



Universitat
de Lleida

#AGILE #SCRUM #MANAGEMENT

GESTIÓN ÁGIL DE PROYECTOS



Jordi Torrecillas Garcia



@jtorrecillas2

AGENDA

- ▶ 9:00 - 12:00 -> Proyectos, Agile y SCRUM (J. Torrecillas)
- ▶ 12:30 - 14:00 -> Github & Zenhub (A. Corchero)
- ▶ 16:00 - fin -> Taller práctico (J. Gervas)

¿QUIÉN SOY?

FORMACIÓN

ESPECIALIZACIONES

¿EN QUÉ TRABAJO?

INGENIERO TÉCNICO INFORMÁTICO

**MASTER IN
TECHNOLOGY IN BUSINESS**

**MASTER IN
PROJECT MANAGEMENT**

**PMP CERTIFIED
(PROJECT MANAGER PROFESSIONAL)**

CURSOS AGILE, SCRUM

**PMO
(GRUPO ALIMENTARIO ARGAL)**

**PROFESOR ADJUNTO
(UNIVERSITAT DE LLEIDA)**

ÍNDICE

- ▶ Proyectos, empresas y entorno
- ▶ Ágil "*agile*"
- ▶ SCRUM



Universitat
de Lleida

PROYECTOS, EMPRESAS Y ENTORNO

PROYECTOS, EMPRESAS Y ENTORNO

- ▶ Tipos de empresas
- ▶ ¿Qué es un proyecto?
- ▶ Modelos de desarrollo



TIPOS DE EMPRESAS

Proyectos, empresas y entorno

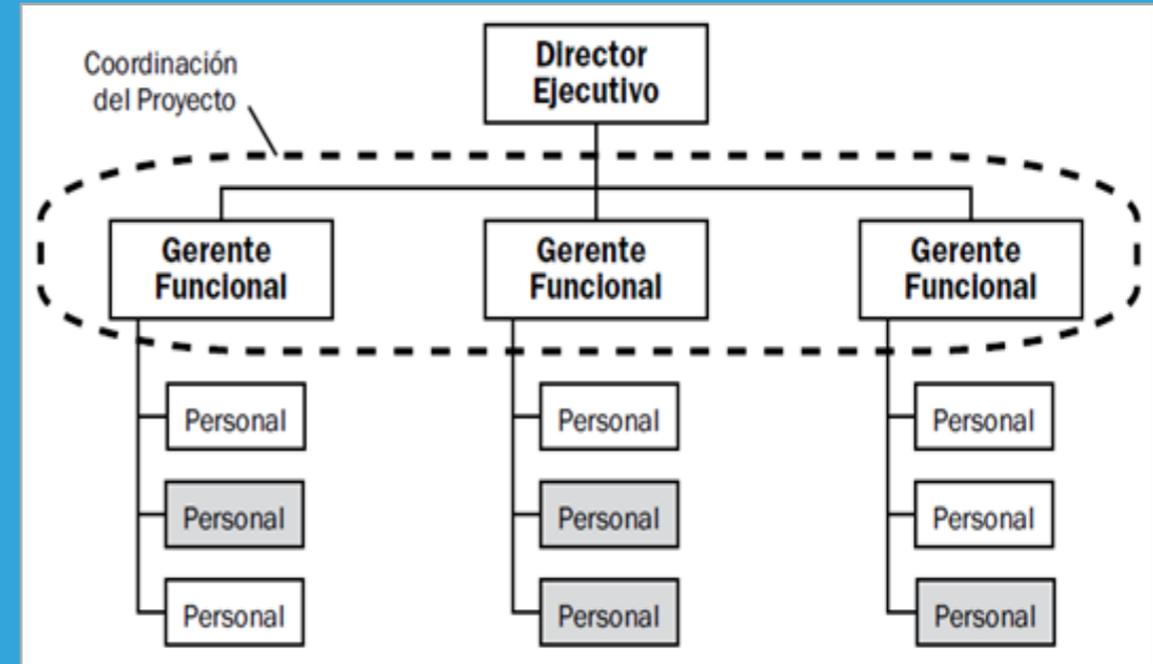
TIPOS DE EMPRESAS

FUNCIONALES

MATRICIALES

PROYECTIZADAS

- ▶ Mucha jerarquía
- ▶ Superior claramente definido.
- ▶ Trabajadores agrupados por especialidades y conocimientos.
- ▶ Departamentos trabajan de forma independiente.
- ▶ Cada director "funcional" coordina y responsabiliza a sus trabajadores.
- ▶ No existe la figura del director de proyectos como tal.



Organigrama funcional

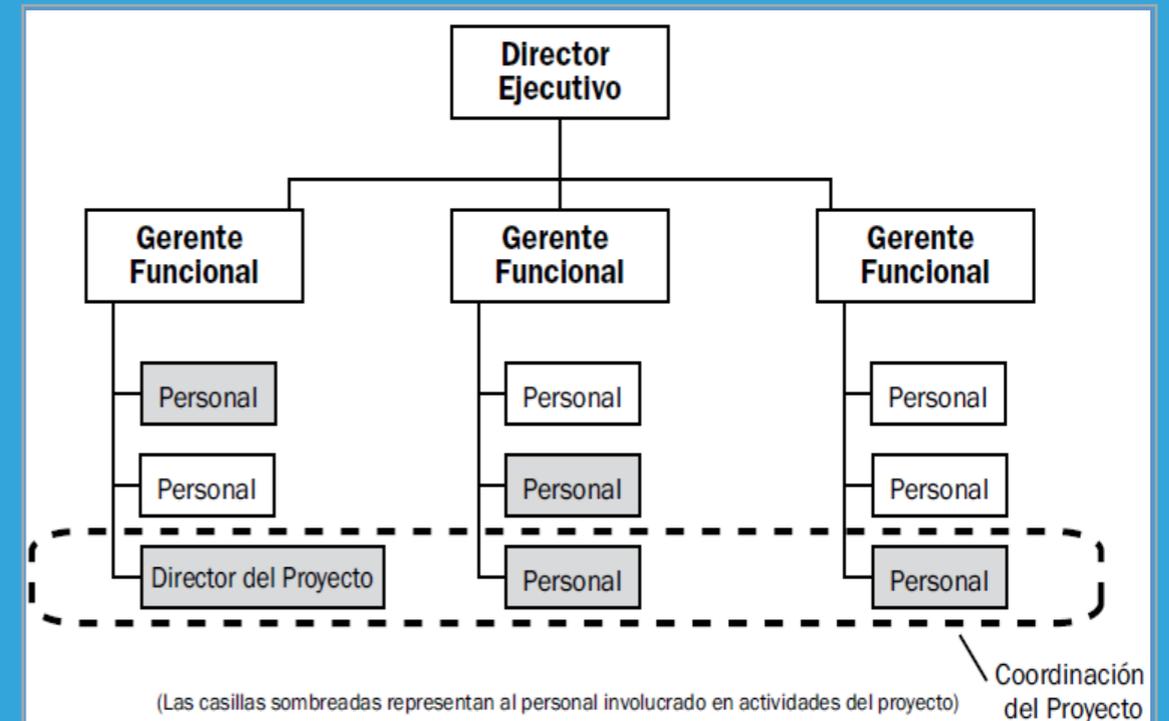
TIPOS DE EMPRESAS

FUNCIONALES

MATRICIALES

PROYECTIZADAS

- ▶ Mezclan características de las organizaciones funcionales y proyectizadas.
- ▶ Se realizan proyectos y se establece una dirección de proyectos; pero con una autoridad moderada.
- ▶ Cada trabajador depende de un director de proyecto y un director funcional.
- ▶ Según el grado: débiles, equilibradas o fuertes.



Organigrama matricial

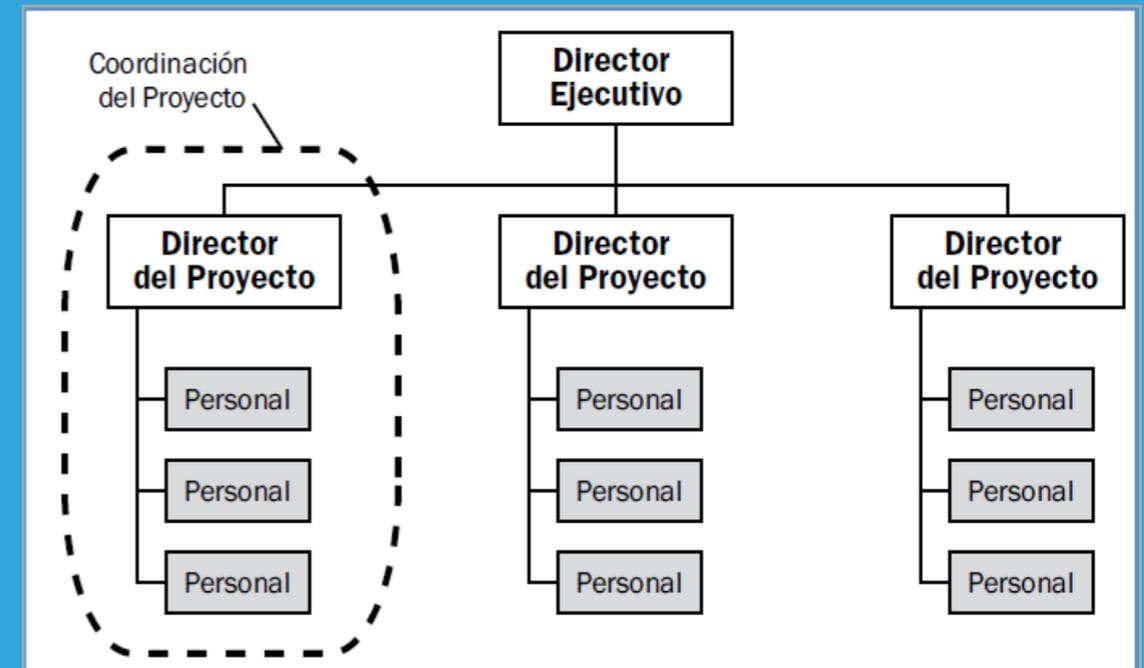
TIPOS DE EMPRESAS

FUNCIONALES

MATRICIALES

PROYECTIZADAS

- ▶ Organización por equipos de trabajo.
- ▶ Equipos multidisciplinarios.
- ▶ Director/responsable de proyecto definido claramente.
- ▶ Existe movimiento entre las personas, equipos y grupos.



Organigrama proyectizado

¿QUÉ ES UN PROYECTO?

Proyectos, empresas y entorno

¿QUÉ ES UN PROYECTO?

¿QUÉ ES UN PROYECTO?

Es un **esfuerzo temporal** que se realiza con el objetivo de crear un producto, servicio o resultado **único**.

Definición PMBoK v5

CARACTERÍSTICAS DE UN PROYECTO

ÚNICO

TEMPORAL

- ▶ **Producto** (edificio, programa, página web...)
- ▶ **Servicio** (asesoramiento o consultoría)
- ▶ **Mejora** (reducir defectos en una línea de producción; tiempo de respuesta web...)
- ▶ **Resultado** (invertir o no invertir; OK / KO; investigación...)

CARACTERÍSTICAS DE UN PROYECTO

ÚNICO

TEMPORAL

- ▶ Tiene un **inicio** y un **final**.
- ▶ Recursos planificados y **limitados**.
- ▶ **No se repite** en el tiempo.
- ▶ Puede durar días o años.

PROYECTO VS TRABAJO OPERATIVO

PROYECTO

TRABAJO OPERATIVO

TEMPORAL

CONTINUADO

RESULTADO ÚNICO

RESULTADOS REPETITIVOS

NUEVO PRODUCTO

MANTENIMIENTO, NEGOCIO,
CADENA PRODUCCIÓN

¿PROYECTO O TRABAJO OPERATIVO?

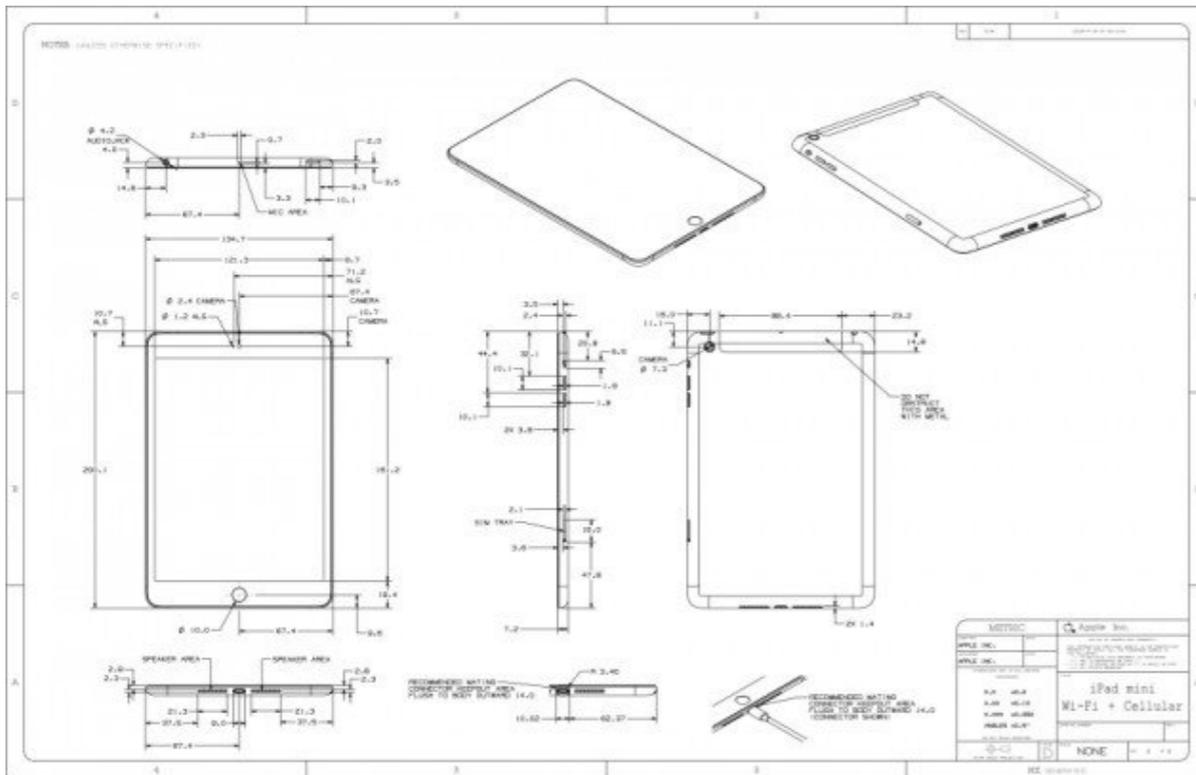


¿PROYECTO O TRABAJO OPERATIVO?



¡DEPENDE!

¿PROYECTO O TRABAJO OPERATIVO?



PROYECTO



TRABAJO OPERATIVO

CONSTITUCIÓN DE UN PROYECTO



NECESIDAD DE
NEGOCIO



AVANCE
TECNOLÓGICO



PETICIÓN CLIENTE



REQUISITO LEGAL

MODELOS DE DESARROLLO.

Proyectos, empresas y entorno

PREDICCIÓN VS ADAPTACIÓN

**MODELOS
PREDICTIVOS**

**MODELOS
ADAPTATIVOS**

LINEALES

ITERATIVOS E INCREMENTALES

POR FASES

ITERACIONES, SPRINTS

REQUISITOS DETALLADOS AL
PRINCIPIO

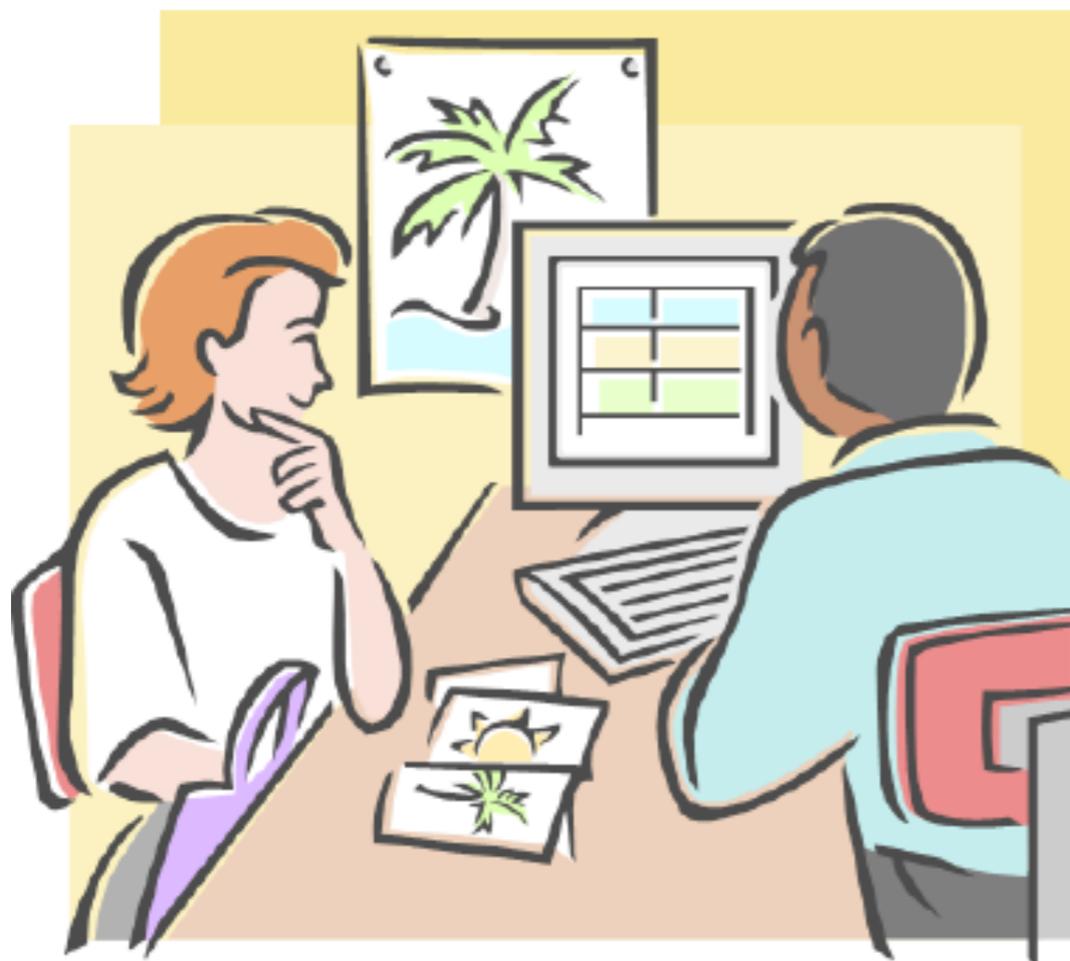
SE AÑADEN REQUISITOS,
CAMBIAN...

PREVISIÓN

ADAPTACIÓN



EN EL MUNDO OFFLINE...



