

MÉTODOS ÁGILES

ROLES, ARTEFACTO y CEREMONIAS

Agenda

1. SCRUM
2. Roles
3. Artefactos
4. Eventos

¿Qué es Scrum?

- Scrum es un **marco de trabajo** de adaptación iterativa e incremental, rápido, flexible y eficaz diseñado para ofrecer un valor significativo de forma rápida en todo el proyecto.
- Scrum es:
 - Ligero.
 - Fácil de entender.
 - Extremadamente difícil de llegar a dominar.
- Usos de Scrum
 - Scrum fue desarrollado inicialmente para gestionar y desarrollar productos. Desde principios de los **años 90**.
 - Scrum se ha usado para desarrollar software, hardware, software embebido, redes de funciones interactivas, vehículos autónomos, escuelas, gobiernos, mercadeo, también para gestionar la operación de organizaciones y casi todo lo que usamos en nuestra vida diaria, como individuo y como sociedad.
 - Scrum demostró ser especialmente efectivo en la transferencia iterativa e incremental de conocimiento.
 - Scrum se usa ahora ampliamente para productos, servicios y gestión de la organización matriz.

Scrum: técnico y scrum avanzado

Scrum técnico

Reglas



Marco de reglas para desarrollo de software

Autores: Ken Schwaber y Jeff Sutherland
"Scrum Development Process OOPSLA'95" 1995

Scrum avanzado

Valores



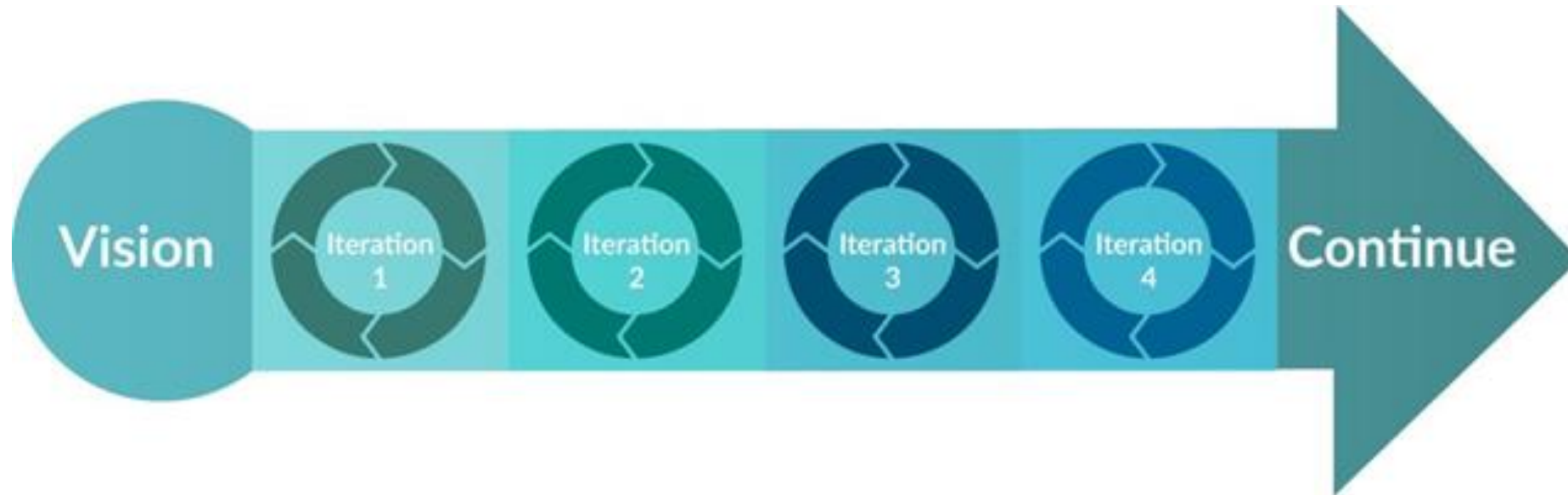
Concepto original scrum

Autores: Hirotaka Takeuchi e Ikujiro Nonaka
"The New New Product Development Game" 1986

Aplicación de reglas definidas	Aplicación de valores ágiles
<p>Roles</p> <ul style="list-style-type: none">▪ Dueño de producto▪ Equipo de desarrollo▪ Scrum Master <p>Eventos</p> <ul style="list-style-type: none">▪ El Sprint▪ Reunión de planificación▪ Scrum diario▪ Revisión de sprint▪ Retrospectiva de sprint <p>Artefactos</p> <ul style="list-style-type: none">▪ Pila de product▪ Pila de sprint▪ Incremento	<ul style="list-style-type: none">▪ Personas > procesos▪ Resultado > documentación▪ Colaboración > negociación▪ Cambio > planificación <p>... "Para avanzar en scrum"</p> <ul style="list-style-type: none">▪ Incertidumbre▪ Autoorganización▪ Fases de desarrollo solapadas▪ "Multiaprendizaje"▪ Control sutil▪ Difusion del conocimiento

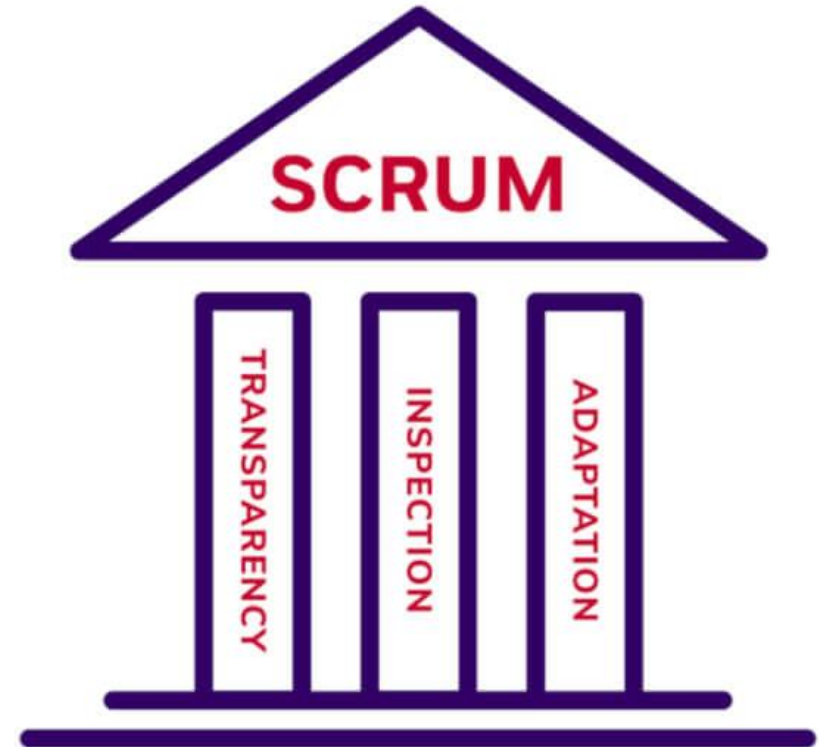
Iterativo

- Scrum se basa en la teoría de control de procesos empírica o empirismo.
- El empirismo asegura que el conocimiento procede de la experiencia y de tomar decisiones basándose
- en lo que se conoce.
- Scrum emplea un enfoque iterativo e incremental para optimizar la predictibilidad y el control del riesgo.



Tres Pilares de Scrum

- **Transparencia:** Visibilidad a los stakeholders
- **Inspección:** revisar el progreso y aprender
- **Adaptación:** Ajustar el proceso con lo aprendido



Transparencia

- Los aspectos significativos del proceso deben ser visibles para aquellos que son responsables del resultado.
- La transparencia requiere que dichos aspectos sean definidos por un estándar común, de tal modo que los observadores compartan un entendimiento común de lo que se están viendo.

Inspección

- Los usuarios de Scrum deben inspeccionar frecuentemente los artefactos de Scrum y el progreso hacia un objetivo para detectar variaciones indeseadas.
- Su inspección no debe ser tan frecuente como para que interfiera en el trabajo.
- Las inspecciones son más beneficiosas cuando se realizan de forma diligente por inspectores expertos en el mismo lugar de trabajo.



Adaptación

- Uno de los tres pilares del control del proceso empírico; la retroalimentación se usa para hacer un ajuste al producto de trabajo que se está desarrollando o al proceso por el cual se está desarrollando.

Los Valores de Scrum

- Compromiso (Equipo, organización)
- Coraje. (Motivación)
- Foco. (Meta)
- Apertura. (Cambios)
- Respeto. (entre todos)

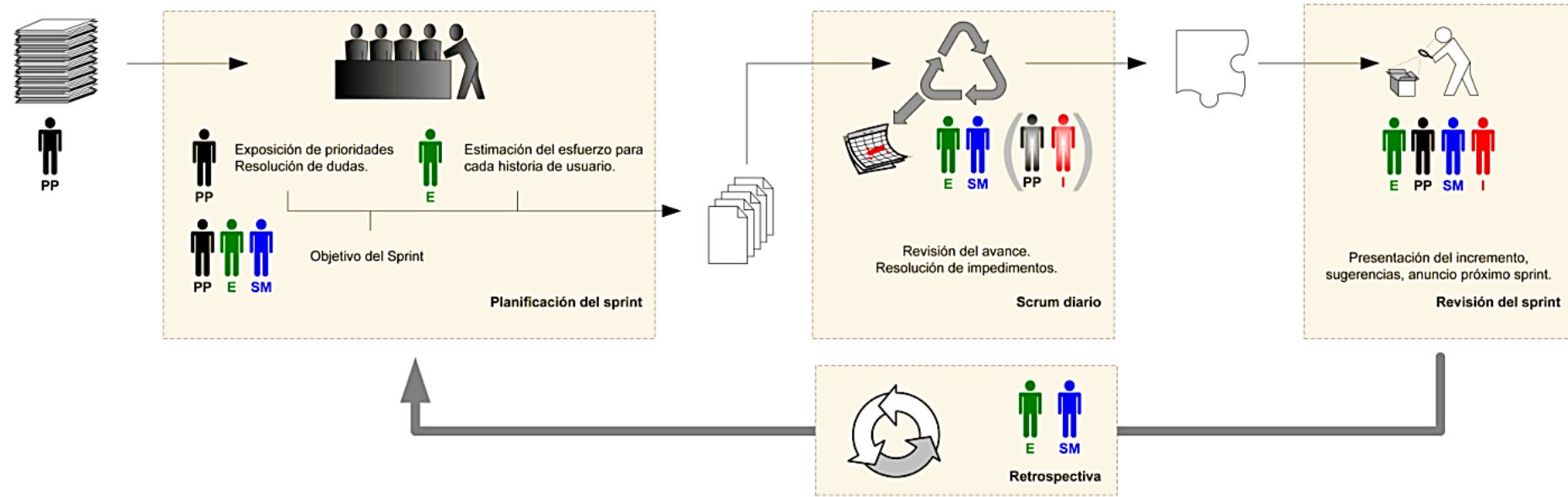
Los miembros del Scrum Team aprenden y exploran estos valores a medida que trabajan en los eventos, roles y artefactos de Scrum.



