

Ingeniería de software ágil 1

UT2. Marcos de gestión ágiles

Depto. de Ingeniería de Software - Facultad de Ingeniería
Universidad ORT Uruguay

Temario

- Scrum:
 - Funcionamiento general.
 - Roles, ceremonias, artefactos.
- Scrum en un eje de tiempo.
- Tableros Scrum.
- Definition of Ready y Definition of Done.
- Dual-track agile/scrum.



Scrum

Scrum

- Es un marco de trabajo para la gestión de proyectos, más que una metodología.
- Proporciona:
 - claridad de responsabilidades a través de roles.
 - visibilidad a través de artefactos.
 - oportunidades de inspección y adaptación a través de eventos o “ceremonias”.
- Dentro de esta estructura, Scrum es un contenedor para otros procesos y herramientas que son apropiados para satisfacer las necesidades específicas de un equipo, organización o producto.

Scrum – Roles, artefactos, ceremonias

- Proporciona un marco para "hacer" ingeniería de software sin prescribir las técnicas de ingeniería que deben utilizarse.
- Roles:
 - Product Owner, Scrum Master, Development Team.
- Artefactos:
 - Product backlog, Sprint backlog, Incremento.
- Ceremonias:
 - Sprint, Sprint planning, Daily scrum, Sprint review, Sprint retrospective.

Scrum – algo de terminología

- Producto:
 - El producto de software que está desarrollando el equipo Scrum.
- Product Owner:
 - Miembro del equipo responsable de identificar las características y atributos del producto.
 - Representa a todos los interesados (*stakeholders*) del proyecto.
 - Revisa el trabajo realizado y ayuda a probar el producto.
 - Es una persona, no un comité, y es un miembro dedicado y de tiempo completo del equipo scrum.

Scrum – algo de terminología

- Product backlog:
 - Lista de tareas pendientes tales como errores para corregir, características (*features*) y mejoras al producto que el equipo de Scrum aún no ha completado.
- Equipo de desarrollo:
 - Un pequeño equipo autoorganizado de cinco a ocho personas que son responsables del desarrollo del producto.
- Sprint:
 - Un período corto, generalmente de dos a cuatro semanas, durante en cual se desarrolla un incremento de producto.

Scrum – algo de terminología

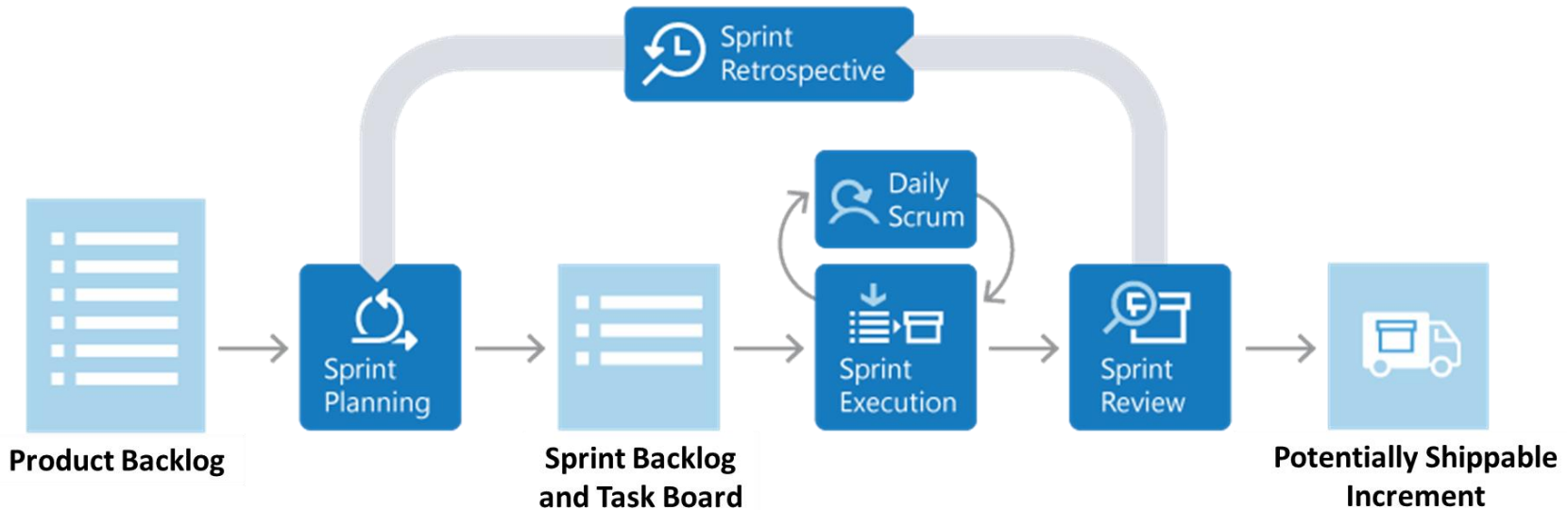
- Daily Scrum:
 - una reunión diaria del equipo donde se revisa el progreso y se discute y acuerda el trabajo a realizar ese día.
- Scrum Master:
 - un entrenador (coach) de equipo que guía al equipo en el uso efectivo de Scrum.
- Incremento de producto potencialmente entregable:
 - Potentially shippable product increment.
 - el resultado de un sprint que es de calidad lo suficientemente alta como para ser implementado para uso del cliente.

Scrum – algo de terminología

- Velocity o Velocidad:
 - Una estimación de cuánto trabajo puede hacer un equipo en un solo sprint.

Scrum – Esquema general

• ..



Scrum

- La idea fundamental que subyace al proceso Scrum es que el software debe desarrollarse en una serie de *sprints*.
- Un *sprint* es una actividad de duración fija, usualmente de dos a cuatro semanas.
- Durante un *sprint*, el equipo tiene reuniones diarias (*daily scrums*) para revisar el trabajo realizado hasta el momento y acordar las actividades de ese día.
- La *sprint backlog* se utiliza para realizar un seguimiento del trabajo que se realizará durante ese sprint.

Scrum

- La planificación del *sprint* se basa en el *product backlog*, que es una lista de todas las actividades que deben completarse para terminar el producto que se está desarrollando.
- Antes de que comience un nuevo sprint, se revisa el *product backlog*.
- Los elementos de mayor prioridad se seleccionan para su implementación en el *sprint* que va a comenzar.