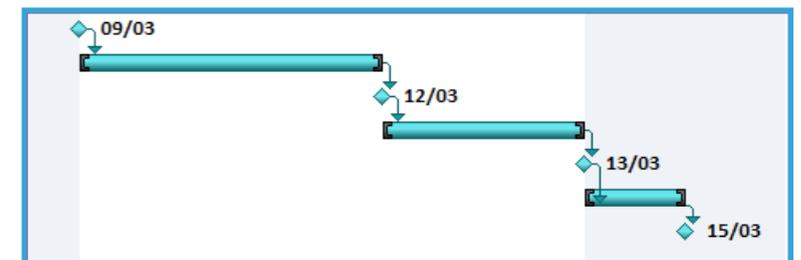
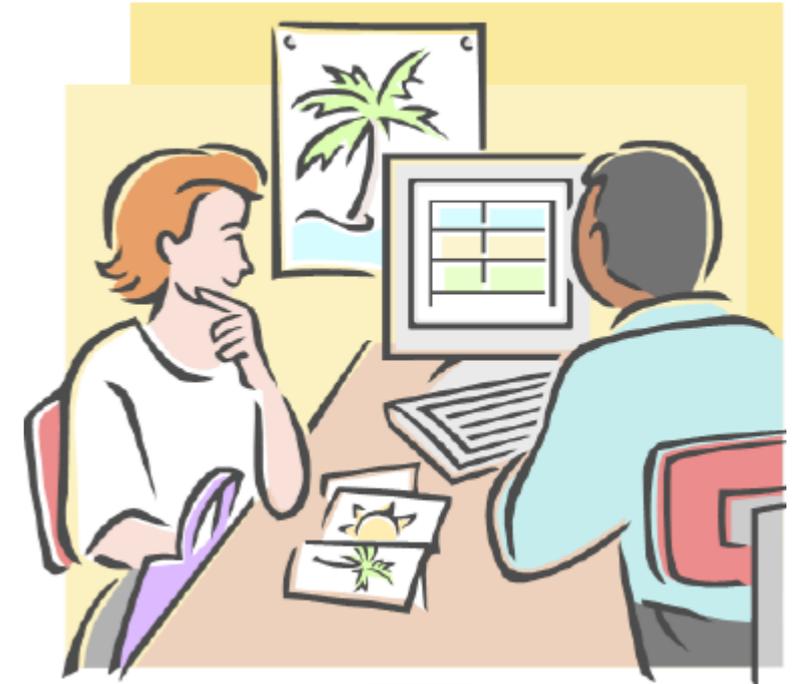
VIAJERO PREDICTIVO



VIAJERO ADAPTATIVO

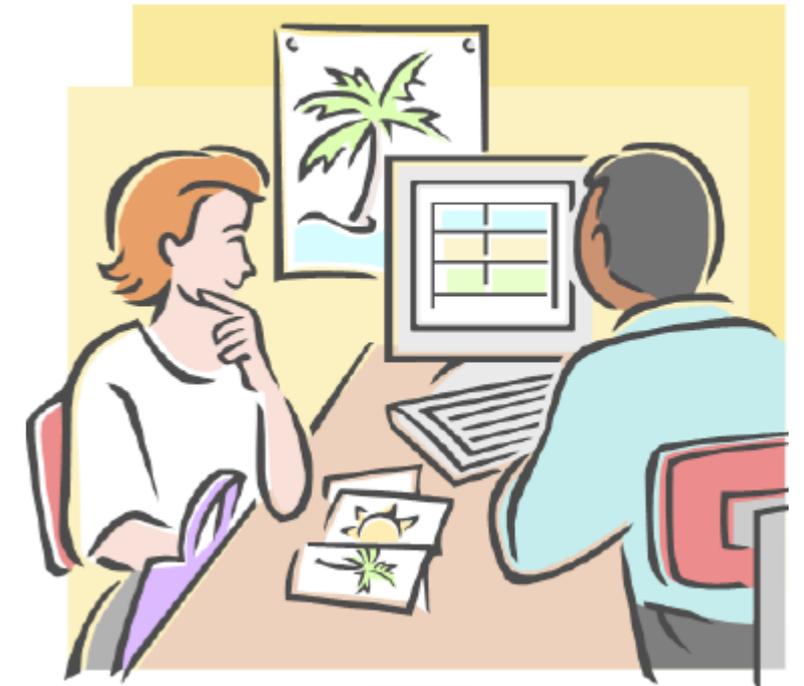
EL VIAJERO PREDICTIVO

- ▶ Billeto avión Vueling - 110€
- ▶ 3 noches hotel Ibis Londres - 120€
- ▶ Billeto Ryanair Londres -> Dublín - 50€
- ▶ 2 noches Hilton Dublín - 100€
- ▶ Billeto avión Iberia Dublín - Madrid - 85€
- ▶ 1 noche NH Madrid - 65€
- ▶ Billeto AVE Madrid - Barcelona - 60€



PRESUPUESTO:
590 €

EL VIAJERO PREDICTIVO



PLANIFICACIÓN



Ya he reservado mi viaje, y esta vez con 3 meses de antelación ya lo tengo todo.

¡¡¡Deseo mis vacaciones!!!!

EL VIAJERO PREDICTIVO



REALIDAD



He perdido el vuelo... a Londres; 1 noche menos para verlo. Llueve todos los días, me quedo en el hotel todo el día. Hay mucha tempestad quizás se anula el vuelo a Dublín también... perderé la noche allí.

!!!No contraté ningún seguro!!!!

EL VIAJERO ADAPTATIVO

- ▶ Billete avión Vueling - 110€
- ▶ 1 noche hotel Ibis Londres - 40€
- ▶ (pendiente de concretar)



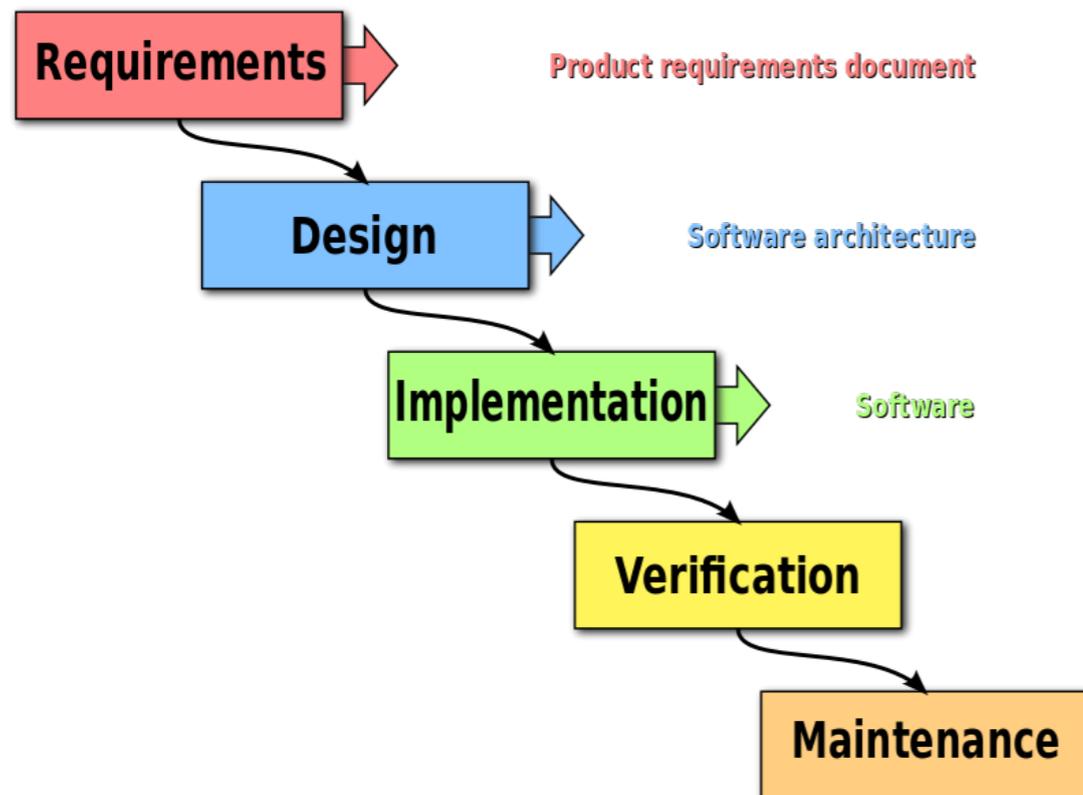
Si me gusta y hace buen tiempo me quedaré más noches, si no me gusta me iré antes; los vuelos entre semana a Dublín no son caros.

En Dublín me he anotado 3 opciones de hoteles, he llamado y parece que tienen disponibilidad para las fechas.

...

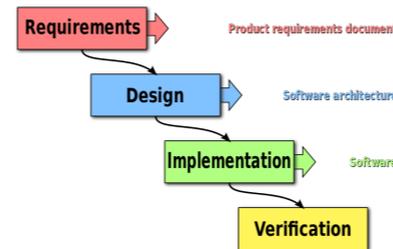
PRESUPUESTO LÍMITE: 590 €

MODELOS EN CASCADA



- ▶ Secuencial; hasta que no termina una fase no empieza la siguiente.
- ▶ No empieza el diseño hasta tener todos los requisitos acabados...
- ▶ Muy formal, cierre de etapas y no volvemos al punto anterior.

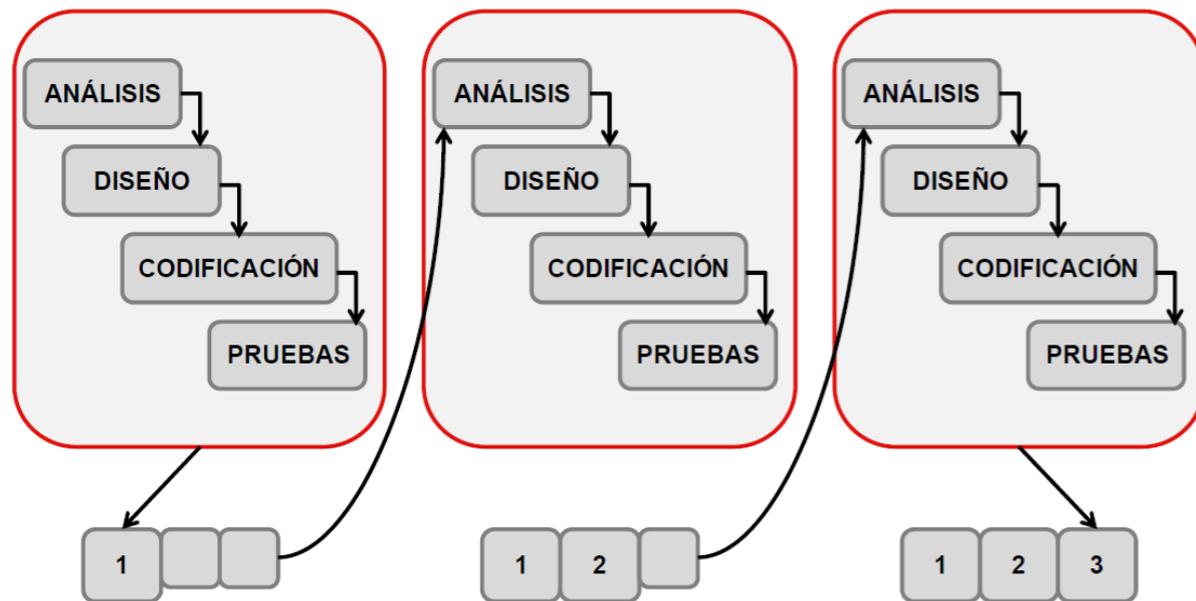
MODELOS EN CASCADA



- ▶ Mucho esfuerzo en etapas iniciales; un cambio y detección de un error en el inicio es menos costoso que al final.
- ▶ Mucha documentación, formalidad... es bueno en caso de disputas, abandono de algún miembro del equipo.... no volvemos al punto anterior.

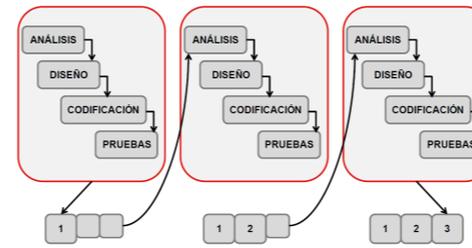
- ▶ IRREAL considerar que una vez terminado una fase ya no habrá cambios en fases pasadas.
- ▶ Existe algún proyecto donde no aparezcan requisitos nuevos?
- ▶ No tenemos un producto para testar hasta el final; si necesitamos cambios o detectamos errores de diseño serán muy costosos.
- ▶ El cliente no ve el producto y no da feedback hasta las últimas fases.

MODELOS ITERATIVOS



- ▶ Múltiples iteraciones.
- ▶ Cada iteración pasa por las distintas fases.
- ▶ Cada iteración nos permite redefinir y ajustar cosas de las distintas fases; aunque el peso dedicado a cada fase es distinto según la iteración.
- ▶ Discusión o presentación al fin de cada iteración.

MODELOS ITERATIVOS



- ▶ Se adapta mejor a los cambios y nos permite correcciones y mejora.
- ▶ Igual que en cascada sigue habiendo mucha documentación y formalidad.

- ▶ “Pesado” en muchas ocasiones y demasiado formal.
- ▶ Mucha documentación; difícil de gestionar y que ocupa mucho esfuerzo.
- ▶ Se pactan el número y duración de las iteraciones antes de empezar el proyecto.

MODELOS ÁGILES

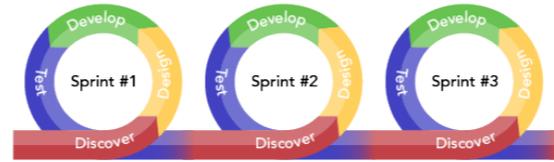


Agile Method

©2013 think interactive inc.

- ▶ Totalmente iterativo e incremental.
- ▶ El producto va creciendo en cada iteración hasta llegar al producto final.
- ▶ Se adapta a todo tipo de cambios.
- ▶ El cliente participa activamente a lo largo del proyecto.

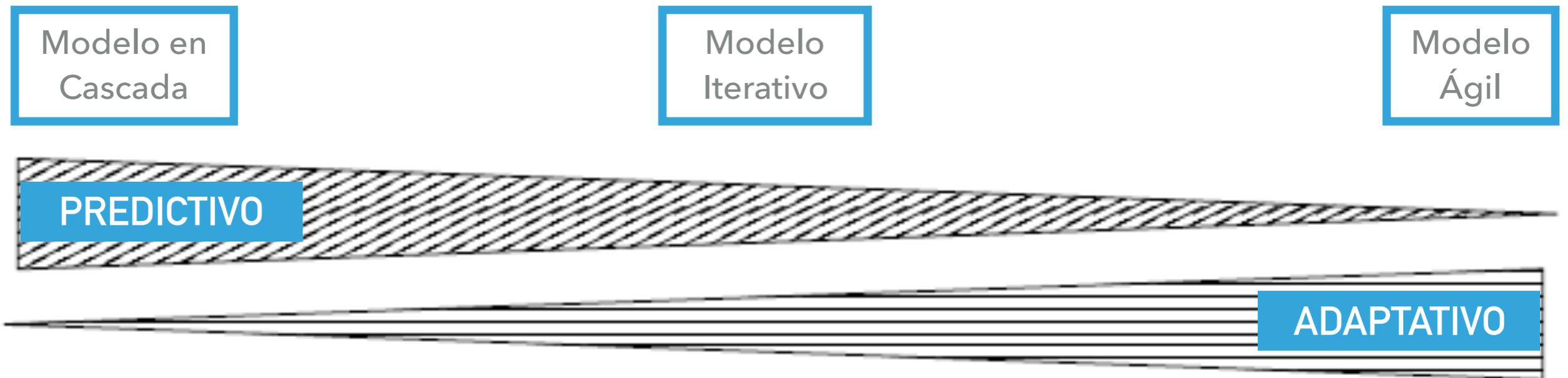
MODELOS ÁGILES



- ▶ Totalmente adaptable a los cambios y nuevos requisitos.
- ▶ El cliente ve el producto y nos da feedback desde los inicios.
- ▶ Testeo desde el momento 0.
- ▶ Auto-gestión, cómodo y transparente.

- ▶ El cliente siempre quiere más.
- ▶ En ocasiones el poco formalismo puede ser un problema.
- ▶ Más difícil de vender y pactar a nivel comercial.
- ▶ Difícil y no aconsejable para grandes proyectos con muchos implicados.

PREDICCIÓN VS ADAPTACIÓN



¿QUÉ MODELO UTILIZAR?

- ▶ Existen múltiples modelos de desarrollo y ciclos de vida de proyectos.
- ▶ **No hay uno mejor que otro.**
- ▶ **Sí hay uno que se adapta mejor que otro para un proyecto/producto concreto.**
- ▶ Múltiples factores a tener en cuenta:
 - ▶ Producto, cliente, tiempo, presupuesto, requisitos...

¿QUÉ MODELO UTILIZAR?

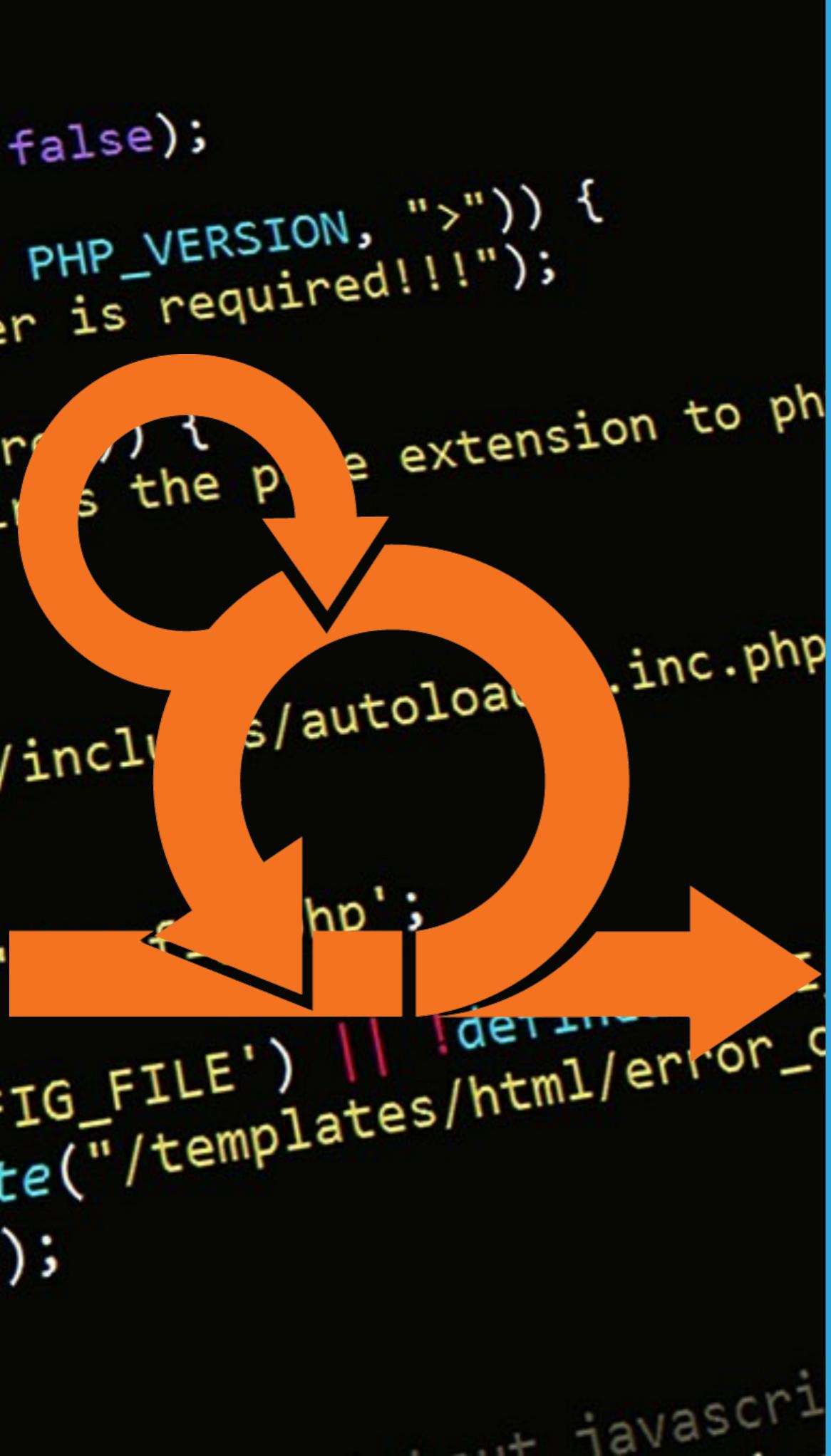
	Cascada	Iterativo	Ágil
Conozco todos los requisitos	SI	SI	SI
Requisitos inciertos	NO	MEDIO	SI
Cientos implicados y equipos grandes	SI	MEDIO	NO
Riesgo para el cliente	BAJO	BAJO	BAJO (REALISTA)
Riesgo para empresa realiza proyecto	ALTO	MEDIO	BAJO

¿QUÉ MODELO UTILIZAR?