La Esencia de Scrum

- La esencia de Scrum es un **pequeño equipo de personas**. El equipo individual es altamente flexible y adaptativo. Estas fortalezas continúan operando en un equipo, en varios, en muchos y en redes de equipos que desarrollan, liberan, operan y mantienen el trabajo y los productos de trabajo de miles de personas.
- Cuando las palabras “develop” y “development” se usan en la Guía de Scrum se está haciendo referencia a trabajo complejo, tales como estos identificados anteriormente.

MARCO



ROLES



PROPIETARIO DEL PRODUCTO

Determina las prioridades.
Una sola persona.

EQUIPO DE DESARROLLO

Construye el producto.

SCRUM MASTER

Gestiona y facilita la ejecución de las reglas de Scrum

INTERESADOS

Resto de implicados. Asesoran y observan.

ARTEFACTOS



PILA DEL PRODUCTO

Relación de requisitos del producto, no es necesario excesivo detalle. Priorizados. Lista en evolución y abierta a todos los roles. El propietario del producto es su responsable y quien decide.



PILA DEL SPRINT

Requisitos comprometidos por el equipo para el sprint con nivel de detalle suficiente para su ejecución.



INCREMENTO

Parte del producto desarrollada en un sprint, en condiciones de ser usada (pruebas, codificación limpia y documentada).

EVENTOS



PLANIFICACIÓN DEL SPRINT

1 jornada de trabajo (máx.) El propietario del producto explica las prioridades. El equipo estima el esfuerzo de los requisitos prioritarios y se elabora la pila del sprint. El equipo define en una frase el objetivo del sprint.



SPRINT

Ciclo de desarrollo básico en el marco estándar de scrum, de duración recomendada inferior a un mes y nunca mayor de 6 semanas.



SCRUM DIARIO

15 minutos máximo. Responsabilidad del equipo. Cada miembro expone: Lo que hizo ayer. Lo que va a hacer hoy, si tiene o prevé problemas. Se actualiza la pila del sprint.



REVISIÓN DEL SPRINT

Informativa, máx. 4 horas, presentación del incremento, planteamiento de sugerencias y anuncio del próximo sprint.



RETROSPECTIVA

El equipo autoanaliza la forma de trabajo. Identificación de fortalezas y debilidades. Refuerzo de las primeras, plan de mejora de las segundas.

Roles

Roles

Un conjunto cohesivo de responsabilidades que pueden ser cumplidas por una o más personas. Los tres roles de Scrum son Product Owner, Scrum Master, y Development Team.



Scrum Team

- El Scrum Team consiste en un **Product Owner, el Development Team y un Scrum Master.**
- El Scrum Team son autoorganizados y multifuncionales.
- El modelo de equipo en Scrum está diseñado para optimizar la flexibilidad, la creatividad y la productividad.
- El Scrum Team ha demostrado ser cada vez más efectivo para todos los usos anteriores y cualquier trabajo complejo.
- El Scrum Team entregan productos de forma iterativa e incremental, maximizando las oportunidades de obtener retroalimentación.
- Las entregas incrementales del producto "Done" aseguran que siempre estará disponible una versión potencialmente útil y funcional del producto.



Product Owner

El Product Owner (PO) representa la voz del cliente, y es el encargado de maximizar el valor del producto.

- Un PO siempre debe mantener una visión dual.
- El debe entender y apoyar las necesidades e intereses de todos los Stakeholders.
- Comprende las necesidades y el funcionamiento del Development Team.



Responsabilidades del Product Owner



Responsabilidades del Product Owner

Ayudar en la elección
del Scrum Master y
de los miembros del
Development Team

Responsable por la
administración del
Product Backlog

Ayudar a crear y a
aprobar los User
Story

Explicar los User
Story al
Development Team

Definir los criterios
de aceptación

Participar en la
retrospectiva del
Sprint y el proyecto



UNIVERSIDAD
DE LIMA



Características de un Product Owner



Scrum Master

- El Scrum Master es responsable de promover y apoyar Scrum como se define en la Guía de Scrum.
- Los Scrum Masters hacen esto ayudando a todos a entender la teoría, prácticas, reglas y valores de Scrum.
- El Scrum Master es un líder que está al servicio del Scrum Team. El Scrum Master ayuda a las personas externas al Scrum Team a entender qué interacciones con el Scrum Team pueden ser útiles y cuáles no.
- El Scrum Master ayuda a todos a modificar estas interacciones para maximizar el valor creado por el Scrum Team.



Responsabilidades del Scrum Master con el Product Owner



Responsabilidades del Scrum Master con la Organización



Responsabilidades del Scrum Master con el Development Team



Development Team

- El Development Team consiste en los profesionales que realizan el trabajo de entregar un Incremento de producto "Done" que potencialmente se pueda poner en producción al final de cada Sprint. Un Incremento "Done" es obligatorio en el Sprint Review. Solo los miembros del Development Team participan en la creación del Incremento. La organización es la encargada de estructurar y empoderar a los Development Teams para que estos organicen y gestionen su propio trabajo. La sinergia resultante optimiza la eficiencia y efectividad del Development Team.
- El tamaño óptimo del Development Team es lo suficientemente pequeño como para permanecer ágil y lo suficientemente grande como para completar una cantidad de trabajo significativa. Tener menos de tres miembros en el Development Team reduce la interacción y resulta en ganancias de productividad más pequeñas.
- Los Development Teams más pequeños podrían encontrar limitaciones en cuanto a las habilidades necesarias durante un Sprint, haciendo que el Development Team no pudiese entregar un Incremento que potencialmente se pueda poner en producción. Tener más de nueve miembros en el equipo requiere demasiada coordinación.

Responsabilidades del Development Team



Responsabilidades del Development Team



Características Development Team

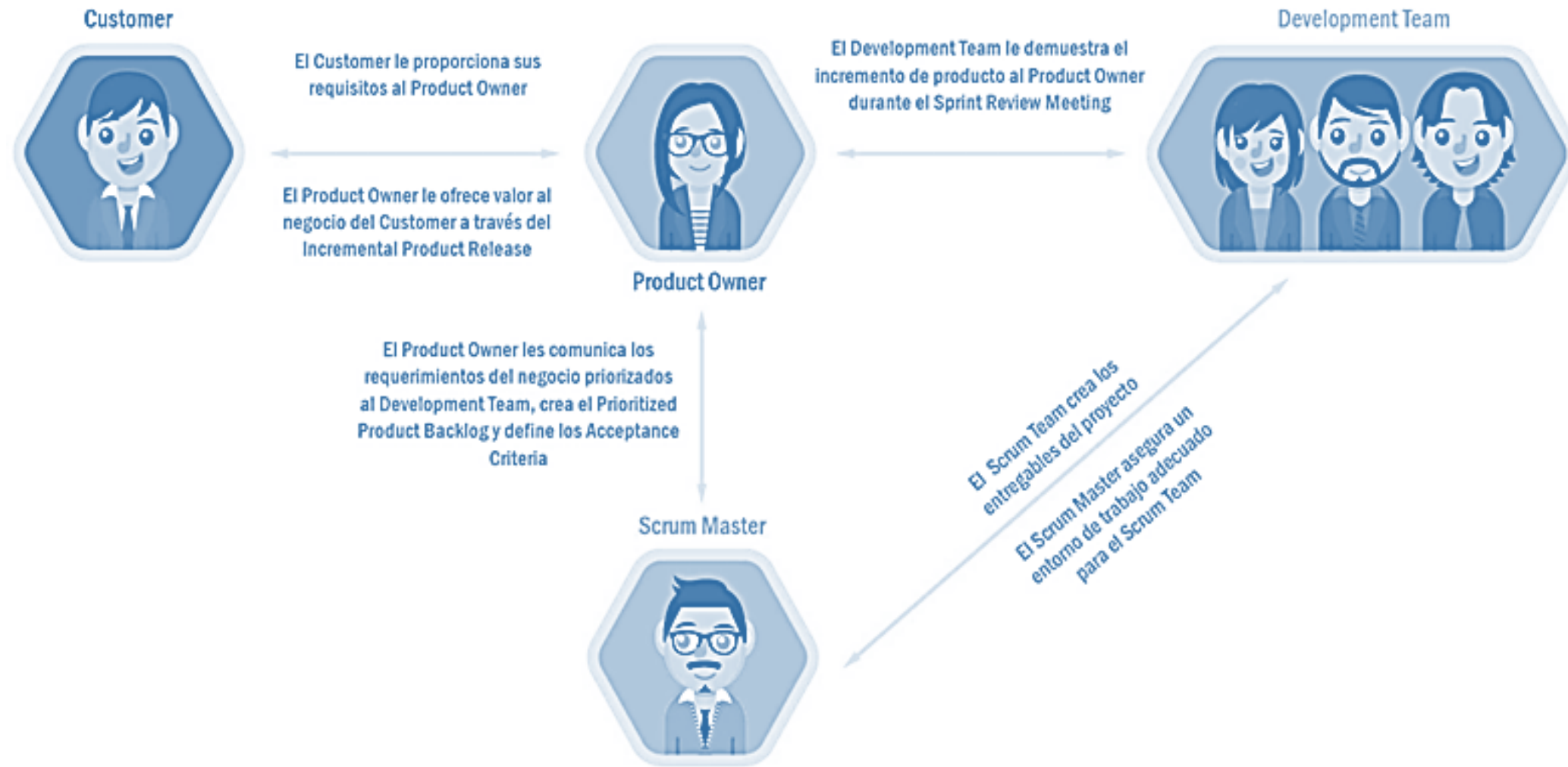


Stakeholders

Una persona, grupo u organización que afecta o puede verse afectado por las acciones de una organización.