Scrum

- Los miembros del equipo trabajan juntos para planificar el *sprint* analizando los elementos seleccionados del *product backlog* para crear el *sprint backlog*.
- Este es una lista de actividades que se completarán durante el *sprint*.
- Durante la implementación, el equipo implementa tantos elementos *sprint backlog* como sea posible en el período de tiempo fijo asignado para el *sprint*.

Scrum

- Los elementos incompletos del *sprint backlog* se devuelven al *product backlog*.
- Los *sprints* nunca se extienden para terminar un elemento incompleto.
- Un *sprint* produce:
 - un incremento entregable de producto (*shippable product increment*) que se puede entregar a los clientes
 - un producto interno.

Scrum

- Los entregables internos, como un prototipo de producto o un diseño arquitectónico, brindan información para futuros *sprints*.
- Si el resultado del *sprint* es parte del producto final, debe estar completo.

Scrum

- Al finalizar un *sprint*, se lleva a cabo una reunión de revisión (*sprint review*) en la que participan todos los miembros del equipo.
- El propósito de la *sprint review* es que el *product owner* obtenga comentarios del cliente/usuario sobre si el producto se está desarrollando en la dirección correcta.
- Esta revisión también es una oportunidad para que el equipo de desarrollo muestre lo que ha logrado. Los miembros del equipo obtienen todo el crédito por lo que han logrado y por lo que no.

Scrum

- Esta reunión al final de cada iteración garantiza que las partes interesadas:
 - estén al día sobre lo que se logró en la iteración
 - tengan un ámbito para hacer comentarios (*feedback*) directamente al *producto owner*, con el equipo de desarrollo escuchando.
 - tengan un producto funcionando.

Scrum

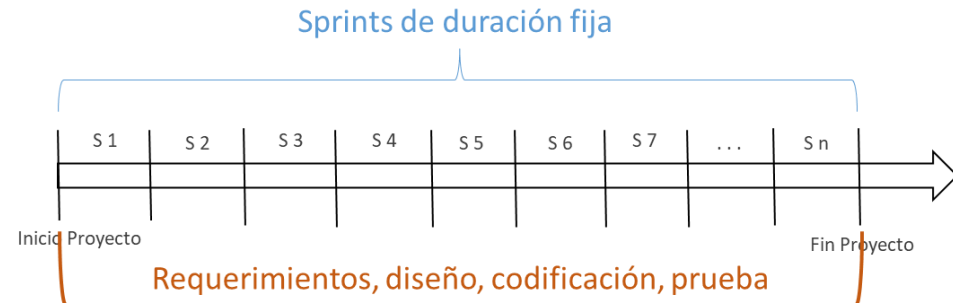
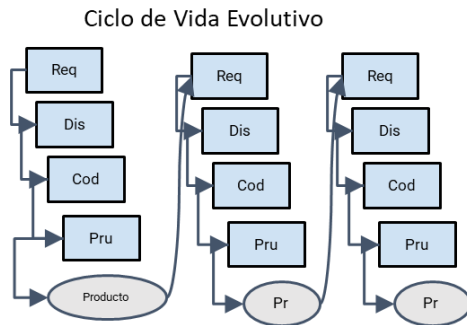
- Al finalizar cada *sprint* se lleva a cabo la *sprint retrospective*.
- Su propósito es tener una oportunidad para que el equipo de scrum (*scrum master*, *product owner* y equipo de desarrollo) evalúe qué salió bien en el sprint que acaba de terminar y qué se puede mejorar.
- Es un proceso de inspección y adaptación, con un enfoque en las personas, los procesos y las herramientas que utiliza el equipo scrum.
- El resultado de la retrospectiva debe ser un plan de acción para mejorar continuamente...



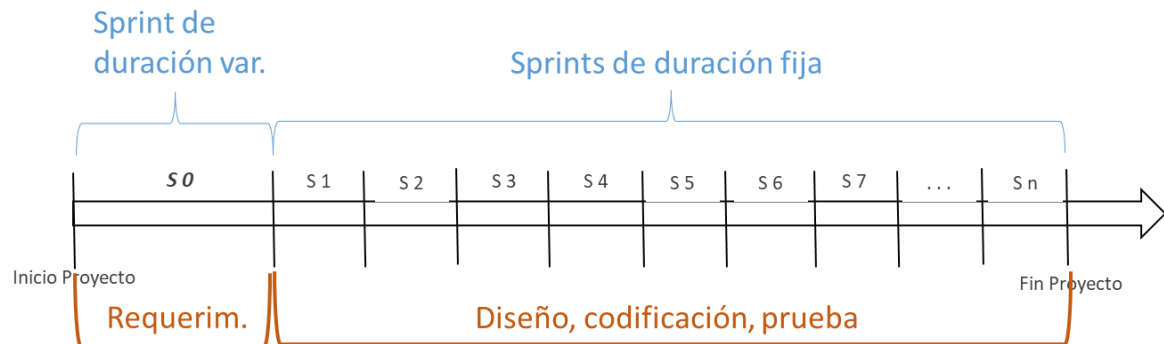
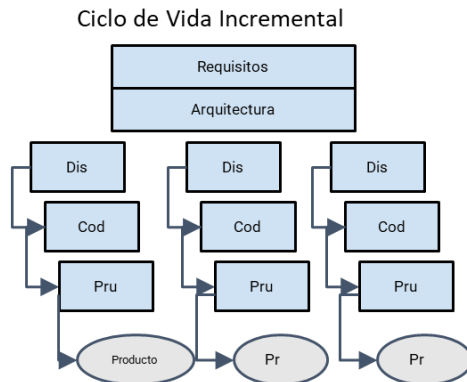
Scrum en un eje de tiempo