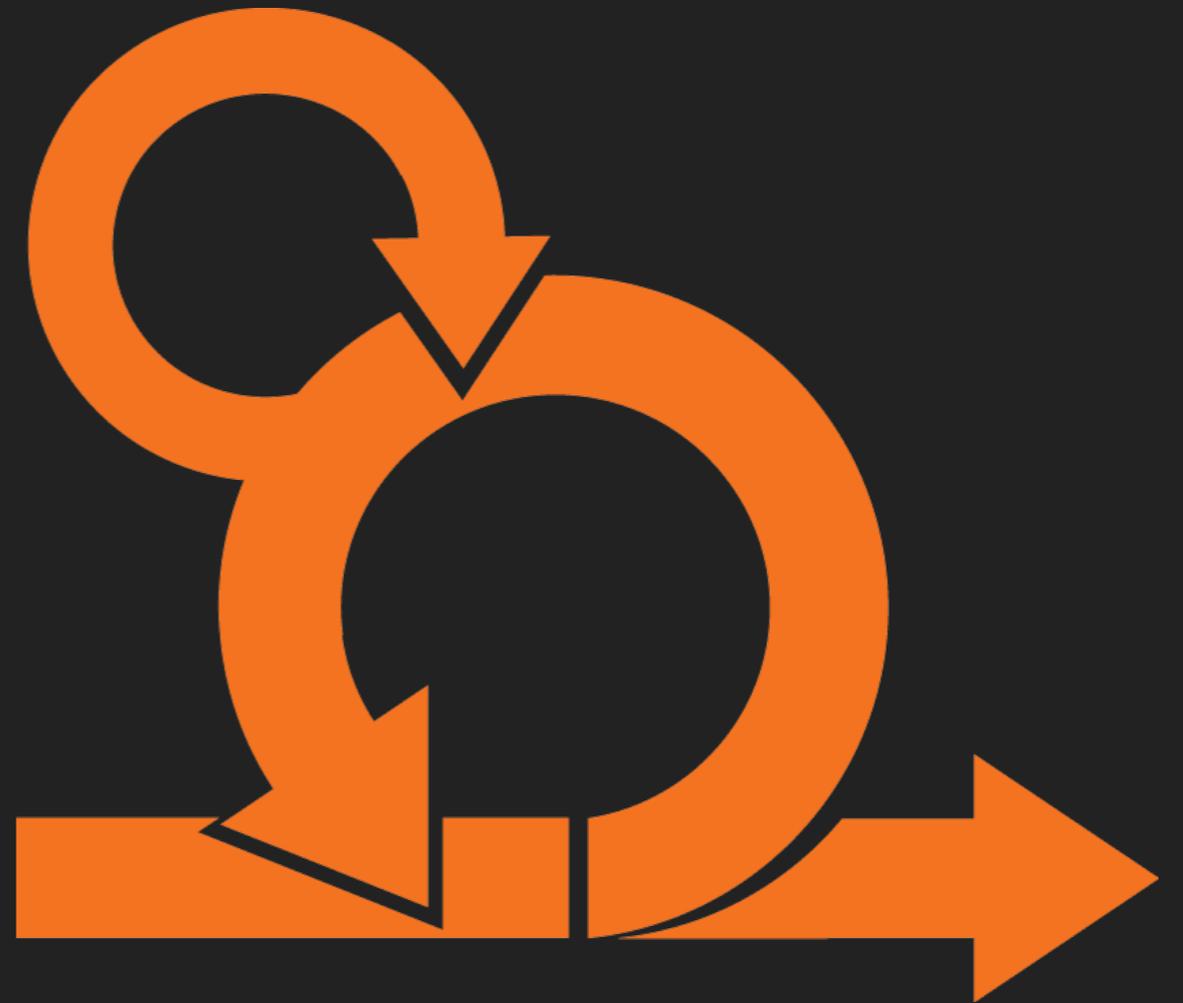


AGILE



AGILE

- ▶ Nuevas necesidades
- ▶ Manifiesto ágil
- ▶ Características
- ▶ Lean y TPS
- ▶ eXtreme Programming



NECESIDADES

Agile

NUEVAS NECESIDADES

- ▶ La forma clásica y predictiva de gestionar proyectos no es válida y tiene muchas dificultades en el entorno actual.
- ▶ Necesidad de nuevos productos y rapidez para sacarlos al mercado.
- ▶ Mayor necesidad de adaptarse al cliente; fuerte competencia.

UN POCO DE HISTÓRIA...

- ▶ A finales de los 80 Hirotaka Takeuchi y Ikujiro Nonaka publican en la revista Harvard Business Review el artículo, "**The New New Product Development Game**"; habla de:
 - ▶ Equipos auto-gestionados
 - ▶ Mayor flexibilidad
 - ▶ Reducir costes y tareas que no aporten valor
 - ▶ Critica la pesadez de la gestión actual de proyectos
 - ▶ Ponían ejemplos de **Toyota** y **Canon**.
- ▶ En los 90 Ken Schwaber y Jeff Sutherland hablan de una metodología de desarrollo de proyectos de software.
- ▶ En 2001 se escribe "**The agile manifesto**"

PRINCIPALES PROBLEMAS



PRINCIPALES PROBLEMAS



VELOCIDAD



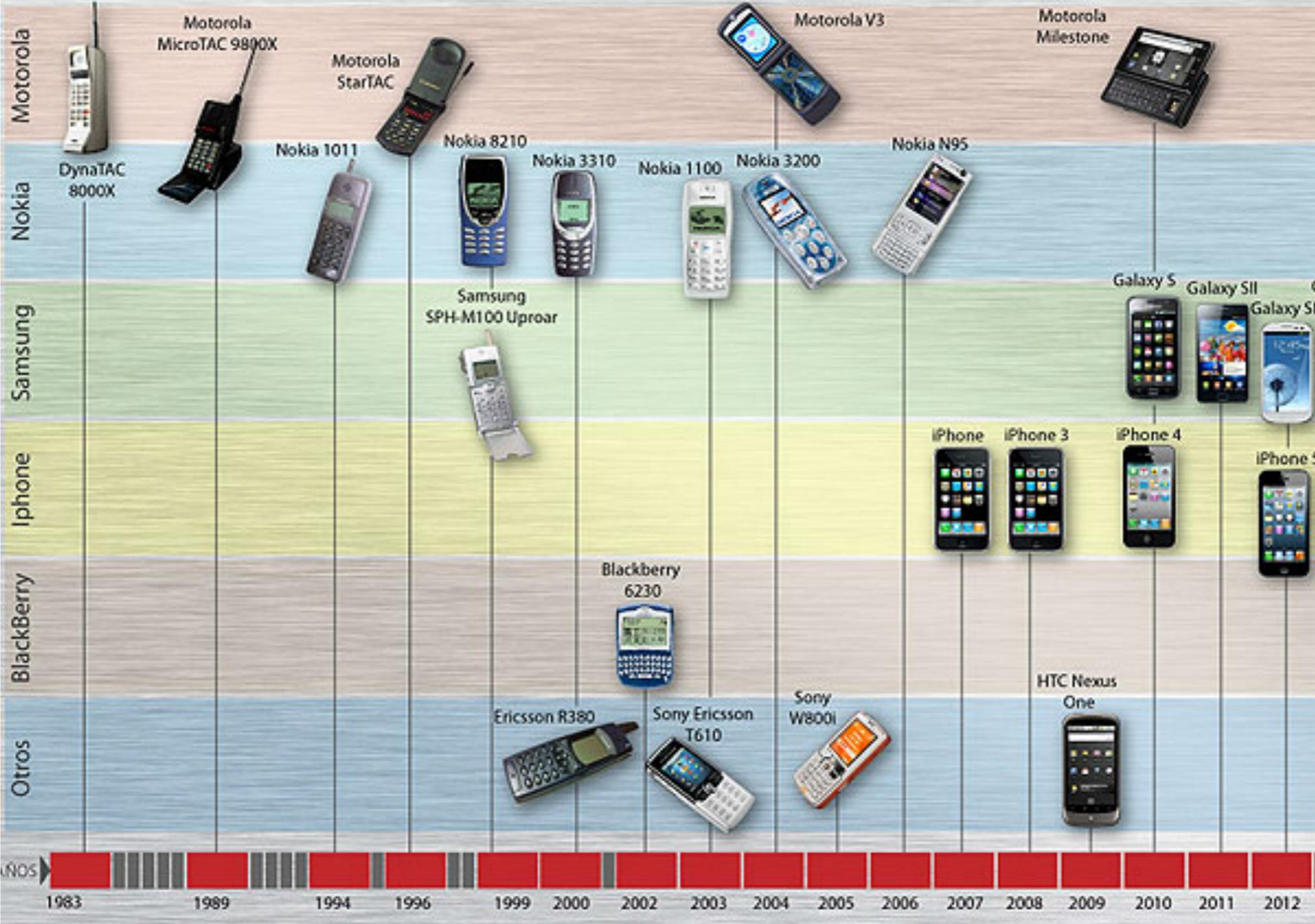
INCERTIDUMBRE



VELOCIDAD

- ▶ Ciclos de vida de los productos muy cortos.
- ▶ Actualizaciones constantes; nuevas tecnologías, lenguajes, navegadores...
- ▶ Mucha competencia.

VELOCIDAD...



VELOCIDAD

NOKIA Connecting People Know our past. Create the future...

1982 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006

1982: Mobile Starter "The first mobile phone ever" (Nokia 2110)

1984: "The first mobile phone with a color display" (Nokia 2110)

1985: "The first mobile phone with a color display" (Nokia 2110)

1986: "The first mobile phone with a color display" (Nokia 2110)

1987: "The first mobile phone with a color display" (Nokia 2110)

1988: "The first mobile phone with a color display" (Nokia 2110)

1989: "The first mobile phone with a color display" (Nokia 2110)

1990: "The first mobile phone with a color display" (Nokia 2110)

1991: "The first mobile phone with a color display" (Nokia 2110)

1992: "The first mobile phone with a color display" (Nokia 2110)

1993: "The first mobile phone with a color display" (Nokia 2110)

1994: "The first mobile phone with a color display" (Nokia 2110)

1995: "The first mobile phone with a color display" (Nokia 2110)

1996: "The first mobile phone with a color display" (Nokia 2110)

1997: "The first mobile phone with a color display" (Nokia 2110)

1998: "The first mobile phone with a color display" (Nokia 2110)

1999: "The first mobile phone with a color display" (Nokia 2110)

2000: "The first mobile phone with a color display" (Nokia 2110)

2001: "The first mobile phone with a color display" (Nokia 2110)

2002: "The first mobile phone with a color display" (Nokia 2110)

2003: "The first mobile phone with a color display" (Nokia 2110)

2004: "The first mobile phone with a color display" (Nokia 2110)

2005: "The first mobile phone with a color display" (Nokia 2110)

2006: "The first mobile phone with a color display" (Nokia 2110)

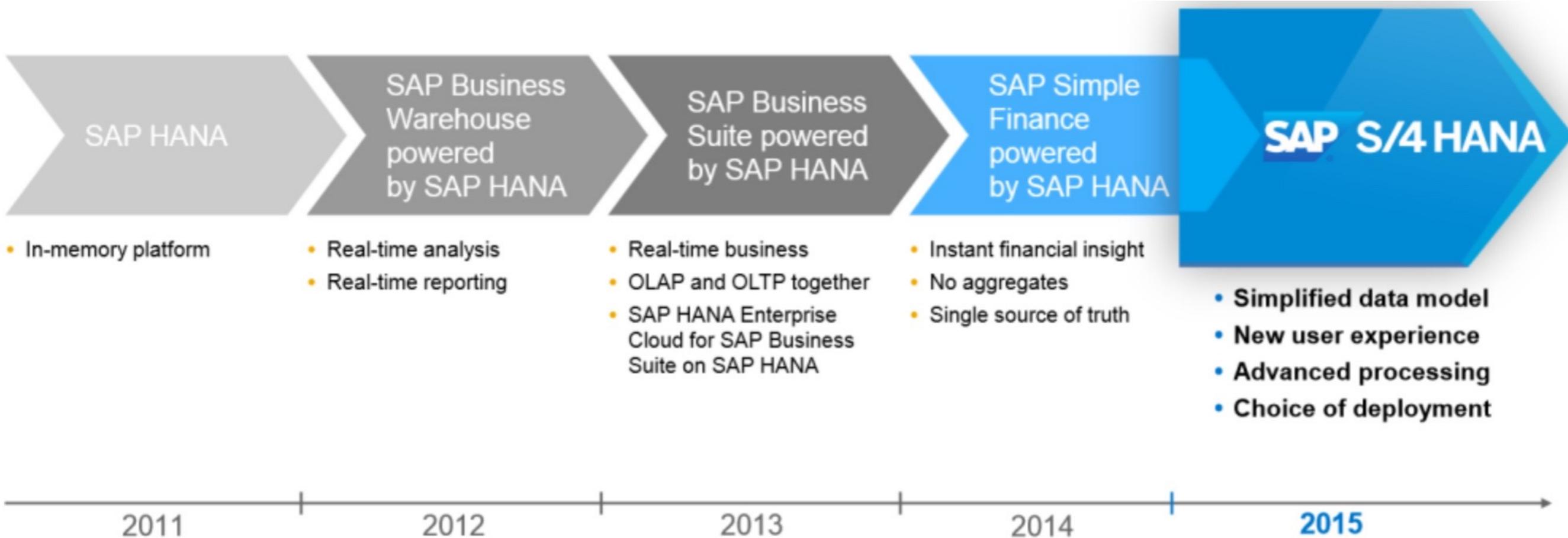
UI: 1994-2006

Logos: Google, Windows, etc.

VELOCIDAD



VELOCIDAD

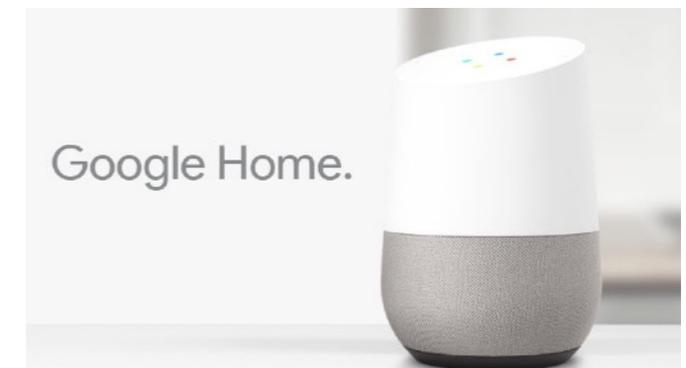
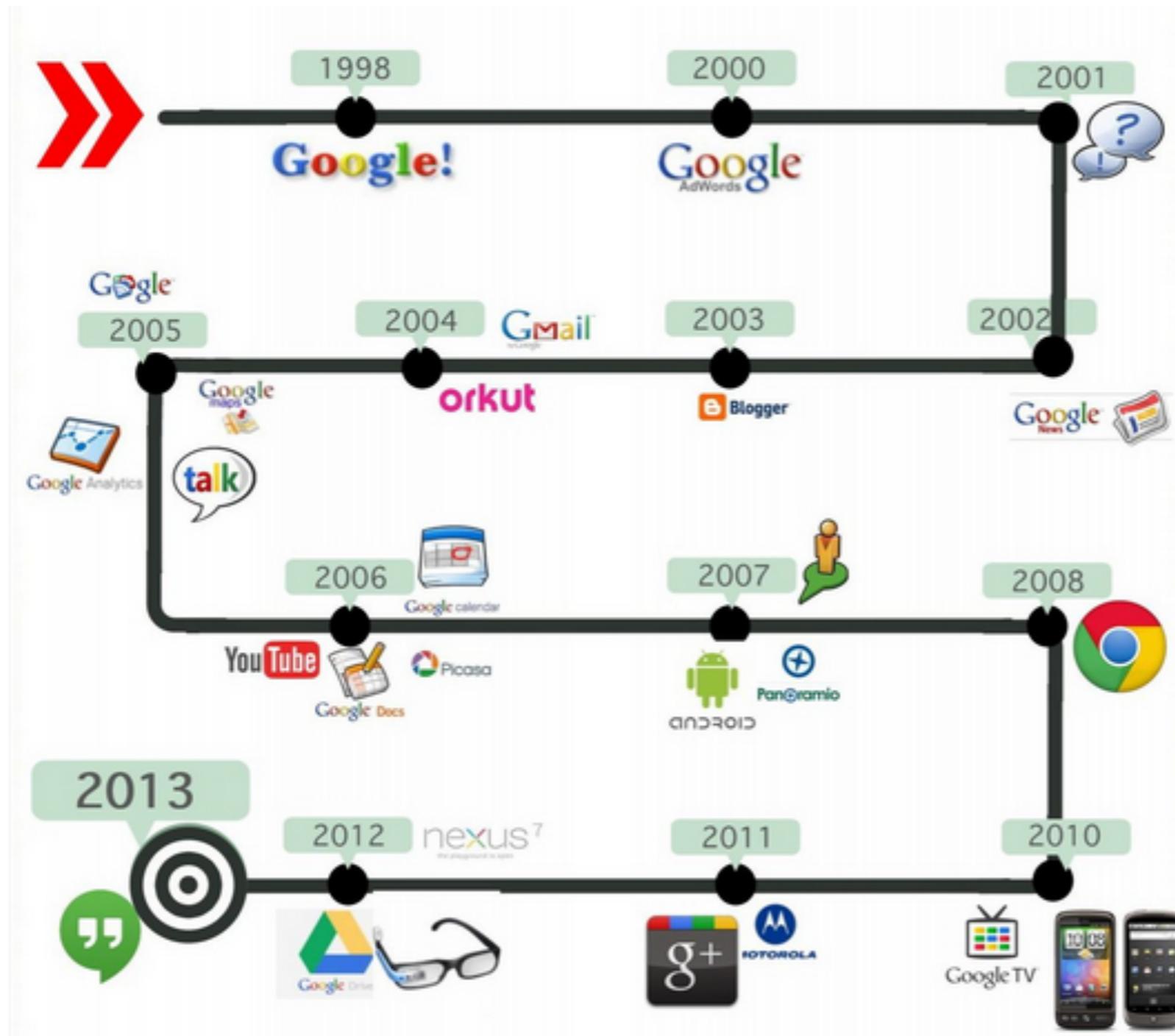


- ▶ Muchos productos.
- ▶ Mercado cambiante.
- ▶ Cliente sobre-presionado.
- ▶ Nuevos productos de la competencia, adaptarte o morir.



