



imatia
innovation

We help you to do more

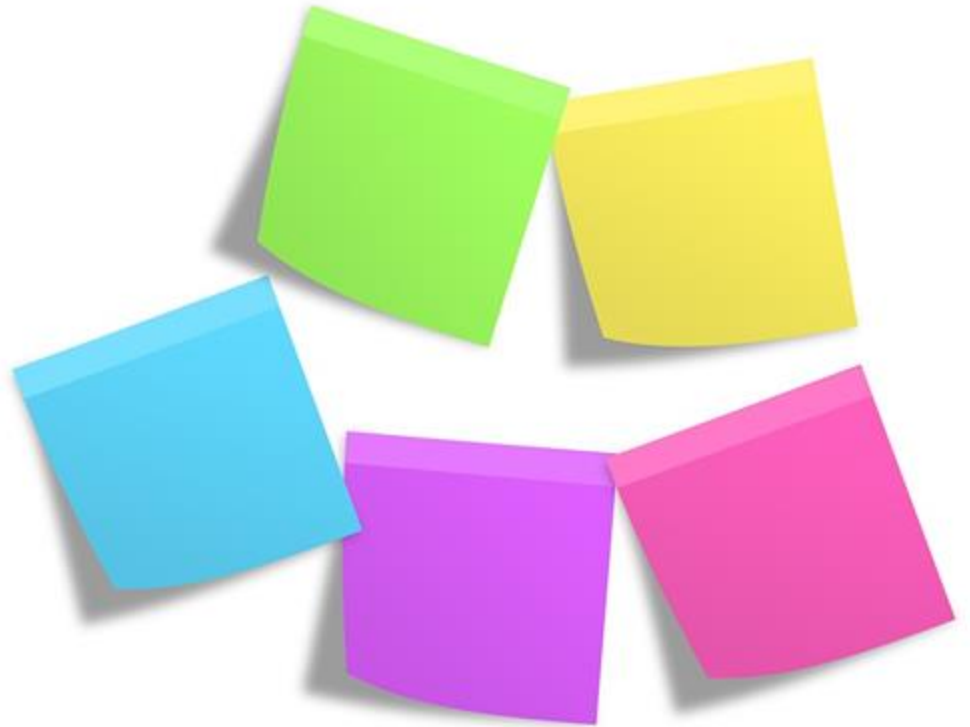


ORGANIZACIÓN DE PROYECTOS A TRAVÉS DE METODOLOGÍAS ÁGILES

im
innovation

We help you to do more

- Introducción
- Mentalidad Agile
- Scrum a primera vista
- Creando un equipo ágil
- Roles de Scrum
- Requerimientos ágiles
- Refinamiento del backlog
- Estimación ágil
- Planificación del sprint
- Reuniones
- Artefactos
- Deuda técnica
- Kanban



INTRODUCCIÓN

imatia
innovation

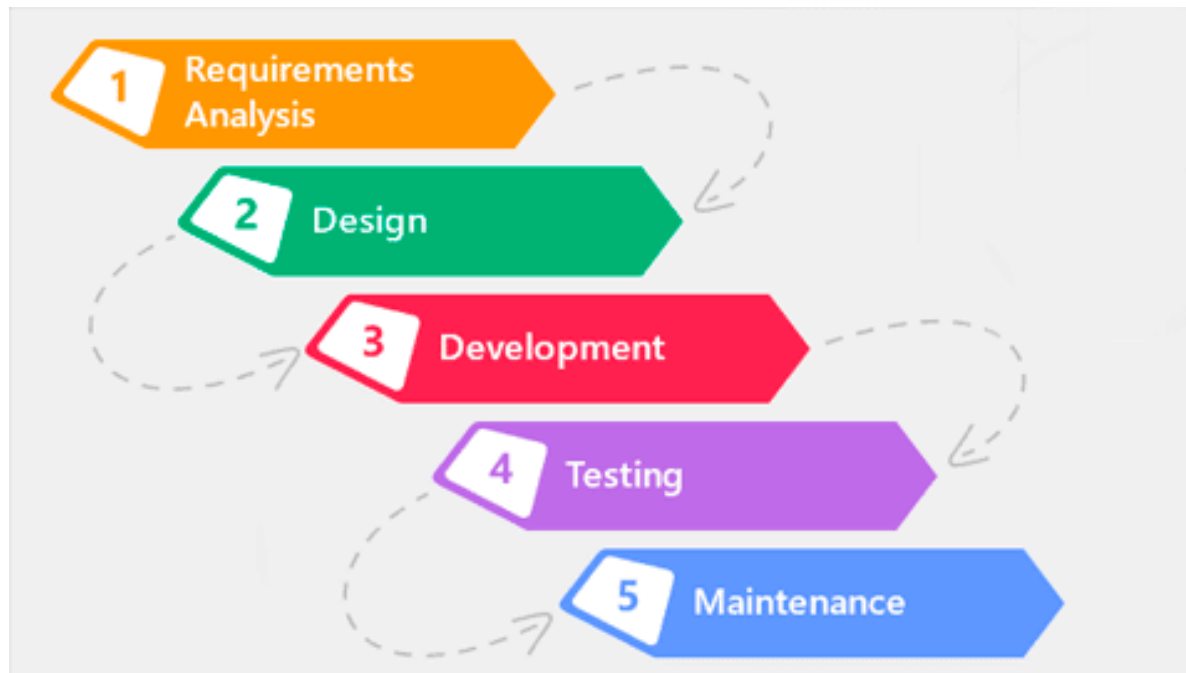
We help you to do more

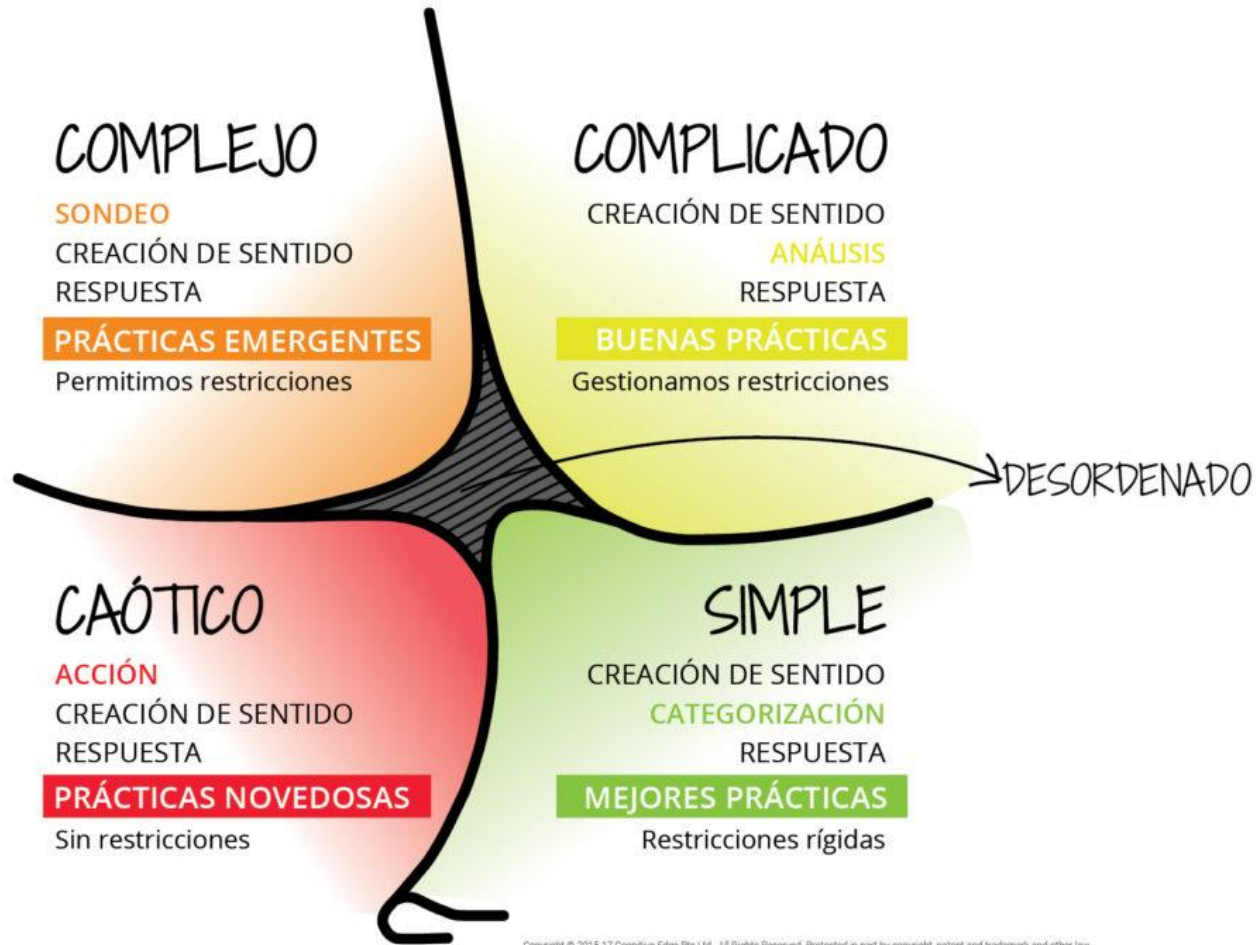
- 10 cosas que ya conoces sobre la agilidad en general

1. _____
2. _____
3. _____
4. _____
5. _____
6. _____
7. _____
8. _____
9. _____
10. _____



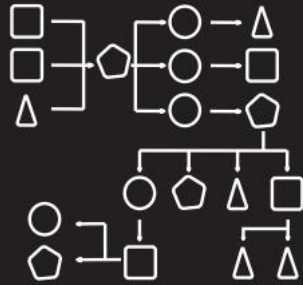
Ciclo de vida del Software



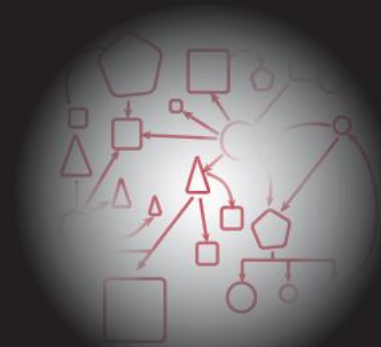


- CYNEFIN
- Marco conceptual desarrollado a principios de los 2000 por IBM
- Define 5 contextos o dominios para la toma de decisiones

¿Complicado o complejo?



COMPLICADO



COMPLEJO

¿Qué contexto incluye la selva del Amazonas y un coche Ferrari?

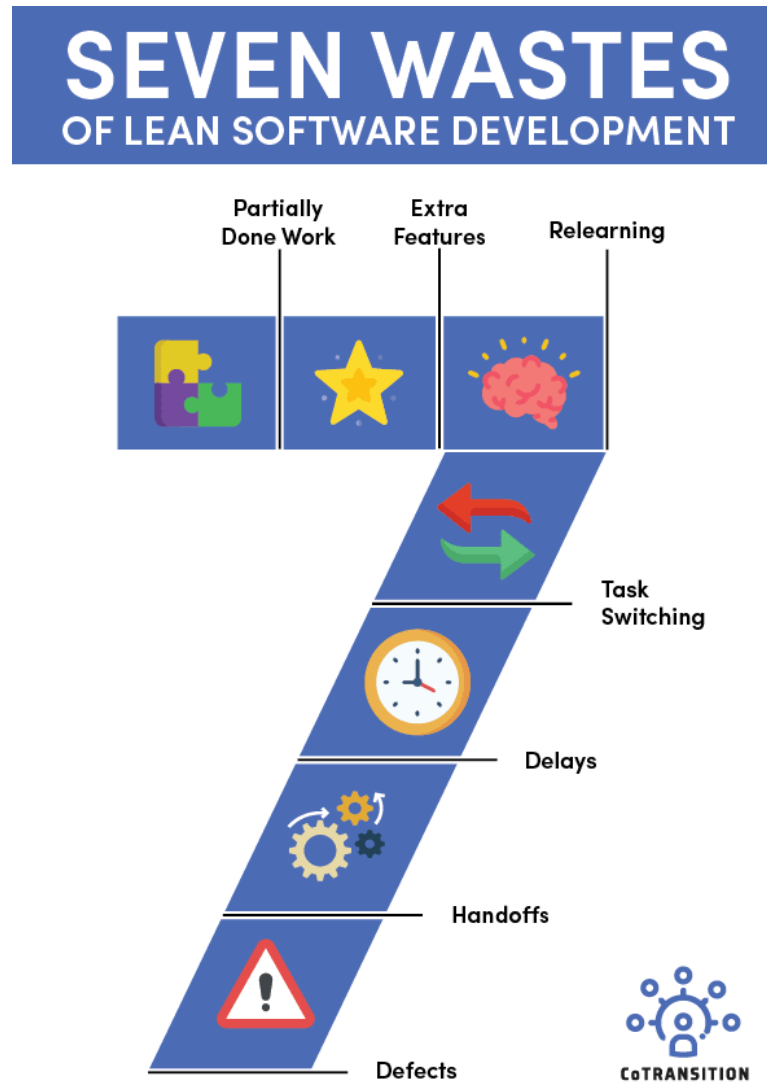


Filosofía Lean



REFERENCIAS: <https://www.cocinaintegral.net/metodologia-lean-menage-confort/>
<https://www.javiergarzas.com/2012/10/lean-software-development-2.html>

Desperdicios LEAN del software

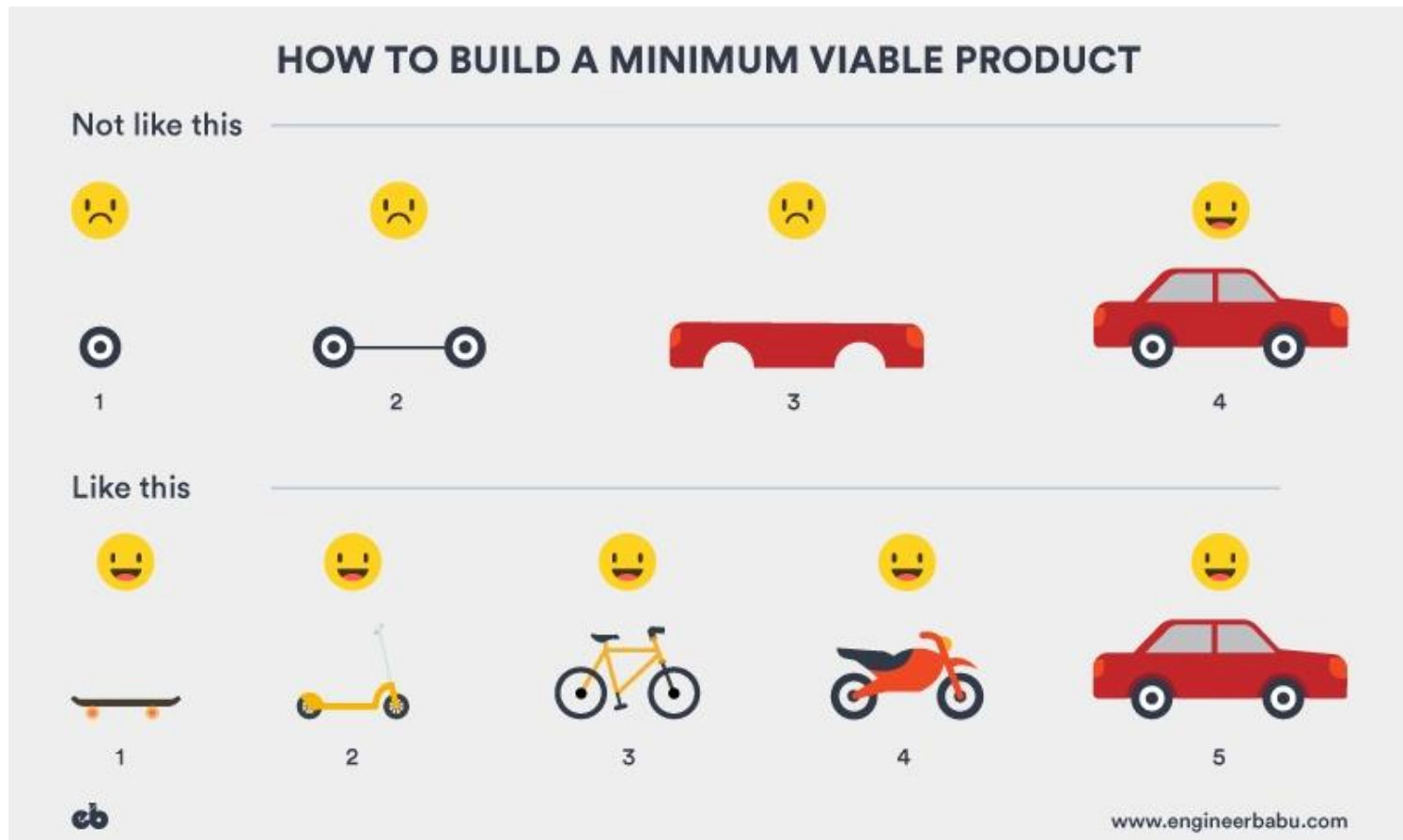


Output vs Outcome

Maximize Value, not Output



¿Como construir un producto mínimo viable (MVP)?



