necesita valor para plantear problemas que dejan en evidencia la falta de eficacia de su equipo; requieren coraje para dejar de hacer algo que no funciona y probar otra cosa; se espera que sean honestos al informar al equipo sobre su progreso, lo que podría dejarlos en una posición de vulnerabilidad o si no cumplen con una fecha de entrega.

Si surgen cambios durante el proceso de XP, se espera que el equipo se adapte y acompañe estos cambios de frente y sin rehuir el nuevo desafío.

5. **Respeto:** Los miembros del equipo deben respetarse mutuamente, comunicarse entre sí, proporcionar y aceptar comentarios que beneficien su relación, y trabajar juntos para identificar diseños y soluciones simples. El respeto es una base importante que promueve la unidad, empatía y la confianza, incluso cuando se expresan las opiniones en las que no se está de acuerdo. Para la programación extrema, estas las expectativas del respeto involucran:

- Respeto mutuo entre los clientes/usuarios y el equipo de desarrollo.
- Respeto mutuo entre los miembros del equipo.
- El reconocimiento de que todos en el equipo aportan algo valioso al proyecto.

Principios de XP

1. **Realimentación rápida (Rapid feedback):** el equipo pide retroalimentación, la entiende y reacciona de inmediato de acuerdo con la información recibida.
2. **Asumir simplicidad (Assume simplicity):** el equipo debe centrarse en el trabajo que es importante en el momento, lo que se ha planificado, lo que se ha comprometido como entregable.
3. **Cambio incremental (Incremental change):** aplicar pequeños cambios de manera continua, es mejor que aplicar grandes cambios hechos de una sola vez.
4. **Abrazar el cambio (Embracing change):** Si un cliente/usuario piensa que un producto necesita ser cambiado, el equipo debe apoyar esta decisión y planificar cómo implementar los nuevos requisitos.
5. **Trabajo de calidad (Quality work):** un equipo que trabaja bien hace un producto valioso, con calidad y se siente orgulloso de ello.

Los principios rigen como guías rectoras de las decisiones que se pueden tomar al implementar la metodología. Es mejor y más fácil elegir una alternativa que se ajuste a los principios de forma directa que una que no lo haga de la misma forma.

Cada principio encarna uno o más valores de la metodología. Un valor puede ser vago y difícil de aplicar, por ejemplo, algo que sea “Simple” para una persona, para otra puede ser complejo así un principio es más concreto, u obtienes feedback rápidamente o no.

Existen principios adicionales a los primeros 5 (principios Core) que son auto explicativos, estos son:

6. Enseñar a aprender
7. Pequeña inversión inicial
8. Juega a ganar
9. Trabajar con los instintos de las personas, no en contra de ellos
10. Responsabilidad aceptada
11. Experimentos concretos
12. Comunicación abierta y honesta
13. Adaptación local
14. Viaja con poco equipaje
15. Medición honesta

Roles de XP

Un equipo de XP incluye seis roles:

1. **El cliente (Customer)** es la persona responsable de escribir historias de usuarios, establecer prioridades y establecer cuáles son los puntos para seguir para poder llegar al producto final y como se debe realizar, en función a sus necesidades y requerimientos.
2. **El programador (Developer)** escribe el código y realiza la totalidad de las tareas del proyecto. Tendrán que distribuir el trabajo y la carga total de información a analizar para poder producir o solucionar fallos.
3. **El entrenador (Coach)** es la persona que vigila el trabajo del equipo, lo controla y enseña a sus miembros a implementar las prácticas más efectivas. Asesorando al grupo para poder realizar correctamente la producción y desarrollo del proyecto solicitado por los clientes y analizado y gestionado por los testers.
4. **Manager (Tracker)** es la persona cuya tarea principal es monitorear el progreso de desarrollo y detectar todos los problemas en él. Hacer seguimiento de las métricas de progreso y de detectar obstáculos y facilitar las soluciones.
5. **El probador (Tester)** es el miembro del equipo responsable de las pruebas del producto. Son los encargados de validar con el cliente que se hayan cumplido sus expectativas y requisitos.