INCERTIDUMBRE

INCERTIDUMBRE



MANIFESTO ÁGIL

Agile

AGILE MANIFESTO - 4 VALORES

VS

PERSONAS E INTERACCIÓN

PROCESOS Y HERRAMIENTAS

SOFTWARE FUNCIONANDO

DOCUMENTACIÓN EXHAUSTIVA

COLABORACIÓN DEL CLIENTE

NEGOCIACIÓN CONTRACTUAL

RESPUESTA AL CAMBIO

SEGUIR EL PLAN

AGILE MANIFESTO - 12 PRINCIPIOS

1. La principal prioridad es **satisfacer al cliente** a través de la entrega rápida y continua de software valor.
2. **Adaptación** y nuevos requisitos son bienvenidos en cualquier momento del proyecto.
3. **Entrega con frecuencia** de software funcionando, semanas en vez de meses.
4. Los desarrolladores y las personas de negocio deben **trabajar conjuntamente** diariamente.
5. Realizar proyectos con **individuos motivados**, dándoles la oportunidad, confianza y el respaldo necesario.
6. La comunicación **cara a cara** es la mas eficiente y que aporta mayor valor.

AGILE MANIFESTO - 12 PRINCIPIOS

7. El **software funcionando** es la mejor prueba y medida del progreso del proyecto.
8. **Trabajo conjunto y constante**; los patrocinadores, desarrolladores y usuarios mantienen un ritmo constante de forma indefinida.
9. Atención permanente a la **excelencia técnica** y el diseño.
10. **Simplicidad** para maximizar la cantidad de trabajo.
11. Equipos **auto-organizados**; mejores diseño, requisitos y arquitectura.
12. Reflexión continua, ajustar y **mejorar**. El equipo realiza "retrospective".

CARACTERÍSTICAS

Agile

IMPORTANCIA DE LAS PERSONAS

equipos:

AUTO-ORGANIZADOS

MOTIVADOS

INNOVACIÓN

MULTIDISCIPLINARES



New York



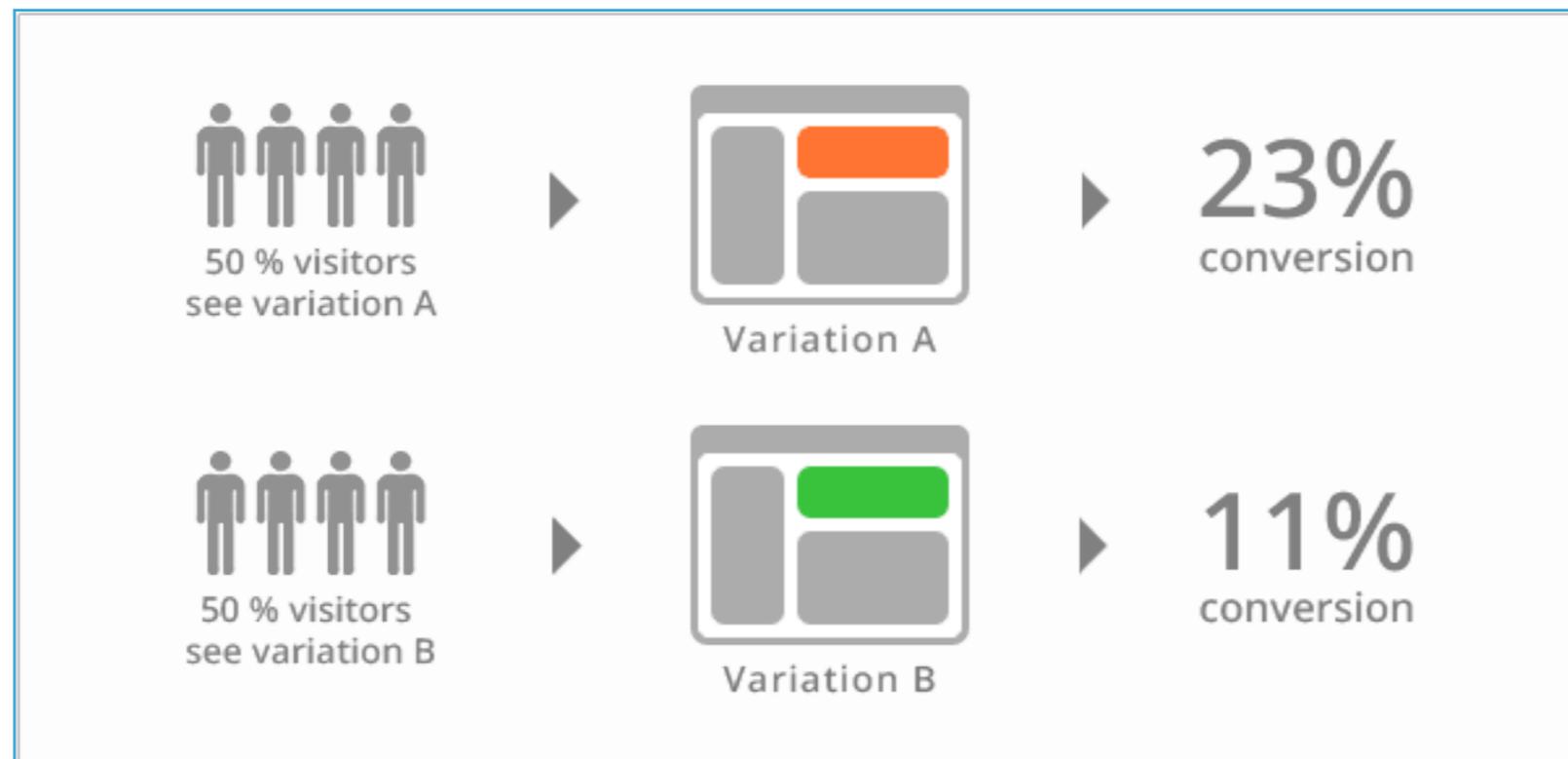
Paris



London

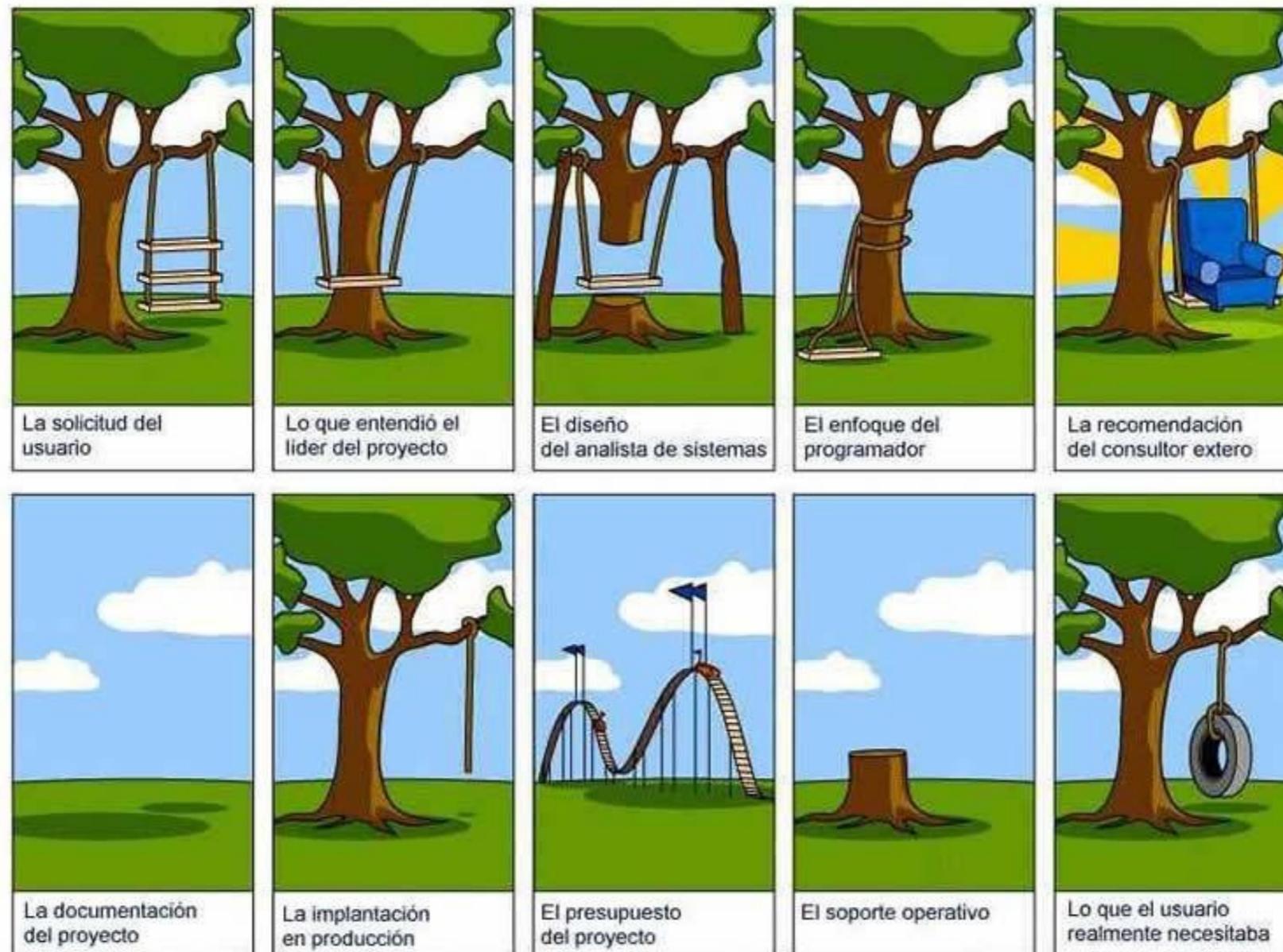
RAPIDEZ Y TEST DE PRODUCTOS

- ▶ Elegir mejor opción.
- ▶ Detectar problemas y errores rápidamente.
- ▶ Maximizar el ROI y el valor del producto.



INTRODUCIR AL CLIENTE EN EL PROYECTO

para evitar esto...

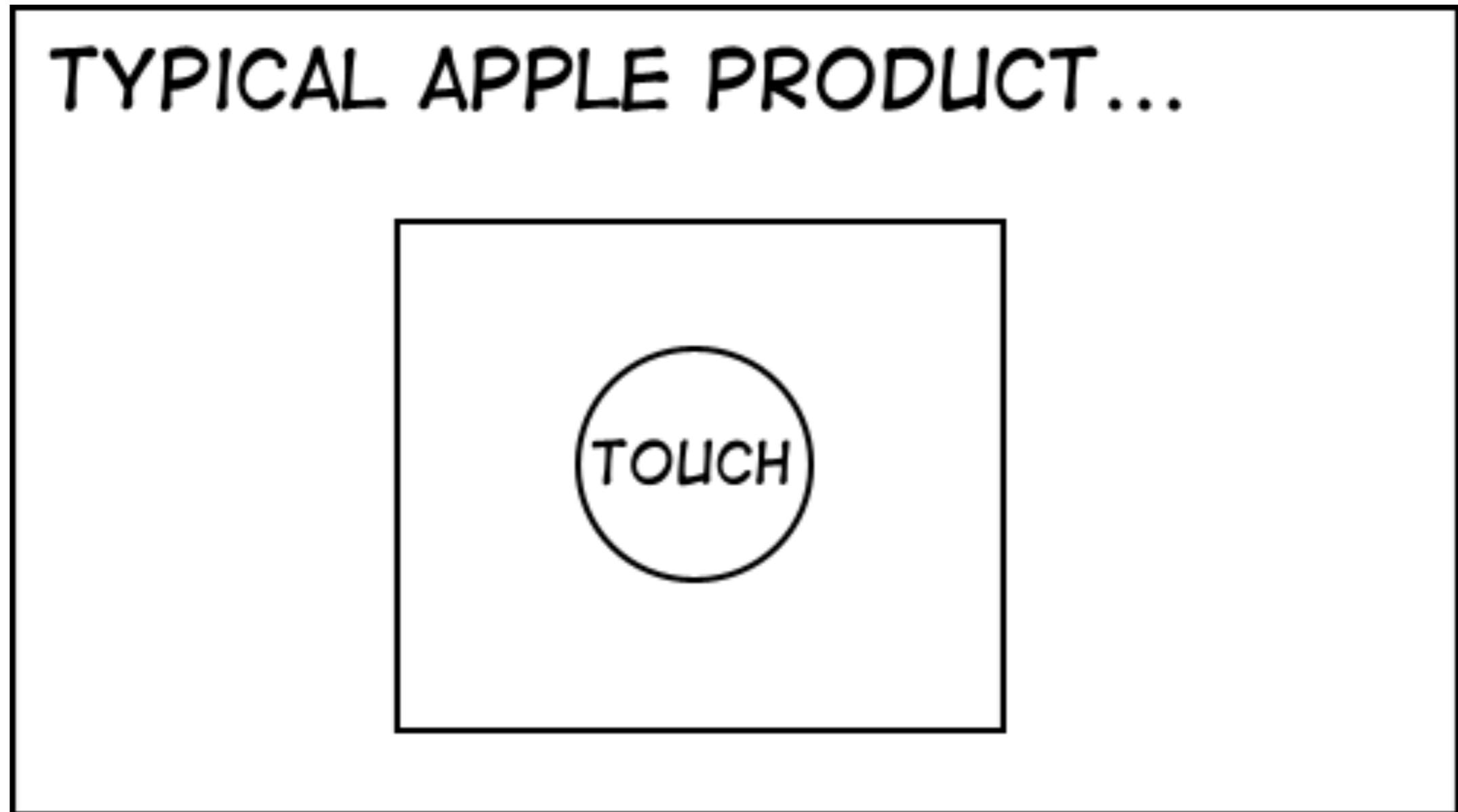


ANÁLISIS DATOS

- ▶ Pruebas y datos que contrastar.
- ▶ Datos y feedback desde el momento 0; por el cliente o herramientas.
- ▶ Incorporar mejoras y correcciones.

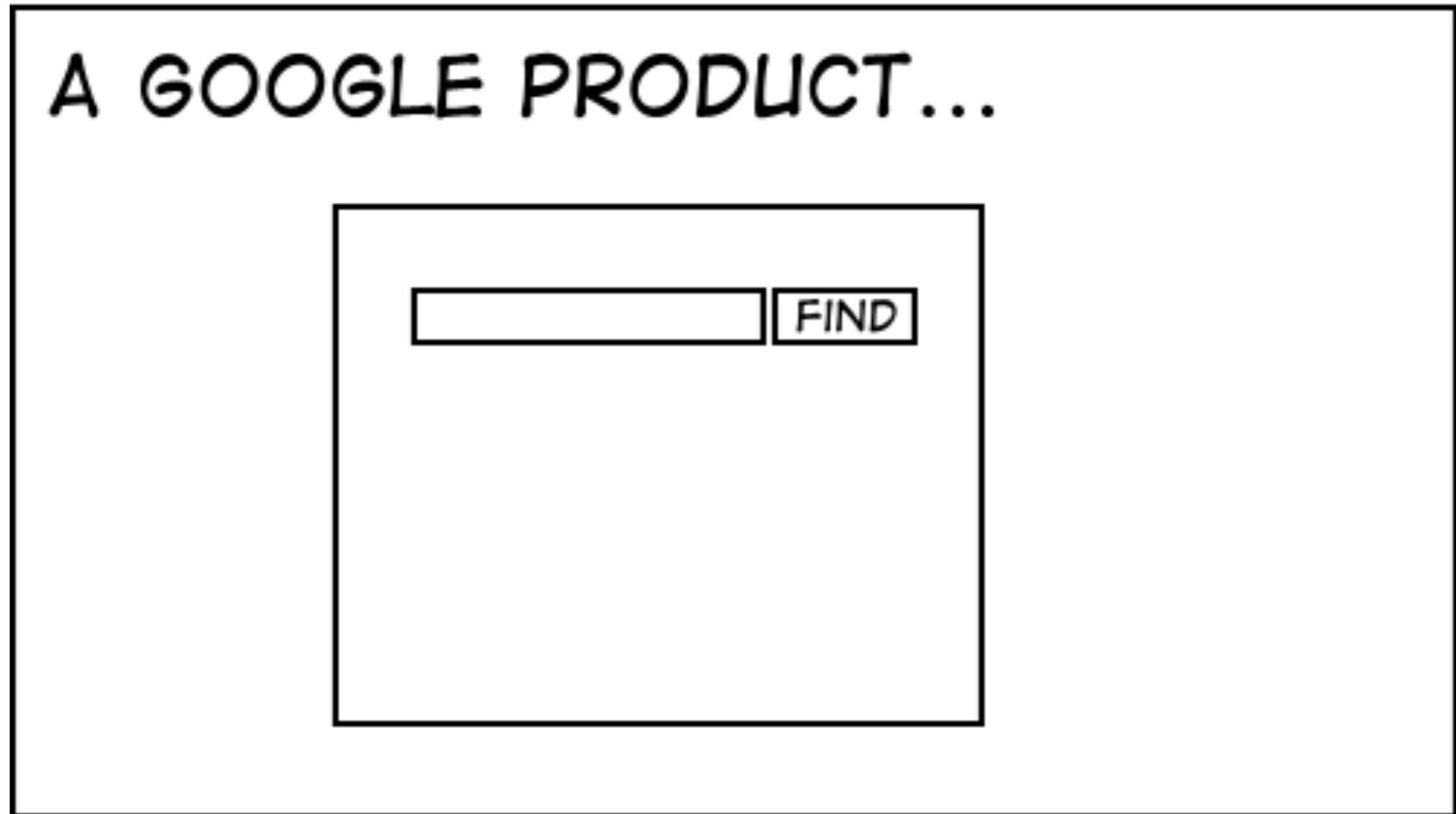


SIMPLICIDAD



- + simple
- + fácil
- manuales
- documentación

SIMPLICIDAD



¿¿¿ SIMPLICIDAD ???

YOUR COMPANY'S APP...

FIRST NAME:	<input type="text"/>	TYPE CD:	<input type="text"/>	4 - K AA2- DK9B KKA? CN3 AA-9
LAST NAME:	<input type="text"/>	TQP STAT:	<input type="checkbox"/> <input type="checkbox"/>	
SSN:	<input type="text"/>	VER:	<input type="text"/>	
ID:	<input type="text"/>	FT/PT:	<input checked="" type="checkbox"/>	
PHONE 1:	<input type="text"/>	CAT CD:	<input type="text"/>	
PHONE 2:	<input type="text"/>	CITY:	<input type="text"/>	
ADDR 1:	<input type="text"/>	STATE:	<input type="text"/>	NEW
ACCT #:	<input type="text"/>	ZIP:	<input type="text"/>	DEL
		ORD #:	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> ? <input type="checkbox"/>	

OKAY APPLY SAVE UNDO HELP DELETE EDIT

SELECT BROWSE ERRORS

STUFFTHATHAPPENS.COM BY ERIC BURKE

Gran necesidad de manuales y documentación!