Scrum en un eje de tiempo

- Ciclo de vida EVOLUTIVO



- Ciclo de vida INCREMENTAL





Tableros Scrum

Tablero Scrum

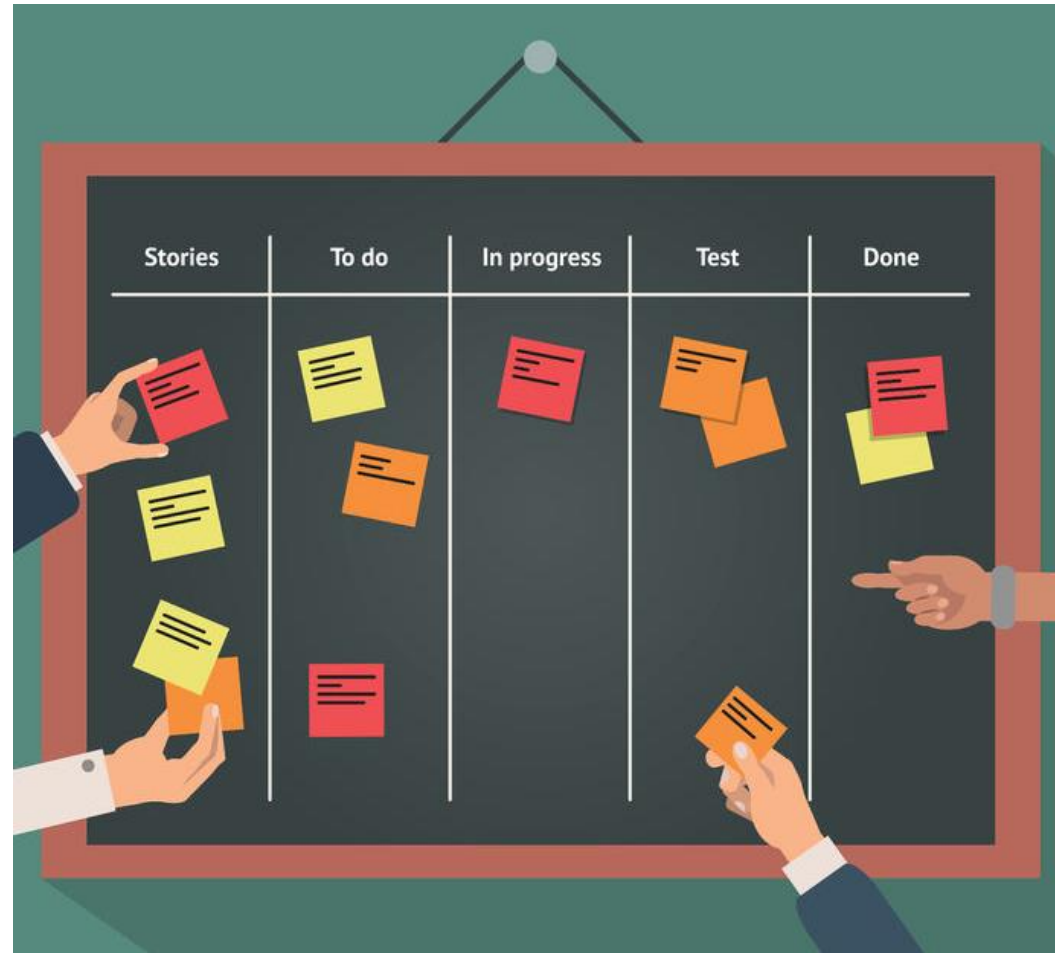
- Un tablero de Scrum se usa para rastrear el trabajo en las iteraciones denominadas *sprints*.
- Antes de que se complete el *sprint*, el objetivo es mover todas las tareas a la columna Listo/Terminado/Done.
- No hay un formato establecido para configurar un tablero de Scrum.
- Se deja al equipo decidir cómo presentar la información que se necesita.

Tablero Scrum

- Por lo general, el tablero se divide en columnas que el equipo usa para rastrear el progreso del trabajo acordado que se completará en el sprint; es decir, el *sprint backlog*.
- Un tablero de Scrum puede incluir estas columnas:
 - **To-Do / Tareas pendientes:** el backlog priorizado de las características del producto planificadas para el sprint actual.
 - **In progress / En curso:** la lista de tareas que se han iniciado.
 - **In test / En prueba:** tareas completadas que se están probando para su verificación.
 - **Done / Terminado:** tareas que se han completado y verificado mediante pruebas.

Tablero Scrum

• ...



Tablero Scrum

- El trabajo al que se le da seguimiento en un tablero de Scrum se limita a la cantidad de tareas que el equipo se ha comprometido a completar en la iteración actual.
- No se pueden agregar elementos nuevos a la lista de **Tareas pendientes** hasta que todo el trabajo llegue a la columna **Terminado**.
- Todos los miembros del equipo deben estar al tanto de:
 - trabajo que se está realizando.
 - los miembros del equipo que tienen asignado ese trabajo.
 - el progreso de ese trabajo.
 - los logros del equipo.

Tablero Scrum

- Un tablero Scrum:
 - **Promueve la interacción y el debate en equipo.** Los miembros del equipo y los interesados miran el tablero a lo largo del día para discutir el progreso y priorizar el trabajo.
 - **Hace que la información sea visual y de fácil acceso.** Ya sea que el tablero sea físico o virtual, la información que muestra es fácil de entender y de fácil acceso.
 - Cualquiera que mire el tablero puede evaluar rápidamente dónde se encuentra el equipo en la iteración y qué es lo que aún debe hacerse para lograr los objetivos.
 - **Apoya el compromiso de todo el equipo.** Cuando el equipo ve todas las tareas en el tablero, evita que se concentren solo en tareas individuales. A medida que se completan las tareas individuales, aquellos con tiempo disponible pueden asumir tareas adicionales.

Tablero Scrum

- Los tableros de Scrum están diseñados específicamente para completar historias de usuarios y tareas específicas dentro de un período de tiempo específico; el *sprint*.
- Es un recordatorio diario del objetivo del sprint y de las tareas que el equipo se ha comprometido a completar.
- Es una herramienta visual para organizar tareas y realizar un seguimiento del progreso.

Tablero Scrum

- En el *daily scrum meeting* los miembros del equipo miran el tablero para mover las tareas que se puedan mover de una columna a la siguiente a medida que se informa el progreso.
- Esta es una manera de ayudar al equipo a mantenerse enfocado en el objetivo y a comprender quién está haciendo qué.