Stakeholders se Dividen en:

- **Cliente:** El cliente es la persona o la organización que adquiere el producto del proyecto, servicio o cualquier otro resultado.
- **Usuarios:** El usuario es el individuo o la organización que utiliza directamente el producto del proyecto, servicio, o cualquier otro resultado; también, en algunas industrias el cliente y los usuarios puede ser lo mismo.
- **Patrocinador:** El patrocinador es la persona o la organización que provee recursos y apoyo para el proyecto, el patrocinador también es el Stakeholder a quien todos le deben rendir cuentas al final.



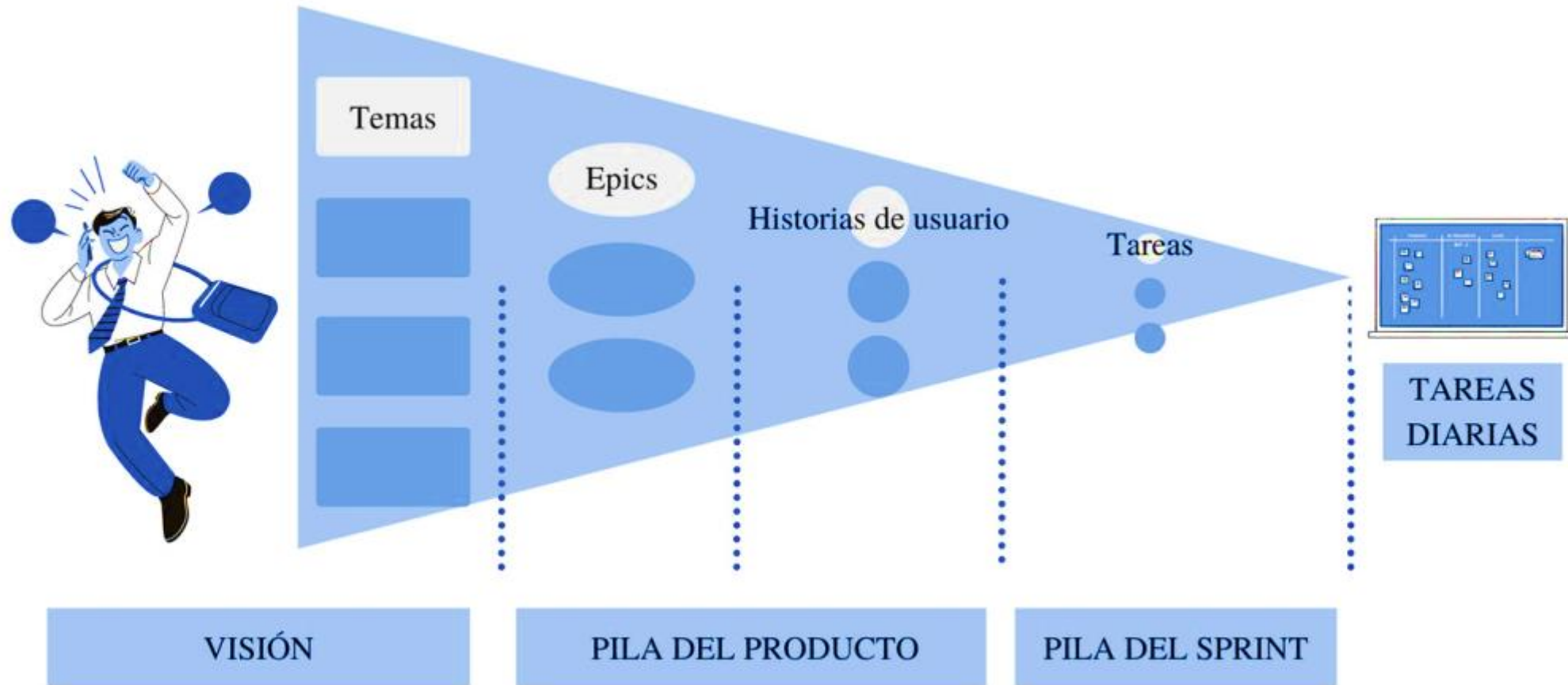
Artefactos

Conceptos Claves

- **Épicas:** Es una historia de usuario que es demasiado grande para caber en un sprint. A menudo, este término se utiliza para describir una gran historia de usuario que tendrá que ser dividido en historias más pequeñas.
- **User Stories:** Es una representación de un requisito del usuario en forma escrita, de una o dos frases, utilizando el lenguaje común del usuario.
- **Task:** Es una representación del requisito que está en lenguaje del usuario, pero de una forma técnica donde está definido cómo se va a trabajar y quiénes van a participar.



Requisitos ágiles

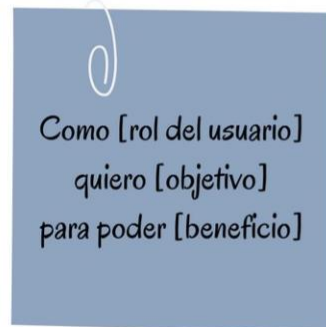


¿Cómo está conformada una User Story?

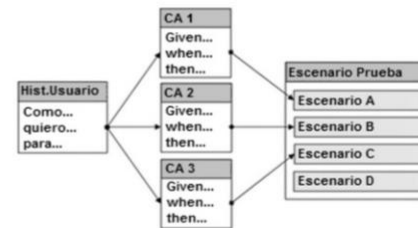
Una historia de usuario debe estar conformada por las 3 C's:

- **Card (tarjeta):** Descripción escrita de lo que necesita el usuario.
- **Conversación:** El PO y el DT aclaran los detalles.
- **Confirmación:** Sirve para determinar lo que se espera.

Card → Conversation → Confirmation



Planificación de sprint
Refinamiento



UNIVERSIDAD
DE LIMA



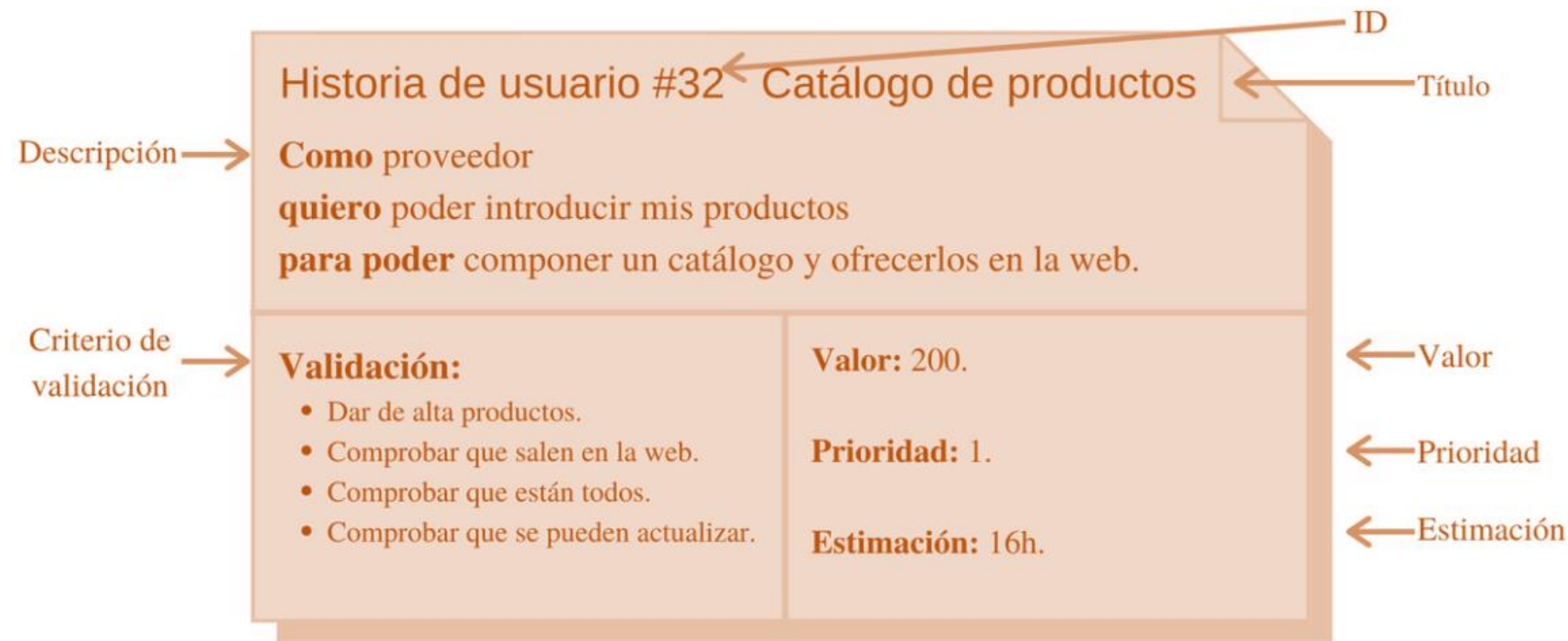
Características: modelo INVEST



- **Independiente:** poder planificar e implementar las historias de usuario en cualquier orden tiene muchas ventajas y facilita el trabajo posterior del equipo. Para que sea posible, cada historia debería ser independiente. Una forma de reducir las dependencias es combinar historias o dividir las de manera diferente.
- **Negociable:** una historia de usuario es una descripción corta de una necesidad, sin detalles. Deben ser negociables ya que los pormenores serán acordados con el cliente o el usuario más adelante. Conviene mantener las historias en términos generales o vagos para no limitar estas conversaciones.
- **Valiosa:** una historia de usuario tiene que aportar valor al cliente o al usuario. Una manera de conseguirlo es que sean ellos quienes la escriban.
- **Estimable:** una buena historia de usuario debe poder estimarse con precisión suficiente para que el propietario del producto pueda priorizar y planificar su implementación. El equipo es quien suele realizar la estimación, y ésta se verá afectada por el tamaño de la historia, pues a mayor tamaño mayor incertidumbre, y por el conocimiento del equipo sobre la necesidad expresada. En caso de falta de conocimiento, serán necesarias más fases de conversación.
- **Pequeña:** las historias de usuario deberían englobar como mucho unas pocas semanas de trabajo por persona. Incluso hay equipos que las restringen a días por persona. Una descripción corta ayuda a disminuir el tamaño de una historia de usuario, facilitando así su estimación.
- **Comprobable:** la historia de usuario debería poder probarse en la fase de confirmación. Si el cliente o usuario no sabe cómo, significa que la funcionalidad no es del todo clara o que no es valiosa, y si el equipo no puede probarla es imposible que sepa si está terminada o no.