LEAN Y TPS

Agile

LEAN

- ▶ ***Lean Manufacturing***; algunas traducciones:
 - ▶ Producción ajustado
 - ▶ Producción sin desperdicios
 - ▶ Producción limpia
- ▶ Modelo enfocado al flujo de trabajo.
- ▶ Utilizar los mínimos recursos necesarios; pero todos los necesarios.
- ▶ Maximizar el valor del producto y la correcta entrega al cliente.

TOYOTA PRODUCTION SYSTEM



- ▶ El ***Lean Manufacturing***, es la extensión y generalización de TPS; sistema de producción integral de Toyota.
- ▶ Con el objetivo de mejorar la producción japonesa.
- ▶ Sigue los principios de:
 - ▶ Henry Ford - *Just-in-time*
 - ▶ Frederick Taylor (valor de los trabajadores; herramientas y métodos eficaces).

TOYOTA PRODUCTION SYSTEM



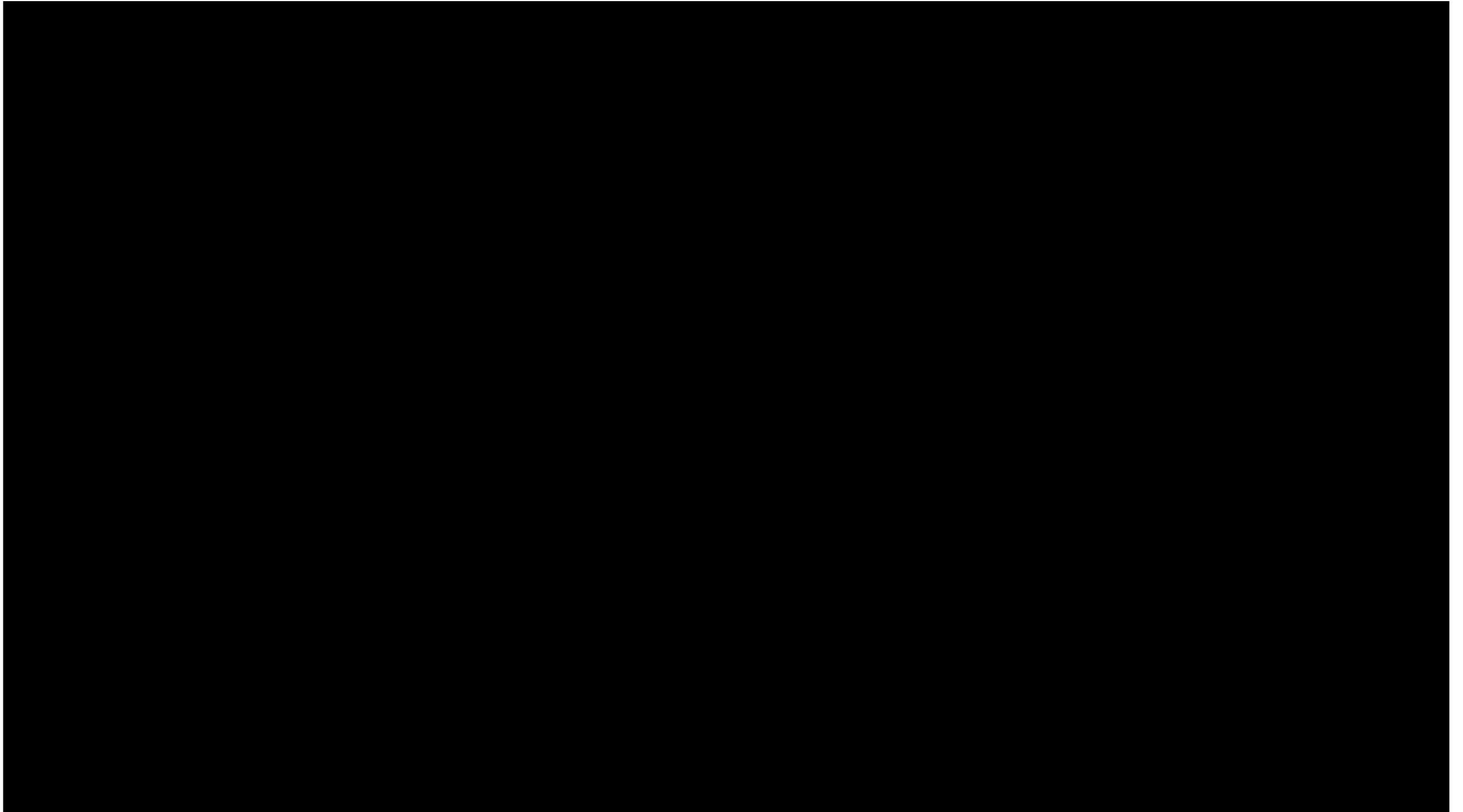
- ▶ Eliminar el despilfarro; "muda" cualquier recurso que no genere valor.
- ▶ Ampliar el aprendizaje cognitivo.
- ▶ Decidir como más tarde mejor.
- ▶ Reaccionar rápido.
- ▶ Entregar rápido.
- ▶ Potenciar el equipo.
- ▶ Ver el conjunto del producto; "piensa en grande, actúa en pequeño".

TOYOTA PRODUCTION SYSTEM



<https://www.youtube.com/watch?v=kce2L23yLcw>

TOYOTA PRODUCTION SYSTEM



Lean Manufacturing

Lean Software

Defectos

Bugs

Exceso de producción

Exceso de funcionalidades

Transporte

Cambios entre tareas

Esperas

Restricciones entre tareas

Inventarios

Tareas a medio hacer, no entregadas pero muy avanzadas

Movimientos

Cambios entre tareas, poca optimización

Procesos innecesarios

Metodologías pesadas, exceso documentación

VAMOS A JUGAR...

Row by row - run

 <u>A - J</u>	 <u>I - X</u>	 <u>1 - 10</u>	"Clock":
A	I	1	01:30
B	II	2	
C	III	3	
D	IV	4	
E	V	5	
F	VI	6	
G	VII	7	
H	VIII	8	
I	IX	9	
J	X	10	
		<u>Total:</u>	

VAMOS A JUGAR...

Row by row - run

	<u>A - J</u>		<u>I - X</u>		<u>1 - 10</u>	"Clock": 01:30
	A		I		1	
	B		II		2	
	C		III		3	
	D		IV		4	
	E		V		5	
	F		VI		6	
	G		VII		7	
	H		VIII		8	
	I		IX		9	
	J		X		10	
<u>Time:</u>	01:24		01:27		01:30	<u>Total:</u> 01:30

EXTREME PROGRAMMING

Agile

EXTREME PROGRAMMING

- ▶ Pair programming
- ▶ Pair review
- ▶ Integración continua
- ▶ Test driven development

```
L', false);  
2", PHP_VERSION, ">")) {  
eater is required!!!");  
"pcre")) {  
requires the pcre extension to php in  
T. '/includes/autoloader.inc.php';  
on  
OT. '/config.php';  
CONFIG_FILE') || !defined('PSI_DEB  
plate("/templates/html/error_conf  
ch());
```

```
out javascript
```

PAIR PROGRAMMING

- ▶ 2 programadores trabajando juntos sobre el mismo código.
 - ▶ Uno programa; el **driver**
 - ▶ El otro revisa el trabajo; **observer**
- ▶ Se elige la mejor solución conjuntamente.
- ▶ Se corrigen errores al instante.
- ▶ Comunicación continua entre los miembros.
- ▶ Los papeles de los programadores se van intercambiando.

PAIR REVIEW

- ▶ El código elaborado por un miembro del equipo es corregido por otro miembro.
- ▶ 4 ojos ven más que 2.
- ▶ Trabajas mejor y más cuidadoso si sabes que alguien va a supervisar tu código.
- ▶ Se optimiza muchísimo el código.

INTEGRACIÓN CONTÍNUA

- ▶ Al trabajar en equipos y ser entregas tan continuas, es muy importante la integración constante.
- ▶ Utilizar software de control de versiones, repositorios...
- ▶ Herramientas de automatización de tests; compilaciones automáticas...

TEST DRIVEN DEVELOPMENT

▶ **Escribir el test a pasar**

Escribir un test que compruebe alguna característica específica que nuestro programa deba cumplir.

▶ **Codificar**

Una vez escrito el test; crear un programa que pase el test.

▶ **Refactor**

Optimizar el código. Eliminar redundancias, simplificar y obtener un mejor diseño.



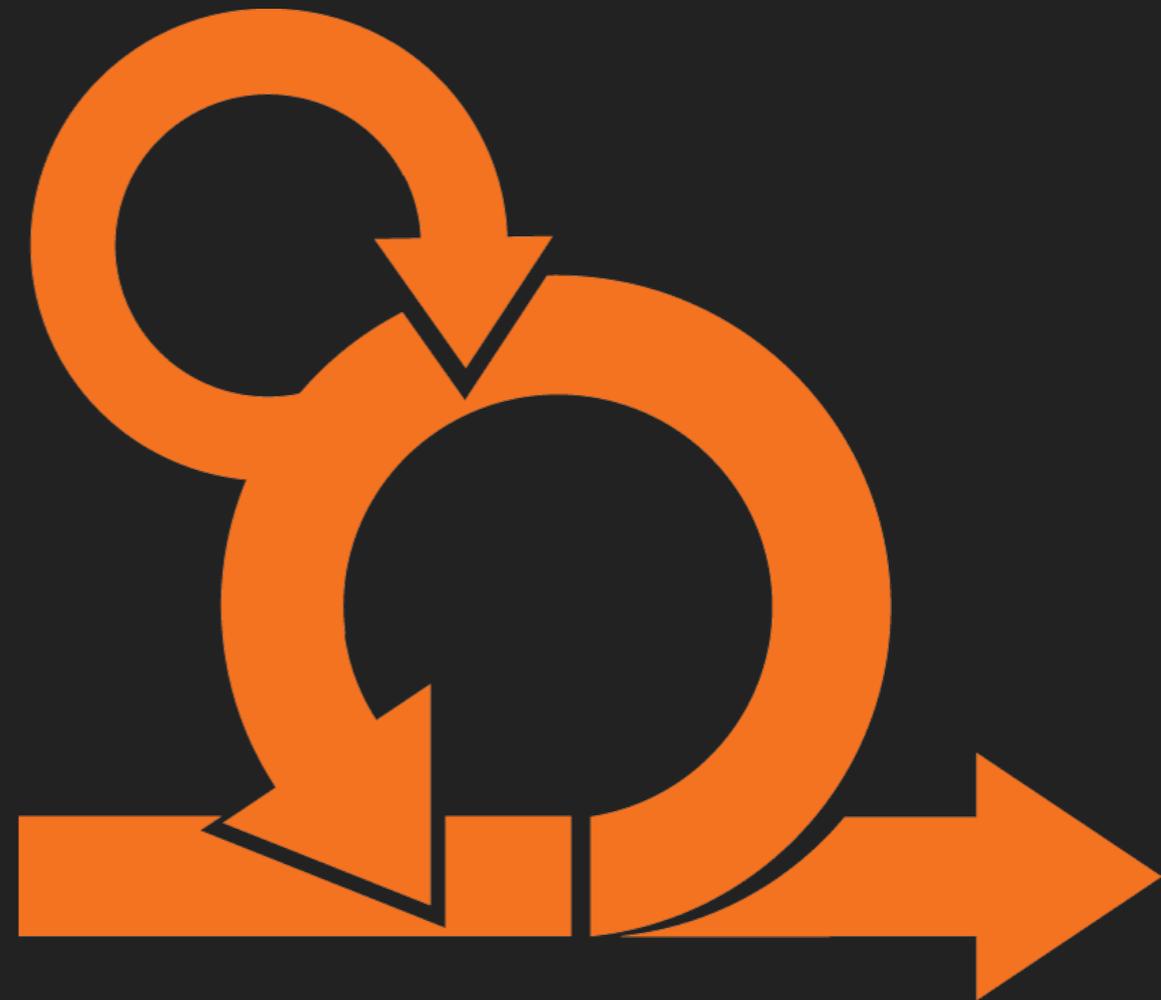
Universitat
de Lleida

SCRUM



SCRUM

- ▶ SCRUM
- ▶ Elementos
- ▶ Roles
- ▶ Reuniones
- ▶ Artifacts



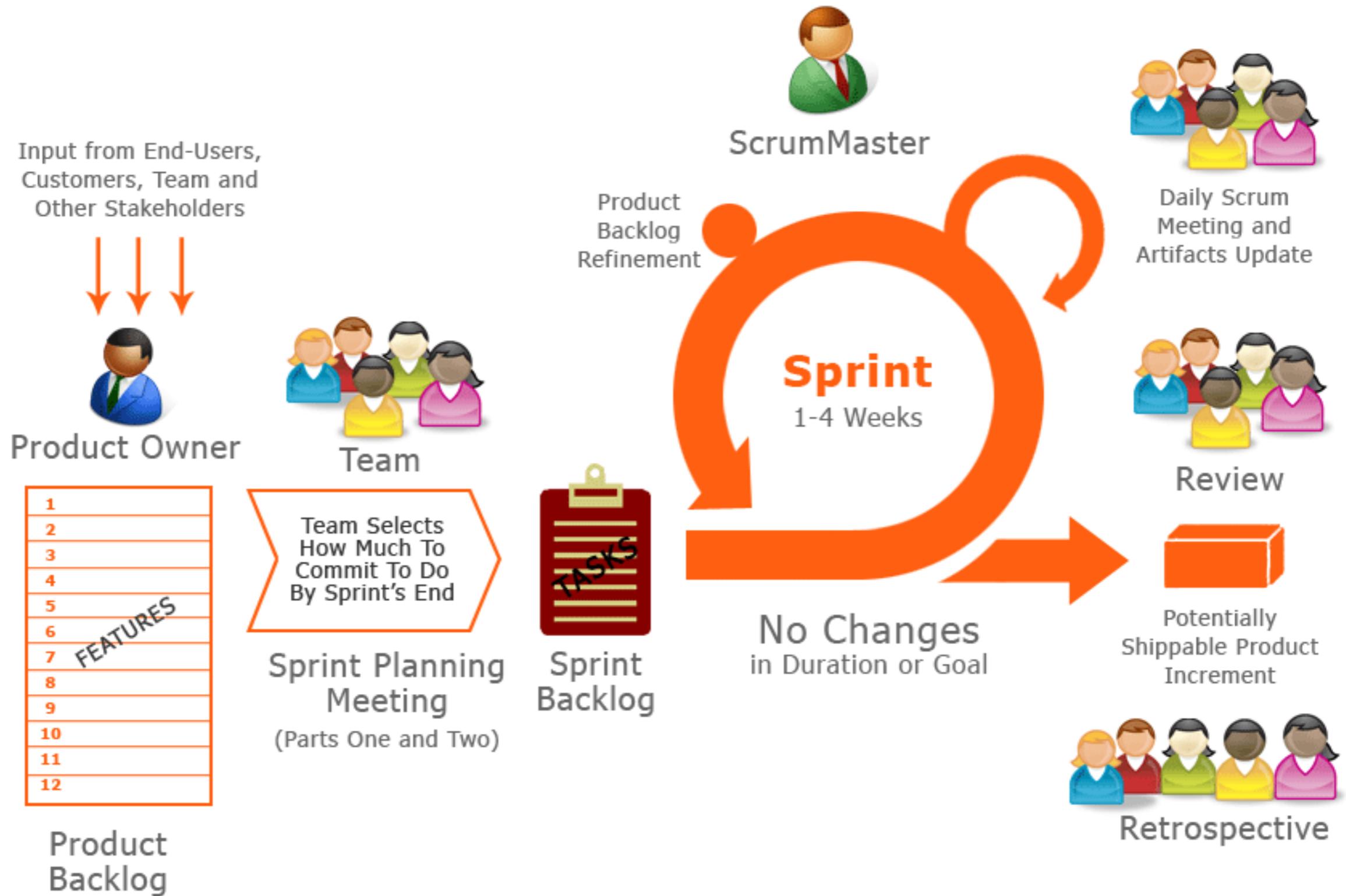
SCRUM

Metodología ágil para el desarrollo de productos (principalmente software) con el objetivo de maximizar el valor del producto y proyecto.

Define prácticas, roles y herramientas que ayudarán al proceso de desarrollo del producto.

Metodología / *Framework* / buenas prácticas

SCRUM

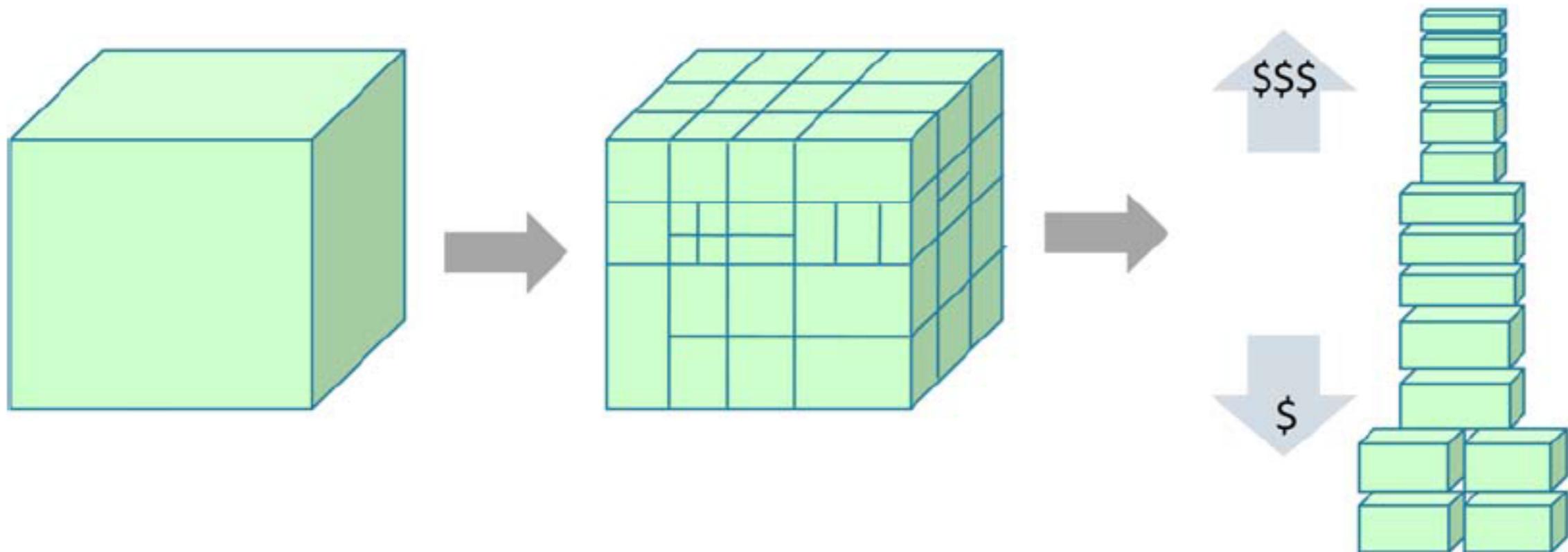


BENEFICIOS DE SCRUM

- ▶ Los beneficios que aportan los desarrollos ágiles:
 - ▶ Menor tiempo de desarrollo para obtener un producto testeable.
 - ▶ Podemos llegar a lanzar productos no finalizados.
 - ▶ Agilidad y flexibilidad; adaptación nuevos requisitos y cambios.