Esto SI es un MVP

HOW NOT TO BUILD A MINIMUM VIABLE PRODUCT



1



2



3



4

ALSO HOW NOT TO BUILD A MINIMUM VIABLE PRODUCT



1



2



3



4

HOW TO BUILD A MINIMUM VIABLE PRODUCT



1



2

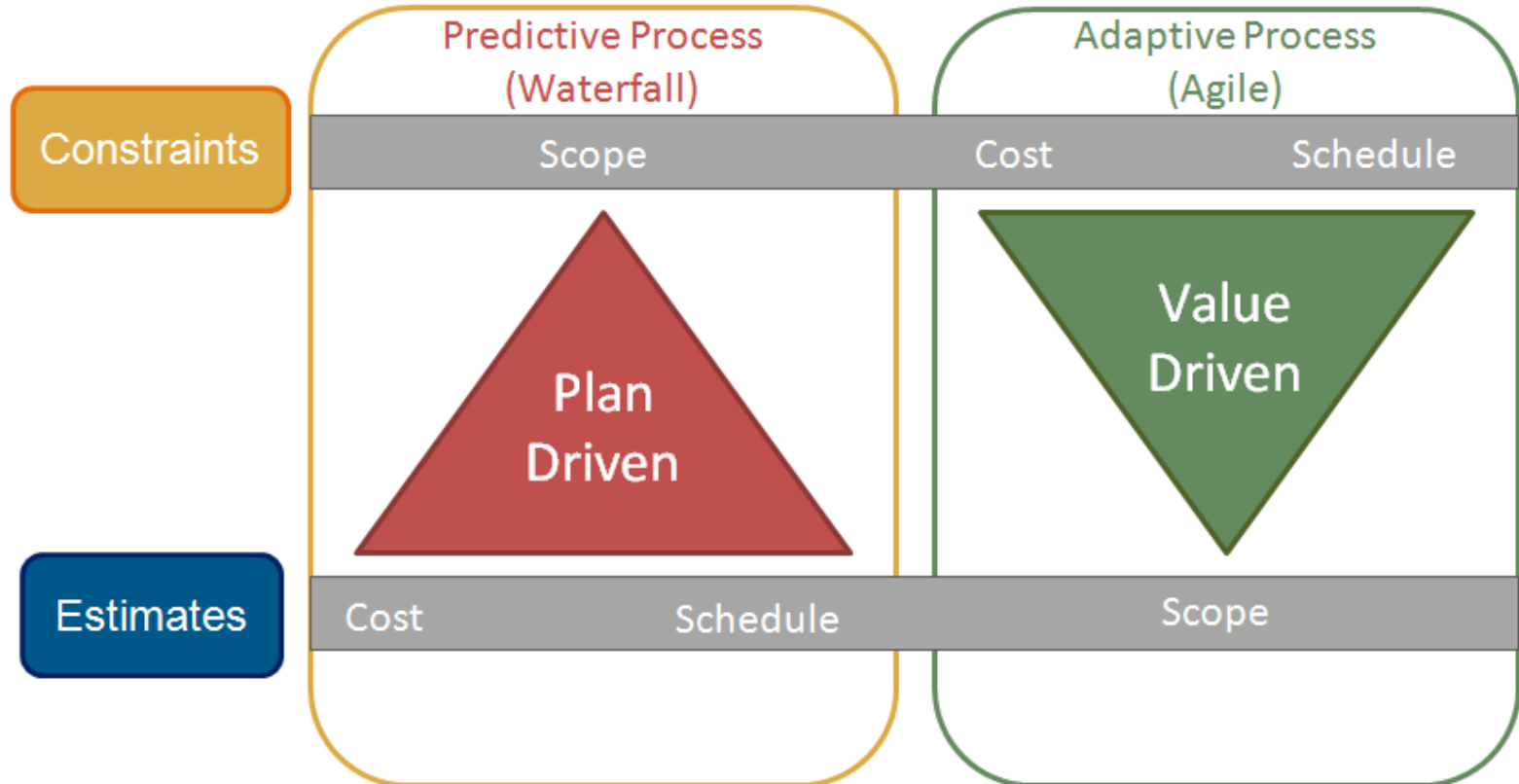


3



4

Alcance, coste y plazo: El triángulo de hierro

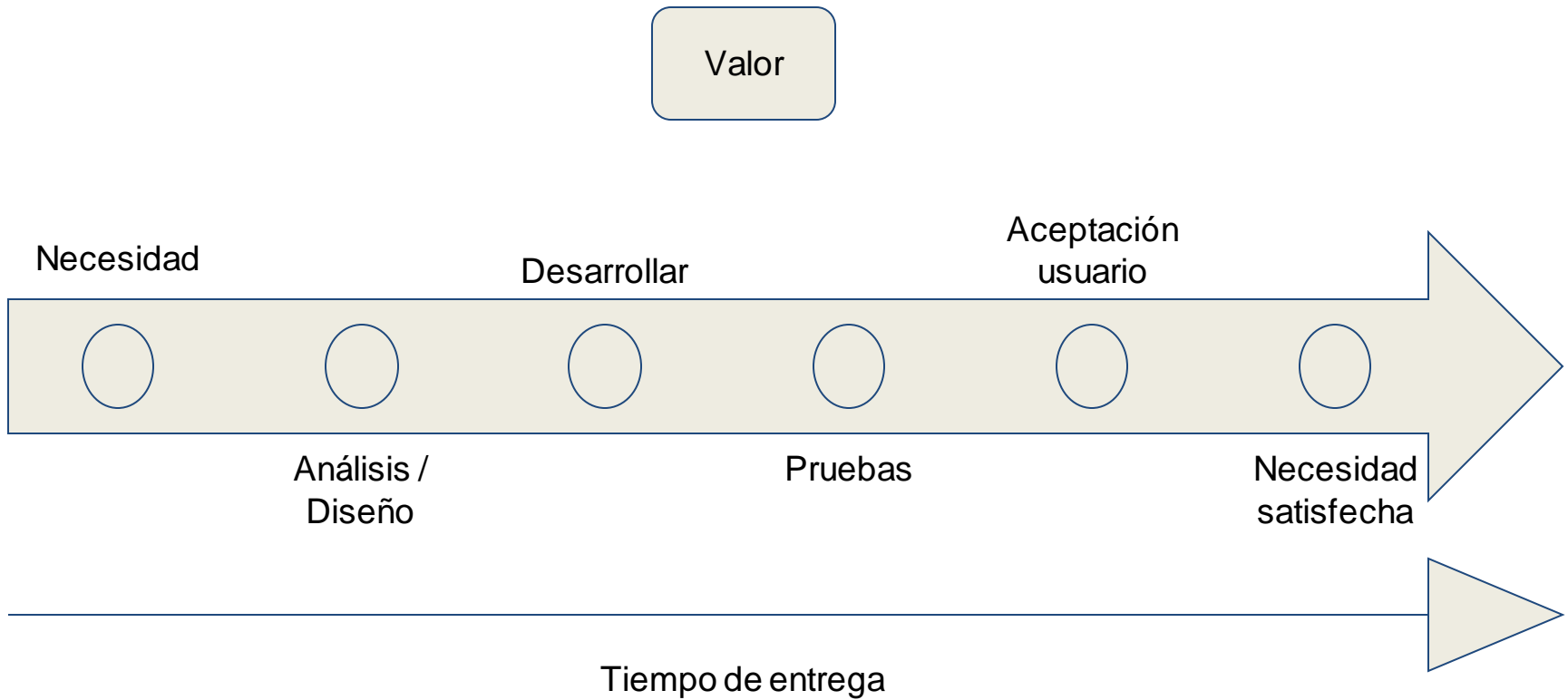


MENTALIDAD AGILE

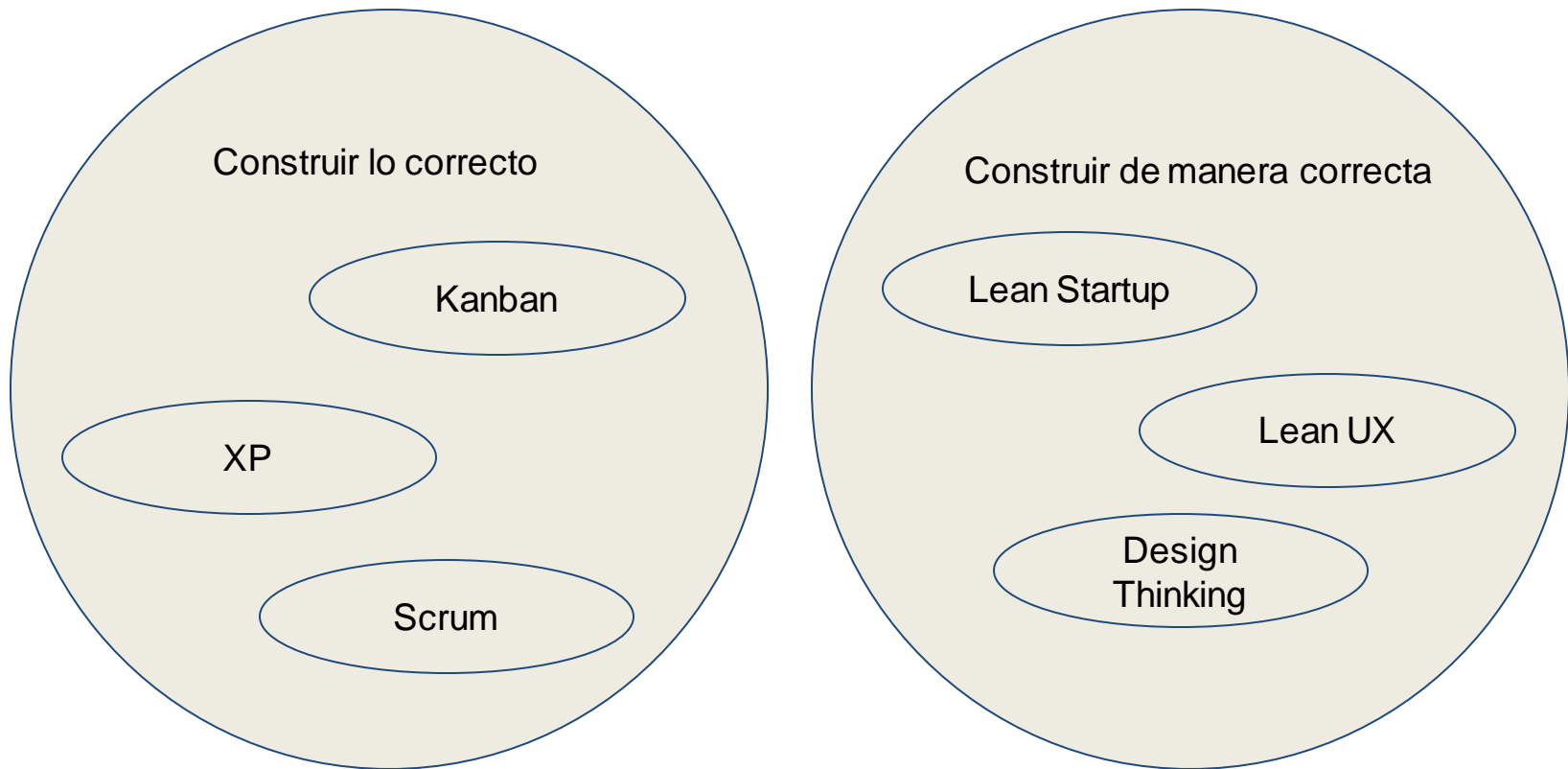
imatia
innovation

We help you to do more

- Entrega de valor al cliente



- Organización Lean-Agile



- Principios, prácticas y herramientas



- Manifiesto ágil

Manifiesto por el Desarrollo Ágil de Software

Estamos descubriendo formas mejores de desarrollar software tanto por nuestra propia experiencia como ayudando a terceros. A través de este trabajo hemos aprendido a valorar:

Individuos e interacciones sobre procesos y herramientas
Software funcionando sobre documentación extensiva
Colaboración con el cliente sobre negociación contractual
Respuesta ante el cambio sobre seguir un plan

Esto es, aunque valoramos los elementos de la derecha, valoramos más los de la izquierda.

- Principios del manifiesto ágil

Seguimos estos principios:

Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.

Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.

Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.

Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.

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.

El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.

El software funcionando es la medida principal de progreso.

Los procesos Ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.

La atención continua a la excelencia técnica y al buen diseño mejora la Agilidad.

La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.

Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.

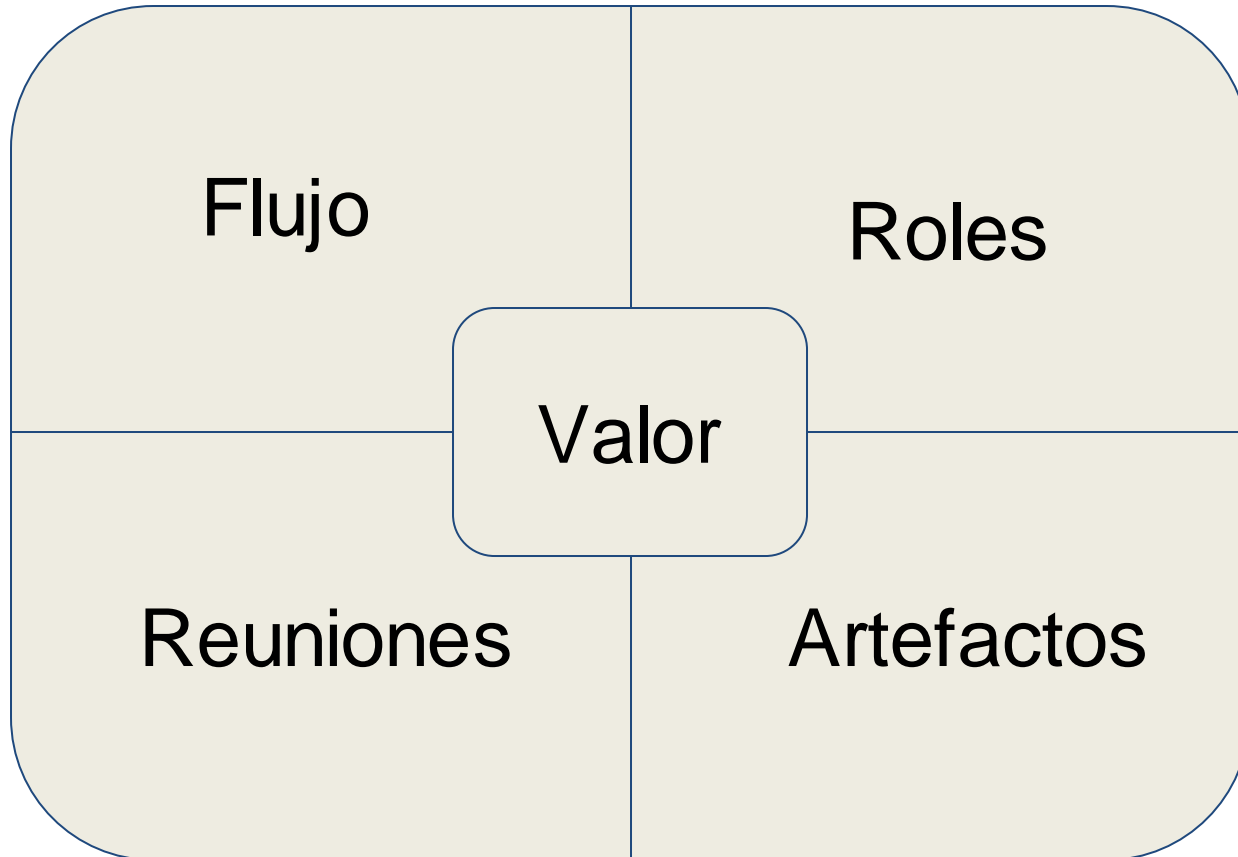
A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.

SCRUM A SIMPLE VISTA

imatia
innovation

We help you to do more

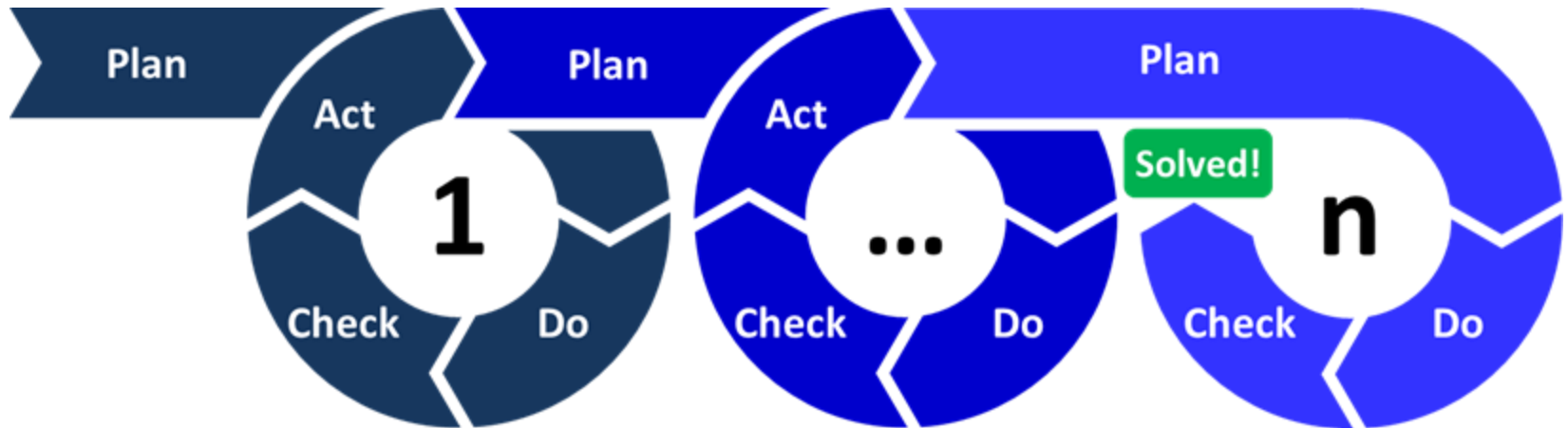
- Scrum



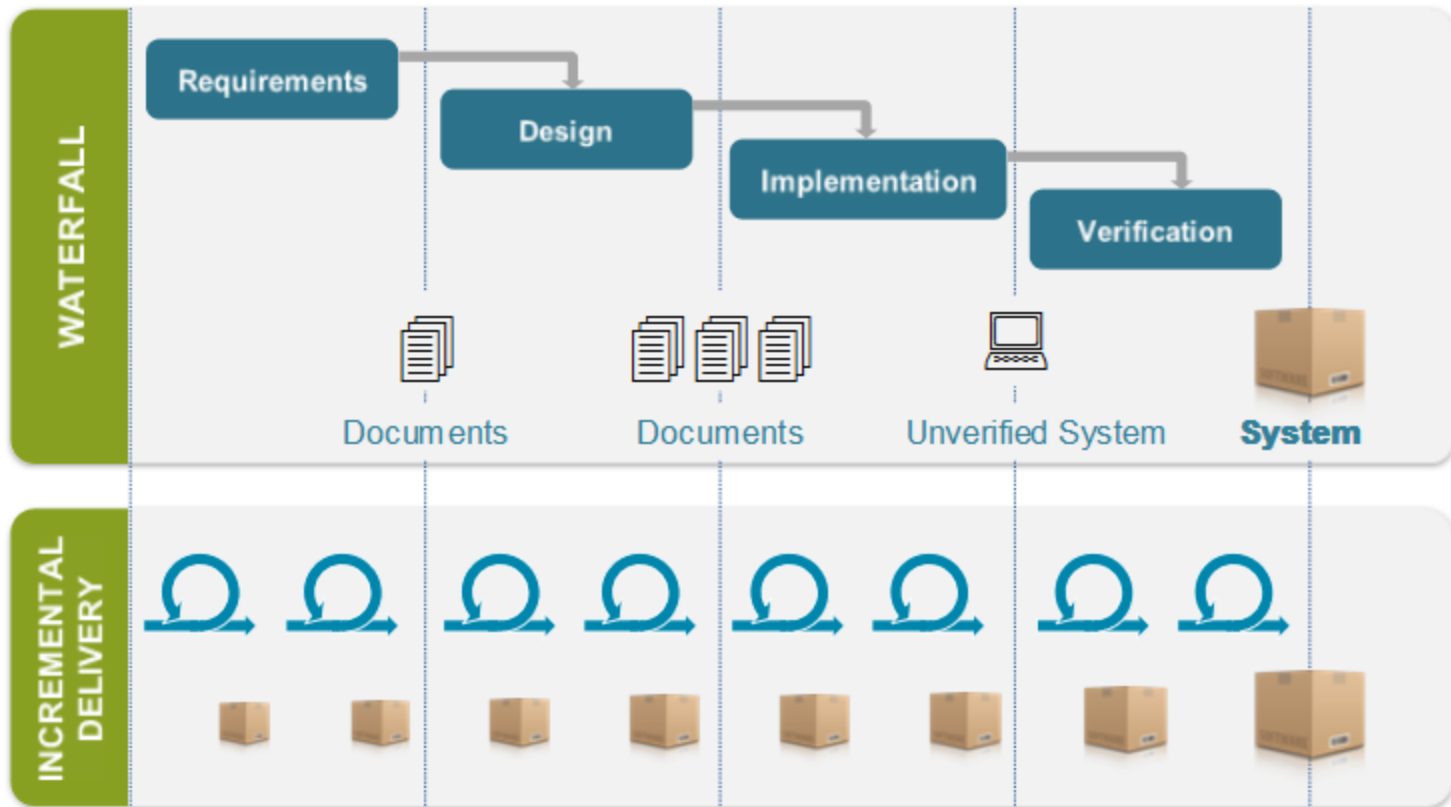
- Cuestión de principios



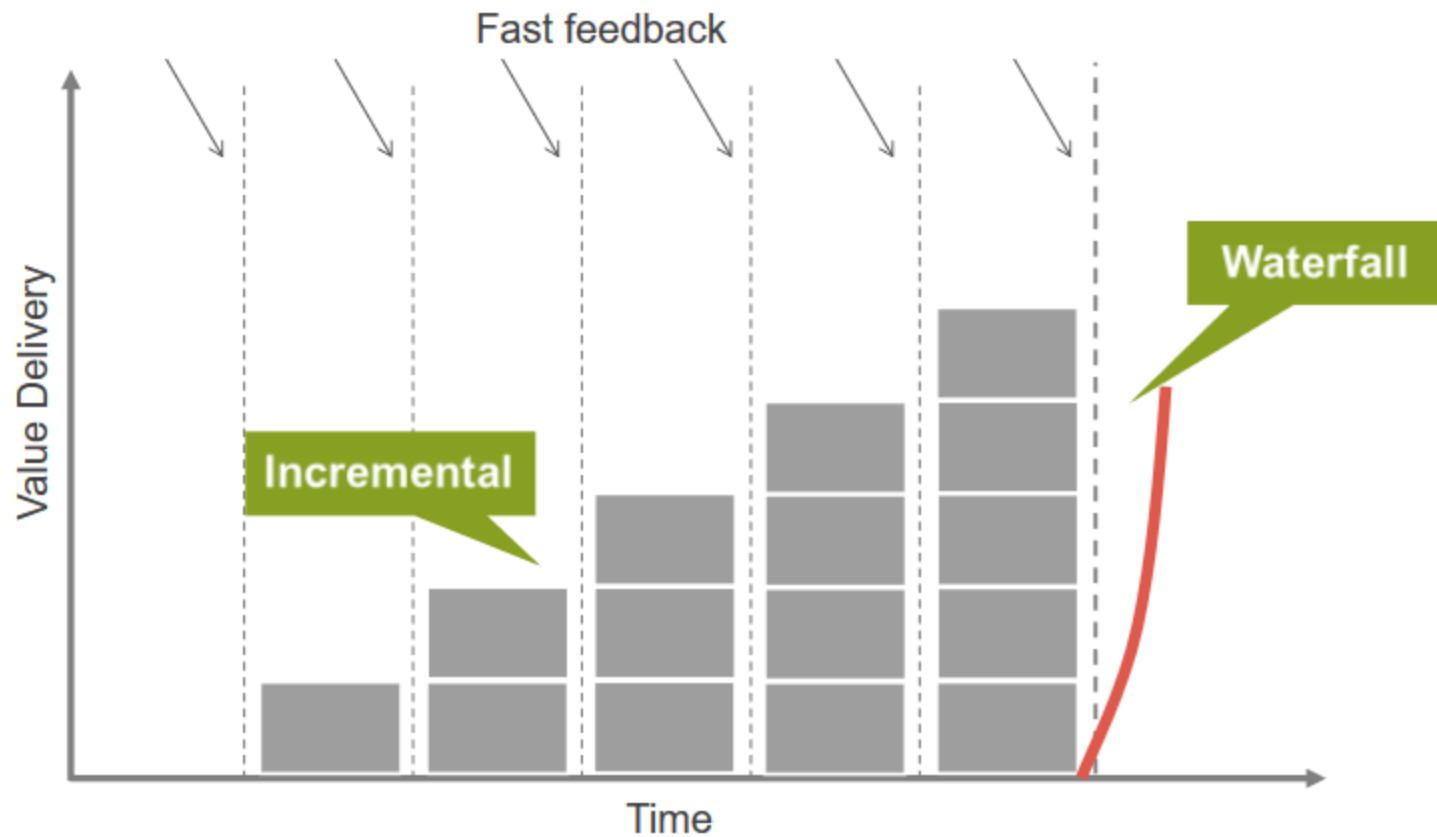
- Ciclo de Deming (PDCA)