Tablero Scrum

Sprint Backlog

Producto Web de Compra de Libros	Pendiente	En Progreso	Finalizado
Nº Sprint: 04		 	 
Objetivo del Sprint <i>El objetivo de este Sprint es que el usuario pueda completar una búsqueda de libros por autor y ordenarla por precio de compra ascendente y descendente, así como por año de publicación.</i>			   
	 	 	 
	 		
	   		

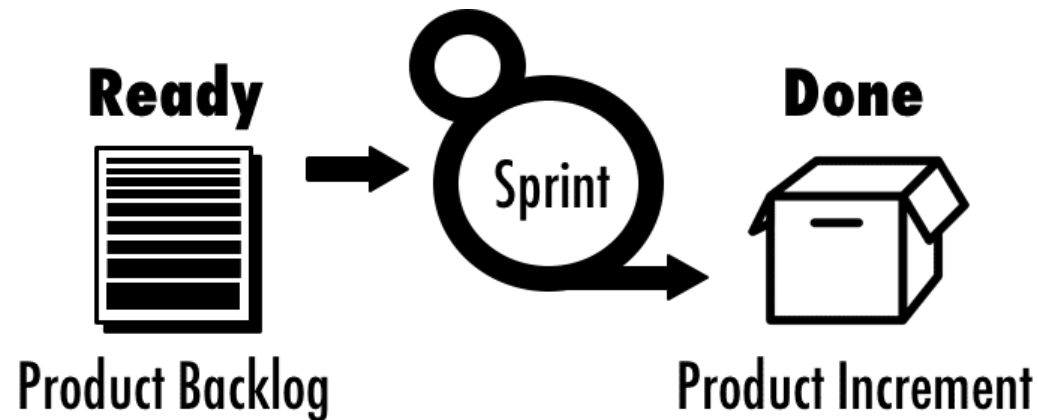


Definition of Ready

Definition of Done

DoR y DoD

- *Definition of Ready*: condiciones que debe cumplir una historia de usuario para poder “entrar” a un sprint.
- *Definition of Done*: condiciones que debe cumplir un incremento del producto (usualmente una historia de usuario) para considerarse Terminado.



Definición de Ready

- Una Definición de Listo / Definition of Ready permite a un equipo especificar ciertas condiciones previas que deben cumplirse antes de que se permita que una historia de usuario entre en una iteración.
- Si las historias de usuario en las que es probable que se trabaje en el próximo sprint no están listas, entonces el equipo tendrá dificultades para:
 - establecer un compromiso o pronóstico realista en la reunión de planificación del sprint y cumplir por completo el objetivo del sprint.

Definición de Ready

- ¿Qué significa que una historia de usuario está “ready”?
- Claridad:
 - Una historia es clara si todos los miembros del equipo Scrum tienen un entendimiento compartido de lo que significa.
 - Escribir historias de usuarios de forma colaborativa y agregar criterios de aceptación a las de alta prioridad facilita la claridad.
 - Los equipos de desarrollo que son nuevos en el producto o dominio suelen necesitar historias más claras y detalladas.
 - A medida que aumenta el conocimiento de los miembros del equipo, las historias pueden volverse menos detalladas.

Definición de Ready

- ¿Qué significa que una historia de usuario está “ready”?
- Comprobable / Testable:
 - Una historia de usuario se puede probar si existe una forma eficaz de determinar si la funcionalidad funciona como se esperaba.
 - Los **criterios de aceptación** aseguran que cada historia pueda ser probada.
- Factible:
 - Una historia de usuario es factible si se puede completar en un sprint, de acuerdo con la Definición de Hecho / Definition of Done.

Definición de Ready

- Ejemplos:
 - La historia de usuario tiene definidos sus *criterios de aceptación*.
 - Su tamaño ha sido definido.
 - Sus dependencias han sido identificadas.

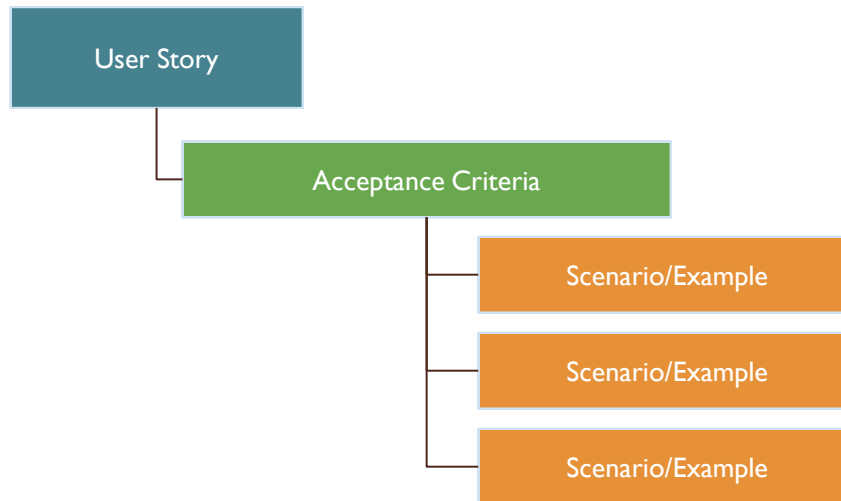
Definición de Done

- Una Definición de Hecho / Definition of Done permite a un equipo especificar ciertas condiciones que deben cumplirse antes de que se considere que una historia de usuario está completada antes de finalizar el sprint.
- Es un entendimiento compartido de lo que significa que una tarea o una historia está “Terminada”.
- Que el equipo cumpla los criterios del DoD asegura que se están entregando tareas que están realmente hechas, no sólo en términos de funcionalidad, sino también en términos de calidad.

Definición de Done

- Ejemplos:
 - El código escrito cumple con el estándar de codificación (ha sido revisado).
 - Todas las pruebas unitarias son correctas.
 - Todos los errores funcionales identificados han sido corregidos.
 - Se cumplen todos los *criterios de aceptación*.

Criterios de aceptación



Título: (una línea que describe la Historia de usuario)

Narrativa:

Como [rol]

Quiero [función o feature]

Para [valor o beneficio]

Criterios de Aceptación:
(presentados como Escenarios)

Escenario 1: Título

Dado [contexto]

Y [más contexto]...

Cuando [evento]

Entonces [resultado]

Y [otros resultados]...

Escenario 2: ...

Criterios de aceptación: escenario genérico

Título: Transferencia de fondos entre cuentas.

Como cliente del banco

Quiero transferir fondos entre mis cuentas cuando necesite

Para poder manejar mi dinero de manera más eficientemente.

Escenario: Transferencia de fondos desde una cuenta corriente a una caja de ahorro.

Dado que mi cuenta corriente tiene saldo

Y el saldo de mi cuenta corriente es mayor o igual que el monto a transferir

Cuando transfiero fondos desde mi cuenta corriente a mi caja de ahorro

Entonces debería disminuir mi saldo en mi cuenta corriente

Y debería aumentar mi saldo en mi caja de ahorro.

Criterios de aceptación: escenario concreto

Título: Transferencia de fondos entre cuentas.

Como cliente del banco

Quiero transferir fondos entre mis cuentas cuando necesite

Para poder manejar mi dinero de manera más eficientemente.