EVOLUTIVO E INCREMENTAL

- ▶ Dividimos el trabajo en partes más pequeñas. Centrándonos y poniendo esfuerzos en las partes de mayor valor.



ELEMENTOS

SCRUM

ELEMENTOS

PRODUCT BACKLOG

SPRINT BACKLOG

SPRINT

INCREMENTO

ELEMENTOS

PRODUCT BACKLOG

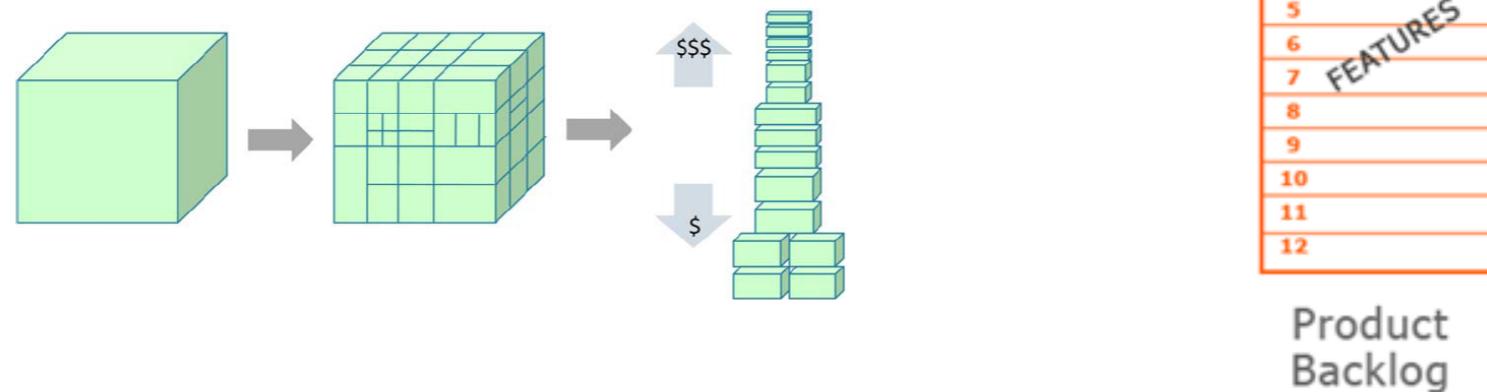
SPRINT BACKLOG

SPRINT

INCREMENTO

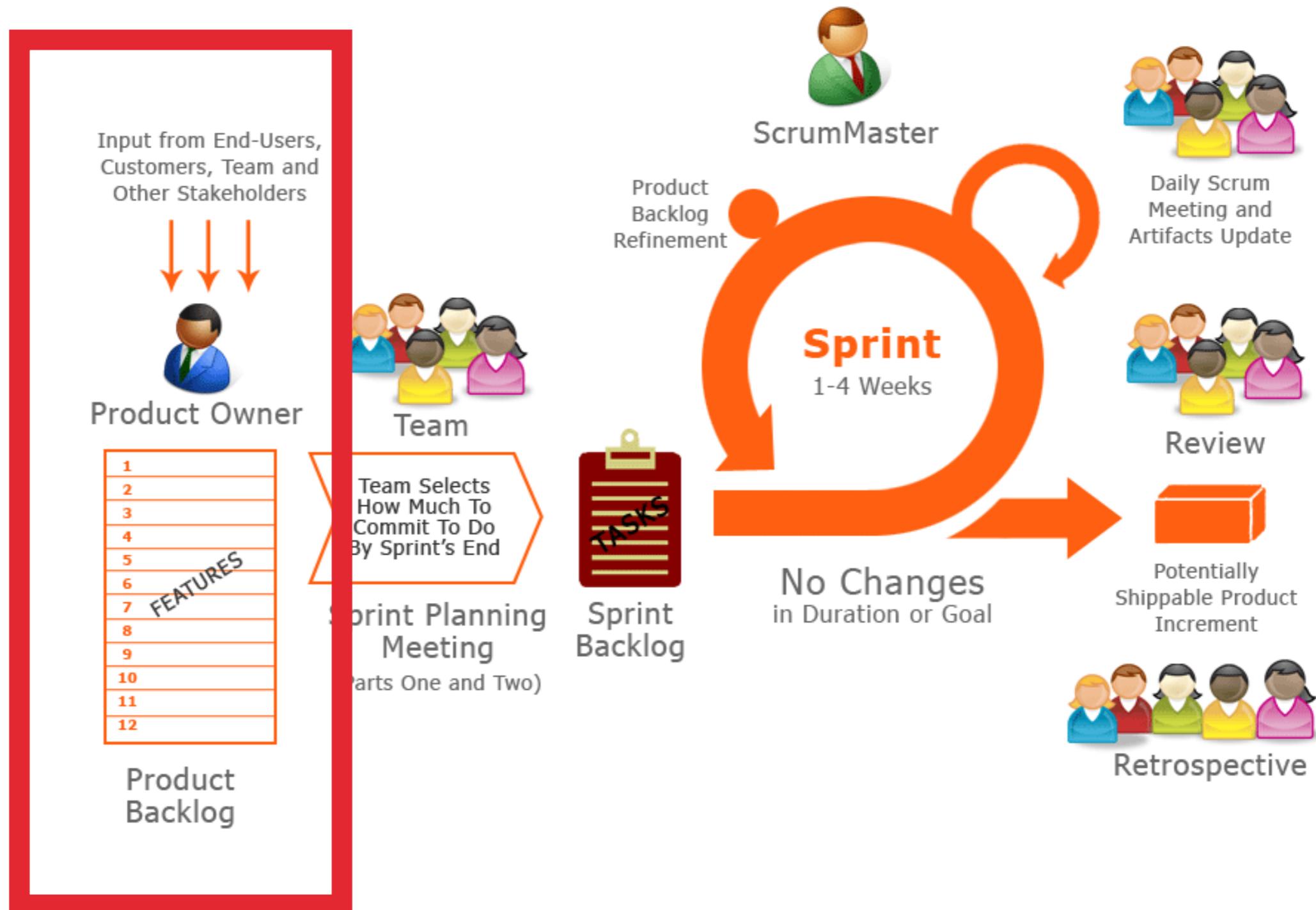
PRODUCT BACKLOG

Lista ordenada según prioridad sobre los requisitos que debe tener el producto a realizar.



- ▶ Se elabora conjuntamente con el cliente.
- ▶ El *Product Owner* es el máximo responsable.
- ▶ Lenguaje natural y comprensible por todos: **User Stories**

PRODUCT BACKLOG



PRODUCT BACKLOG USER STORIES

Forma de expresar los requisitos en lenguaje natural:

Como <**tipo de usuario**>

Quiero <**necesidad a implementar**>

Para <**beneficio u objetivo a conseguir**>

*Como **usuario registrado** de la web, debo poder **cambiar la contraseña**;
con el objetivo de **personalizarla y recordarla** mejor.*

*Como **usuario registrado** de la web **IKEA**, debo poder **guardar mi diseño**
para **seguir trabajando** en otro momento.*

PRODUCT BACKLOG

USER STORIES

1.1.1 As a Project manager

User story ID	I want to...	so that...
1.	Add project members to my project	the project members can register hours on the project
2.	View report on total hours spent on the project	I can track the project and see if we are on budget

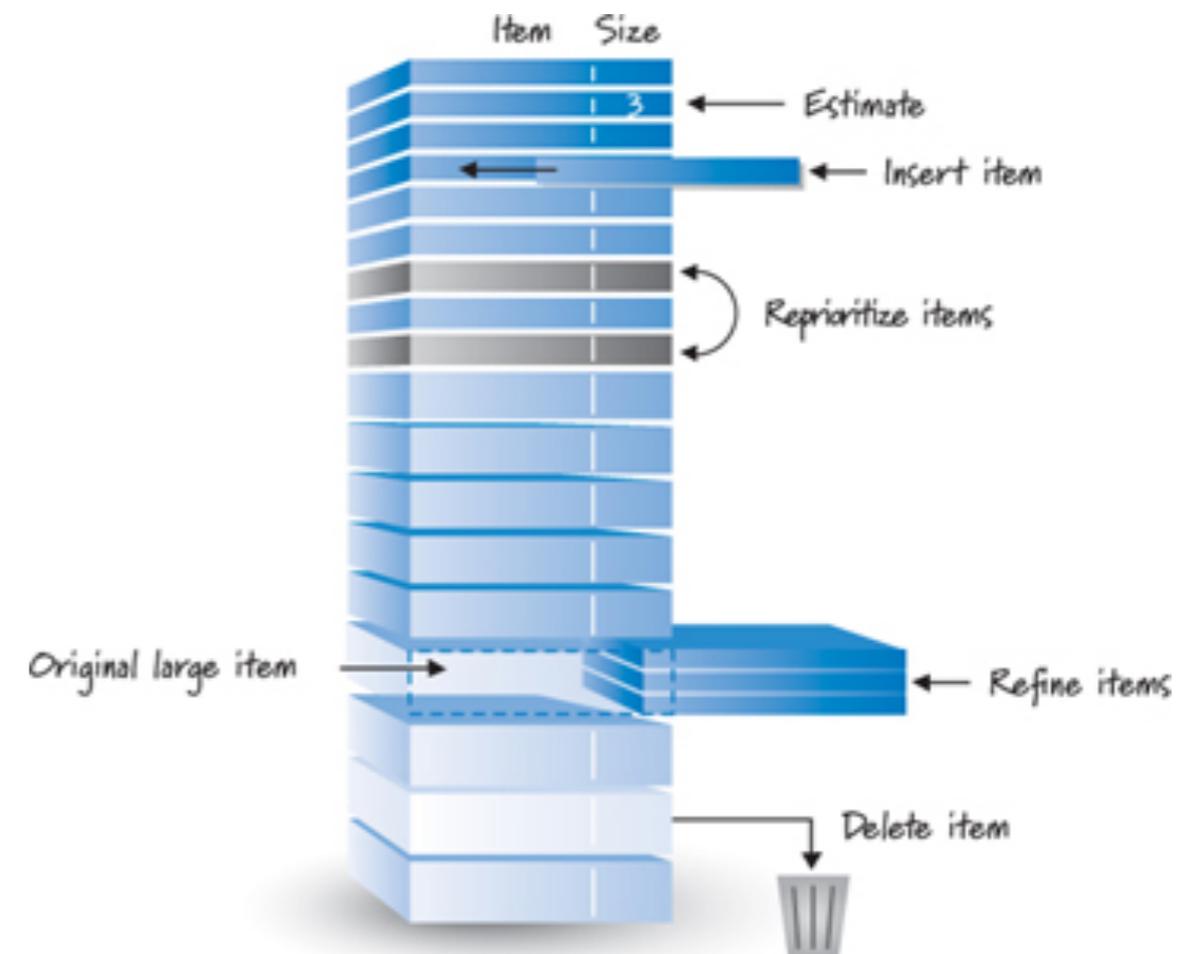
1.1.2 As a Project member

User story ID	I want to...	so that...
3.	Register hours on a project	my project manager know how much time I have spent on the project
4.	View a weekly report on total hours spent	I can see how much I have been working

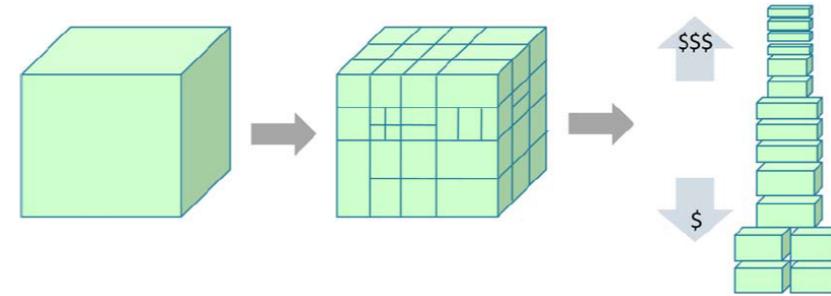
PRODUCT BACKLOG

A lo largo del proyecto el Product Backlog puede variar:

- ▶ Añadimos nuevas necesidades
- ▶ Cambiamos prioridades
- ▶ Eliminamos elementos
- ▶ Recalculamos elementos

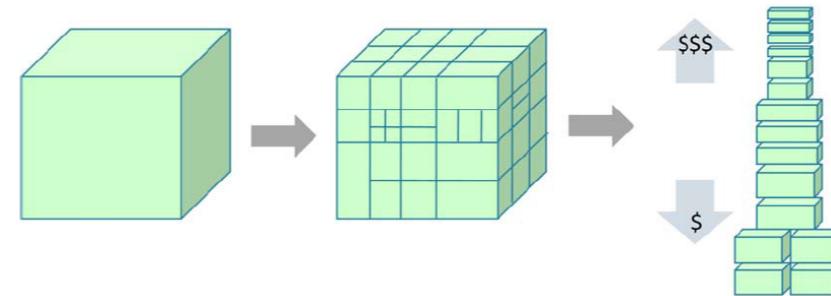


PRODUCT BACKLOG PRIORIZACIÓN



User Storie	Nivel necesidad	Horas	Valor	PRIORIDAD
Storie A	IMPORTANTE	10	50M €	
Storie B	IMPRESCINDIBLE	5	100M €	
Storie C	PRESCINDIBLE	5	10M €	
Storie D	IMPORTANTE	10	600M €	
Storie E	IMPRESCINDIBLE	20	50M €	

PRODUCT BACKLOG PRIORIZACIÓN



User Storie	Nivel necesidad	Horas	Valor	PRIORIDAD
Storie A	IMPORTANTE	10	50M €	4
Storie B	IMPRESCINDIBLE	5	100M €	1
Storie C	PRESCINDIBLE	5	10M €	5
Storie D	IMPORTANTE	10	600M €	3
Storie E	IMPRESCINDIBLE	20	50M €	2