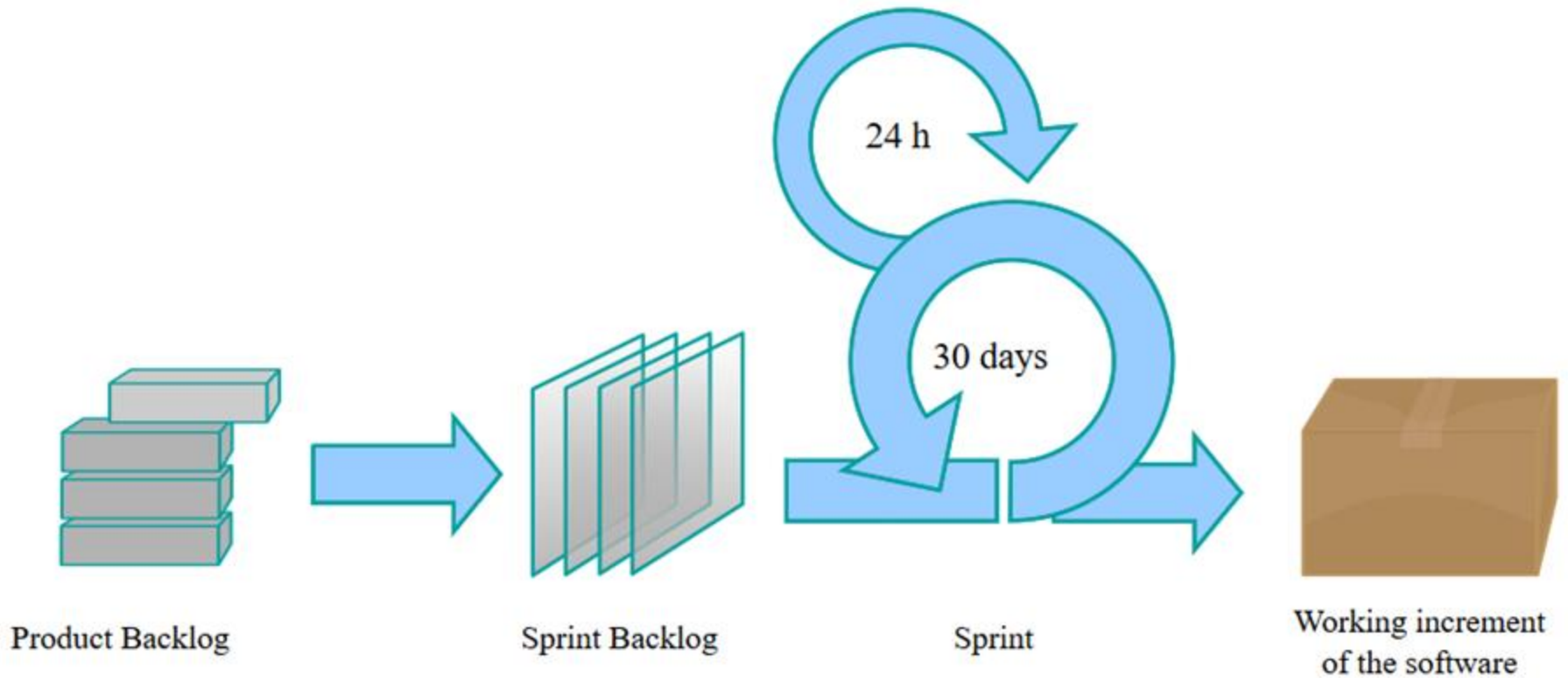
- Construcción iterativa e incremental vs. Cascada



- Construcción iterativa e incremental vs. Cascada



- Flujo de Scrum



CREANDO UN EQUIPO ÁGIL

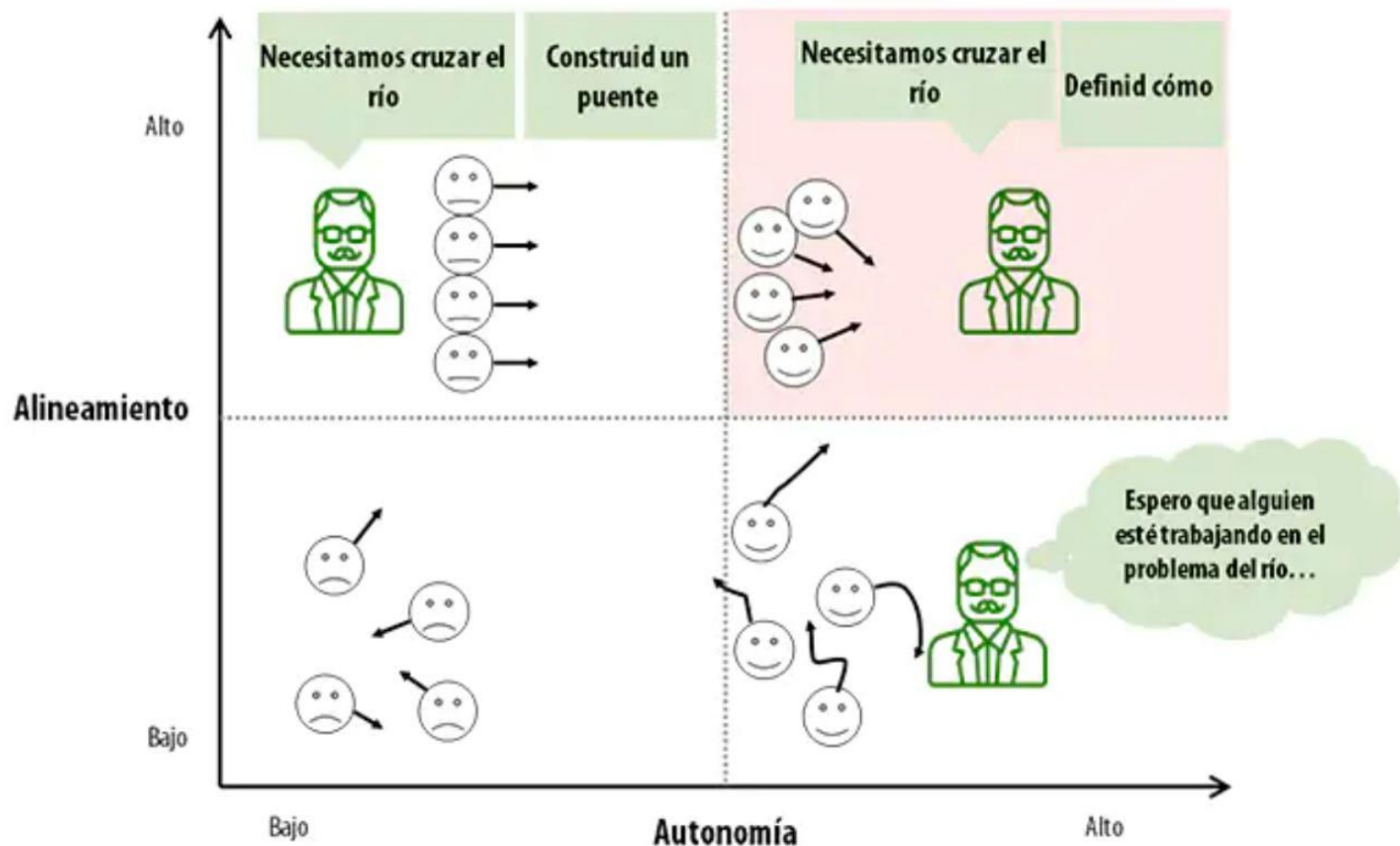
imatia
innovation

We help you to do more

- Centralizado Vs Descentralizado



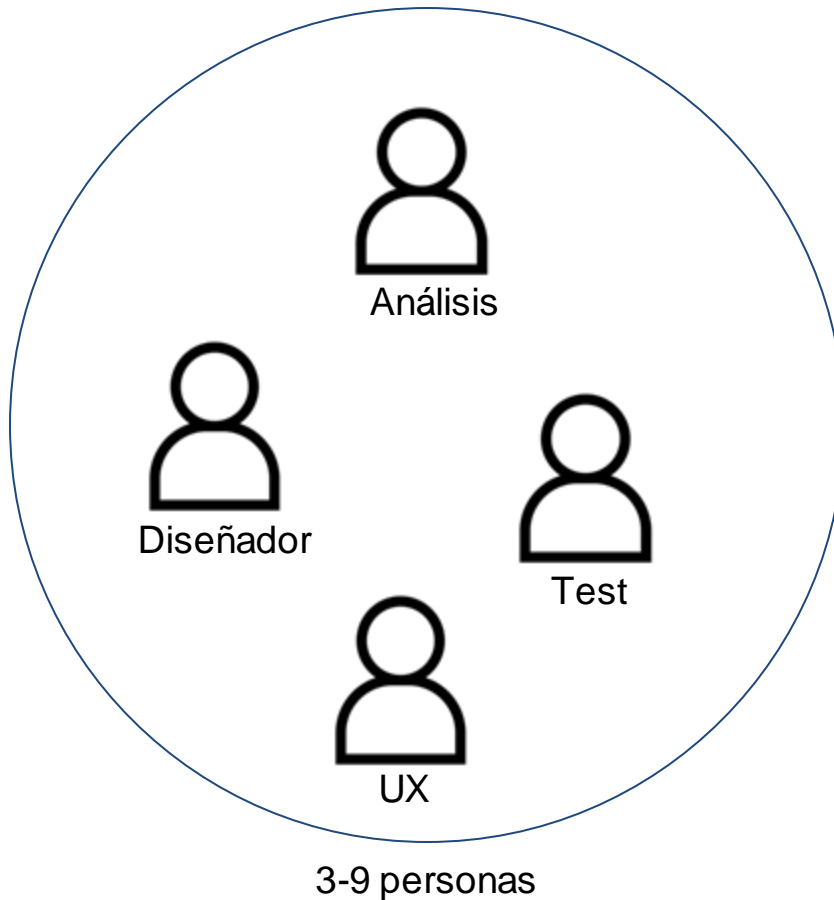
El alineamiento permite la autonomía



- Elementos para la auto-organización



- Equipo auto-organizado y multi-funcional

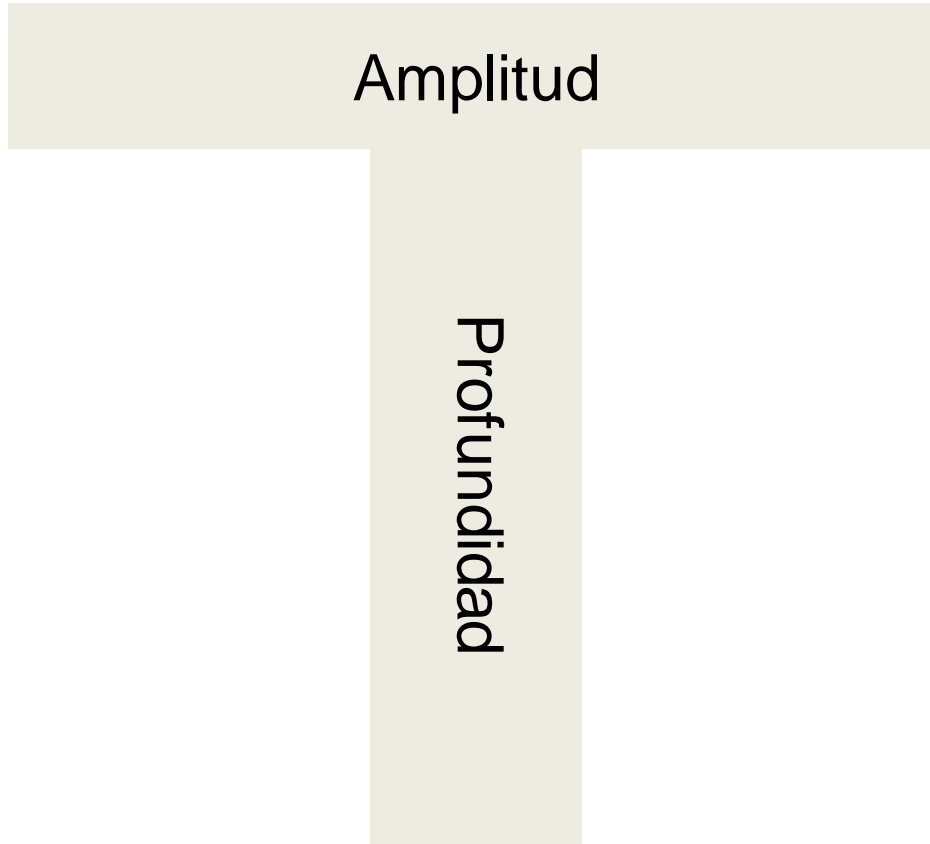


Objetivo común definido externamente

Deciden:

- Cómo planificar su trabajo
- Cuando entregar
- Cómo hacer el trabajo
- Quien hace qué
- Qué estándares siguen

- Perfil “T”



- Habilidades de codificación
- Habilidades de arquitectura
- Habilidades de testing
- Habilidades de documentación
- Habilidades de despliegue
- Habilidades de comunicación
- Habilidades de organización

- Actividad: El nudo humano

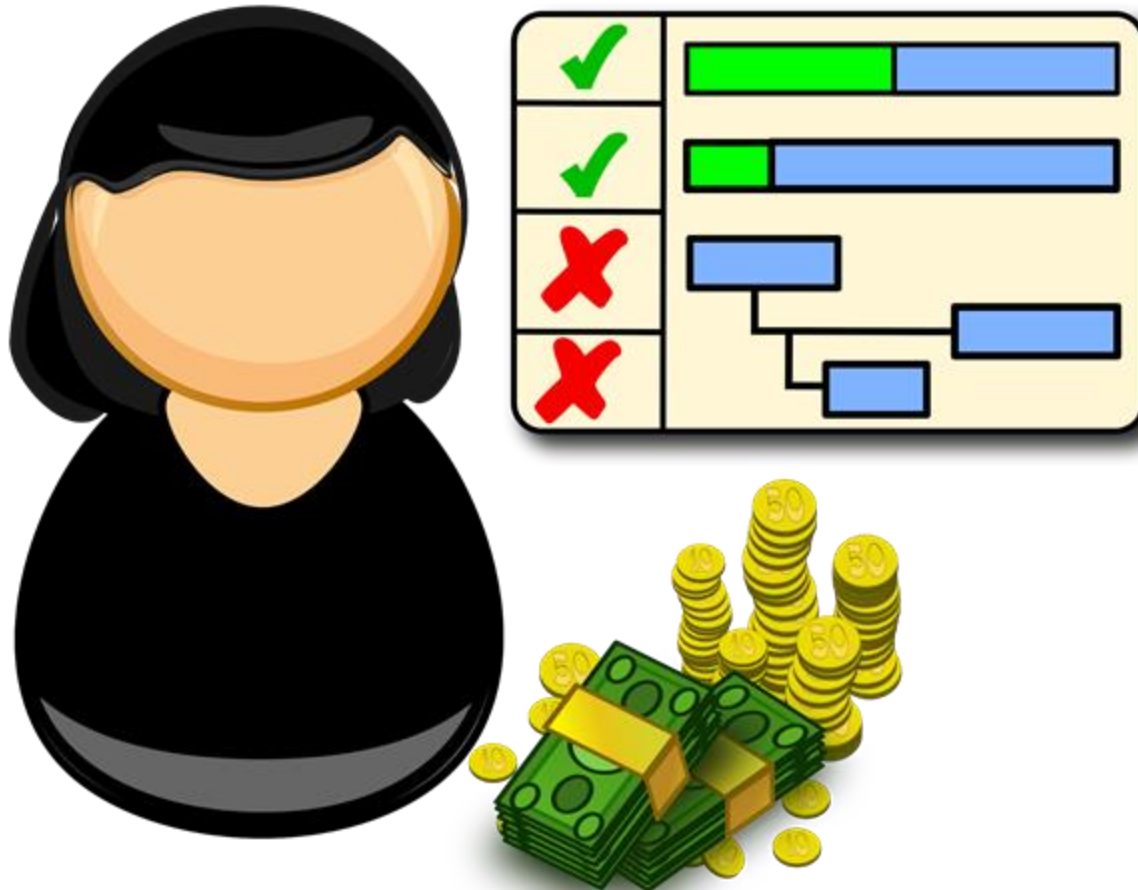


ROLES DE SCRUM

imatia
innovation

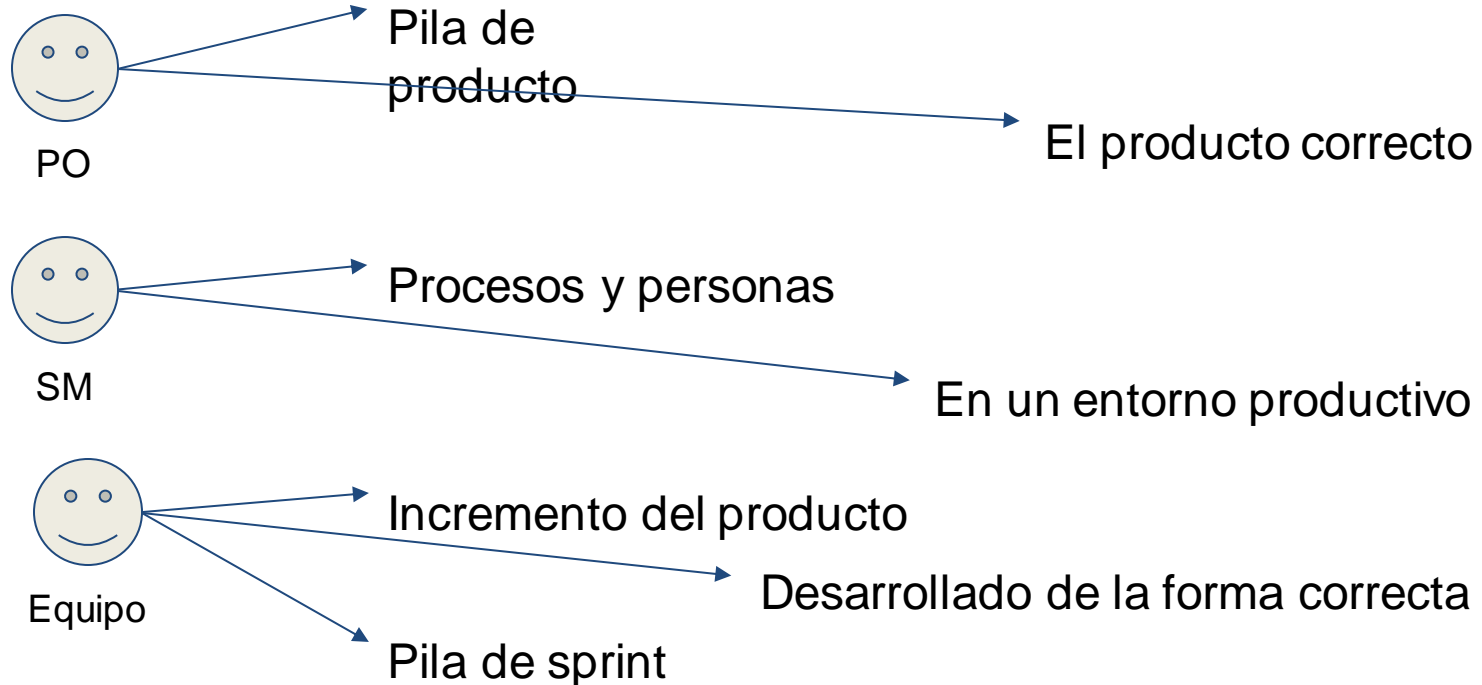
We help you to do more

- Actividad: Identificar las responsabilidades de un Project Manager



- Macrogestión, microgestión, personas y procesos

Macrogestión	Product Owner
Microgestión	Equipo
Personas, procesos	Scrum Master



Roles de Scrum

By Javier Garzás, Ana María García (2014)