Estructura de User Story



Descripción

Como [rol del usuario], **quiero** [objetivo], **para poder** [beneficio].

- **Como [rol del usuario]:** ¿para quién estamos construyendo? No nos referimos a un cargo o un rol profesional, sino a la persona tras la necesidad descrita, según esté representada en su user persona. El equipo debe tener un entendimiento compartido de quién es, cómo trabaja, cómo piensa y siente; en definitiva, sentir empatía.
- **Quiero [objetivo]:** ¿qué es lo que está intentando conseguir realmente? Es importante recordar que en este campo se explica su intención, no la funcionalidad que usará.
- **Para poder [beneficio]:** ¿cómo encaja su necesidad inmediata en lo que le rodea? ¿cuál es el beneficio general que está tratando de lograr? ¿cuál es el gran problema que necesita resolver? Se trata de un beneficio que va relacionado con la visión del producto.

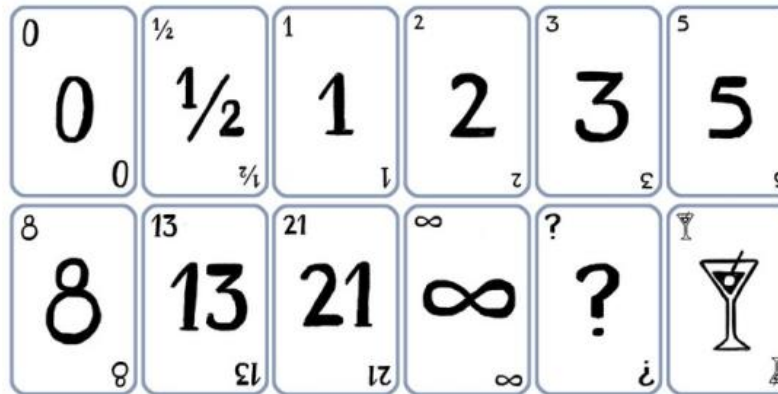
Estimación

Estimar el esfuerzo ayuda al propietario de producto a **priorizar** y al equipo a decidir qué historias de la pila caben en el **sprint**.

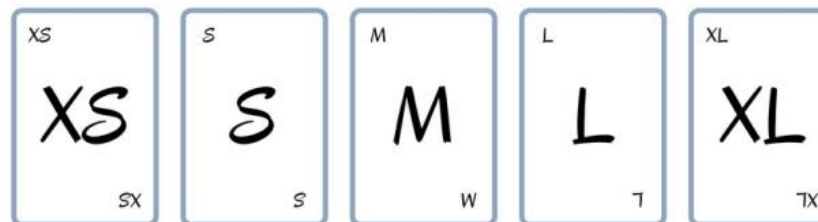
Un hecho a considerar es que la incertidumbre para estimar historias de usuario **pequeñas es mucho menor que para historias grandes**. Además, aunque el tamaño de las historias crece linealmente, la incertidumbre lo hace exponencialmente.

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...

- **0** para historias que requieren un esfuerzo casi nulo
- **1/2** para historias muy pequeñas.
- Algunas barajas sustituyen el 21 por un 20, ya que decir que una historia tiene un tamaño de 21 da una falsa sensación de precisión.
- A partir del 21 las barajas suelen pasar a incluir un símbolo de **infinito** (∞)
- se siguen, incluyen valores redondos como 40 y 100.



La distancia entre los números de la serie refleja el hecho de que la incertidumbre inherente a la estimación crece con el tamaño de la historia.



UNIVERSIDAD
DE LIMA



Estimación Planning Póker

Esta es una de las técnicas más reconocidas en Scrum, ya que es muy sencilla, divertida y eficaz, donde el Development Team (DT) estima como grupo el esfuerzo a realizar en el Sprint.



Prioridad

Aunque todas las historias de usuario puedan ser importantes, para focalizarnos de forma eficiente hay que destacar **las que den más valor al sistema**. El propietario del producto debe asignar un valor a cada historia que intervenga en el sistema de priorización, teniendo en cuenta las siguientes variables:

- Beneficios de implementar la funcionalidad.
- Pérdida o coste que derive de posponer su implementación.
- Riesgos de implementarla.
- Coherencia con los intereses del negocio.
- Valor diferencial con respecto a productos de la competencia.

técnica *MoSCoW*

- *Must have*, es necesario: la funcionalidad debe estar implementada en la solución, si no ésta fallará o no podrá considerarse un éxito.
- *Should have*, es recomendable: se debería implementar, ya que es una funcionalidad de alta prioridad. La solución no fallará si no existe, pero debería haber causas justificadas.
- *Could have*, podría implementarse: es deseable, sería conveniente tener esta funcionalidad en la solución, pero dependerá de los tiempos y el presupuesto del proyecto.
- *Won't have*, no lo queremos por ahora: se trata de una funcionalidad de muy baja prioridad o descartada, pero que en el futuro puede ser relevante. En tal caso pasaría a alguno de los otros estados

Criterios de aceptación

Después de años de historia de ingeniería de software se ha llegado a la conclusión de que los criterios de aceptación, que a veces se traducen en pruebas, son un excelente **lenguaje para detallar requerimientos funcionales**, y es por ello que toman una gran importancia en las historias de usuario.

Para medir la calidad de un criterio de aceptación se utiliza el método SMART, por el que se han de ser, en la medida de lo posible:



Criterios de aceptación

Una opción excelente es escribirlos con la técnica del comportamiento por escenarios propia de BDD (Behavior Driven Development) y con **gherkin**, un lenguaje creado para las descripciones de comportamiento de software. La sintaxis de gherkin es la siguiente:

(Scenario) **Escenario** [número de escenario] [título del escenario]:
(Given) **Dado que** [contexto] **y adicionalmente** [contexto],
(When) **Cuando** [evento],
(Then) **Entonces** [resultado / comportamiento esperado].

Historia de usuario

Como cliente
quiero retirar dinero del cajero
para poder evitar ir al banco a hacer cola.

Criterios de aceptación

Escenario 1: cuenta tiene crédito.

Dado que la cuenta tiene crédito
Y que la tarjeta es válida
Y que el cajero tiene dinero disponible
Cuando el cliente pide dinero
Entonces la cuenta es debitada
Y el dinero es entregado al cliente
Y el cliente recupera su tarjeta

Escenario 2: la cuenta excede el límite negativo acordado con el banco.

Dado que la cuenta excede el límite negativo acordado con el banco
Y que la tarjeta es válida
Cuando el cliente pide dinero
Entonces el cajero muestra un mensaje negando el pedido
Y el dinero no es entregado al cliente
Y el cliente recupera su tarjeta



UNIVERSIDAD
DE LIMA



Definición de Done

Son los acuerdos del PO con los Stakeholders que contiene todas las condiciones que deben de cumplir los ítems del Product Backlog para considerar un Sprint completado o finalizado.



Definición de hecho

La definición de hecho o DoD es un artefacto de scrum que responde la pregunta **¿qué tiene que cumplir una historia de usuario para que esté hecha?** Es una herramienta relacionada con la calidad del software y que tiene que ser puesta en común por el equipo y el propietario del producto.

- Consiste en una serie de criterios y acciones que agregan valor verificable y demostrable al producto.
- La definición de hecho es un documento que explica qué es necesario para considerar una historia como hecha.
- Permite que todos los implicados compartan esta definición.
- Diferenciarla de los criterios de aceptación, que son específicos para cada historia.

Definición de hecho

A nivel de sprint se suele incluir que la historia de usuario cumpla con los criterios de aceptación.

Tarea:

- Implementada.
- El código escrito cumple estilos y directrices.
- Hechas las pruebas unitarias.
- Integrada en el repositorio.
- Integra con el resto en el *build*.
- Gestor de tareas actualizado.
- Satisfechas ciertas métricas: cobertura, análisis estático, etc.
- Aprobada por el *tester*.

Release:

- Todas las historias planificadas están hechas.
- Medios de distribución producidos.
- Comprobada la documentación de usuario.
- Cumple con los requerimientos no funcionales y estándares.
- Instrucciones de instalación/despliegue de la release hechas.
- Pruebas de seguridad, despliegue y regresión hechas.
- Incidencias críticas y de resolución necesaria para despliegue resueltas.
- Comunicación de release y formación a usuarios hechas.
- Aprobada por el propietario del producto.

Historia de usuario:

- Todas las tareas asociadas están hechas.
- Incidencias detectadas en fase de desarrollo resueltas.
- Código documentado y/o comentado.
- Documentación y otros requisitos del proyecto/producto hechos.
- Satisface los criterios de aceptación.
- Pruebas de integración hechas.
- Superadas las pruebas unitarias acumuladas y de aceptación de forma manual o automática.
- Sigue los estándares de ingeniería/arquitectura.
- Cumple con los requerimientos no funcionales.
- Está instalada en un entorno tipo *test*, *staging* o preproducción y ha pasado las pruebas de humo.
- Aprobada por el propietario del producto.

Sprint:

- Todas las historias planificadas están hechas.
- Hecha la revisión de sprint y obtenido feedback de los interesados.
- Hecha la retrospectiva e identificada una acción de mejora.
- Aprobado por el propietario del producto.



Task



- En Scrum se puede definir como el trabajo técnico que realiza el Development Team para completar un ítem del Product Backlog.
- La mayoría de las tareas se definen como pequeñas, lo que representa no más de unas pocas horas de un día.