ELEMENTOS

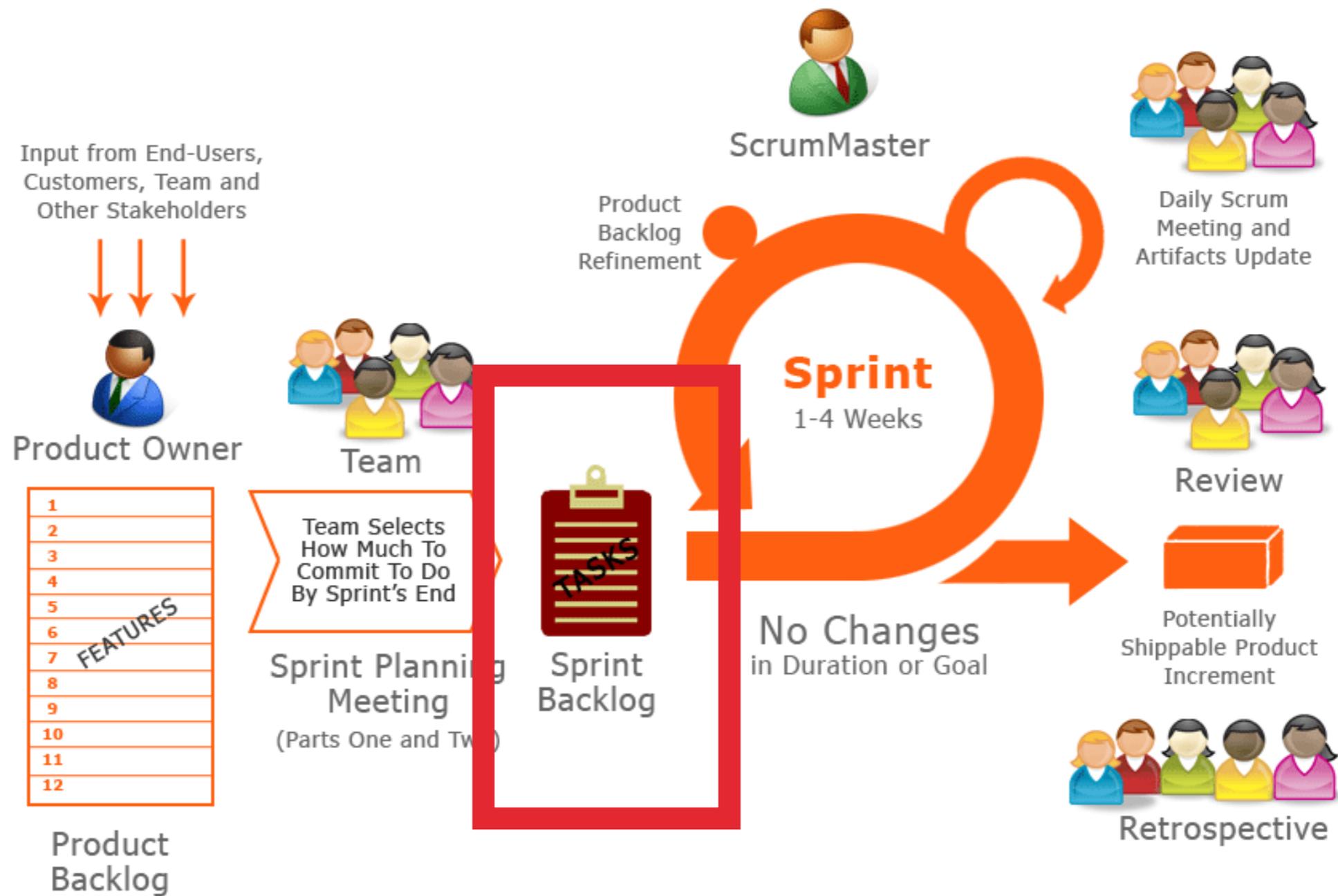
PRODUCT BACKLOG

SPRINT BACKLOG

SPRINT

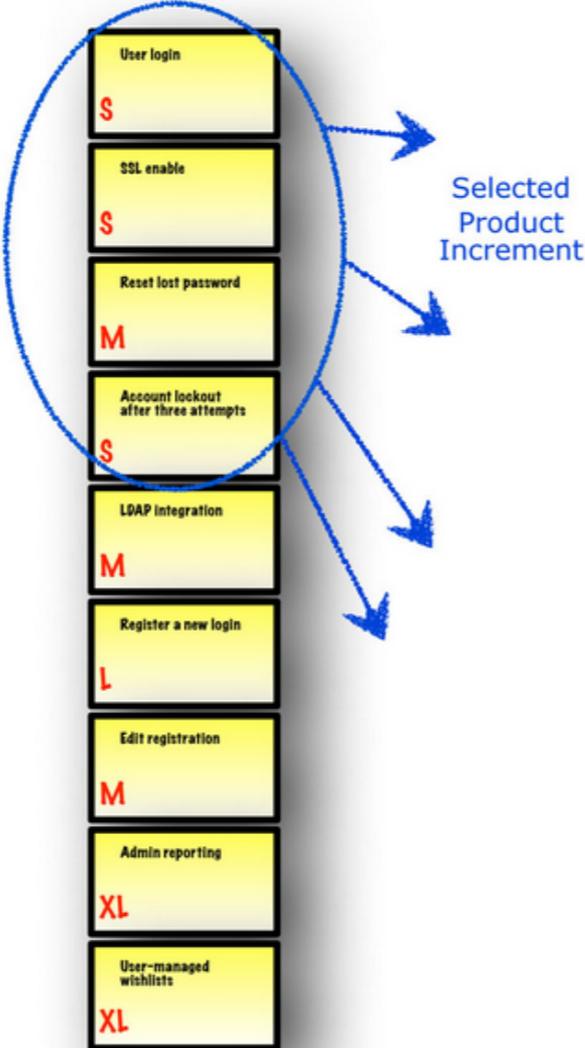
INCREMENTO

SPRINT BACKLOG

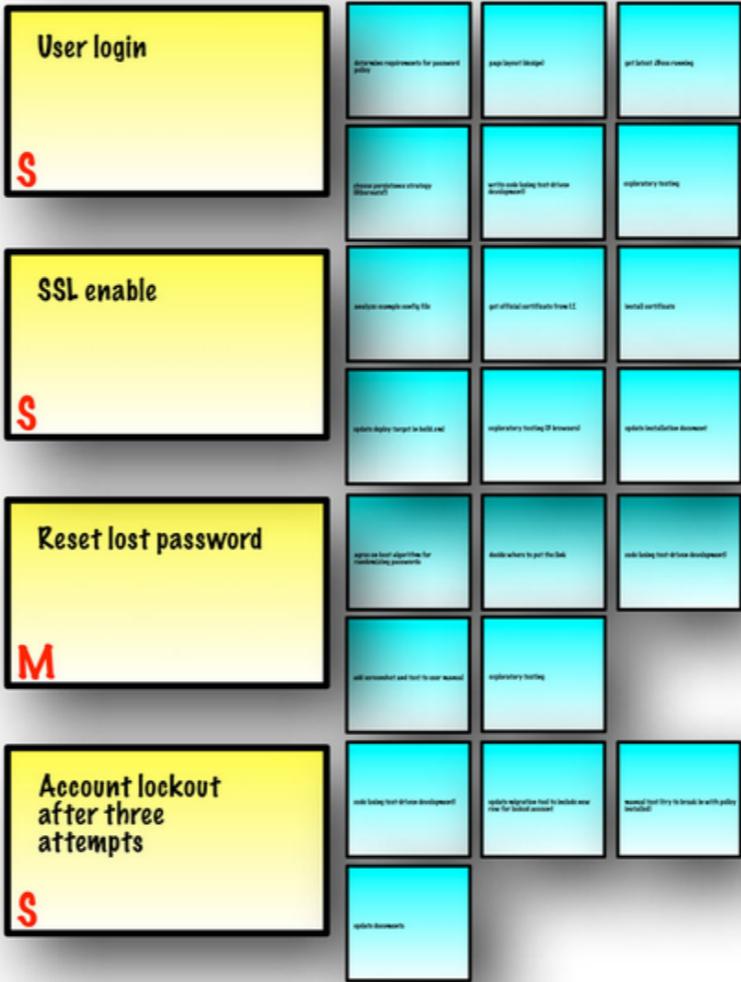


SPRINT BACKLOG

Product Backlog



Sprint Backlog



ELEMENTOS

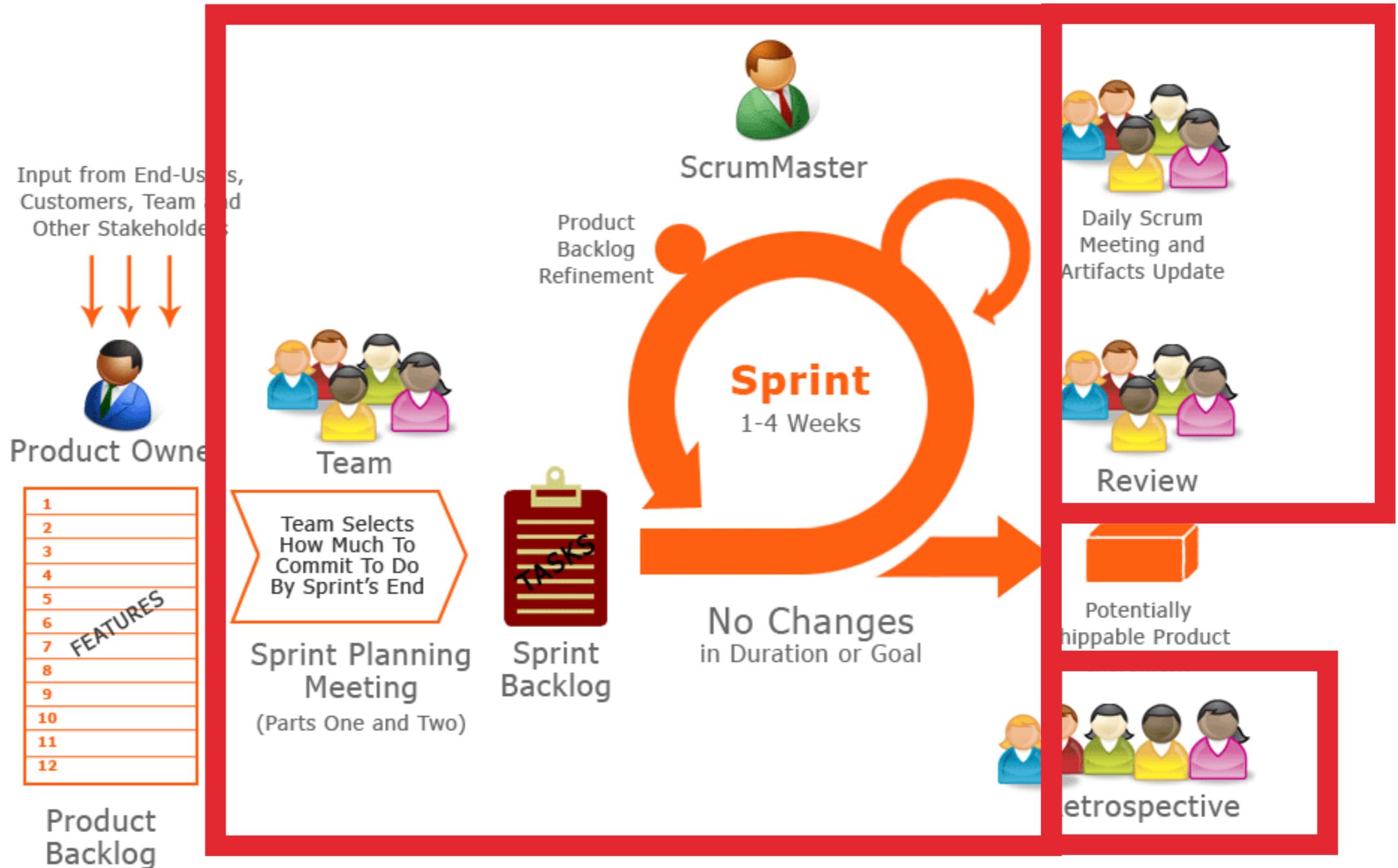
PRODUCT BACKLOG

SPRINT BACKLOG

SPRINT

INCREMENTO

SPRINT



SPRINT



El "sprint" es el core del SCRUM.

- ▶ Podemos decir que son las iteraciones del proyecto.
- ▶ Los sprints no acostumbran a durar mucho, entre 1 y 4 semanas (dependiendo del proyecto).
- ▶ Cada sprint tiene un objetivo claro "**Sprint Goal**" y un **Incremento** claro a entregar.
- ▶ El trabajo a realizar dentro de un sprint NO cambia durante la duración del sprint.
- ▶ Objetivos cortos, reales y alcanzables.

ELEMENTOS

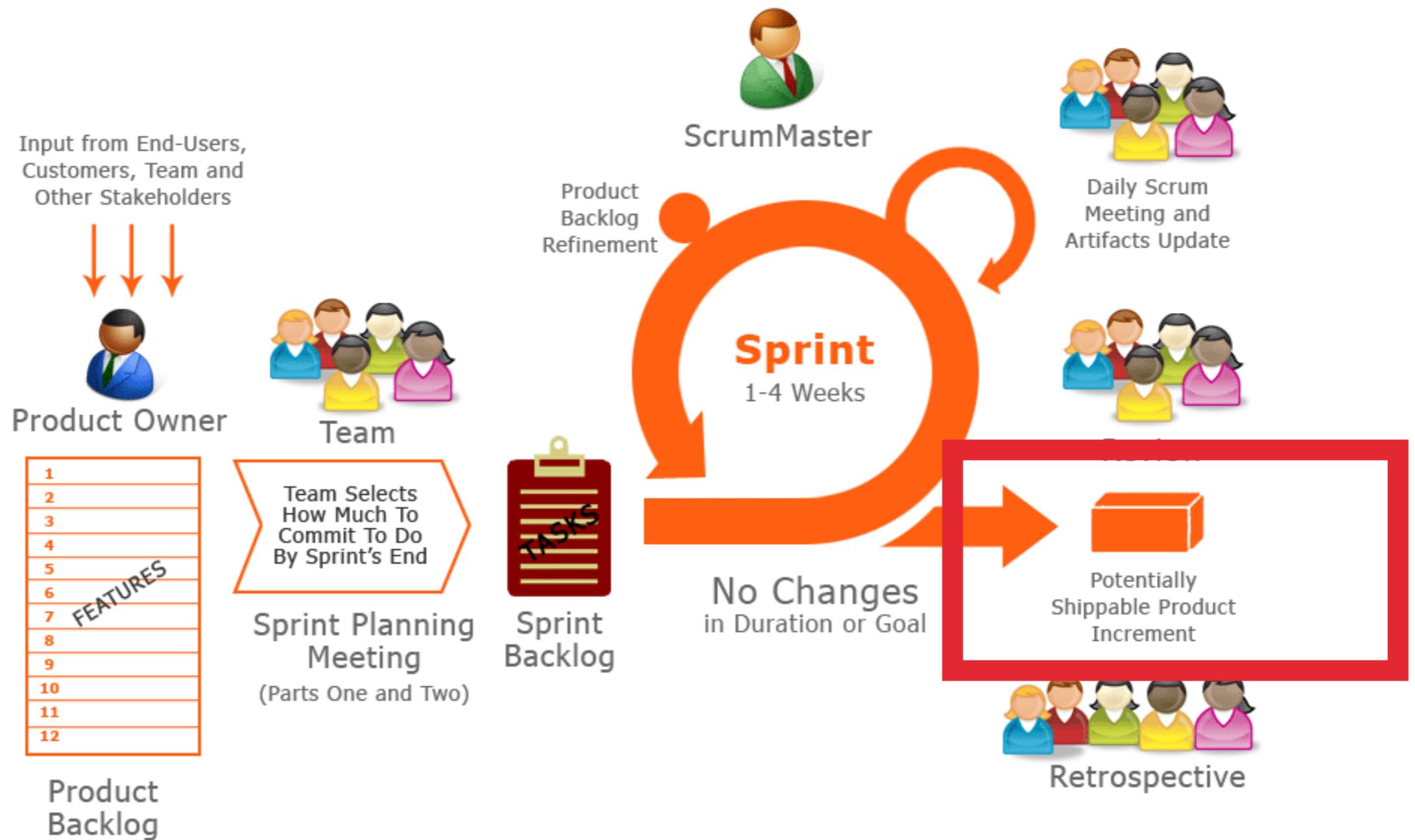
PRODUCT BACKLOG

SPRINT BACKLOG

SPRINT

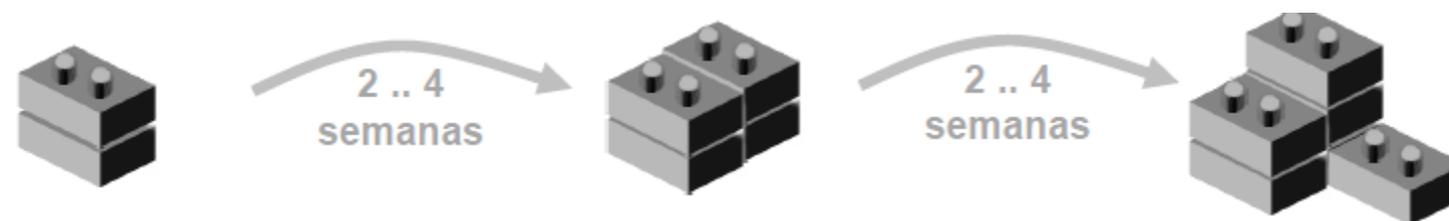
INCREMENTO

INCREMENTO DEL PRODUCTO



INCREMENTO DEL PRODUCTO

- ▶ Es el resultado de cada Sprint. Las dos características más importantes que debe reunir este incremento son:
 - ▶ **“Acabado”**. Potencialmente se pueda poner en producción. El Manifiesto Ágil lo deja claro, el incremento debe ser algo funcional.
 - ▶ Debe aportar valor a nuestros Clientes / Usuarios.



SPRINT

ENTREGA DE PRODUCTOS “ACABADOS”



Table of Sprints

01

Android navigation definition

- Login
- DashBoards

02

Web application

- Database
- Web application
 - Login
 - Navigation
 - Data management

03

Android UI and WS

- Bluetooth control
- Web services definition and implementation
- Android WS integration
- Android location services

04

Final sprint

- Push notifications
- News
- Geofencing

ROLES

SCRUM

ROLES

PRODUCT OWNER

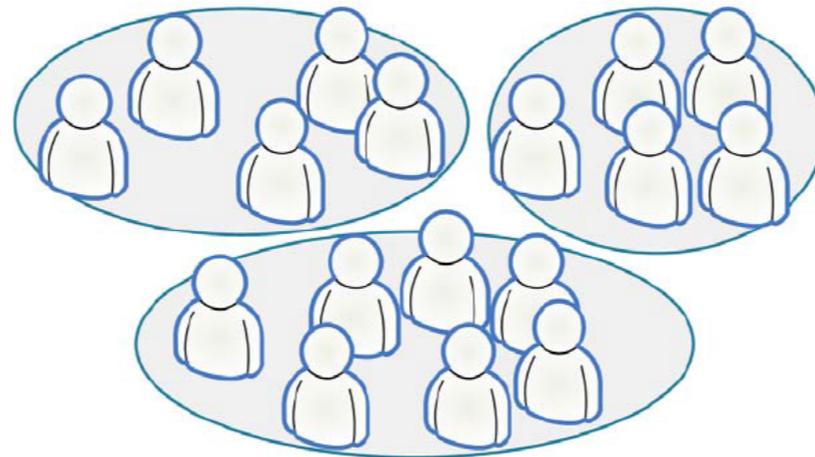
EQUIPO DE TRABAJO

SCRUM MASTER



EQUIPO DE TRABAJO

- ▶ Equipos pequeños y auto-gestionados.



- ▶ Identificamos los siguientes roles:



Product Owner



Team



ScrumMaster

ROLES

PRODUCT OWNER

EQUIPO DE TRABAJO

SCRUM MASTER



PRODUCT OWNER



Product Owner

Es el responsable autorizado por la parte del cliente; su misión es la de maximizar el valor del producto.

- ▶ Máximo responsable de la lista de producto *"Product Backlog"*.
- ▶ Prioriza los requisitos.
- ▶ Indica los requisitos y realiza las peticiones que el cliente busca en el producto final.
- ▶ No tiene perfil técnico, realiza las peticiones en lenguaje natural - *"User stories"*.

ROLES

PRODUCT OWNER

EQUIPO DE TRABAJO

SCRUM MASTER

DEVELOPMENT TEAM



Es el grupo de profesionales que realizan el trabajo para conseguir la entrega del producto.

- ▶ Entre 3 y 9 personas
- ▶ Equipos multidisciplinares
- ▶ Auto-organizados
- ▶ Mucha confianza
- ▶ Poca jerarquía
- ▶ Comunicación y transparencia

ROLES

PRODUCT OWNER

EQUIPO DE TRABAJO

SCRUM MASTER

SCRUM MASTER



ScrumMaster

Es el responsable de asegurar que se utiliza la metodología correctamente, que se entiende por todos los actores y actúa como facilitador entre las partes .

- ▶ Líder para el resto del equipo.
- ▶ Maximiza el valor del proyecto y producto.
- ▶ Asegura que se cumplan los tiempos, las reuniones, actualiza documentación.
- ▶ Trata errores y problemas con el cliente, abstrae al equipo de desarrollo de posibles problemas.

MEETINGS.

SCRUM

MEETINGS

SPRINT PLANNING MEETING

DAILY SCRUM MEETING

SPRINT REVIEW

SPRINT RETROSPECTIVE

MEETINGS

SPRINT PLANNING MEETING

DAILY SCRUM MEETING

SPRINT REVIEW

SPRINT RETROSPECTIVE

SPRINT PLANNING MEETING

Reunión que se realiza al inicio de cada *sprint*. Se valora, elige y planifica el trabajo que se debe realizar en las próximas semanas (el sprint).

- ▶ ¿Que incluye el siguiente **incremento** del producto?
- ▶ ¿Cómo se reparte el trabajo?
- ▶ ¿Cómo nos organizamos?

SPRINT PLANNING MEETING

En esta reunión participan todos los miembros del equipo SCRUM, incluido el cliente representado por el P.O.

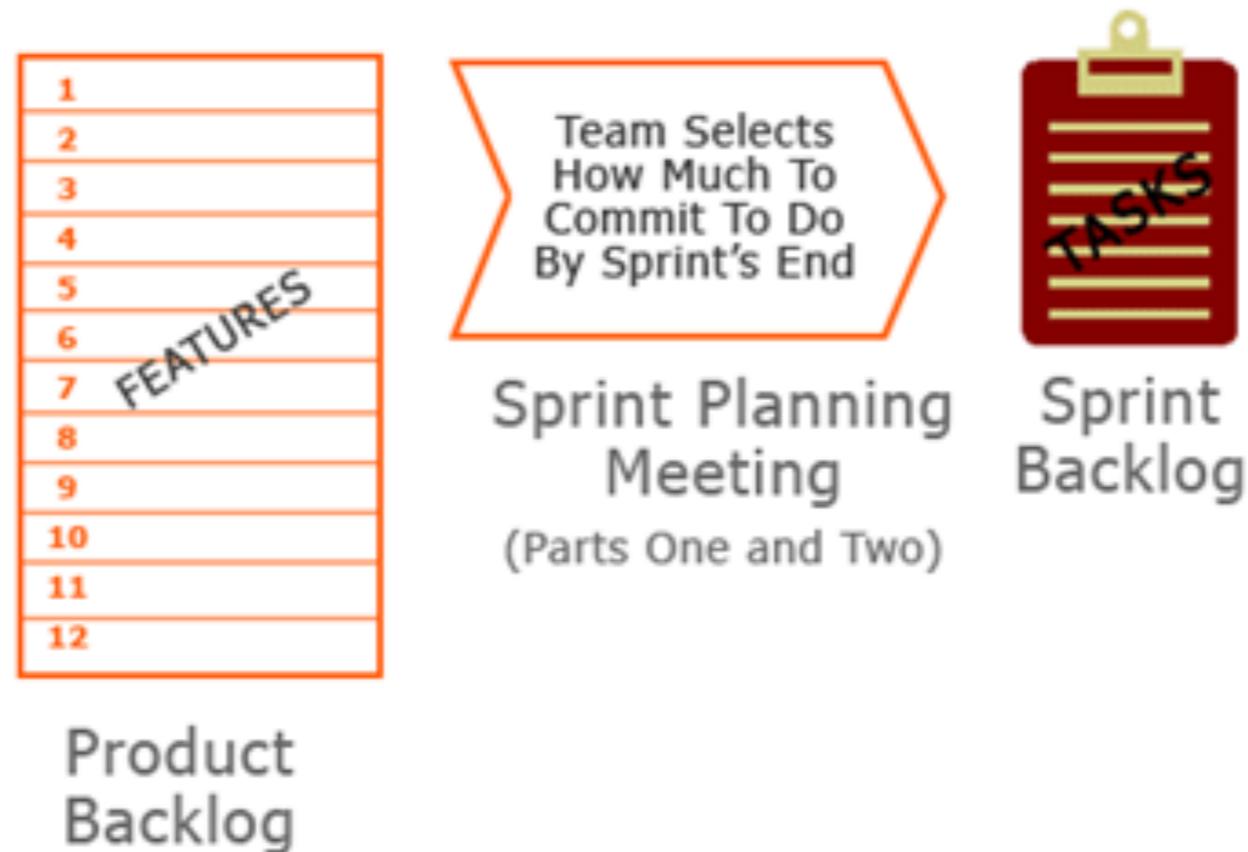
Puede llegar a durar algunas horas; pero debemos salir...