Responsabilidades






Product Owner

Debe participar en las reuniones

-  Decidir qué construir... ¡y que no!
-  Recoger y tener claros los requisitos del software.
-  Definir buenas historias de usuario.
-  Fijar criterios de aceptación para cada historia de usuario.
-  Ordenar y priorizar los items del Product Backlog.
-  Definir el producto mínimo viable.
-  Acordar junto al resto del equipo una definición de DONE.
-  Definir el plan de releases.
-  Validar entregas (Sprint Review)

» El Product Owner no debe dar ordenes al equipo.
» El equipo no debe trabajar en otros requisitos distintos a los que el Product Owner incluya en el Backlog.

-  Estar disponible y accesible para el equipo.
-  Es el responsable de cancelar el sprint si ocurre un imprevisto extremo.
-  Asegurarse de que el Product Backlog es visible para todo el mundo.
-  Asegurarse de que todo el mundo entiende los items del Product Backlog.

El Product Owner debe conocer la velocidad del equipo, para realizar estimaciones de cuando estaran implementadas las necesidades en el producto.

javiergarzas.com

233 grados de TI



Daily meeting
(Participación optativa)



Sprint Planning



Sprint Review



Sprint Retrospective

- Roles en un equipo Agile



- Equipo

- Responsable de transformar la pila de sprint (sprint backlog), en un incremento de funcionalidad del software
- El equipo decide a que se compromete y como hacer lo mejor para cumplir con lo comprometido

Roles de Scrum

By Javier Garzías, Ana María García (2014)

Responsabilidades



Planifica la implantación de Scrum junto con la organización.



Ayuda a la organización a entender qué interacciones con el equipo aportan valor y cuáles no.



Ayuda al Product Owner a entender la agilidad.



Ayuda al Product Owner a maximizar el valor de negocio.



Enseña al Product Owner a priorizar y gestionar efectivamente el Product Backlog.



Ayuda al equipo de desarrollo a convertirse en auto-organizado y multifuncional.



Soluciona posibles impedimentos que pudieran surgir durante el Sprint.



Se asegura de que haya una definición de DONE



Scrum Master

Debe participar en las reuniones y asegurarse de que cumplan el tiempo y el objetivo establecido



Daily meeting



Sprint Planning



Sprint Review



Sprint Retrospective



El Scrum Master no es lo mismo que el Product Owner. El Product Owner tiene una visión más de negocio, mientras que el Scrum Master se encarga de que todo el equipo entienda Scrum y lo aplique correctamente.



Ayuda a que las posibles mejoras detectadas en la retrospectiva del Sprint se lleven a cabo.



Junto con el equipo de desarrollo, actualiza el trabajo en progreso (burndown chart).



Se asegura y promueve buenas prácticas de programación.

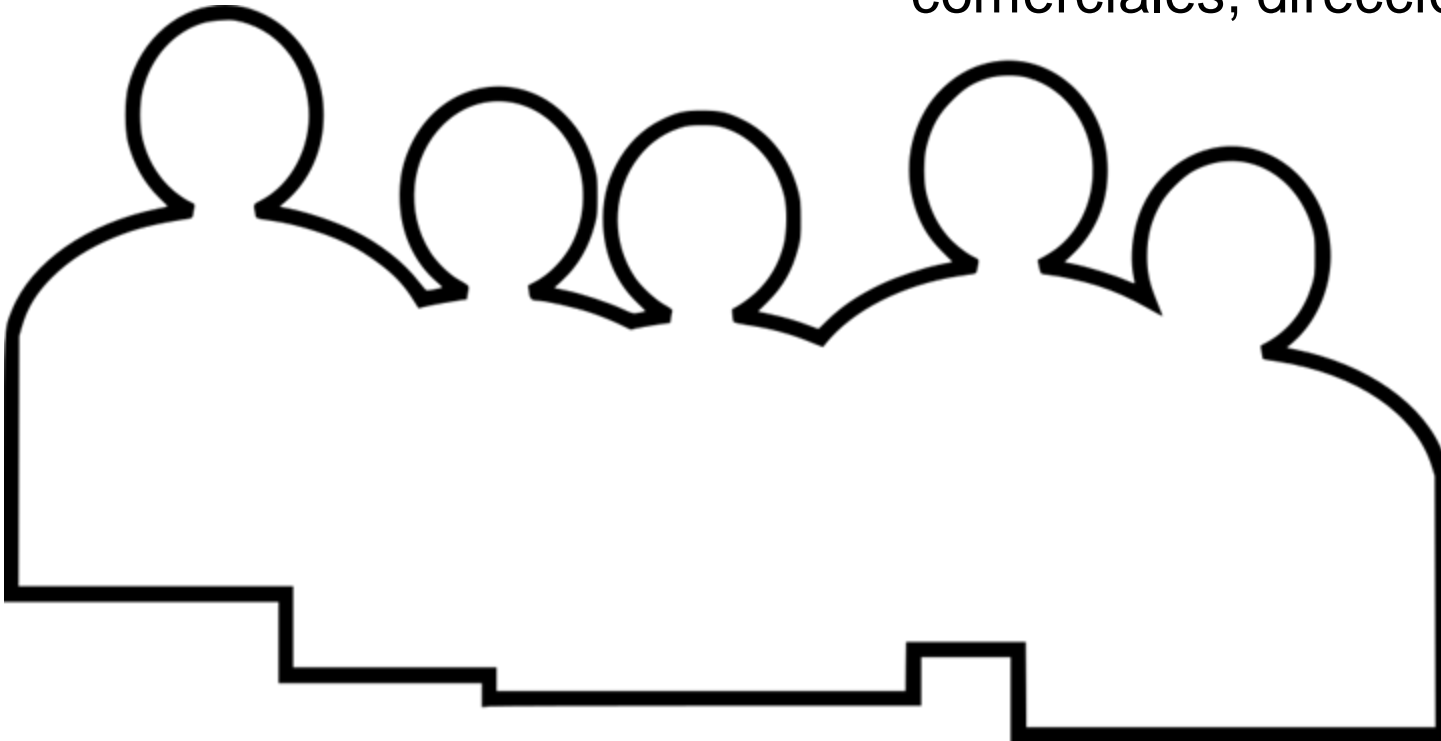


Realiza cursos para aprender Scrum si es necesario.



El Scrum Master es el responsable de asegurar que se sigue Scrum.

- Roles en un equipo Agile
 - Partes interesadas (Stakeholders)
 - Clientes, usuarios, administrativos, marketing, comerciales, dirección...



- Actividad: Roles en Scrum

Responsabilidad	Equipo	PO	SM
Recopilar requisitos / funcionalidades			
Estimar el tamaño de los ítems de la pila			
Motivar al equipo			
Desarrollar y entregar incrementos de software “terminado”			
Eliminar impedimentos			
Trabajar iterativamente y medir su propio progreso			
Aconsejar al PO sobre cómo maximizar el valor de negocio entregado			
Dar el visto bueno al incremento resultante de cada iteración			
Determinar cómo hacer el trabajo comprometido			
Ser un líder servicial			
Compartir la visión global del producto			
Transformar la Pila del Sprint en un incremento			
Gestionar y priorizar la Pila de producto			

REQUERIMIENTOS ÁGILES

imatia
innovation

We help you to do more

- Requerimientos tradicionales vs. requerimientos ágiles

¿Tradicional vs Ágil?

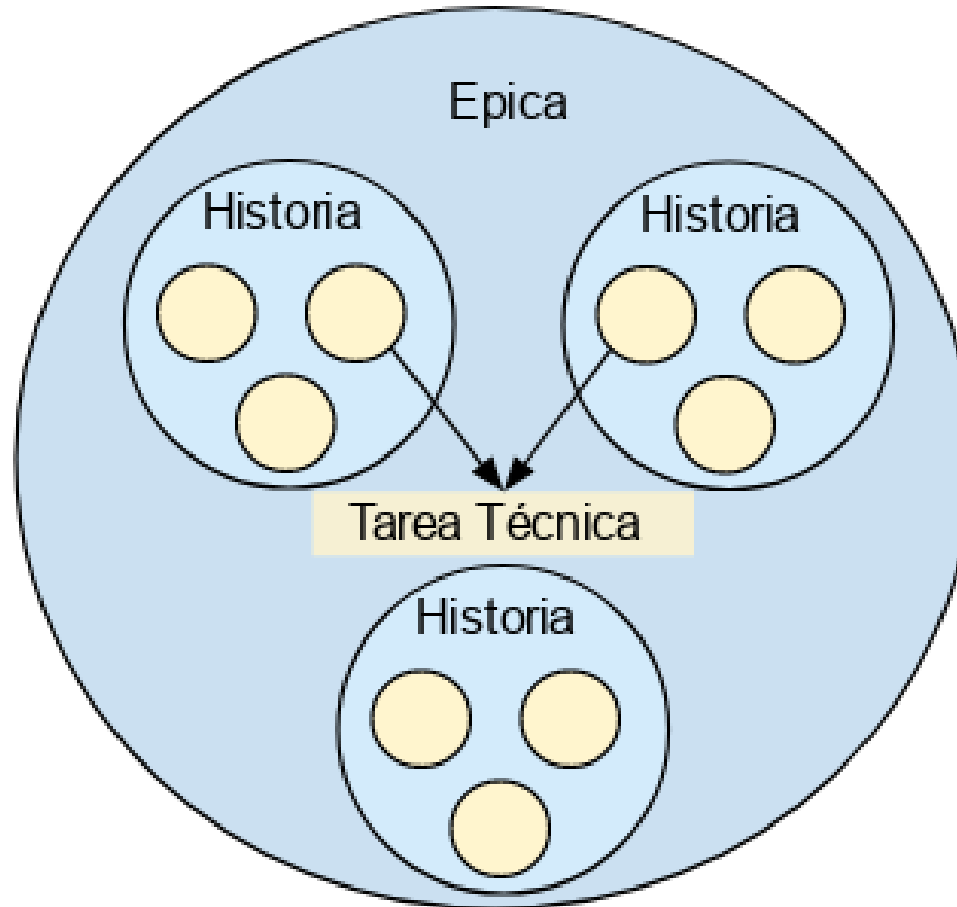
	Tradicional	Ágil
Prioridad	Cumplir el Plan	Entregar valor
Enfoque	Ejecución (¿Cómo?)	Estrategia (¿Por qué? ¿Para qué?)
Definición	Detallados y cerrados Descubrimiento al inicio	Esbozados y evolutivos Descubrimiento progresivo
Participación	Sponsor, stakeholders de mayor poder e interés	Colaborativo con stakeholders de mayor interés (usuarios finales, clientes)
Equipo	PM, BA, Área de Procesos	Equipo multidisciplinar
Herramientas	Entrevistas, Observación, formularios	Principalmente prototipado Técnicas de facilitación para descubrir
Documentación	Procesos detallados Matriz de trazabilidad de requisitos	Historias de usuario Visual Story Map
Producto final	Definido en Alcance	Identificado progresivamente
Proceso	Estable, adverso al cambio	Incertidumbre, abierto al cambio

- Visión

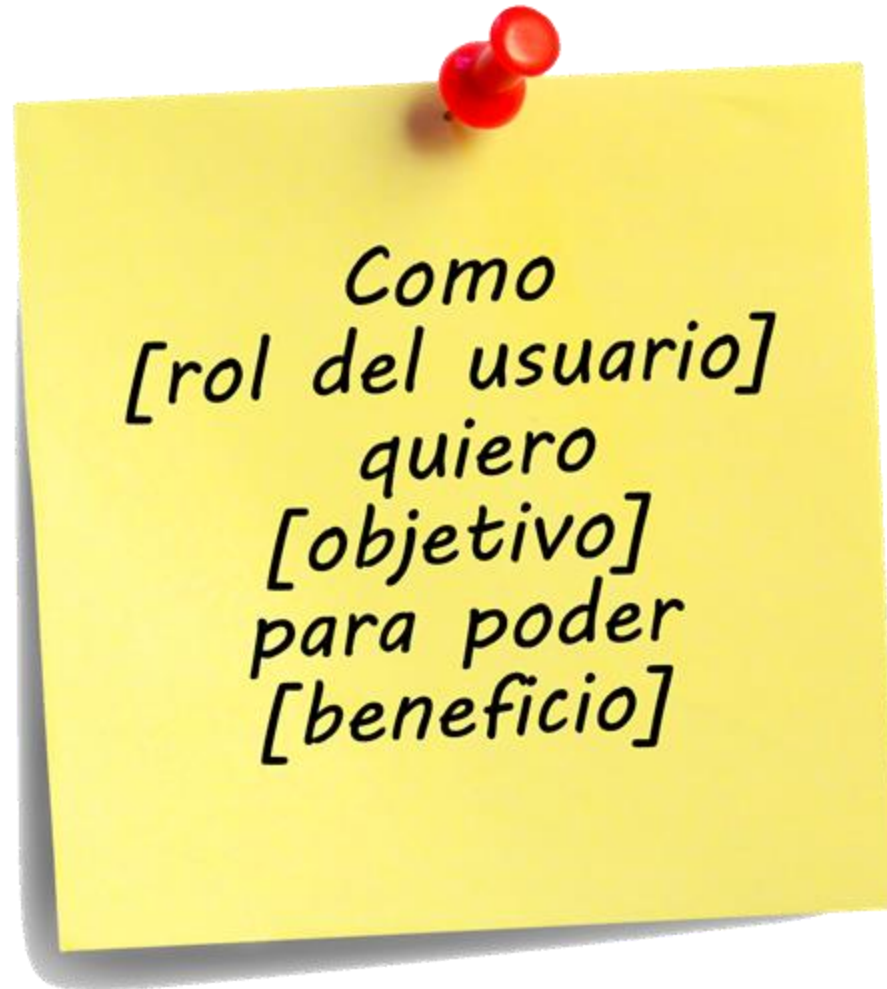
¿Tradicional vs. **ÁGIL**?



- De Épicas a Tareas



- Historias de usuario



- INVEST: Como hacer las historias de usuario

I Independiente
N Negociable
V Valioso
E Estimable
S Small (Pequeño)
T Test (Probable)



- Problemas usuales con las historias de usuario
 - Solapamiento de trabajo
 - Dependencias entre historias
 - Chapado de oro
 - Adición de detalles a las historias
 - Inclusión de la interface de usuario
 - Grandes esfuerzos en “requisitos de ingeniería”
 - No preparar el sprint
 - Problemas al establecer las prioridades

- Actividad: De requisitos a historias de usuario
 - 1. Pago de anuncios
 - a. Los anunciantes podrán pagar sus anuncios utilizando varias formas de pago a partir de la página “Listado de anuncios”
 - b. Los anunciantes seleccionarán el método de pago y pulsarán el botón “Pagar ahora”
 - c. Si la forma de pago es con tarjeta de crédito, el sistema mostrará sólo las opciones VISA y Mastercard. El sistema puede saber qué tipo de tarjeta es a partir de los primeros dígitos del número de a tarjeta
 - i. Si la compra es superior a 100 EUR, se solicitará el CVV
 - ii. El sistema preguntará al usuario si quiere que el número de tarjeta sea almacenado para futuras compras
 - iii. Se almacenará también la fecha de expiración

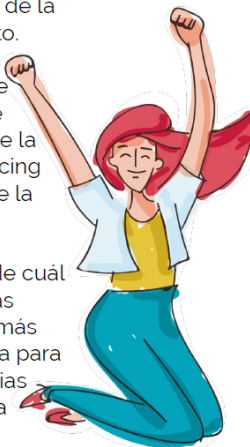
- Actividad: Dividir historias de usuario

Juego de las cartas de Slicing


¿Cómo jugar?

Dada una historia de usuario aparentemente muy grande o compleja:

1. Repartimos todas las cartas entre los participantes de la reunión de refinamiento.
2. Cada participante elige entre sus cartas la que considera que contiene la mejor estrategia de slicing posible y la pone sobre la mesa.
3. Por consenso se decide cuál de todas las estrategias sobre la mesa será la más conveniente y se aplica para crear las nuevas historias derivadas de la historia original.