¿Cómo está conformada una Task?

Características modelo **SMART**:

S: Specific (Específico).

M: Measurable (Medible).

A: Achievable (Alcanzable).

R: Relevant (Relevante).

T: Time-boxed (Bloque de tiempo).

Identificación:

Responsable:

Descripción:

Estimación:



UNIVERSIDAD
DE LIMA



Artefactos

Los artefactos de Scrum representan trabajo o valor en diversas formas que son útiles para proporcionar transparencia y oportunidades para la inspección y adaptación.

Los artefactos definidos por Scrum están diseñados específicamente para maximizar la transparencia de la información clave, necesaria para asegurar que todos tengan el mismo entendimiento del artefacto.

- Product Backlog.
- Sprint Backlog.
- Increment.

Product Backlog

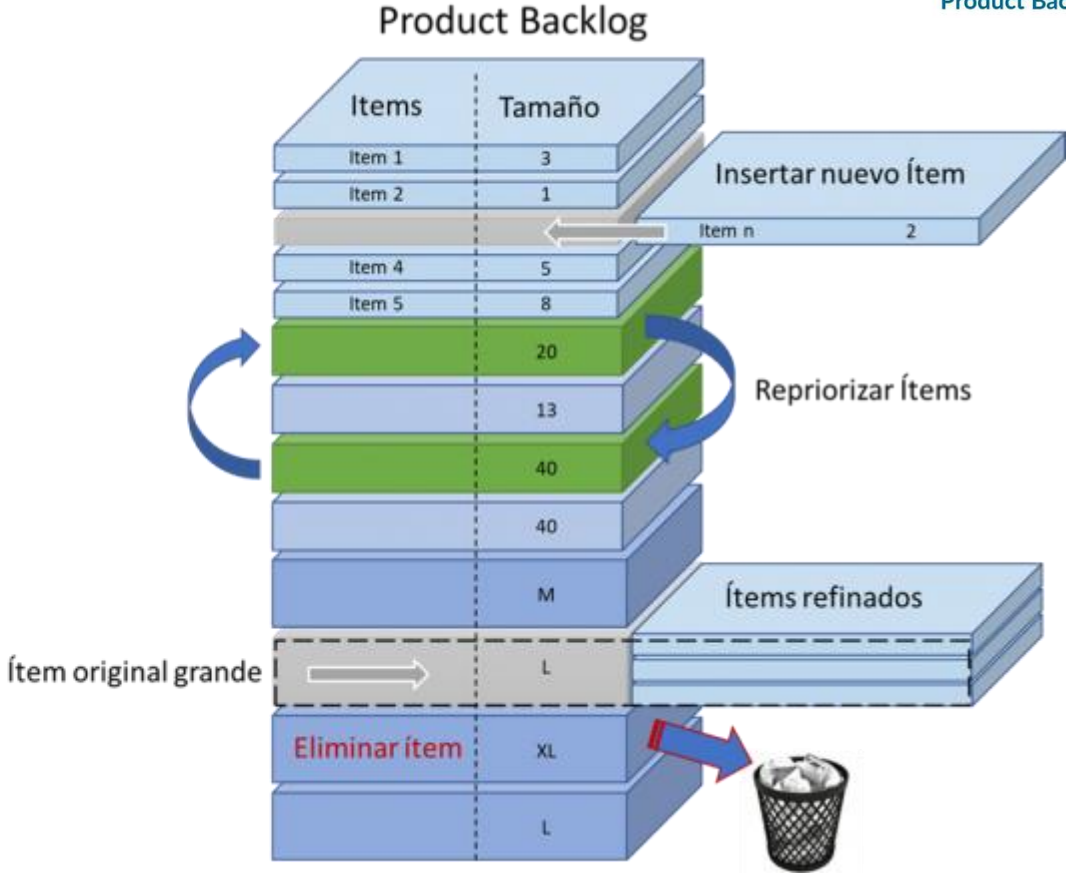
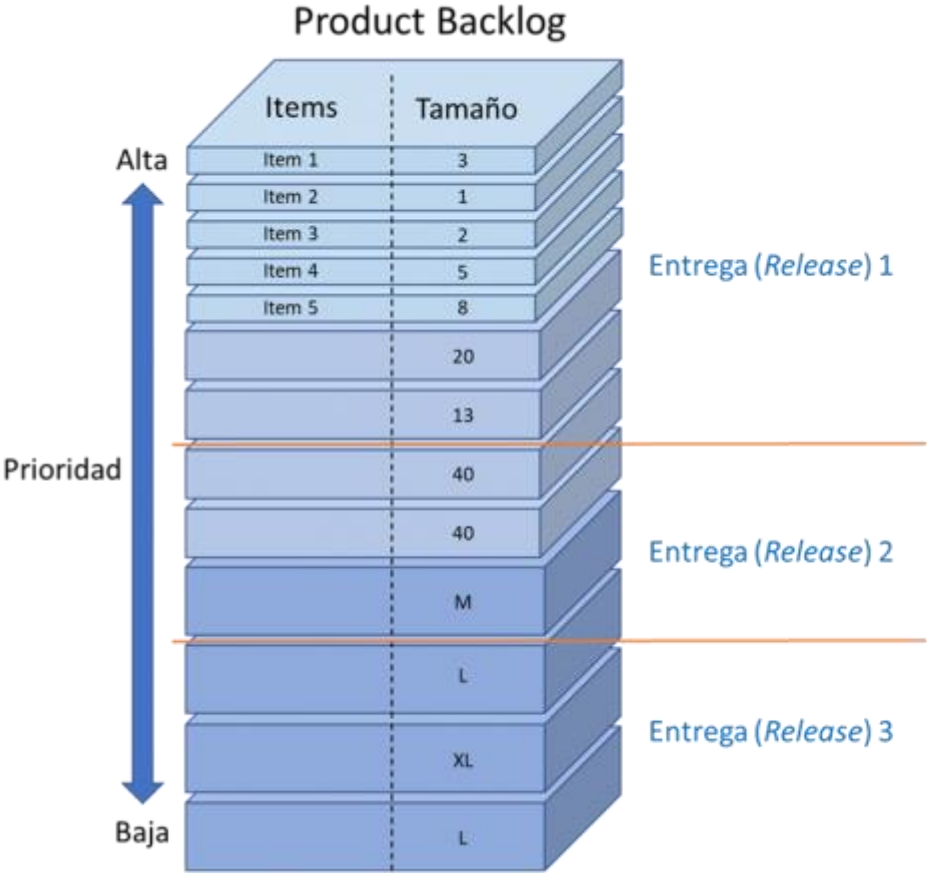
- El Product Backlog es una lista ordenada de todo lo que se conoce que es necesario en el producto.
- Es la única fuente de requisitos para cualquier cambio a realizarse en el producto. El Product Owner es el responsable del Product Backlog, incluyendo su contenido, disponibilidad y ordenación.
- El Product Backlog es dinámica; cambia constantemente para identificar lo que el producto necesita para ser adecuado, competitivo y útil.
- El Product Backlog enumera todas las características, funcionalidades, requisitos, mejoras y correcciones que constituyen cambios a realizarse sobre el producto para entregas futuras.

PROYECTO			Sáb 07 Ene	Dom 08 Ene	Lun 09 Ene	Mar 10 Ene	Mié 11 Ene
Inicio	Fin	Jornada					
7-ene-12	1-abr-12	40 hs					
Tareas pendientes			15	15	14	14	11
Horas pendientes			172	162	148	142	124
Fecha de Cierre			12 ene	12 ene	12 ene	13 ene	16 ene

PILA DEL PRODUCTO					OBJETIVO DEL SPRINT				
Categoría	Tarea	Responsable	Estimado en horas	Estado	Crear y publicar versión básica del sitio web público				
Diseño	Crear diseño de base de datos	Juan	24	Completo	24	16	8	4	
Diseño	Validar diseño de base de datos	Pedro	4	Completo	4	4	4	4	
Desarrollo	Contratar servicio de hosting	Pedro	4	Completo	4	2			
Desarrollo	Crear layout y estilos de sitio web	Maria	16	Activo	8	8	4	2	
Desarrollo	Crear página principal	Maria	24	Pendiente	24	24	24	24	24
Desarrollo	Mostrar resúmenes de noticias por sección	Juan	16	Pendiente	16	16	16	16	8
Desarrollo	Crear banners de publicidad	Luis	24	Pendiente	24	24	24	24	24
Desarrollo	Visualizar un Artículo	Luis	8	Pendiente	8	8	8	8	8
Desarrollo	Imprimir un Artículo	Luis	4	Pendiente	4	4	4	4	4



Product Backlog



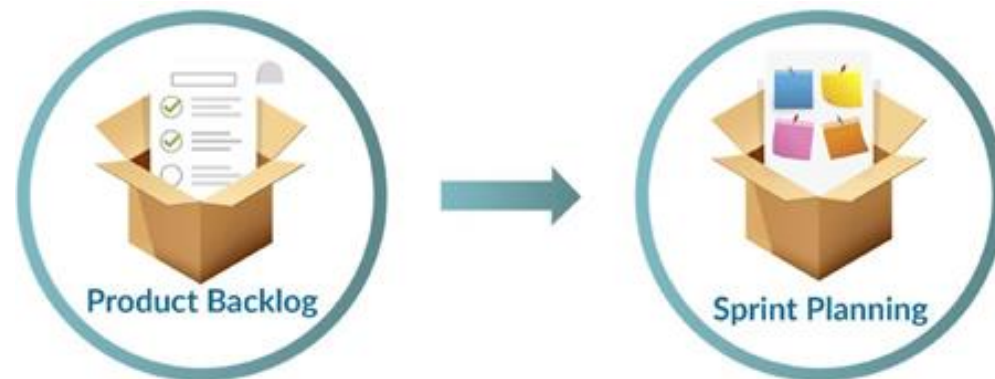
Refinamiento del Product Backlog

- **El refinamiento** (refinement) del Product Backlog es el acto de **añadir detalle, estimaciones y orden** a los elementos del Product Backlog.
- Durante el refinamiento del Product Backlog, se examinan y revisan sus elementos.
- El Scrum Team decide cómo y cuándo se hace el refinamiento.
- Este usualmente consume no más del 10% de la capacidad del Development Team. Sin embargo, los elementos del Product Backlog pueden actualizarse en cualquier momento por el criterio del Product Owner.