Escenario: Transferencia de fondos desde una cuenta corriente a una caja de ahorro.

Dado que mi cuenta corriente tiene un saldo de \$1.000

Y mi caja de ahorros tiene un saldo de \$2.000

Cuando transfiero \$500 de cuenta corriente a mi caja de ahorro

Entonces mi cuenta corriente debe queda con saldo \$ 500

Y mi caja de ahorros con saldo \$ 2.500.

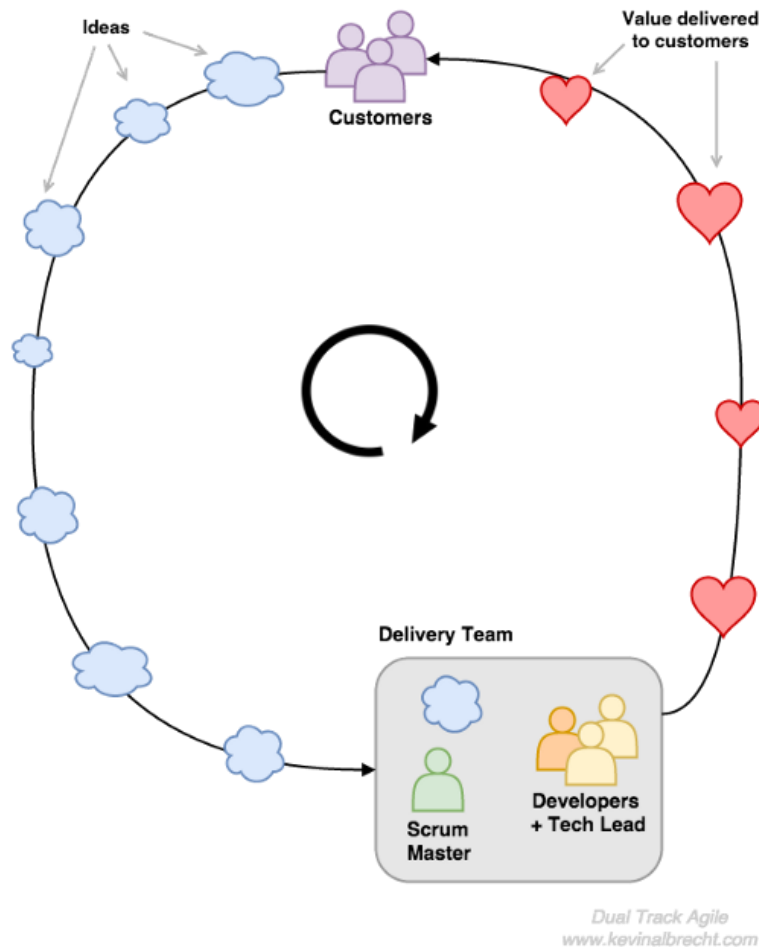


Dual-track agile

Dual-track

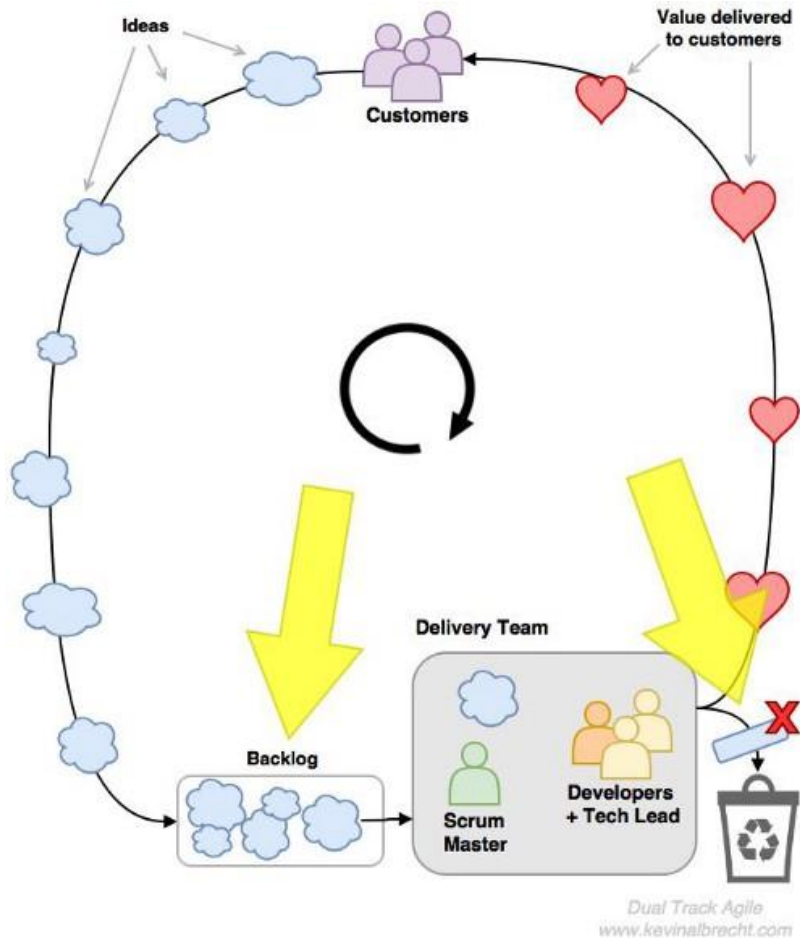
- ...