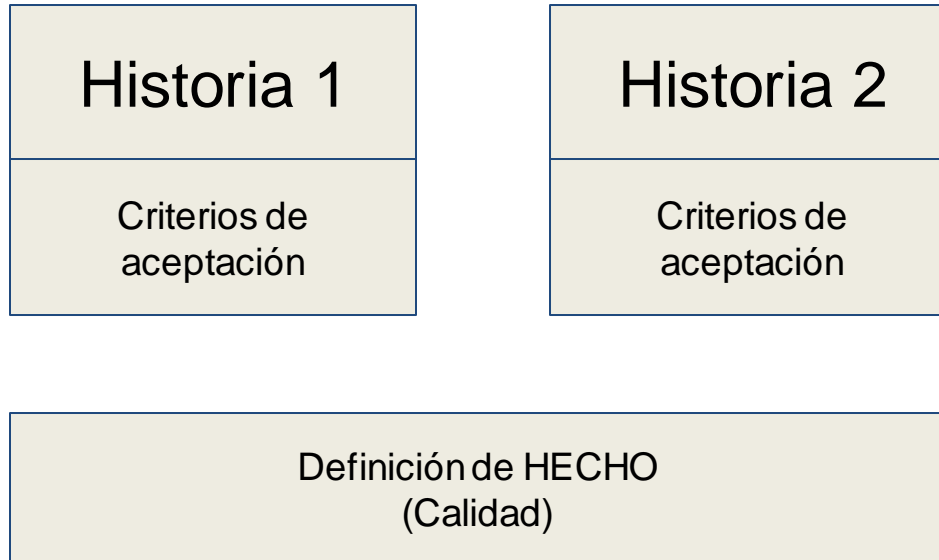


¡A jugar colaborativamente con todo el equipo!



kleer™ [klla/slicing-kards](https://klla.slicing-kards)
v6.0

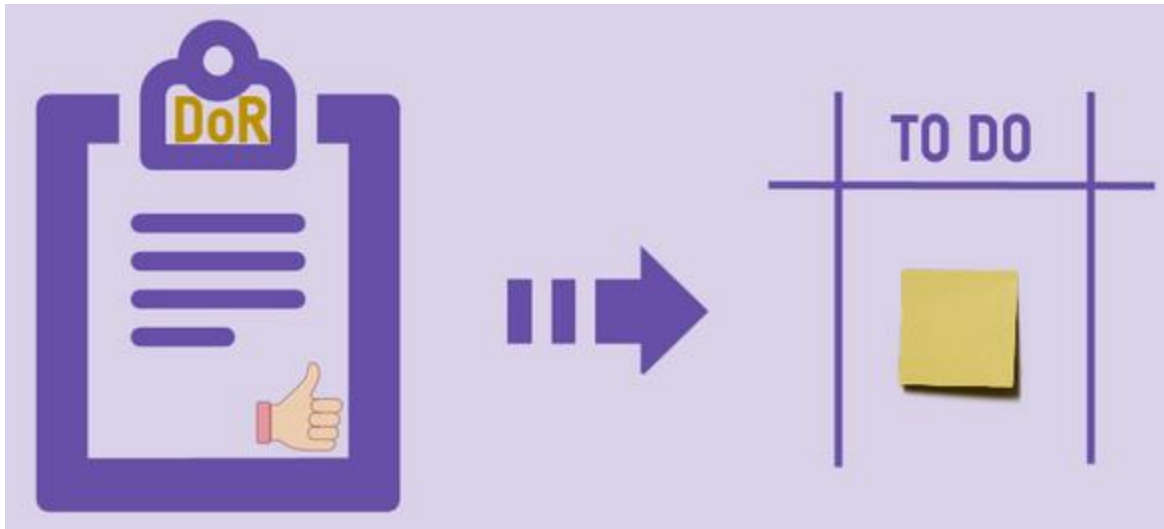
- Criterios de aceptación



- Los criterios de aceptación definen qué es necesario para considerar una historia particular como completa (desde el punto de vista del usuario).
- Determinan límites, eliminan ambigüedades y suelen traducirse en pruebas
- SMART
 - Specific (Específicos)
 - Measurable (Medibles)
 - Achievable (Alcanzables)
 - Relevant (Relevantes)
 - Time-boxed (Limitados en el tiempo)

- Definición de hecho (Definition of Done, DoD)
 - Criterios que tiene que cumplir cualquier historia de usuario para que esté hecha
 - Se refiere a la calidad del software y es responsabilidad del equipo
 - Ejemplos
 - La historia fue analizada y se hizo el diseño
 - El código se ha escrito
 - El código está documentado o comentado
 - El código se integró
 - El código se ha probado apropiadamente (pruebas unitarias, de integración, de regresión, etc.)
 - La historia pasó las pruebas de forma manual o automática según los criterios de aceptación
 - La documentación y otros requisitos que tenga el proyecto/producto están hechos

- Definición de preparado (Definition of Ready, DoR)



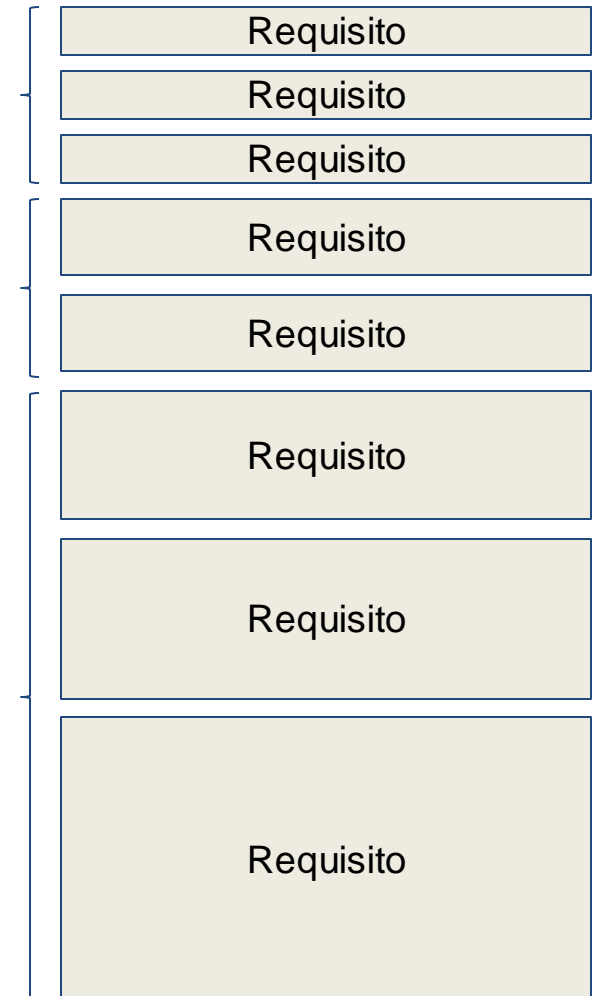
- Qué tiene que estar listo antes de empezar a trabajar en una historia de usuario para que esta pueda entrar en un sprint
- Se refiere a la calidad de los requisitos y es responsabilidad del propietario de producto

Ejemplos:

- Que sea independiente y desarrollable en un solo sprint
- Que los criterios de aceptación que describen la nueva funcionalidad sean detallados
- Que pueda ser testado
- Que no tenga dependencias externas
- Que esté estimada por el equipo
- Aprobación de arquitectura
- Aprobación del usuario

- Requisitos no Funcionales
- A diferencia de los requisitos funcionales, que definen los criterios que se deben cumplir para que el producto desarrollado sea el indicado para su propósito, los requisitos no funcionales especifican criterios que debe de cumplir el producto para que sea adecuado para su uso
 - Comprobabilidad
 - Disponibilidad
 - Extensibilidad
 - Escalabilidad
 - Mantenibilidad
 - Seguridad
 - Usabilidad

- Product Backlog Items (PBIs) Técnicos
- Historias técnicas
 - Historias correspondientes a necesidades técnicas para poder implementar las historias de usuario
- Deuda técnica
 - Historias para resolver la deuda técnica adquirida
- Spikes
 - Historias de investigación o exploración para adquirir conocimiento y reducir riesgos técnicos

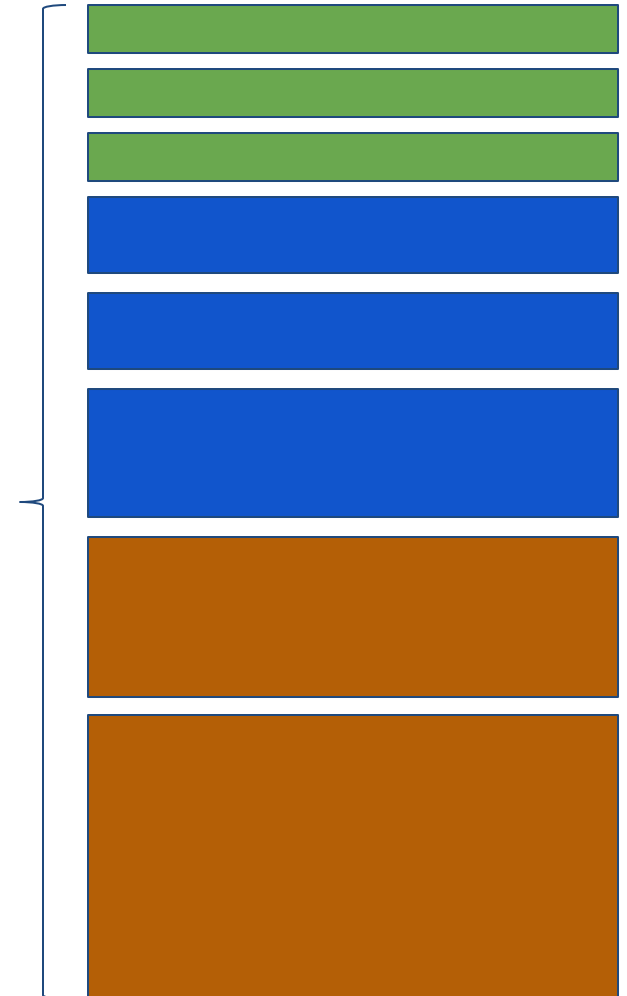


REFINAMIENTO DEL BACKLOG

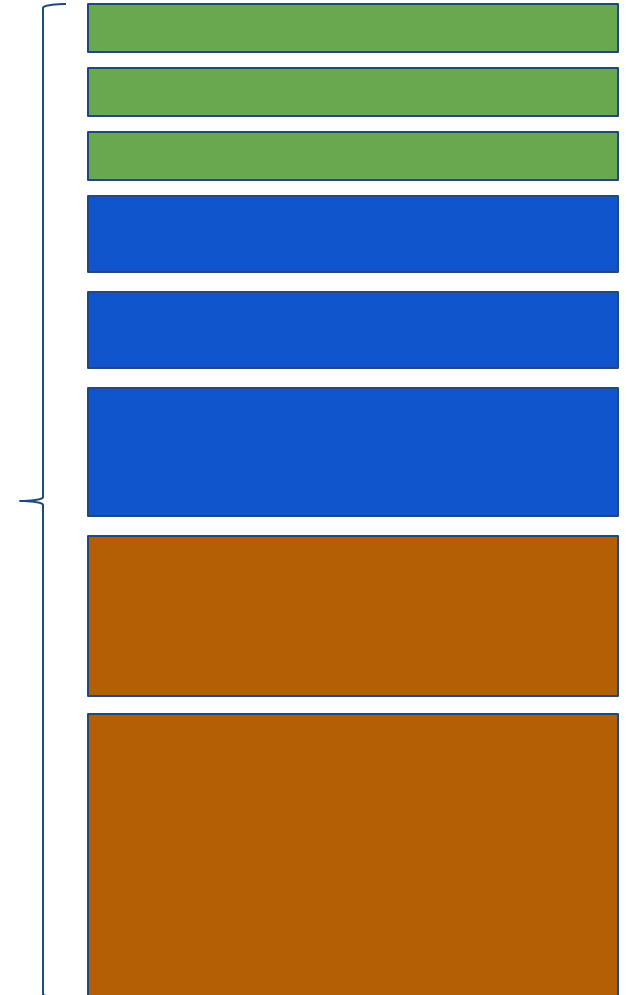
imatia
innovation

We help you to do more

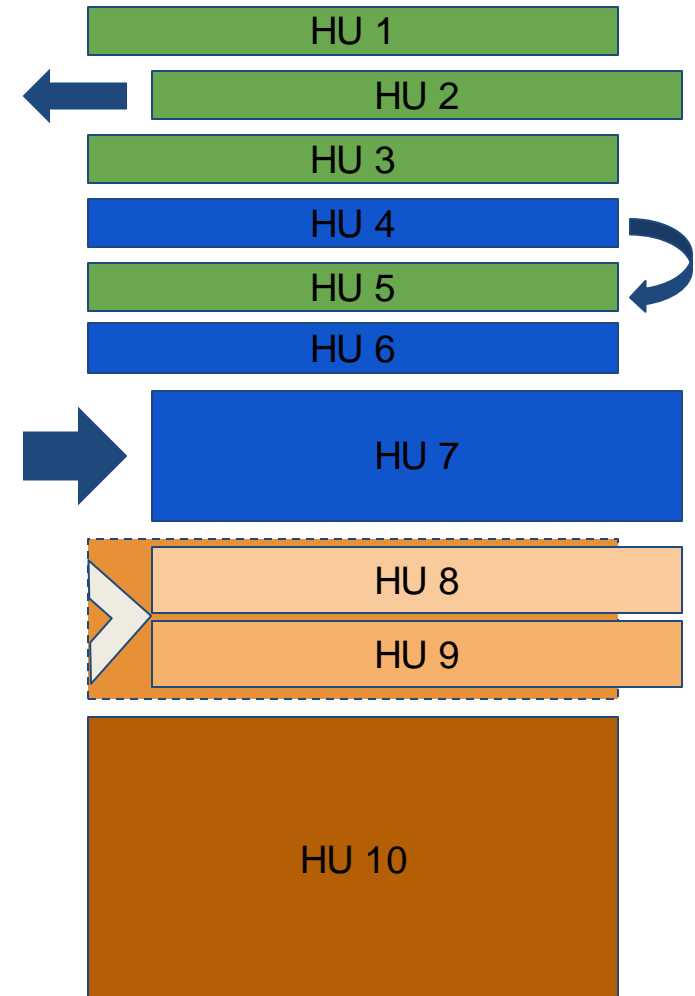
- Pila de producto
- La Pila de Producto es una lista ordenada de todo lo que podría ser necesario en el producto es la única fuente de requisitos para cualquier cambio a realizar en el producto.
- La Pila de Producto nunca está completa.
- La Pila de Producto enumera todas las características, funcionalidades, requisitos, mejoras y correcciones
- Los elementos de la Pila de Producto de orden más alto son generalmente más claros y detallados que los de menor orden.



- Pila de producto
 - ¿Quién la gestiona?
 - El Product Owner
 - ¿Quién puede acceder a ella?
 - Todo el mundo
 - ¿De quién proceden los requisitos?
 - De los Stakeholders
 - ¿Cómo se prioriza?
 - Por valor de negocio, urgencia, tamaño, riesgo
 - Valor negocio / Tamaño



- Refinamiento
 - Actividad desarrollada por el PO y el equipo de preparación para los siguientes Sprints, consistente en
 - Mantener la pila ordenada
 - Modificar la prioridad de elementos no importantes
 - Dividir elementos en ítems más pequeños
 - Combinar ítems en un ítem más grande
 - Estimar ítems



- Actividad: Análisis y división de historias
 - Como alumno quiero poder acceder a los foros de Moodle para poder exponer mis dudas.
 - Como empleado quiero poder dar de alta anticipos para poder recibir el dinero para gastos por anticipado.
 - Como equipo de desarrollo quiero que el usuario pueda dar de alta incidencias para que no llame directamente.
 - Siendo cliente del banco quiero poder retirar efectivo del cajero automático para poder tener dinero en horario no laboral.

ESTIMACION ÁGIL

imatia
innovation

We help you to do more

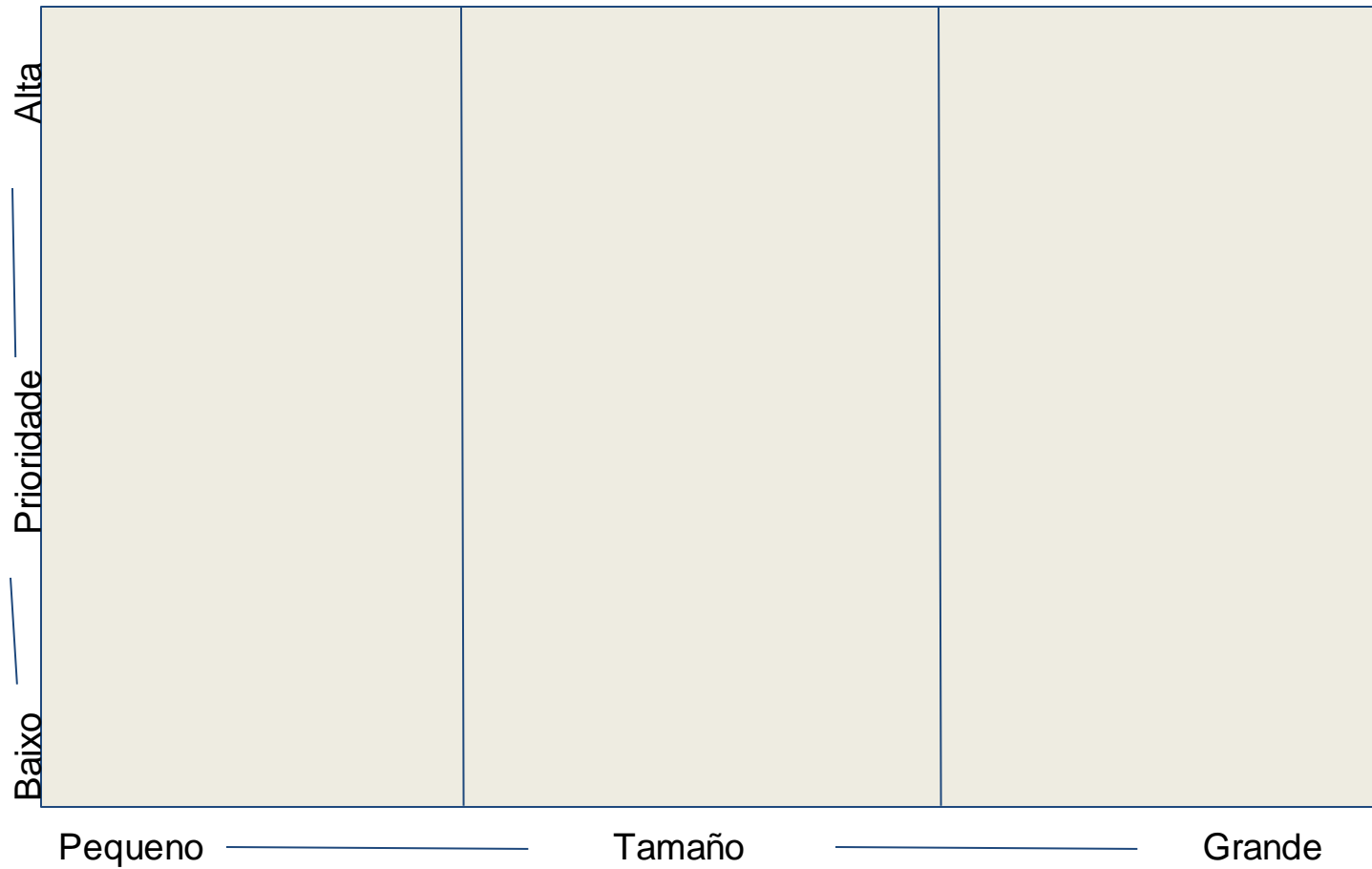
- Estimación relativa



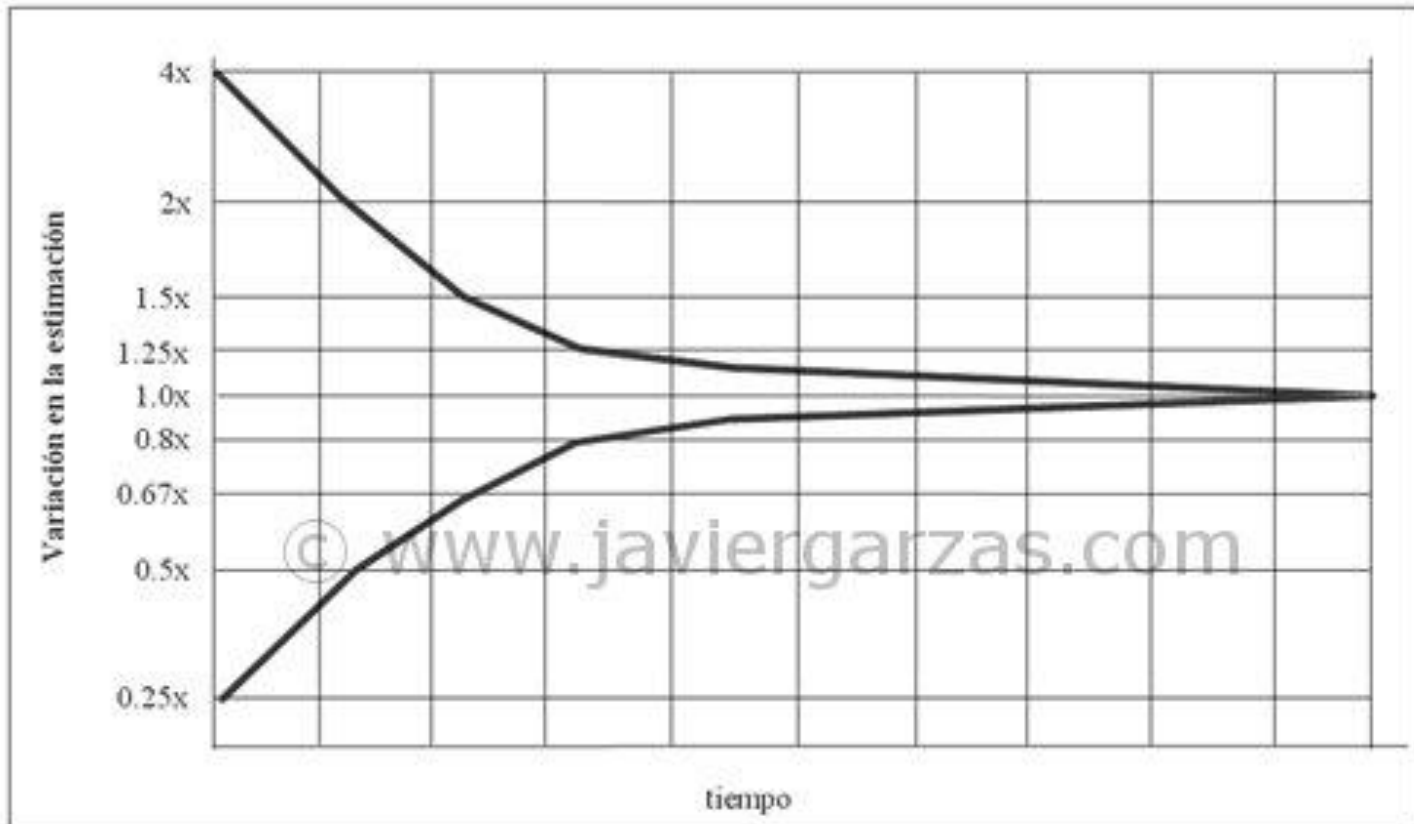
- Planning poker



- Big Wall



- Cono de incertidumbre



- Ejercicio: Estimacion en equipo

Como regla general, es mejor conservar a un enemigo intacto que destruirlo. Captura a sus soldados para conquistarlos y domina a sus jefes.