Sprint Backlog

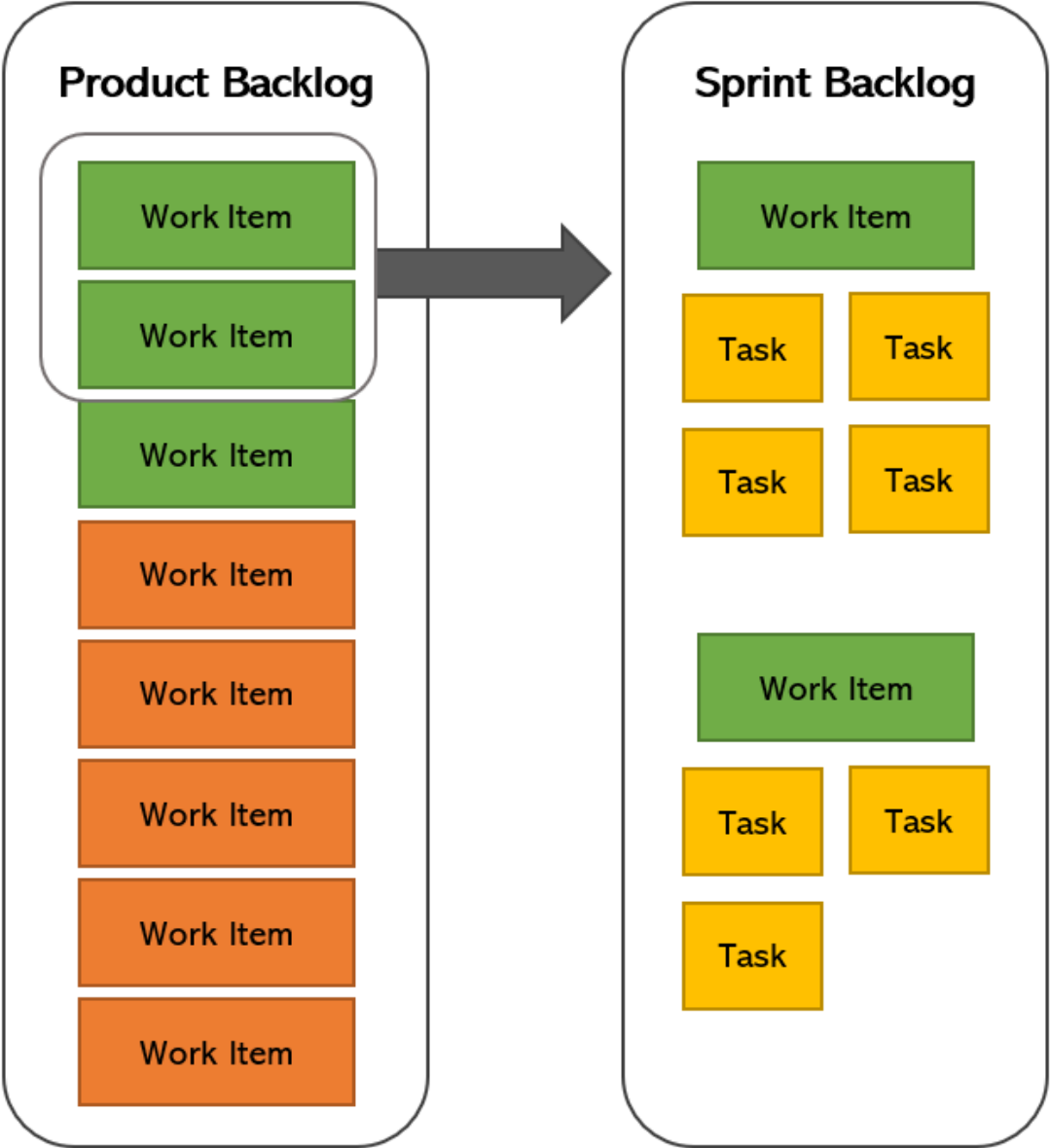
- El Sprint Backlog es el **conjunto de elementos** del Product Backlog seleccionados para el Sprint, más un plan para entregar el Incremento de producto y conseguir el Sprint Goal.
- El Sprint Backlog es una predicción hecha por el Development Team acerca de qué funcionalidad formará parte del próximo Incremento y del trabajo necesario para entregar esa funcionalidad en un Incremento “Done”.
- El Sprint Backlog del Sprint hace visible todo el trabajo que el Development Team identifica como necesario para alcanzar el Sprint Goal.



UNIVERSIDAD
DE LIMA



Sprint Backlog



SPRINT																
INICIO			DURACIÓN													
1			1-mar-07 12													

SPRINT BACKLOG																
Tarea	Estado	Responsal	ESFUERZO													
Descripción de la tarea 1	Terminada	Luis	16	16	16	16	16	16								
Descripción de la tarea 2	Terminada	Luis	12	8												
Descripción de la tarea 3	Terminada	Luis	4	4	4	4	4									
Descripción de la tarea 4	Terminada	Elena	8	4												
Descripción de la tarea 5	Terminada	Elena	16	16	4											
Descripción de la tarea 6	Terminada	Elena	6	6	2											
Descripción de la tarea 7	Terminada	Antonio	16	4												
Descripción de la tarea 8	Terminada	Antonio	16	16	20	12	4									
Descripción de la tarea 9	Terminada	Antonio	12	2												
Descripción de la tarea 10	En curso	Luis	12	12	12	12	12	12	12	12	12	12	12	12	12	
Descripción de la tarea 11	Pendiente	Luis	8	8	8	8	8	8	8	8	8	8	8	8	8	
Descripción de la tarea 12	Terminada	Luis	14	14	14	14	14	14								
Descripción de la tarea 13	En curso	Antonio	8	8	8	8	8	6								
Descripción de la tarea 14	Pendiente	Antonio	16	16	16	16	16	16	16	16	16	16	16	16	16	
Descripción de la tarea 15	Pendiente	Antonio	16	16	16	16	16	16	16	16	16	16	16	16	16	
Descripción de la tarea 16	Terminada	Elena	8	8	8											
Descripción de la tarea 17	Terminada	Elena	12	12	12	8	4									
Descripción de la tarea 18	En curso	Elena	16	16	16	16	16	10	10	10	10	10	10	10	10	
Descripción de la tarea 19	Terminada	Elena	12	12	12	12	12									
Descripción de la tarea 20	Pendiente	Elena	16	16	16	16	16	16	16	16	16	16	16	16	16	
Descripción de la tarea 21	Pendiente	Elena	12	12	12	12	12	12	12	12	12	12	12	12	12	
Descripción de la tarea 22	Pendiente	Antonio	8	8	8	8	8	8	8	8	8	8	8	8	8	
Descripción de la tarea 23	Pendiente	Antonio	12	12	12	12	12	12	12	12	12	12	12	12	12	

Priorización

- El Product Owner es una única persona, no un comité.
- El Product Owner podría representar los deseos de un comité en el Product Backlog, pero aquellos que quieran cambiar la prioridad de un elemento de la Lista deben hacerlo a través el Product Owner.
- Para que el Product Owner pueda hacer bien su trabajo, toda la organización **debe respetar sus decisiones**.
- Las decisiones del Product Owner se reflejan en el contenido y en la priorización del Product Backlog.
- Nadie puede forzar al Development Team a que trabaje con base en un conjunto diferente de requisitos.



Seguimiento

- Seguimiento del Progreso Hacia los Objetivos.
- Seguimiento del Progreso del Sprint.

Sprint Burn Down Chart



Sprint Burn Up Chart



Increment

- **El Incremento** es la suma de todos los elementos del Product Backlog completados durante un Sprint y el valor de los incrementos de todos los Sprints anteriores.
- Al final de un Sprint el nuevo Increment debe estar “Done”.
- Un Increment es un cuerpo de trabajo inspeccionable y terminado que respalda el empirismo al final del Sprint.
- El Increment debe estar en condiciones de utilizarse sin importar si el Product Owner decide liberarlo o no



Eventos

Time-Boxing

Todos los eventos son bloques de tiempo (duración máxima de tiempo), de tal modo que todos tienen una duración máxima.



Ventajas de Time-Boxing

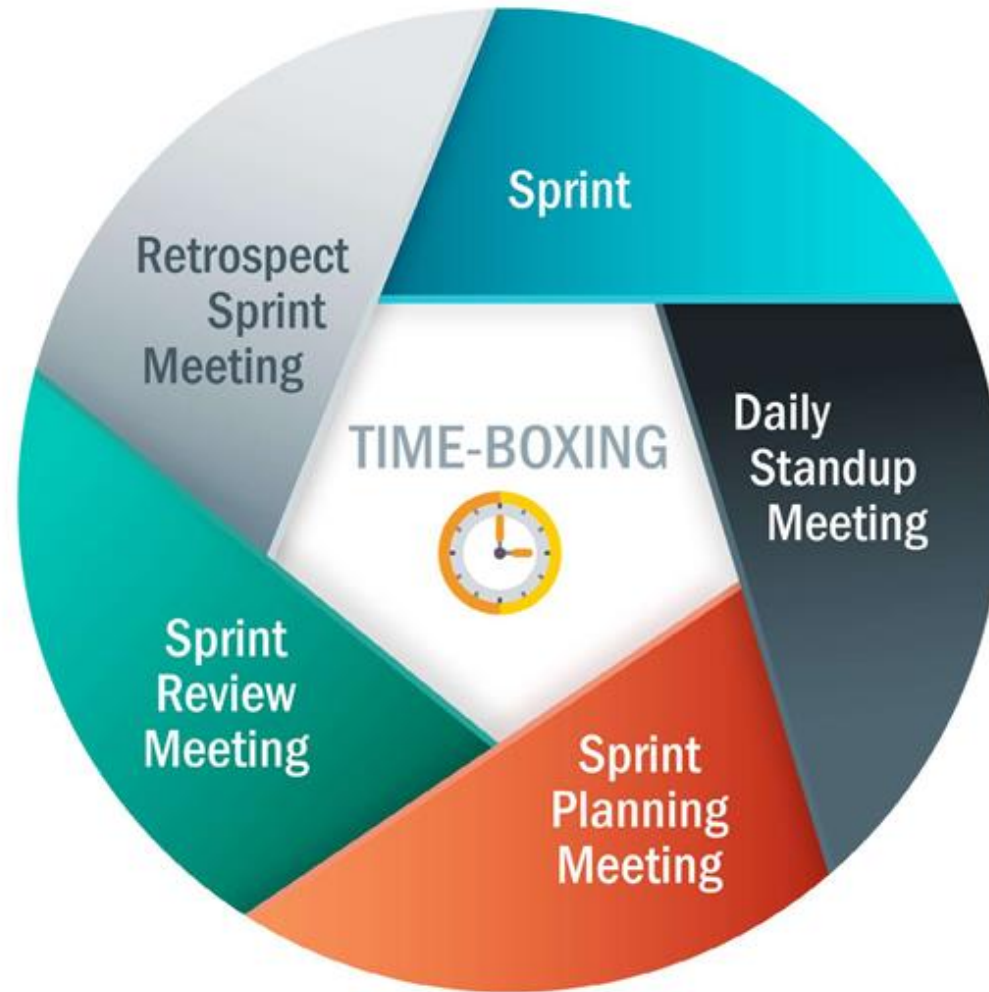
Time-Boxing es una práctica crítica en Scrum y debe aplicarse con cuidado.