1. Ciclo de retroalimentación



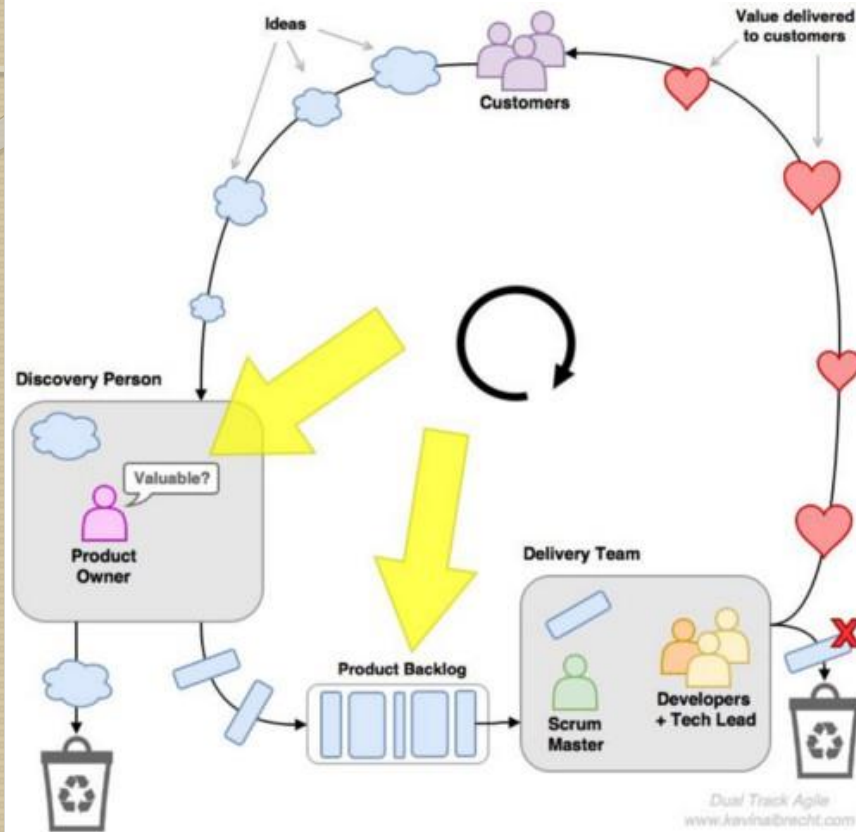
- Se desarrolla un producto de software para resolver un problema de mis clientes.
- Delivery team:
 - su trabajo es brindar una solución al cliente que resuelva su problema.
 - Desarrolladores, Líder técnico, Scrum Master.
- El equipo entrega una versión usable al cliente.
- El cliente responde con “ideas”: defectos, cambios, solicitudes de nuevas funcionalidades.

2. Demasiadas ideas...



- No todas las “ideas” son factibles. Algunas se descartan.
- Llegan mas ideas de las que se pueden implementar y se acumulan en un *backlog*.
- Las “ideas” que van quedando, ¿son todas “valiosas”?
- Se necesita un rol de gestión de esas ideas...

3. El rol del Product Owner

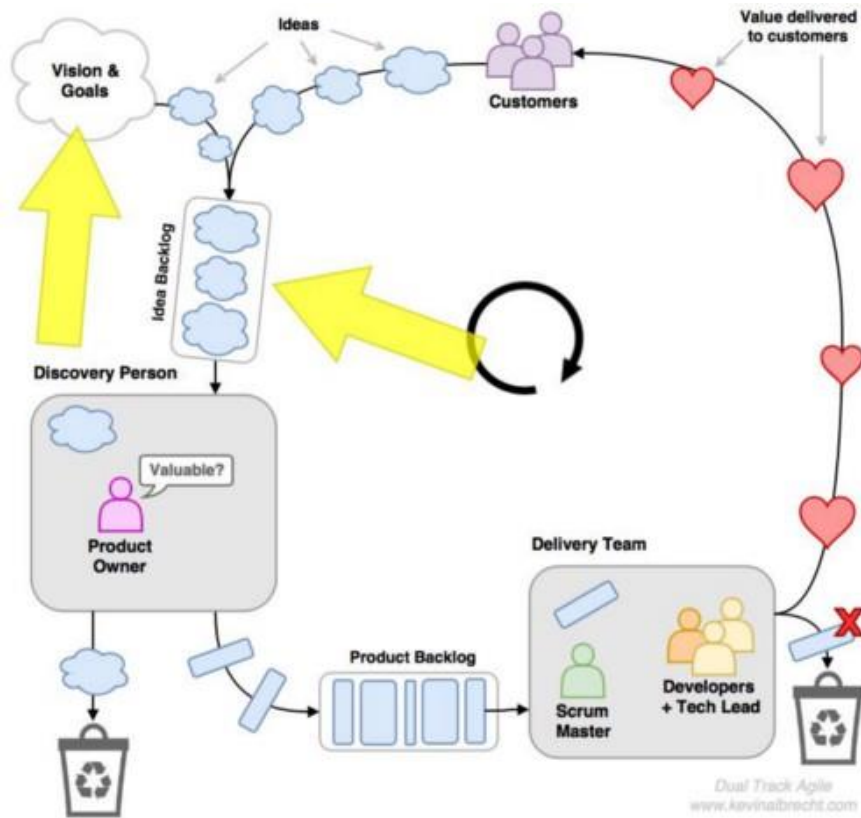


- Se agrega el rol de *product owner* (PO) para mantener el backlog bajo control.
 - Es responsable de analizar cada idea entrante y hacer la pregunta: *¿Es esto valioso?*
 - Si “no”: el PO descarta la idea.
 - Si “si”: el PO toma la idea y crea una historia de usuario y la prioriza en un *product backlog*.
-
- ¿Qué significa “valioso”?
 - ¿“Valioso” para quién?

Visión y objetivos del producto

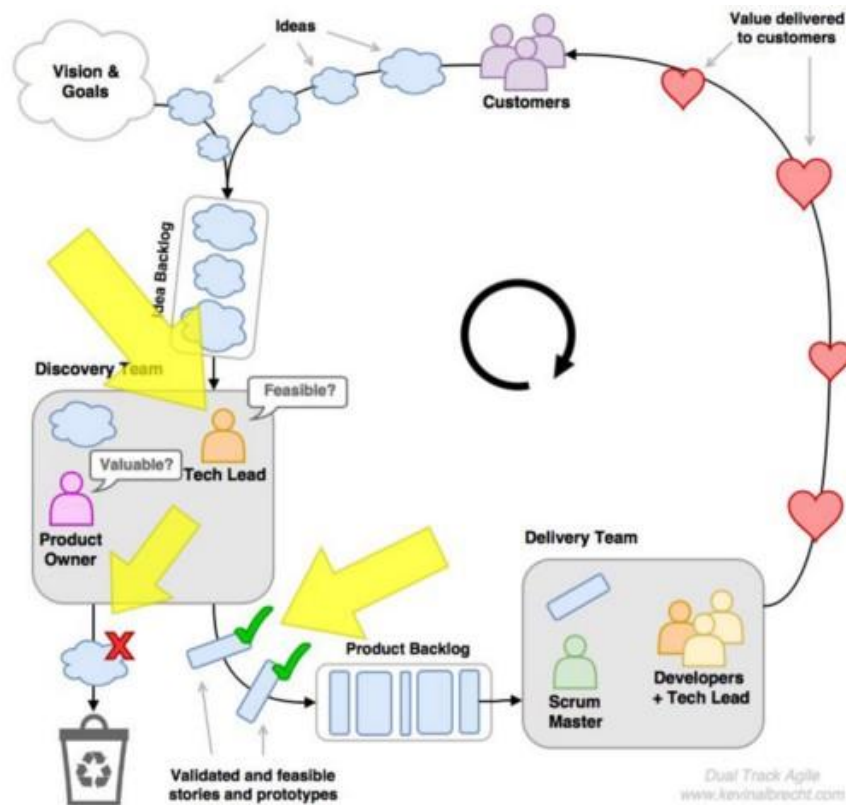
- La “visión del producto” es una descripción de la esencia del producto en desarrollo: cuáles son los problemas que está resolviendo, para quién y por qué ahora es el momento adecuado para construirlo.
- Una visión de producto le da al equipo una imagen más amplia de en qué están trabajando y por qué.
- Los “objetivos del producto” son los hitos en una hoja de ruta del producto en los que los equipos deben enfocar su esfuerzo, ya que el logro de estos objetivos aumenta gradualmente el valor para los usuarios.