Un General decía: "Practica las artes marciales, calcula la fuerza de tus adversarios, haz que pierdan su ánimo y dirección, de manera que aunque el ejército enemigo esté intacto sea inservible: esto es ganar sin violencia. Si destruyes al ejército enemigo y matas a sus generales, asaltas sus defensas disparando, reúnes a una muchedumbre y usurpas un territorio, todo esto es ganar por la fuerza."

Por esto, los que ganan todas las batallas no son realmente profesionales; los que consiguen que se rindan impotentes los ejércitos ajenos sin luchar son los mejores maestros del Arte de la Guerra.

- Ejercicio: Planning poker animal



Pinguino

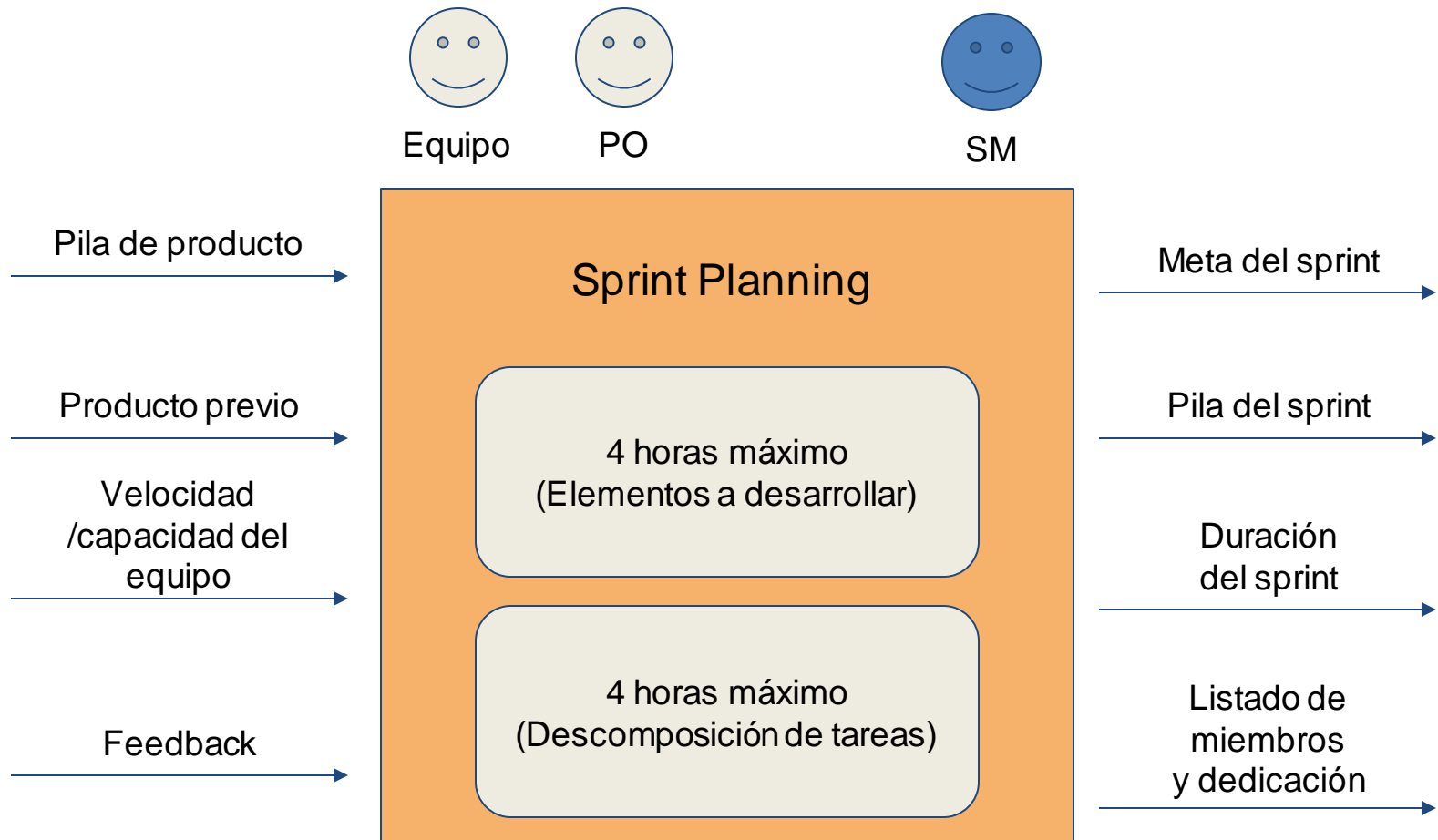
1 zoo-punto

EVENTOS DEL SPRINT: PLANIFICACION

im
innovation

We help you to do more

- Reunión de planificación

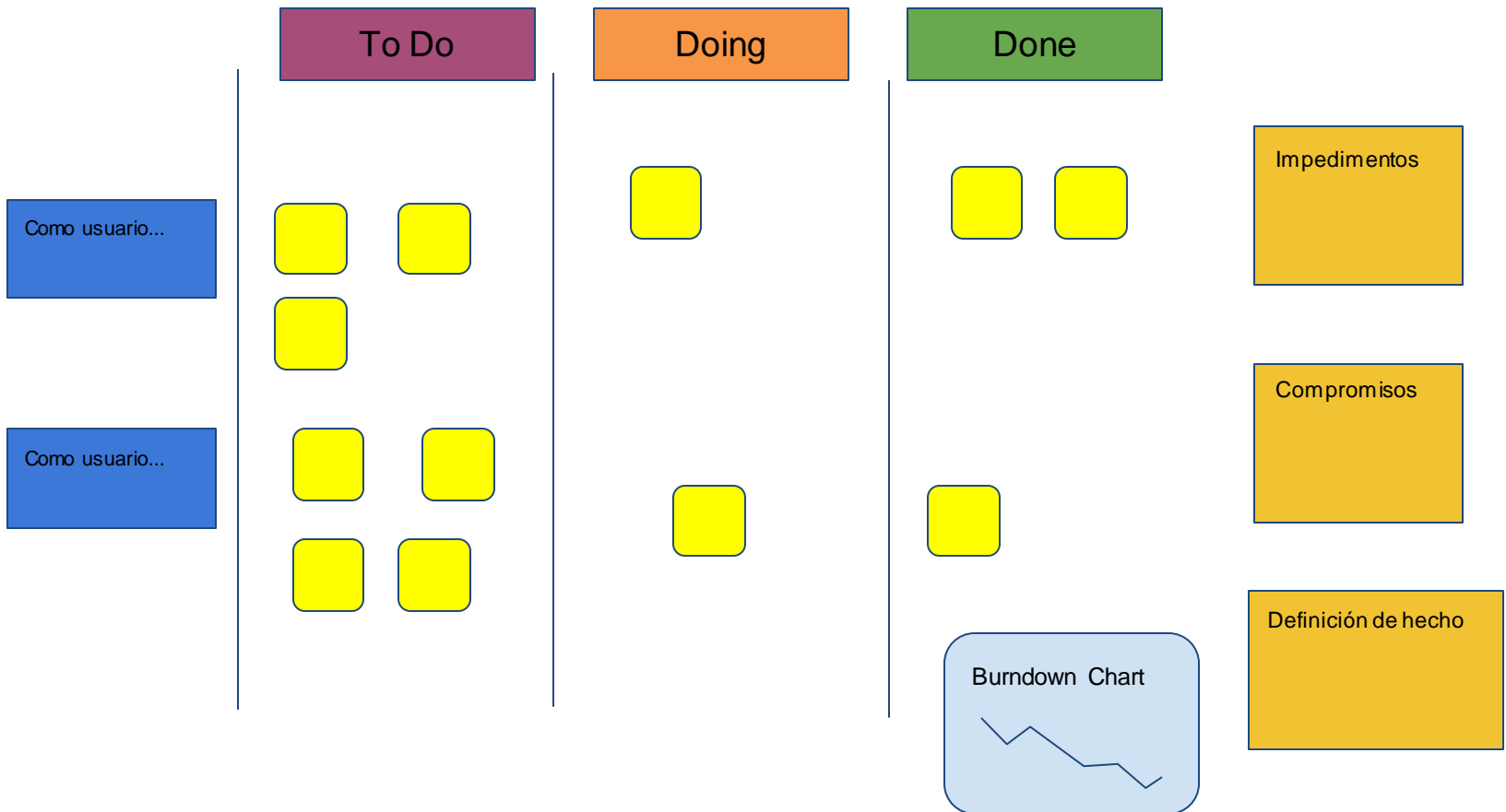


- Desglose en tareas



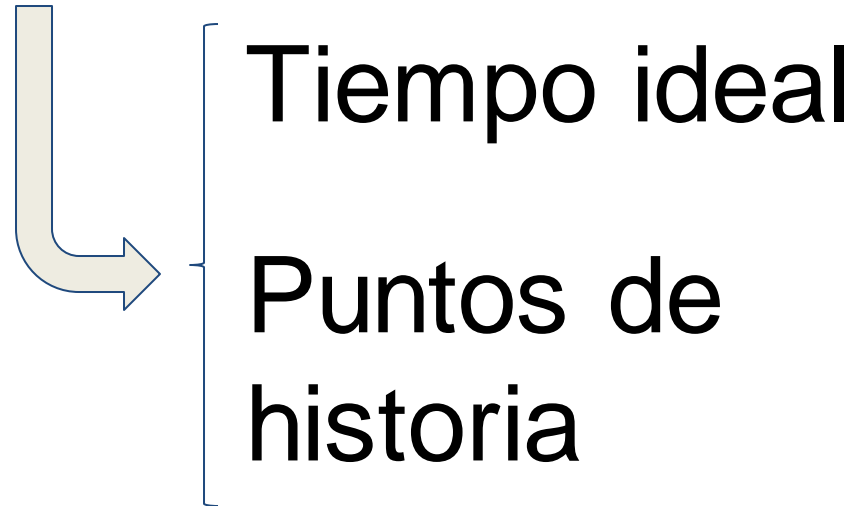
- Pila del sprint (Sprint Backlog)
 - Es una lista dinámica de tareas en las que se descomponen las funcionalidades de la Pila de Producto elegidas para la construcción de un incremento
 - ¿Quién las gestiona?
 - El equipo
 - ¿Quién puede acceder a ella?
 - Todo el mundo
 - ¿Cómo se prioriza?
 - Lo decide el equipo, atendiendo a la secuencia lógica, dependencias, promoción del aprendizaje...

- Tablero de tareas



- Velocidad

$$\text{Velocidad} = \text{Trabajo} / \text{Sprint}$$



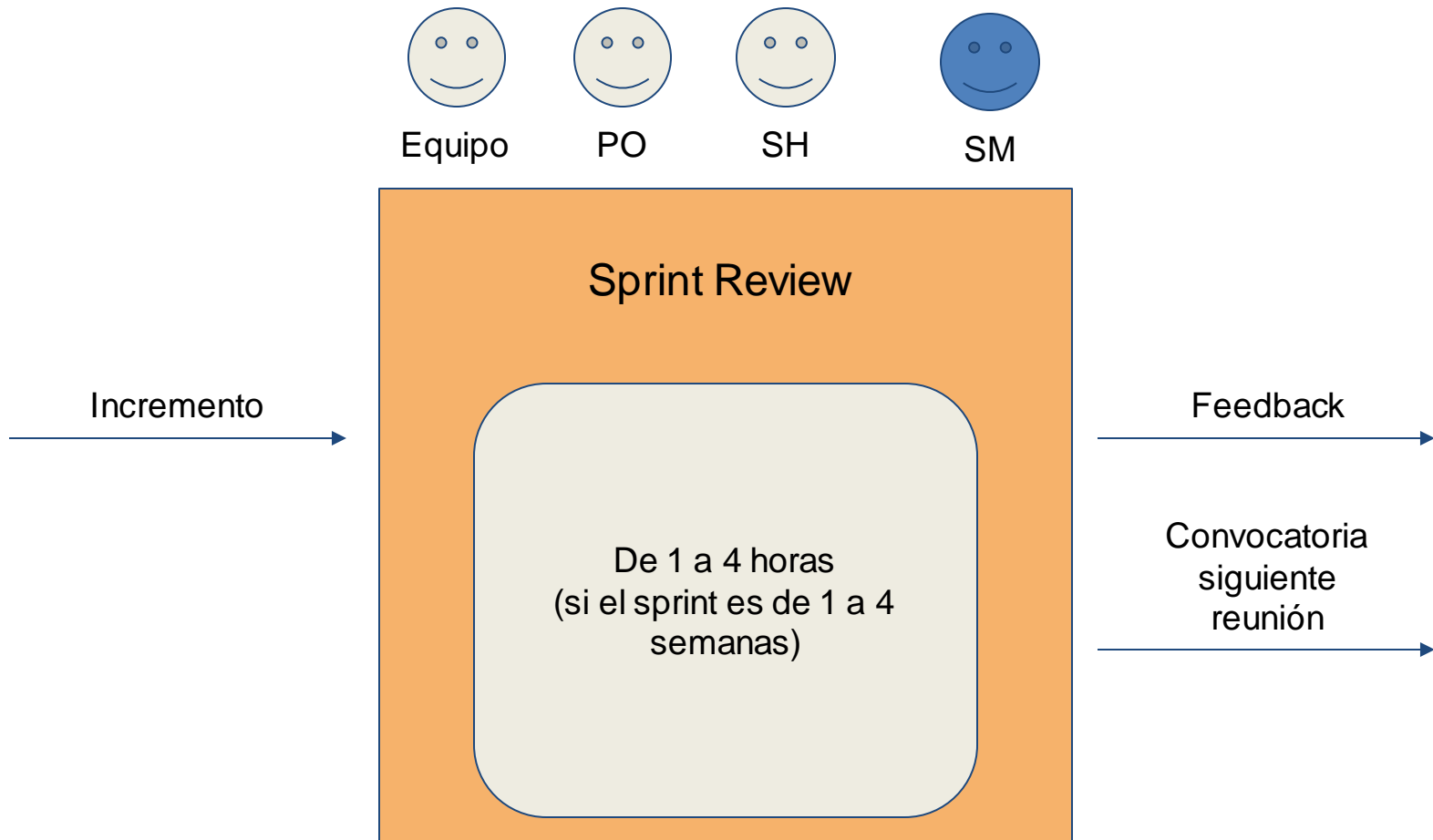
- Evitar el exceso de capacidad
- Explicitar las suposiciones sobre las historias
- No llenar el sprint hasta agotar la capacidad
- Utilizar spikes para reducir la incertidumbre