Prácticas de XP

La programación extrema, además de los valores y principios, propone buenas prácticas de planificación, organización, comunicación y de ingeniería que, sumado a los valores y principios, permiten crear una cultura de equipo de excelencia. Las prácticas de Extreme Programming (XP) más populares son:

1. **El Juego de Planificación (The Planning Game):** La idea principal de esta práctica es compartir las responsabilidades entre el equipo y el cliente. Se basa en un principio simple: los clientes tienen toda la información sobre el valor de los proyectos (requerimientos), y los equipos saben todo sobre su costo, así el juego trata de hacer compatibles las necesidades, tiempos y costos de desarrollo para definir prioridades respecto a las necesidades del cliente.
2. **Pequeños Releases (Small Releases):** La programación extrema realiza entregas pequeñas y de manera periódica para obtener feedback importante durante todo el proceso, las cuales se envían directamente a los clientes, aunque también pueden hacerse internamente.
3. **Metáforas (Metaphor):** El equipo debe tener la capacidad de explicar el diseño del sistema a personas nuevas a través de metáforas en vez de pedirles que lean una gran cantidad de documentos. Se decide en grupo y se usa un lenguaje para expresar cómo debe funcionar el equipo.
4. **Diseño Simple (Simple Design):** La metodología XP está diseñada para ser simple, producirá solo lo necesario y nada más.
5. **Desarrollo guiado por pruebas (TDD, Test Driven Development):** se realiza a través de 2 prácticas que permiten crear un código de calidad, pensando en la arquitectura del software o solución que queremos desarrollar: escribiendo la prueba primero y luego mejorarla a través de la refactorización.
6. **Refactorización (Refactoring):** Mejorar la estructura interna del código existente sin alterar su comportamiento externo. De esta manera se obtendrán diseños simples y de alta calidad.
7. **Programación en parejas (Pair Programming):** Dos programadores trabajando en conjunto en una sola máquina, resolviendo el mismo problema. Uno enfocado en la técnica y/o interfaz y el otro en la estrategia (integración con otros módulos o sistemas y como ir depurándola)
8. **Propiedad Colectiva del código (Collective Code Ownership):** Cualquier par de desarrolladores puede modificar el código en cualquier momento, independientemente de que lo hayan desarrollado ellos o no. En la programación extrema, la codificación se realiza en equipo, y el trabajo de todos se realiza según los estándares más altos.
9. **Integración Continua (Continuous Integration):** Práctica especialmente diseñada para construir o integrar todas las etapas de desarrollo, identificar errores y eliminarlos durante el proceso de desarrollo. Consiste en que el resultado de cada iteración es “acoplado” a la solución en construcción y sometidos a las pruebas de validación para que funcione dentro del constructo general.
10. **Historias de Usuario (User Stories):** La historia es un pequeño documento de texto que escribe el cliente donde especifica en palabras simples las características que el sistema debe poseer, sean requisitos funcionales o no funcionales, incluso para crear documentación.
11. **Estándares de Codificación (Coding Standard):** XP enfatiza la comunicación de los programadores a través del código, para lo cual es indispensable que se sigan ciertos estándares de programación (del equipo, de la organización u otros estándares reconocidos para los lenguajes de programación utilizados). Los estándares de programación (convenciones de código) mantienen el código homogéneo y legible para los miembros del equipo, facilitando los cambios.
12. **Ritmo de trabajo sostenible:** Los equipos deben determinar cuánto trabajo pueden producir a este ritmo por día y por semana, y usarlo para establecer plazos de trabajo. También a esta Práctica se le conoce como 40 horas por semana, ya que indica que es el plazo máximo recomendable (8 horas de trabajo, distribuidos en 5 días de laborales) que hacen que los participantes no se exijan demasiado y estén enfocados en el desarrollo.

Podríamos resumir a la metodología XP como un marco de desarrollo que persigue el feedback continuo, una excelencia técnica y la entrega de valor constante promovido a través de:

- **Valores** que representan la mentalidad del equipo, incentivan el trabajo en equipo y el enfoque hacia un objetivo común.
- **Principios** que reflejan los valores de maneras más concretas.
- **Prácticas** que nos ayudan a conseguir una cultura de equipo y una buena organización a través de la excelencia técnica y la creación de software de valor.