4. La visión y objetivos del producto



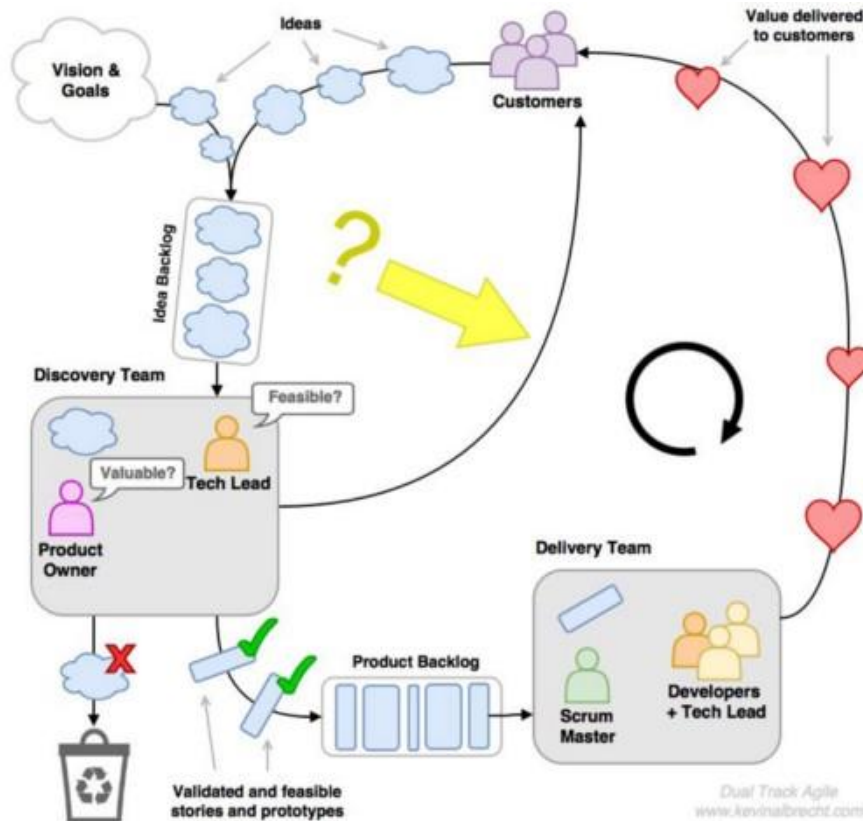
- La visión captura la esencia de lo que desea que logre el producto.
- Si una “idea” de un cliente no se ajusta a la visión y los objetivos del producto, se debe descartar.
- Es un “filtro de ideas”.
- Las ideas que pasan ese filtro se organizan en un “backlog de ideas” o “backlog de oportunidades”.

5. El “líder técnico”



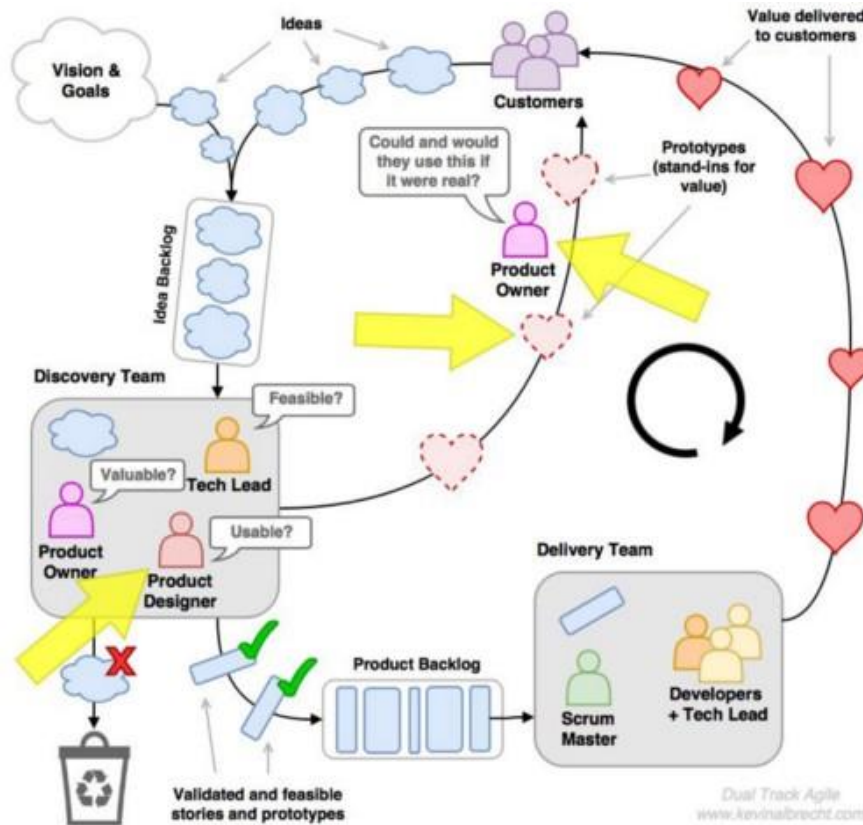
- Discovery team: conformado por el Líder técnico y por el Product owner.
- El Product owner identifica cuales ideas son valiosas.
- El Líder técnico identifica cuales ideas son factibles.
- Cuando el “Delivery team” saca una historia del product backlog, estar seguro de que es posible y valiosa convertirla en un incremento de software.