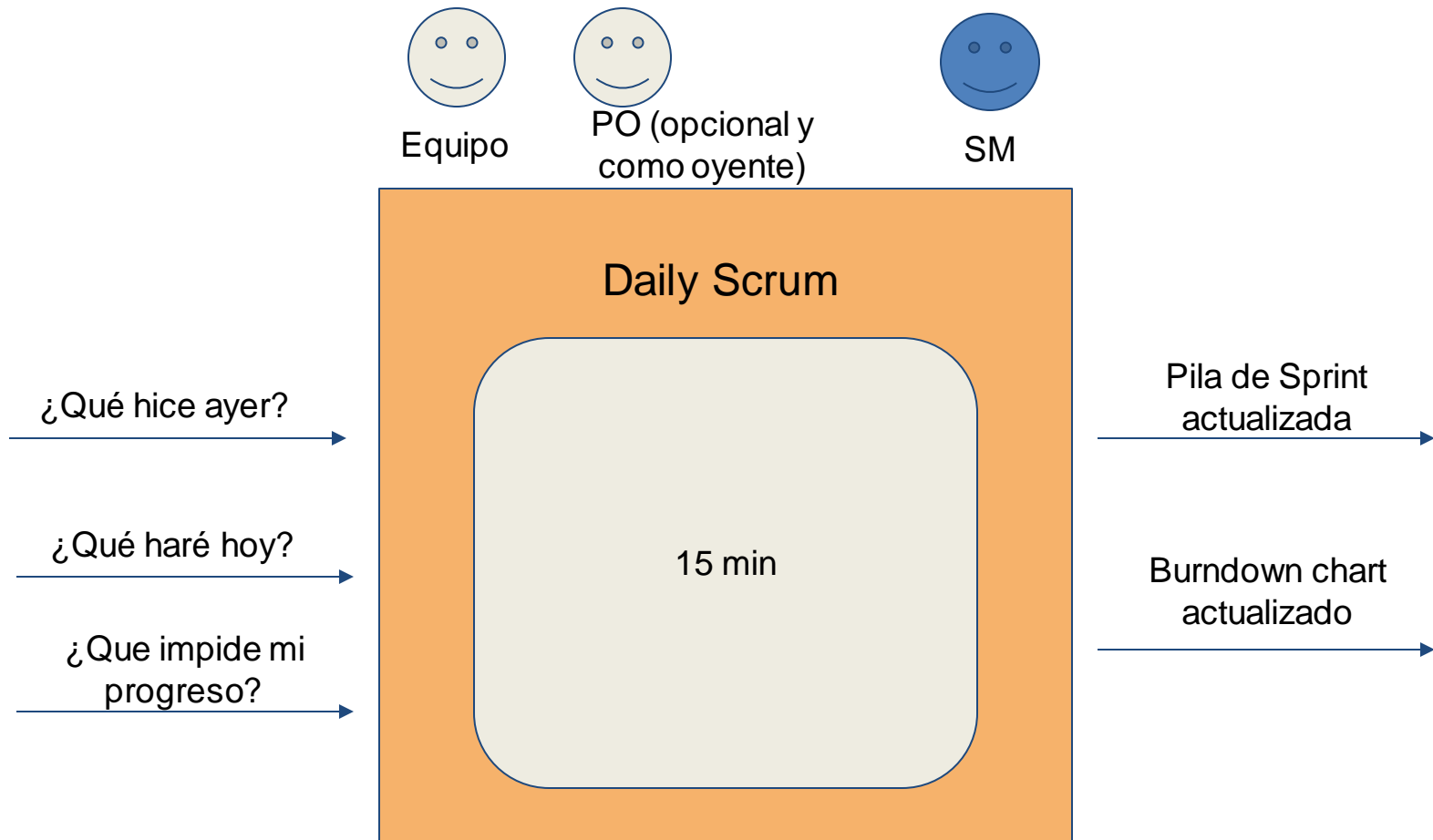
EVENTOS DEL SPRINT: DAILY, SPRINT REVIEW Y RETROSPECTIVA

We help you to do more

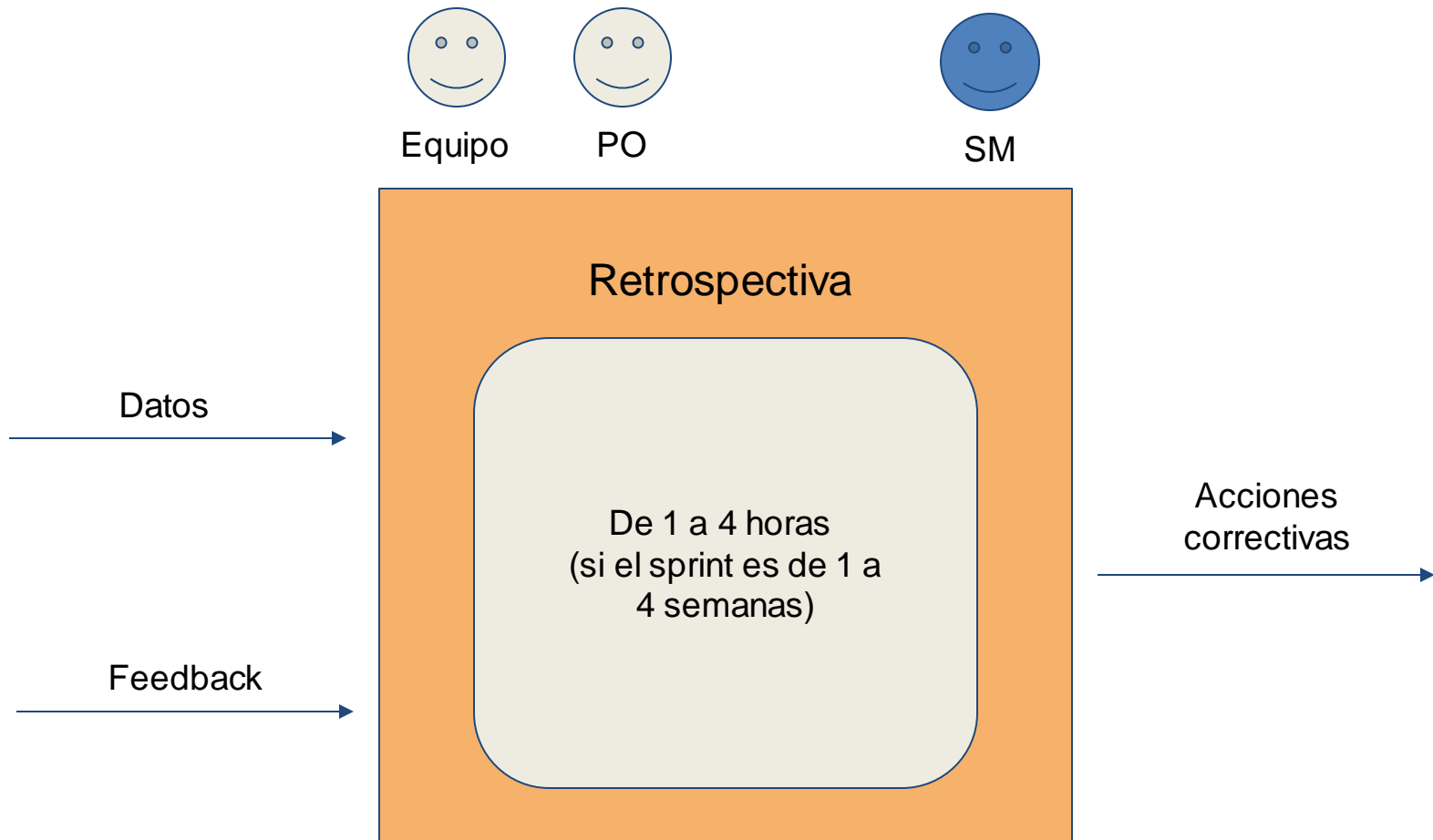
- Revisión del sprint (Sprint review)



- Reunión diaria (Daily Scrum)



- Retrospectiva



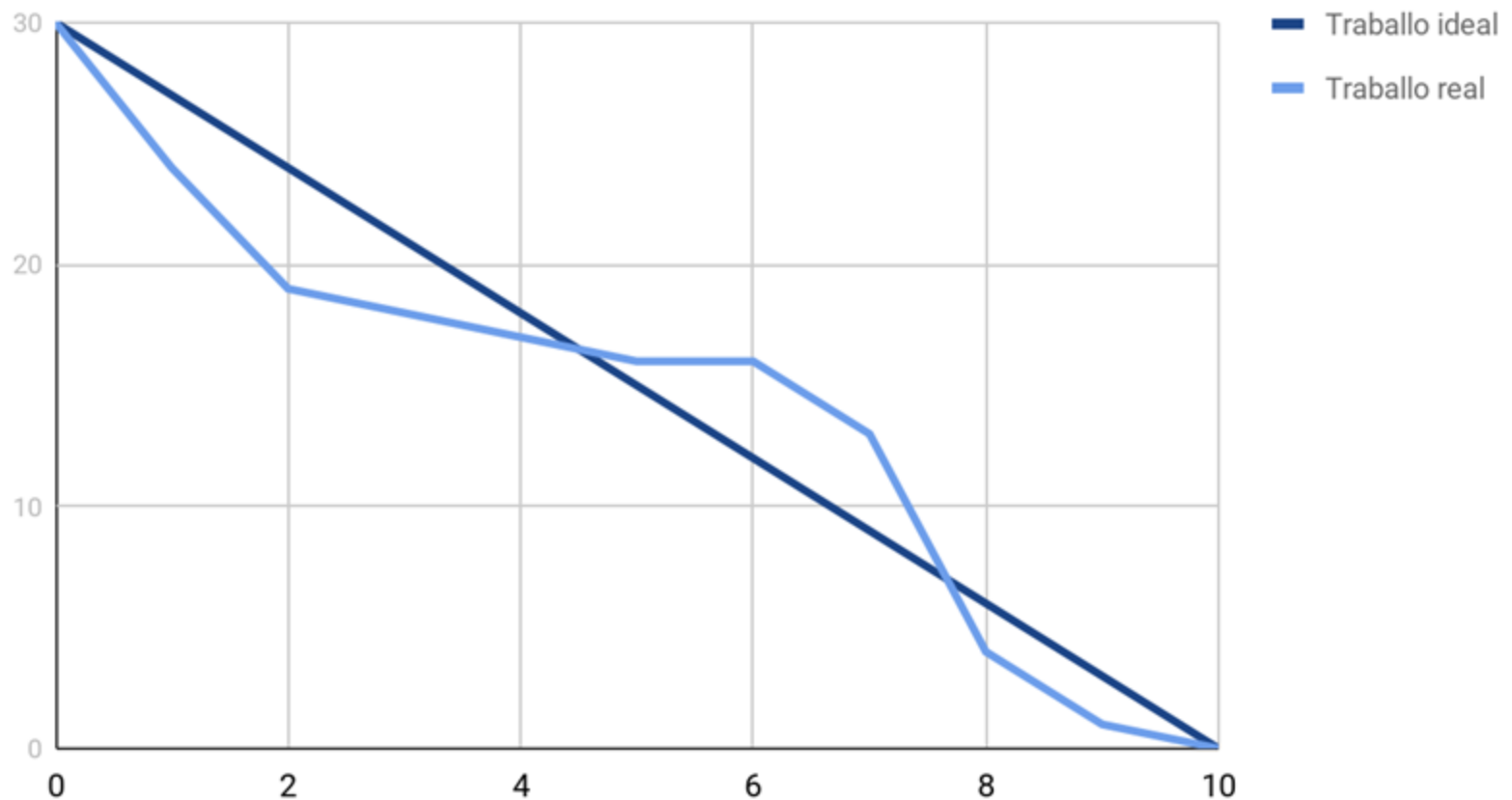
OTROS ARTEFACTOS

imatia
innovation

We help you to do more

- Otros artefactos

Gráfico de burndown



- Otros artefactos

Gráfico de burndown

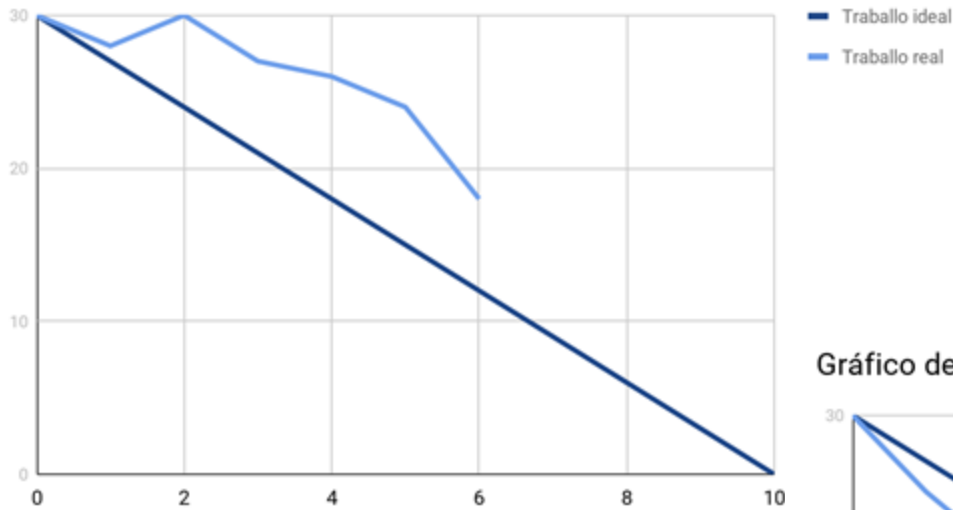
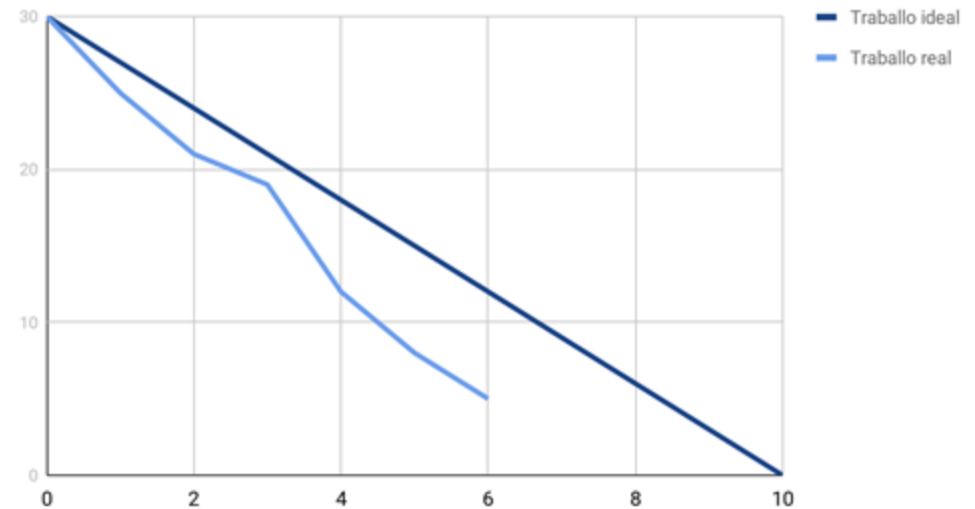


Gráfico de burndown



DEUDA TÉCNICA

imatia
innovation

We help you to do more

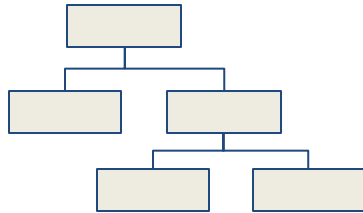
- Deuda técnica



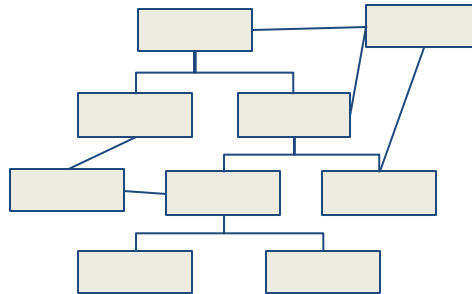
Ejemplos de deuda técnica:

- Código poco claro e ilegible
- Falta de pruebas automatizadas, construcción automatizada, despliegue automatizado...
- Falta de calidad
- Código duplicado
- Arquitectura enmarañada y dependencias innecesariamente complejas (código espaguetti)
- Herramientas ineficaces y lentas
- Código no subido y ramas de larga duración
- Documentación técnica importante inexistente u obsoleta
- Falta de entornos de test y que no sean equivalentes a los entornos de producción
- Ciclos de construcción y test muy largos
- Falta de integración continua

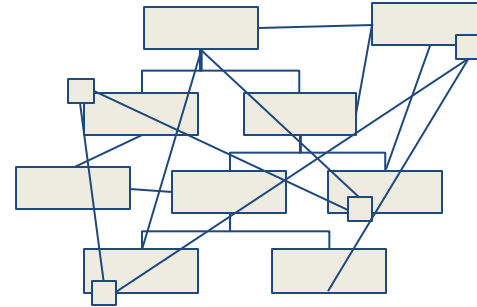
- Deuda técnica



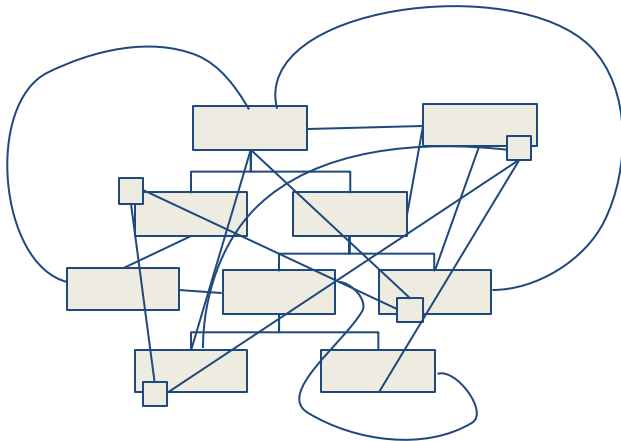
Coste del cambio: C



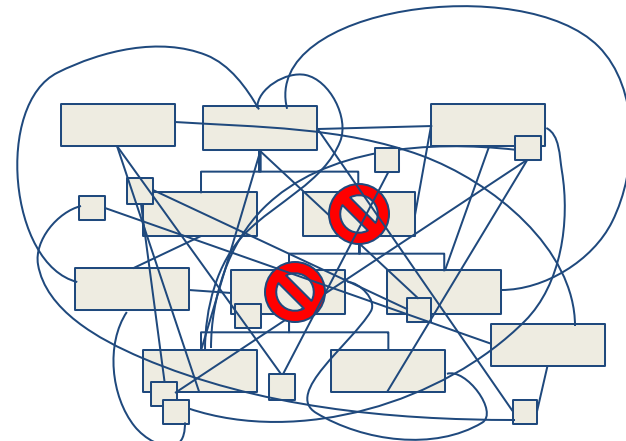
Coste del cambio: $C+n$



Coste del cambio: $C \cdot n$



Coste del cambio: C^n



Coste del cambio: C^{n^n}

- Deuda técnica
- Causas inintencionadas
 - Personas
 - Falta de formación o conocimiento del proceso
 - Habilidades equivocadas
 - Falta de comprensión de las historias
 - Colaboración pobre
 - Miembro nuevo escribiendo código de manera equivocada
 - Técnica
 - Infraestructura equivocada
 - Diseño equivocado
 - Sin convenciones de código
 - Sistema de código descuidado



```

78 // ... trim(preg_replace('/\\\\\\\\/', '/', $in
79 $SESSION['_CAPTCHA']['config'] = serialize($captcha_con
80
81 return array(
82     'code' => $captcha_config['code'],
83     'image_src' => $image_src
84 );
85 }
86
87
88 if (function_exists('hex2rgb')) {
89     function hex2rgb($hex_str, $return_string = false, $separator) {
90         $hex_str = preg_replace("/[^0-9A-Fa-f]/", '', $hex_str);
91         $rgb_array = array();
92         if (strlen($hex_str) == 6) {
93             $color_val = hexdec($hex_str);
94             $rgb_array['r'] = 0xFF & ($color_val >> 0x10);
95             $rgb_array['g'] = 0xFF & ($color_val >> 0x08);
96             $rgb_array['b'] = 0xFF & $color_val;
97         } elseif (strlen($hex_str) == 3) {
98             $rgb_array['r'] = hexdec(str_repeat(substr($hex_str, 0, 1), 2));
99             $rgb_array['g'] = hexdec(str_repeat(substr($hex_str, 1, 1), 2));
100             $rgb_array['b'] = hexdec(str_repeat(substr($hex_str, 2, 1), 2));
101         } else {
102             return false;
103         }
104         return $return_string ? implode($separator, $rgb_array) : $rgb_array;
105     }
106 }
107
108 // Draw the image
109 if (!isset($_GET['captcha'])) {

```

- Deuda técnica
- Causas intencionadas
 - Equipo
 - Presiones comerciales
 - Pereza
 - Dinero
 - Política de soluciones rápidas
 - Técnicas
 - Nombres genéricos de variables
 - Enfoque de modelado de objetos pobre
 - Métodos largos
 - Métodos de usos múltiples
 - Sin convenciones de código o patrones de diseño
 - Chapuza ahora, arréglalo más tarde
 - Falta de pruebas automatizadas
 - Falta de documentación
 - Desarrollo en paralelo de varias ramas en el gestor de versiones