Un **Time-Boxing** arbitrario puede llevar a la desmotivación del equipo y puede tener como consecuencia la creación de un entorno opresivo, por lo que Time-Boxing debe ser utilizado de manera apropiada.

Beneficios:

- Procesos de desarrollo eficiente.
- Menos gastos generales.
- Alta velocidad para los equipos.
- Ayuda a gestionar eficazmente la planificación y ejecución de proyectos.

¿Dónde se utilizan los Time-Boxing?



Eventos Formales

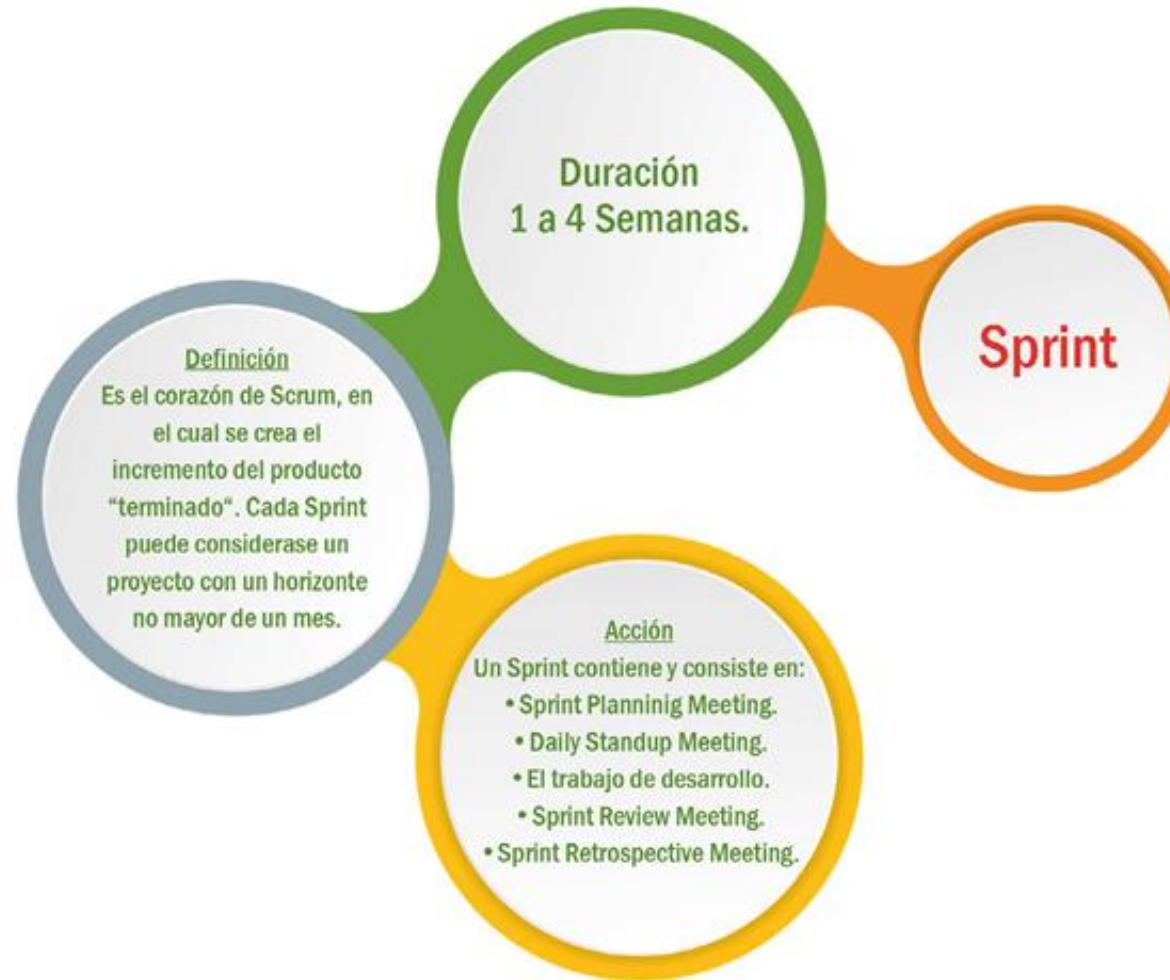
Scrum prescribe cuatro eventos formales, contenidos dentro del Sprint, para la inspección y adaptación:

- Planificación del Sprint (**Sprint Planning**).
- Scrum Diario (**Daily Scrum**).
- Revisión del Sprint (**Sprint Review**).
- Retrospectiva del Sprint (**Sprint Retrospective**).

Eventos Formales

Scrum Event	Time-box (for 1-month Sprint)	Participants
Sprint Planning	8 Hours	Scrum Master, Product Owner, Development Team
Daily Scrum	15 Minutes	Scrum Master (optional), Product Owner (optional), Development Team
Sprint Review	4 Hours	Scrum Master, Product Owner, Development Team and all key stakeholders
Sprint Retrospective	3 Hours	Scrum Master, Product Owner, Development Team

Sprint



Cancelación de un Sprint

Un Sprint puede ser cancelado antes de que el Time-Box llegue a su fin, siempre y cuando el objetivo del Sprint llegara a quedar obsoleto o no tiene sentido seguir con el Sprint. Solo el PO tiene la autoridad para cancelar el Sprint.



Reuniones o Ceremonias de Scrum

Para que cualquier proyecto tenga éxito, la comunicación es importante. Los Scrum Teams emplean una serie de reuniones clave para estructurar el trabajo del equipo:

- Daily Standup Meeting.
- Sprint Planning Meeting.
- Sprint Review Meeting.
- Retrospective Sprint Meeting.

Reunión Diaria (DailySprint)

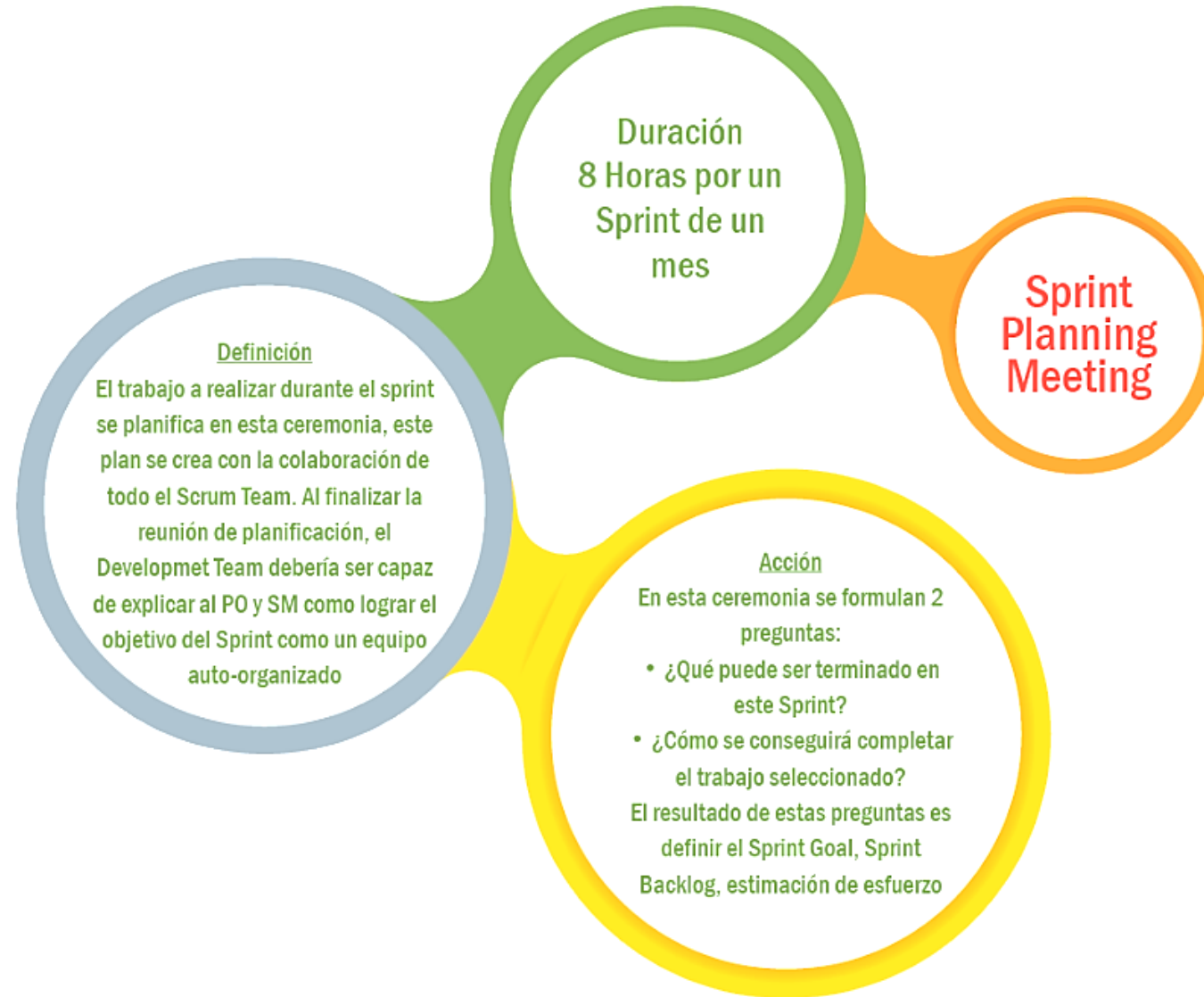


Daily Scrum

- Punto de inspección y adaptación en Scrum. **Máx. 15 minutos.**
- El equipo se reúne para comunicar y entender los estados.
- Esencial para conocer el progreso continuo y evitar bloqueos.
- No tiene como objetivo reportar progreso al Scrum Master, Product Owner o cualquier otro stakeholder.
- El Product Owner podrá participar siempre y cuando su participación sea pasiva.
- El Scrummaster se asegura de que el Development Team mantenga la reunión, pero el Development Team es el responsable de dirigir el Scrum Diario.