6. El feedback del cliente



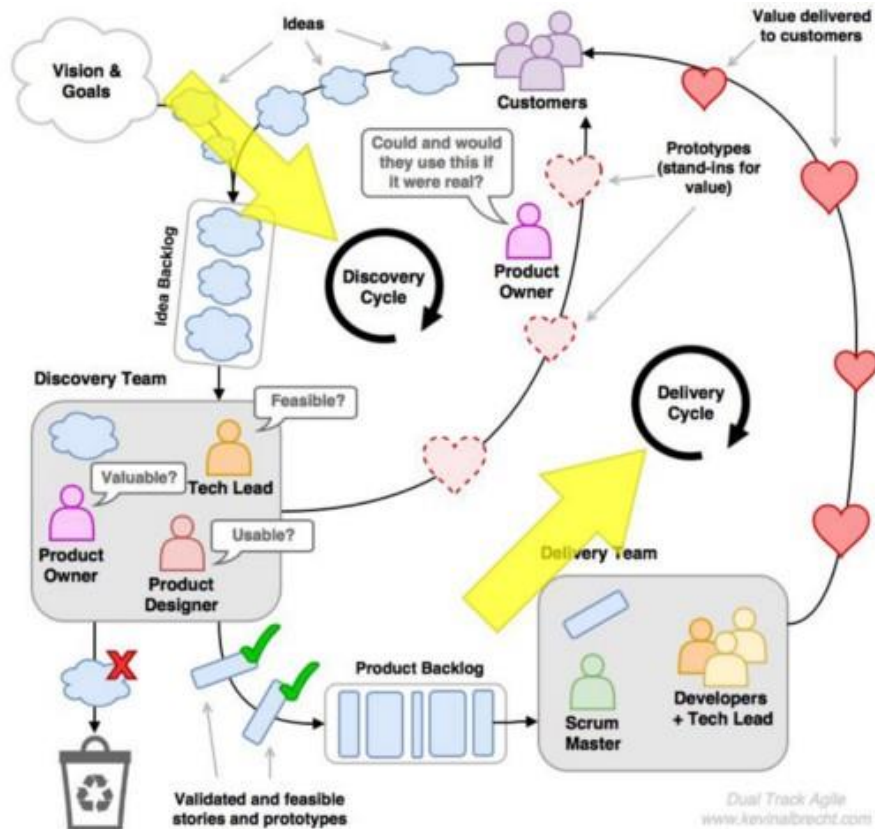
- Obtener retroalimentación del cliente lo más rápido posible.
- Incluso antes de escribir código...

7. Feedback rápido en base a prototipos



- En lugar de esperar a tener el software para recibir comentarios de los clientes, se le puede mostrar mediante *prototipos*.
- El trabajo del Discovery team es crear prototipos del valor *que se podría entregar* a los clientes.
- El PO muestra el prototipo al cliente para obtener respuesta a *"Si tuviéramos que construir esto, ¿el cliente lo usaría?"*

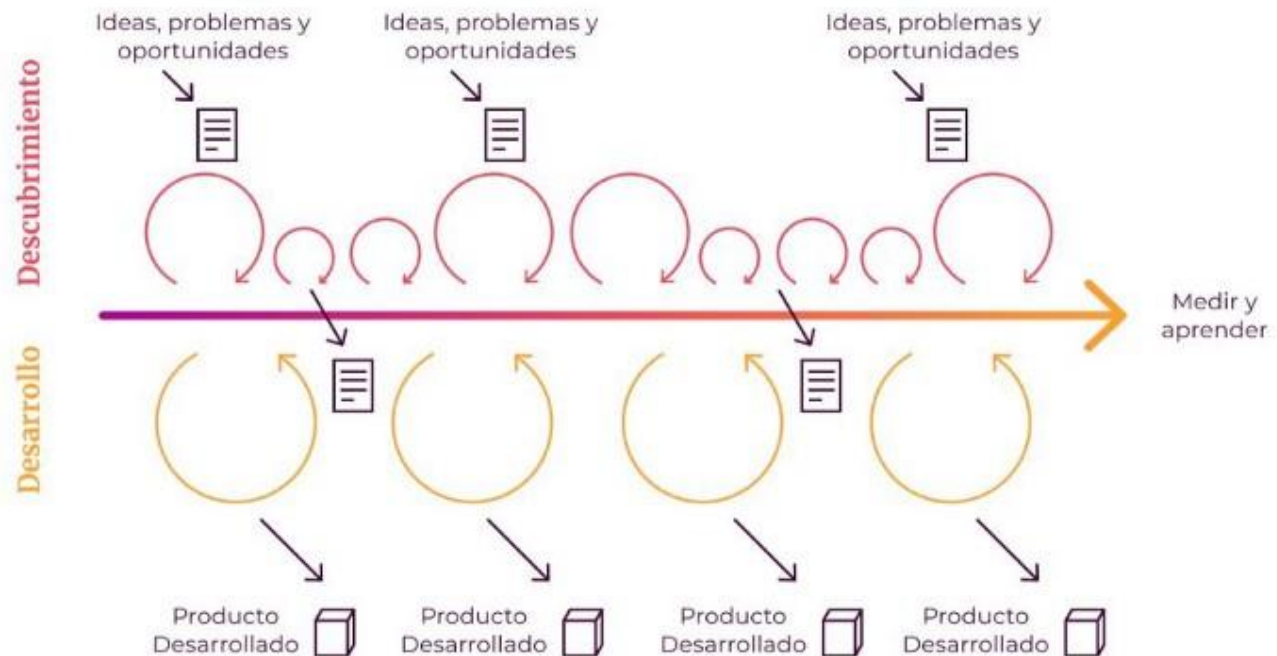
8. Los dos ciclos



- El “ciclo de descubrimiento” valida el valor y la viabilidad de una idea, creando prototipos.
- Es más corto y más rápido que el “ciclo de entrega”.
- El “ciclo de entrega” entrega al cliente el valor de una idea ya validada.
- Los dos “teams” (Discovery y Delivery) deben trabajar como uno solo.

Descubrimiento y Entrega

- El carril de Discovery el problema, trabaja en soluciones, crea prototipos, los prueba con el cliente y, cuando el prototipo está validado, pasa al carril de Delivery donde se empieza a desarrollar el producto.



Descubrimiento y Entrega

- Cada carril puede usar una metodología de trabajo distinta.
- Usualmente, para el carril de Discovery se utiliza Kanban y para el carril de Delivery se utiliza Scrum.