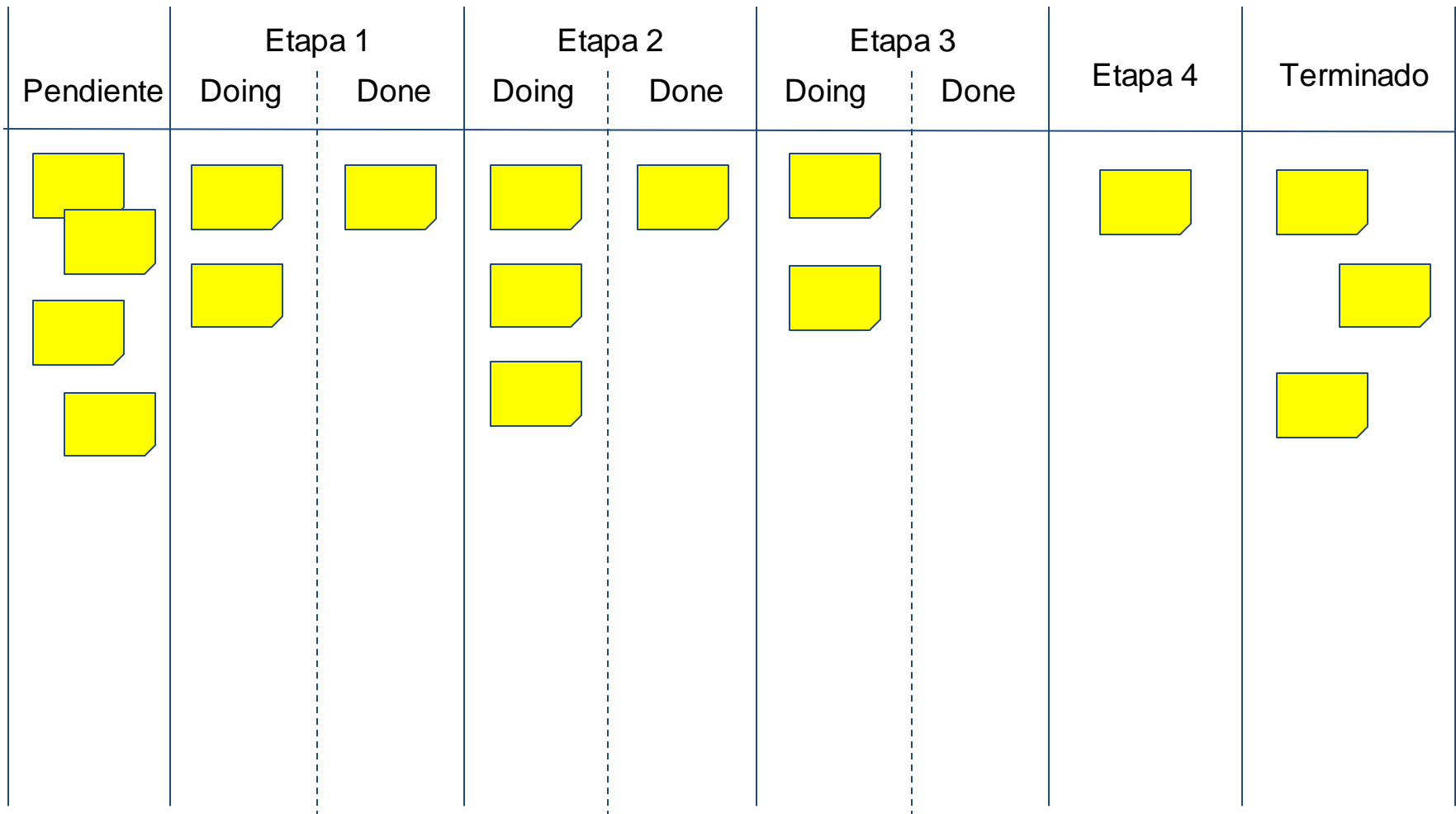
- Deuda técnica
- Consecuencias
 - Falta de motivación
 - Frustración
 - Pérdida de productividad
 - Más dificultades para desarrollar -> más incidencias
- Negocio
 - Mayor tiempo de entrega
 - Incrementa el riesgo
 - Incrementa las malas decisiones
- Gestión
 - Pila
 - Crear PBI de deuda técnica con su correspondiente estimación
 - Crear una pila paralela de deuda técnica
 - Sprint
 - Definir un sprint de deuda técnica (evitar sprints "temáticos")
 - Estrategias de amortización: 10% de los PH para deuda técnica
 - Prevención
 - Utilizar TDD e integración continua

EQUIPOS KANBAN

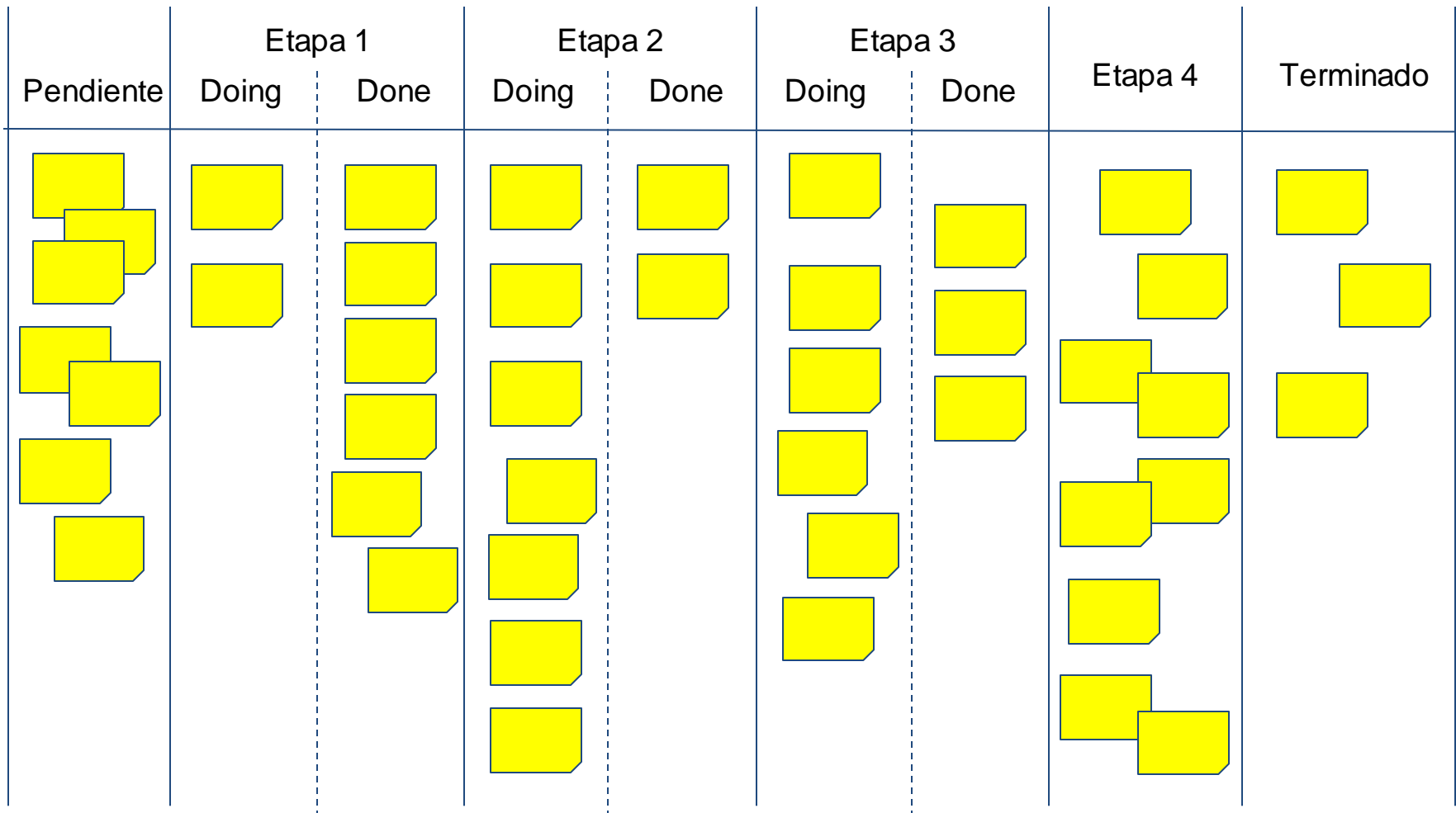
imatia
innovation

We help you to do more

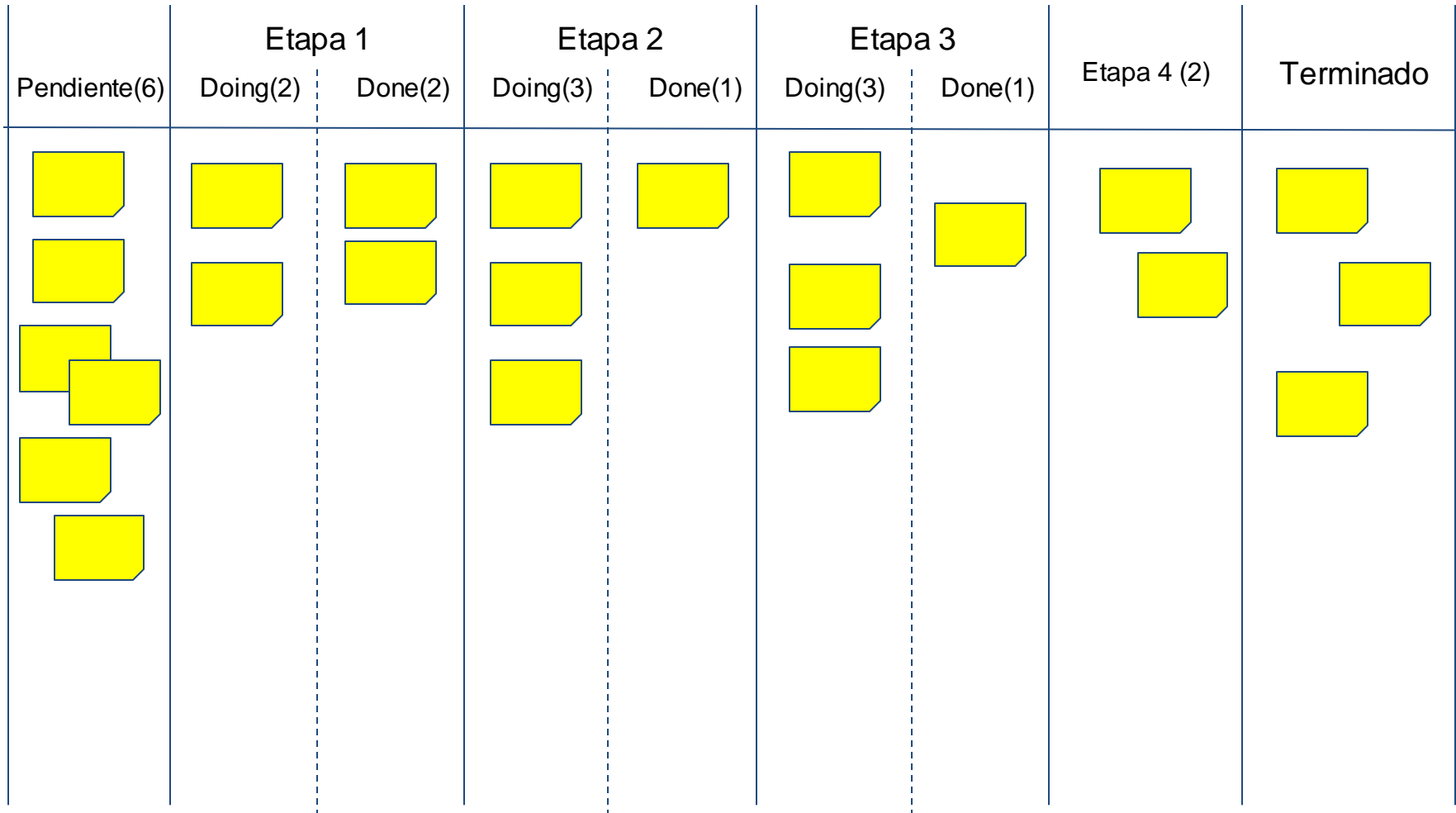
- Tablero Kanban



- Sin límites WIP



- Con sistema de arrastre de límites WIP



- Tarjetas kanban

#412365	Responsable	Fecha fija		
Exportación a PDF				
Backlog	Comienzo	Desarrollo	Fin	Total

- Actividad: Diseñar un tablero kanban
 - Pendiente (oferta de selección pendiente de publicar)
 - Publicar
 - Sin cuestionario: Doing (2), Done (1)
 - Con cuestionario: Doing (2), Done (1)
 - Filtrar CV (2)
 - En espera (4)
 - Para entrevista (2)
 - Descartados
 - Primera selección (2)
 - Concertar entrevista
 - ...
 - Terminado

Ejercicio alternativo:

[KanbanCakes](#)



- Gestión visual de flujo

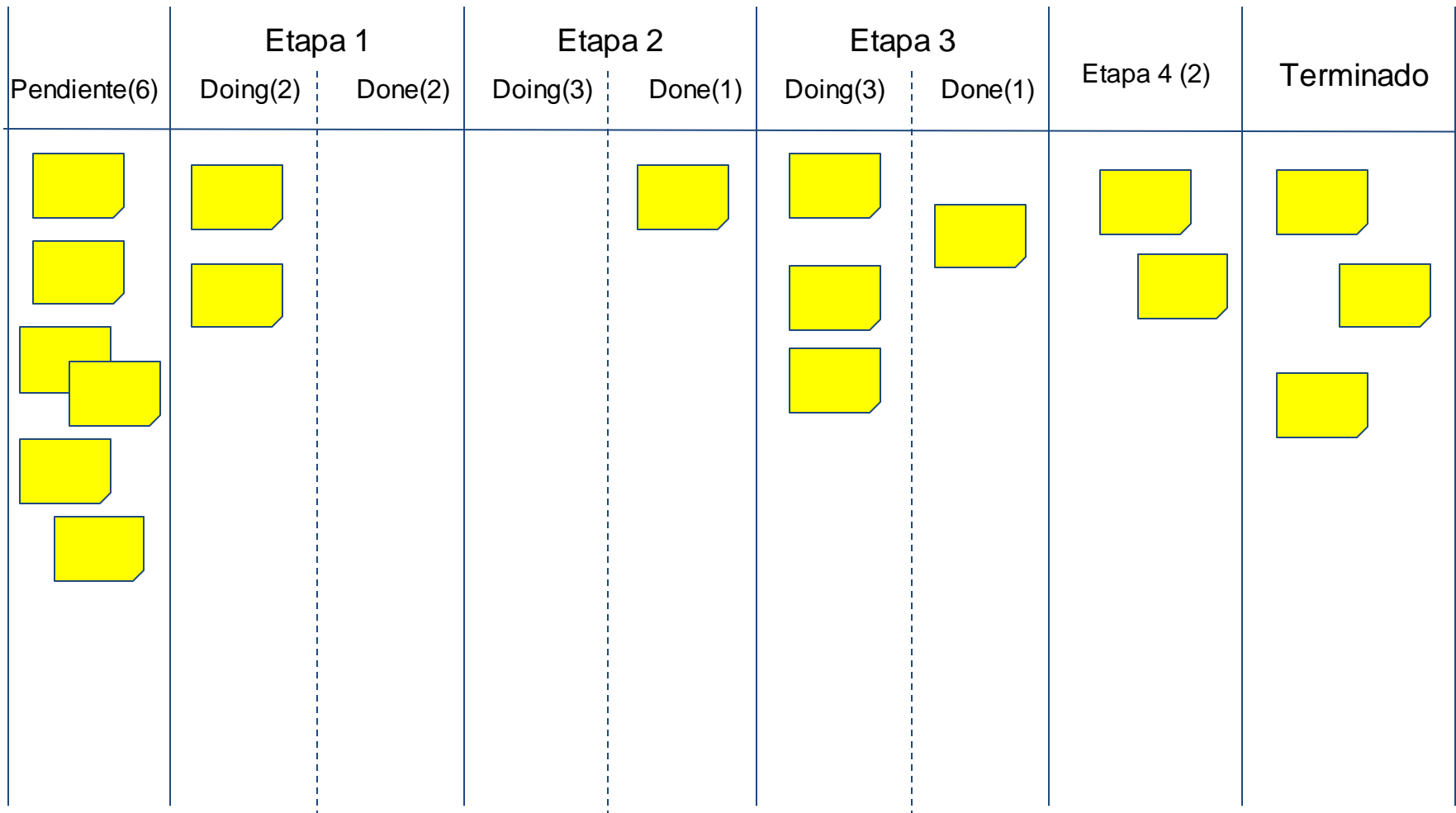
Usando tableros y tarjetas kanban se visualiza el flujo de trabajo, lo que permite:

- Evidenciar colas invisibles de trabajo a medio hacer
- Evidenciar problemas: bloqueos, cuellos de botella
- Favorece la colaboración
- Facilita la visión agregada desde lejos

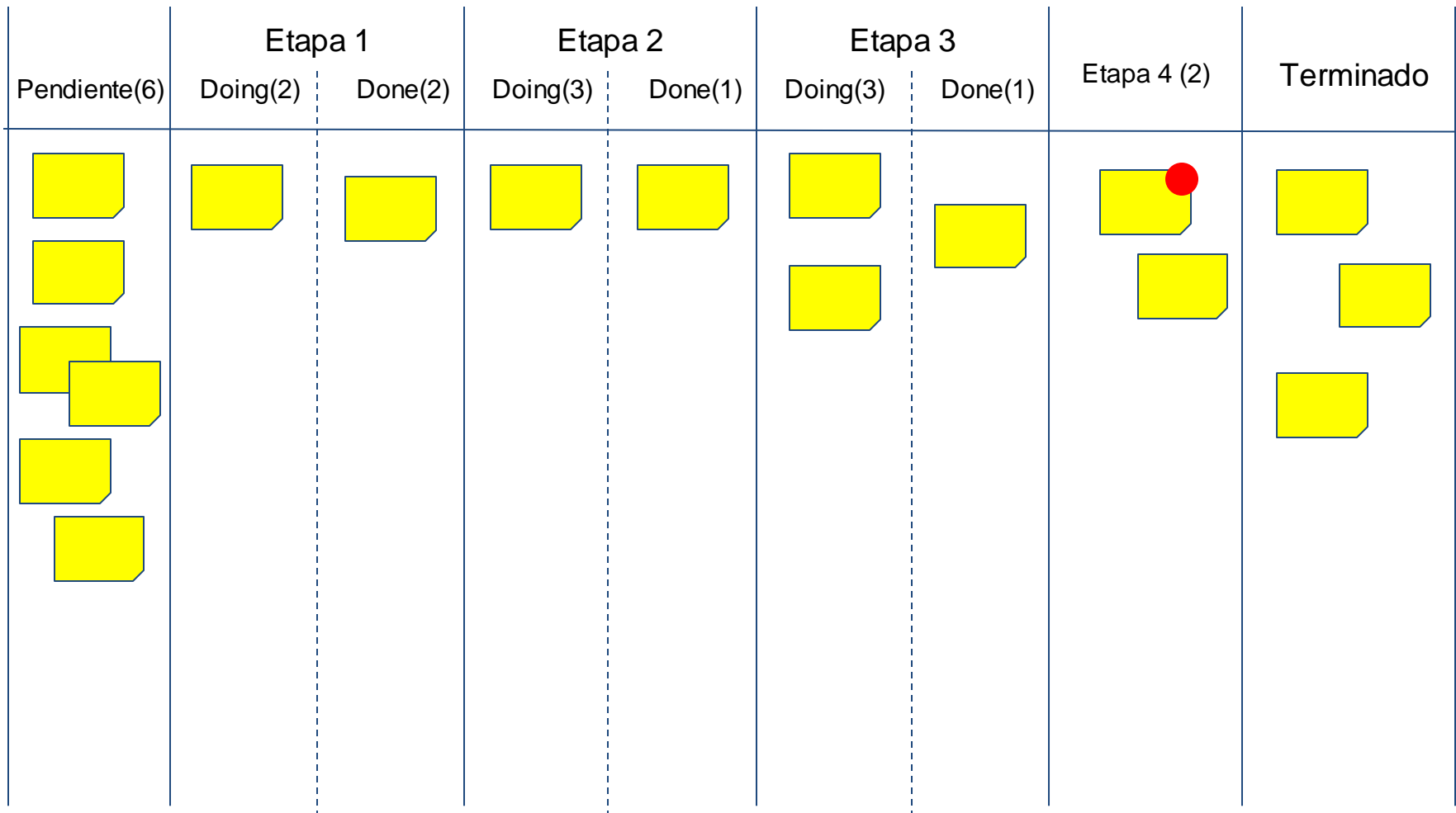
- Medición de progreso
 - Lead time (Tiempo de entrega al cliente)
 - Cycle time (Tiempo de entrega del sistema)



- Hambruna



- Señalización de bloqueos



- Tarjetas, clases y niveles de servicio

Urgente

Ejemplo: Petición de soporte
Nivel: Lo más pronto posible

Fecha entrega
fija

Ejemplo: Nueva funcionalidad
Nivel: 100% a tiempo

Estándar

Cualquiera
Tiempo de entrega máximo garantizado al 95%

- Ejercicio: Simulador de flujo Kanban

Simulador Kanban para comprobar cómo funciona el WIP:
www.kanbansim.org

Simulación interactiva Kanban para jugar con ella:
flowban.herokuapp.com