Imagen 15 Principios XP

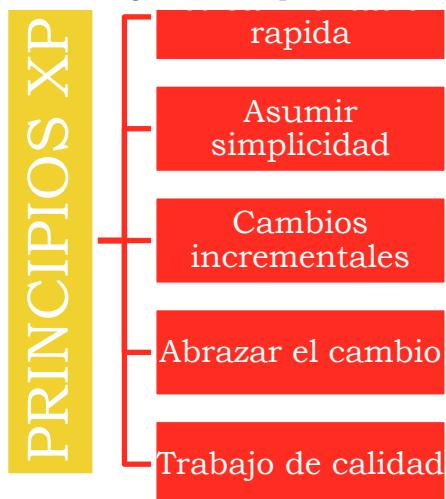
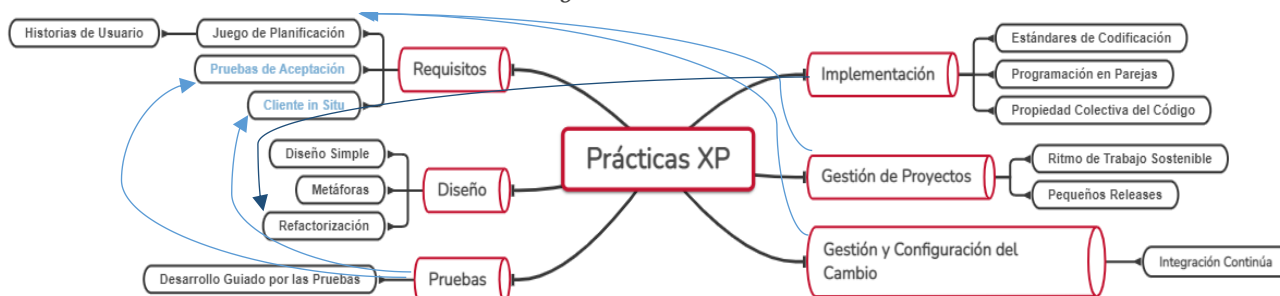


Imagen 16 Valores XP



Imagen 17 Prácticas XP



Ejemplo:

¿Cómo funcionaría la metodología ágil “Extreme programming XP”?

- El **Manager** elabora una lista de requisitos del cliente, una vez el cliente le ha contado sus **historias de usuario** (User stories), las que definen sus necesidades
- Se elabora un **plan de lanzamiento del producto**
- Entrega del producto en **iteraciones cortas**, por ejemplo, cada dos semanas
- El **equipo de desarrollo** trabaja en conjunto y realizan reuniones si surgen problemas
- **Participación activa** del **cliente** con **feedback** directo con historias de usuario
- Esto se repite hasta que el resultado es el requerido y ahí se entrega el producto final (que obedece a la historia de usuario desarrollada)

Listado tablas

Tabla 1 Comparativa Metodología / Marco de Trabajo.....	5
Tabla 2 Ejemplo Tarjeta de Historia de Usuario.....	16
Tabla 3 Ejemplo de tarjeta de KANBAN Productivo.....	19
Tabla 4 Ejemplo de tarjeta de KANBAN de Movimiento o Transporte.....	19

Listado imágenes

Imagen 1 Resumen simplificado Principios Agiles.....	7
Imagen 2 Componentes del Scrum.....	10
Imagen 3 Responsabilidades y características Product Owner.....	11
Imagen 4 Inter relaciones del Scrum Master.....	12
Imagen 5 Características del Equipo de Desarrollo.....	12
Imagen 6 Sprint Planning, características y recomendaciones.....	13
Imagen 7 Scrum Diario, características y cuestionamientos a resolver.....	14
Imagen 8 Ceremonias de Revisión y Retrospectiva del Sprint. principales recomendaciones, interacciones y puntos relevantes.	15
Imagen 9 Eventos del Scrum, sus principales tareas y características.....	15
Imagen 10 Claves del Backlog.....	17
Imagen 11 Tablero KANBAN.....	18
Imagen 12 Beneficios de la Metodología Kanban.....	21
Imagen 13 Ejemplo tablero Kanban.....	22
Imagen 14 Ejemplo de elementos de una tarjeta Kanban.....	22
Imagen 16 Valores XP.....	26
Imagen 15 Principios XP.....	26
Imagen 17 Prácticas XP.....	26

Referencias

- Alistair Cockburn, M. B. (2001). *Manifiesto Agil*. Obtenido de [https:// agilemanifesto.org/iso/es/manifesto.html](https://agilemanifesto.org/iso/es/manifesto.html)
- Arie van Bennekum, M. B. (2001). *Manifiesto Agil*. Obtenido de [https:// agilemanifesto.org/iso/es/principles.html](https://agilemanifesto.org/iso/es/principles.html)
- Deloitte. (2018). *Agile in the public sector*. Obtenido de https://www2.deloitte.com/content/dam/Deloitte/xs/Documents/About-Deloitte/mepovdocuments/mepov25/agility_mepov25.pdf
- Hubspot. (01 de diciembre de 2021). *Metodología Agile: qué es y cómo aplicarla a tu proyecto*. Obtenido de <https://blog.hubspot.es/marketing/metodologia-agile>
- Miguel Carrasco, T. C.-I. (05 de marzo de 2020). *BCG*. Obtenido de <https://www.bcg.com/publications/2020/getting-to-agile-at-scale-public-sector>
- Mike Beedle, A. v. (2001). *Manifiesto Agil*. Obtenido de [https:// agilemanifesto.org/history.html](https://agilemanifesto.org/history.html)



Innova GRD



SENAPRED
Ministerio del Interior
y Seguridad Pública

Gobierno de Chile