Sprint Planning Meeting



¿Qué puede ser terminado?

- Esta pregunta nos ayuda para que el Development Team (DT) trabaje para proyectar la funcionalidad que se desarrollará durante el Sprint, donde se define objetivo del Sprint (Sprint Goal).
- El número de elementos del Product Backlog seleccionados para el Sprint depende únicamente del Development Team (DT).



Sprint Goal

- El Sprint Goal es una meta establecida para el Sprint que puede lograrse mediante la implementación del Product Backlog.
- El Sprint Goal brinda al Development Team cierta flexibilidad con respecto a la funcionalidad implementada en el Sprint.
- El Sprint Goal puede representar otro nexo de unión que haga que el Development Team trabaje en conjunto y no en iniciativas separadas.



SPRINT GOAL

Why are you doing this iteration?



UNIVERSIDAD
DE LIMA

60 años

¿Cómo se conseguirá completar el trabajo seleccionado?

- Una vez que se ha establecido el Sprint Goal y seleccionado los elementos del Product Backlog para el Sprint, el Development Team decide cómo construirá esta funcionalidad para formar un Incremento de producto “Done” durante el Sprint.
- Los elementos del Product Backlog seleccionados para este Sprint, más el plan para terminarlos, recibe el nombre de Sprint Backlog.

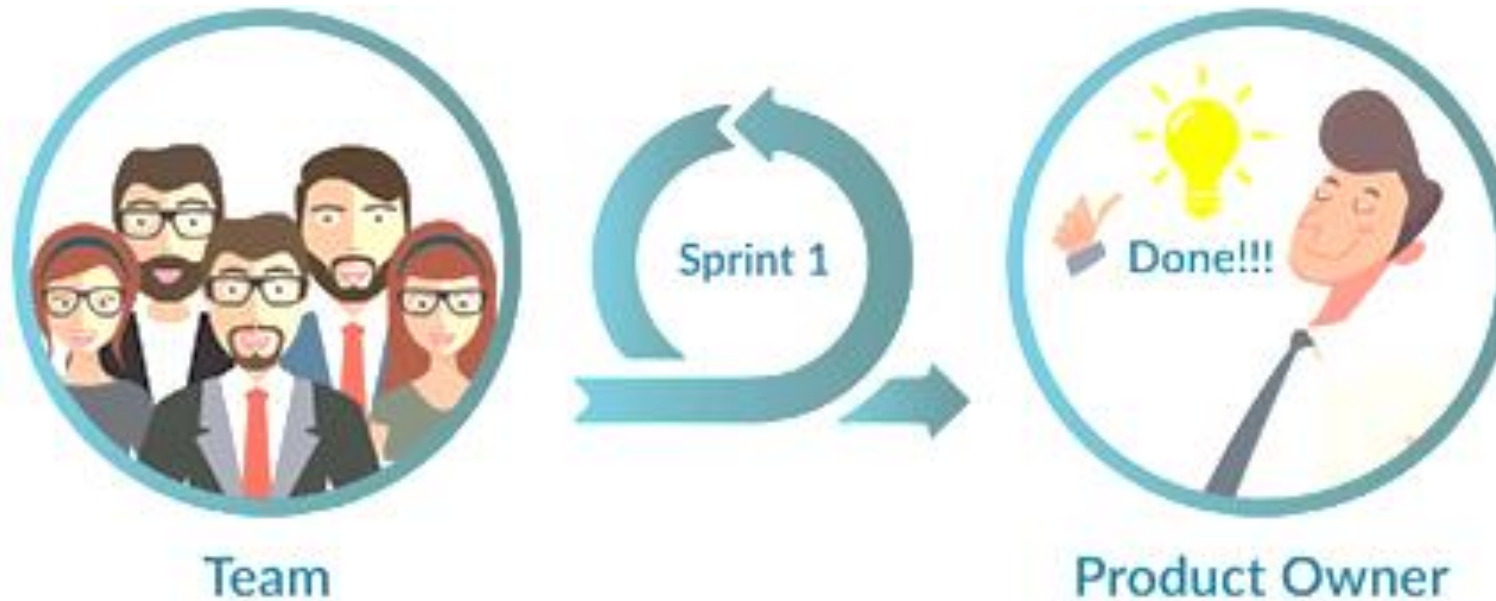


Sprint Review Meeting

- Los asistentes son el Scrum Team y los stakeholders claves invitados por el Product Owner.
- El Product Owner explica qué elementos del Product Backlog se han “Done” y cuales no se han “Done”.
- El Development Team habla acerca de qué estuvo bien durante el Sprint, qué problemas aparecieron y cómo fueron resueltos esos problemas.
- El Development Team hace una demostración del trabajo que ha “Done” y responde preguntas acerca del Incremento.
- El Product Owner habla acerca del Product Backlog en su estado actual. Proyecta objetivos probables y fechas de entrega en el tiempo basándose en el progreso obtenido hasta la fecha (si fuera necesario).
- El grupo completo colabora acerca de qué hacer a continuación, de modo que el Sprint Review proporcione información de entrada valiosa para el subsiguiente Sprint Planning.
- Revisión de cómo el mercado o el uso potencial del producto podría haber cambiado lo que es de más valor para hacer a continuación.
- Revisión de la línea de tiempo, presupuesto, capacidades potenciales y mercado para las próximas entregas de funcionalidad o capacidad prevista del producto.



Sprint Review Meeting



Sprint Retrospective

- El Sprint Retrospective es una oportunidad para el Scrum Team de inspeccionarse a sí mismo y de crear un plan de mejoras que sean abordadas durante el siguiente Sprint.
- El Sprint Retrospective tiene lugar después del Sprint Review y antes del siguiente Sprint Planning.
- Se trata de una reunión de, a lo sumo, tres horas para Sprints de un mes.

El propósito del Sprint Retrospective es:

- Inspeccionar cómo fue el último Sprint en cuanto a personas, relaciones, procesos y herramientas.
- Identificar y ordenar los elementos más importantes que salieron bien y las posibles mejoras.
- Crear un plan para implementar las mejoras a la forma en la que el Scrum Team desempeña su trabajo.

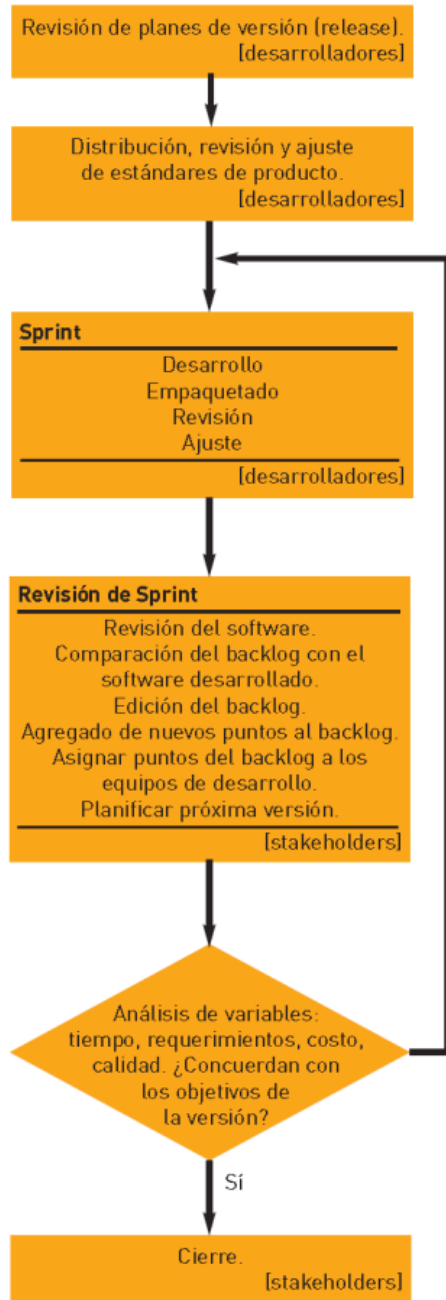


Las 5 Etapas de una Retrospectiva



The 5 Stages of a Retrospective





REVISIÓN DE PLANES DE VERSIÓN:

Se revisa que hay que hacer y en que punto está la distribución actual.

SPRINT:

Es la fase de desarrollo iterativa.

Desarrollo: Análisis, implementación, testing.

Empaquetar: Generar paquetes ejecutables

Revisión: Resolución de problemas y se añaden nuevos ítems.

Ajustes: Uso de los ajustes para mejorar el producto.

SPRINT REVIEW:

Después del Sprint se hace una reunión con el ScrumMaster donde se revisa el producto del Sprint anterior y en el que se pueden añadir puntos nuevos al backlog.

Cierre:

En esta fase se encuentran las típicas actividades de fin de proyecto como, hacer una versión distribuible, testear, marketing etc....

Diagrama fases de Scrum