- ▶ Teniendo claro cual es el objetivo del *sprint*; **Incremento** del producto
- ▶ Con un **sprint backlog** claro y que no variara durante el sprint.

SPRINT PLANNING MEETING

PRODUCT BACKLOG > SPRINT BACKLOG

- ▶ Seleccionamos que parte del Product Backlog vamos a **realizar en un sprint:**

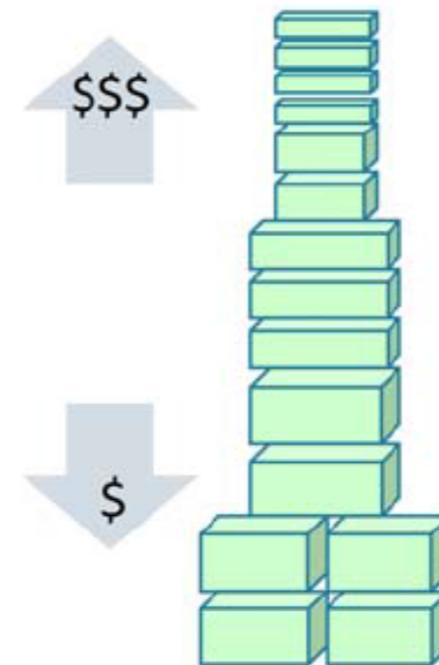


SPRINT PLANNING MEETING

PRODUCT BACKLOG > SPRINT BACKLOG

- ▶ El Cliente (con la figura del Product Owner) con la colaboración y asesoramiento del equipo de desarrollo es el responsable de seleccionar que trabajo desea realizar en el sprint.

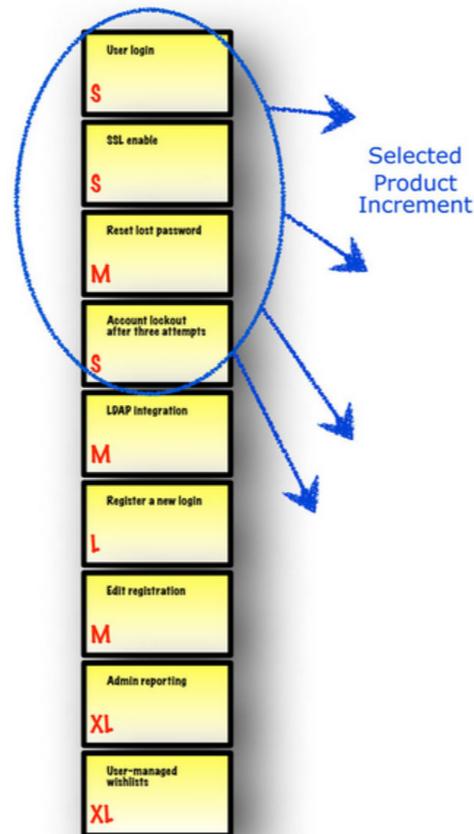
- ▶ Necesidad técnica
 - ▶ Valor de negocio
 - ▶ Coste
 - ▶ Riesgo
- } R.O.I



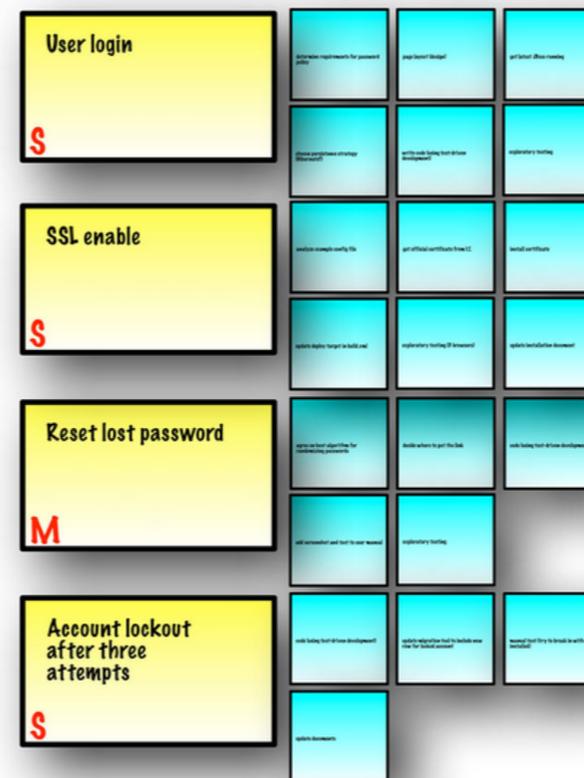
SPRINT PLANNING MEETING PRODUCT BACKLOG > SPRINT BACKLOG

- ▶ Si es necesario, se descomponen las *user stories* elegidas en tareas más pequeñas para facilitar la gestión, repartición y control del trabajo...

Product Backlog



Sprint Backlog



MEETINGS

SPRINT PLANNING MEETING

DAILY SCRUM MEETING

SPRINT REVIEW

SPRINT RETROSPECTIVE

DAILY SCRUM MEETING

- ▶ Reunión diaria
- ▶ Unos 15-20 minutos aproximadamente
- ▶ Reunión distendida y en ambiente informal
- ▶ Solo participa el equipo de desarrollo y el Scrum Master
- ▶ Se habla sobre el trabajo para la próxima jornada y temas concretos.

DAILY SCRUM MEETING

- ▶ Cada miembro del equipo comenta rápidamente:
 - ▶ Trabajo realizado el día anterior
 - ▶ Previsión del trabajo de hoy
 - ▶ ¿Hay algún problema para alcanzar el objetivo del sprint?
- ▶ ¡De pie!
- ▶ Importante respetar tiempo; deben salir los problemas existentes, NO resolverlos.

MEETINGS

SPRINT PLANNING MEETING

DAILY SCRUM MEETING

SPRINT REVIEW

SPRINT RETROSPECTIVE

SPRINT REVIEW

- ▶ Revisión/presentación al final de cada sprint. Se valora y comprueba el incremento del producto.
- ▶ El cliente valida si el trabajo pactado al inicio del sprint se ha realizado y es correcto. El objetivo es conseguir su aprobación.
- ▶ Reunión formal donde pueden atender todos los miembros del equipo scrum.
- ▶ Se repasan líneas de tiempo y coste.

MEETINGS

SPRINT PLANNING MEETING

DAILY SCRUM MEETING

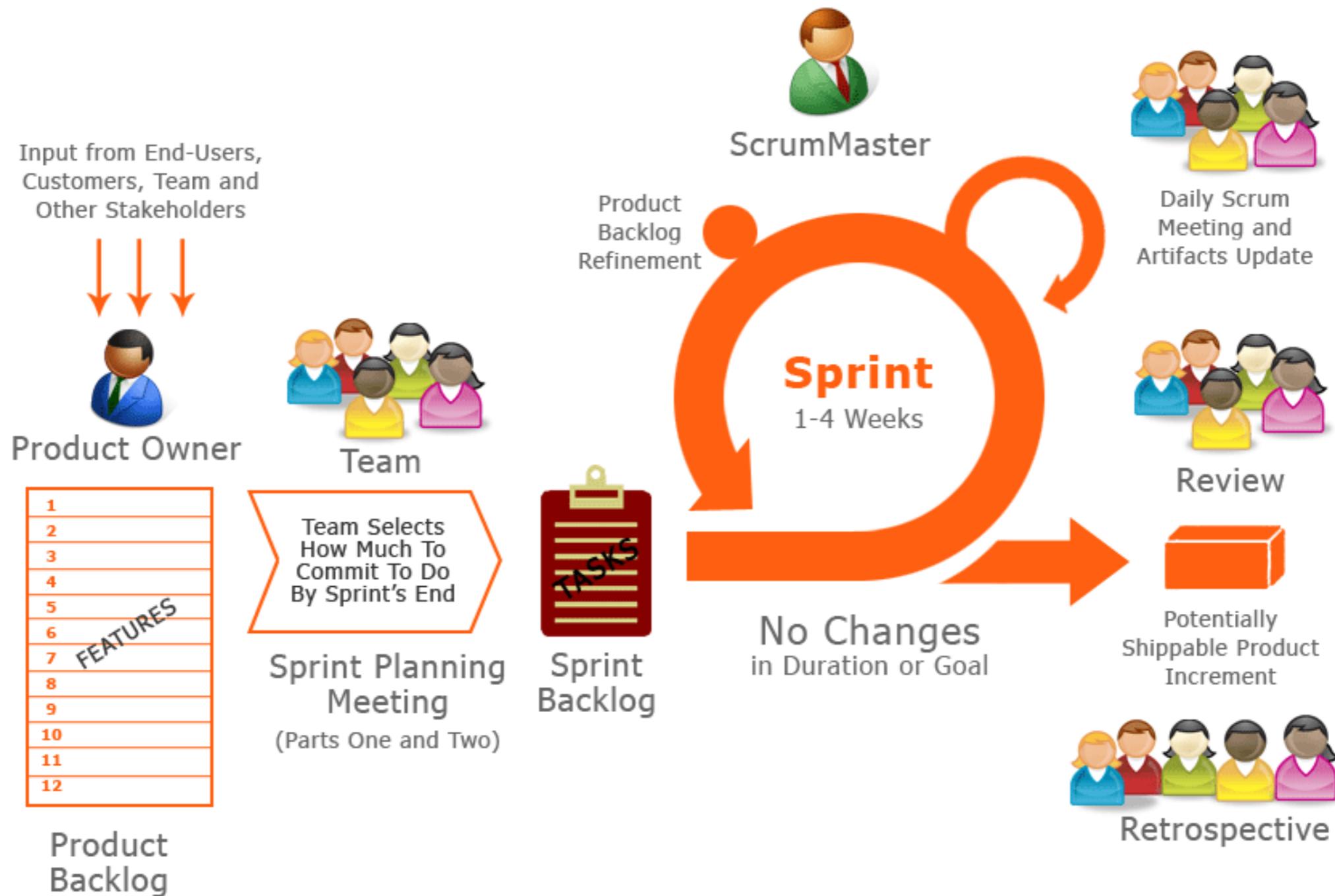
SPRINT REVIEW

SPRINT RETROSPECTIVE

SPRINT RETROSPECTIVE

- ▶ Valoración propia e interna del equipo de desarrollo.
- ▶ Se realiza después de la revisión con el cliente.
- ▶ Se repasan las cosas bien hechas, se comentan y documentan.
- ▶ Que aspectos deben mejorar; o que cosas no debemos repetir.
- ▶ Pretende mejorar la eficiencia, calidad y busca la mejora continua de los equipos.
- ▶ Ventajas para futuros sprints y futuros proyectos.

REPRESENTACIÓN SCRUM



ARTIFACTS

ARTIFACTS

BURNUP CHART

BURNDOWN CHART

KANBAN



ARTIFACTS

BURNUP CHART

BURNDOWN CHART

KANBAN

GRÁFICO BURNUP

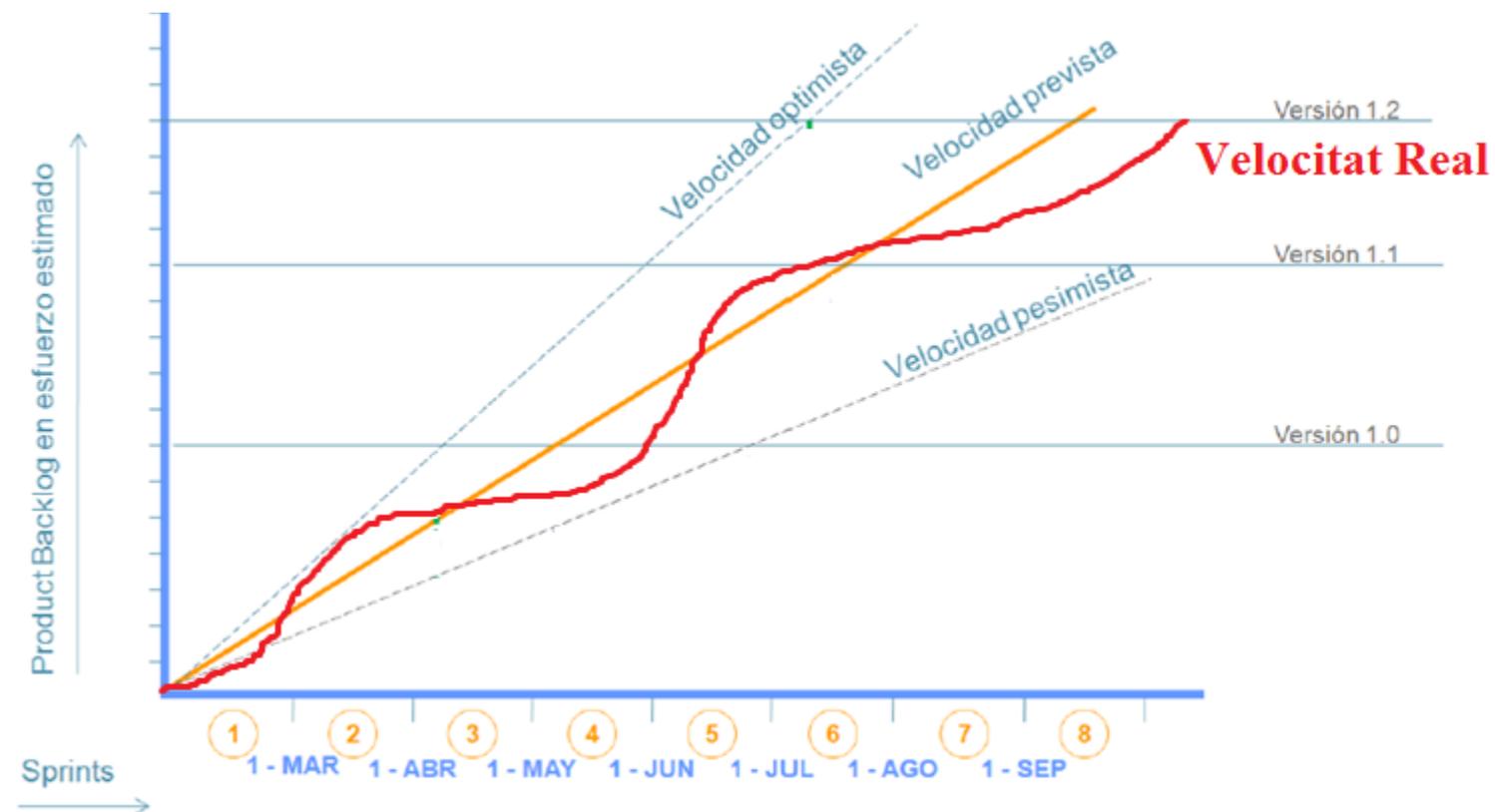
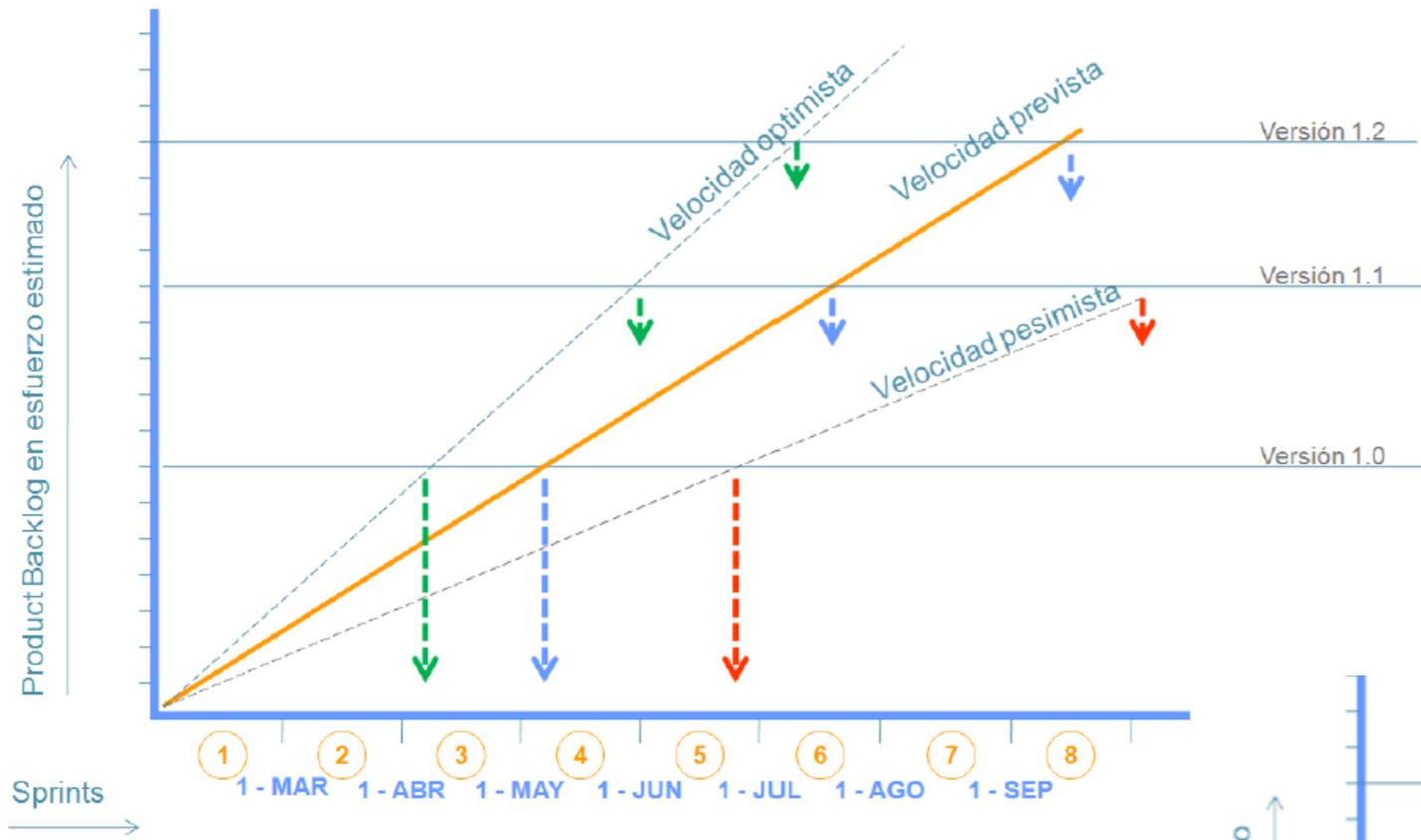
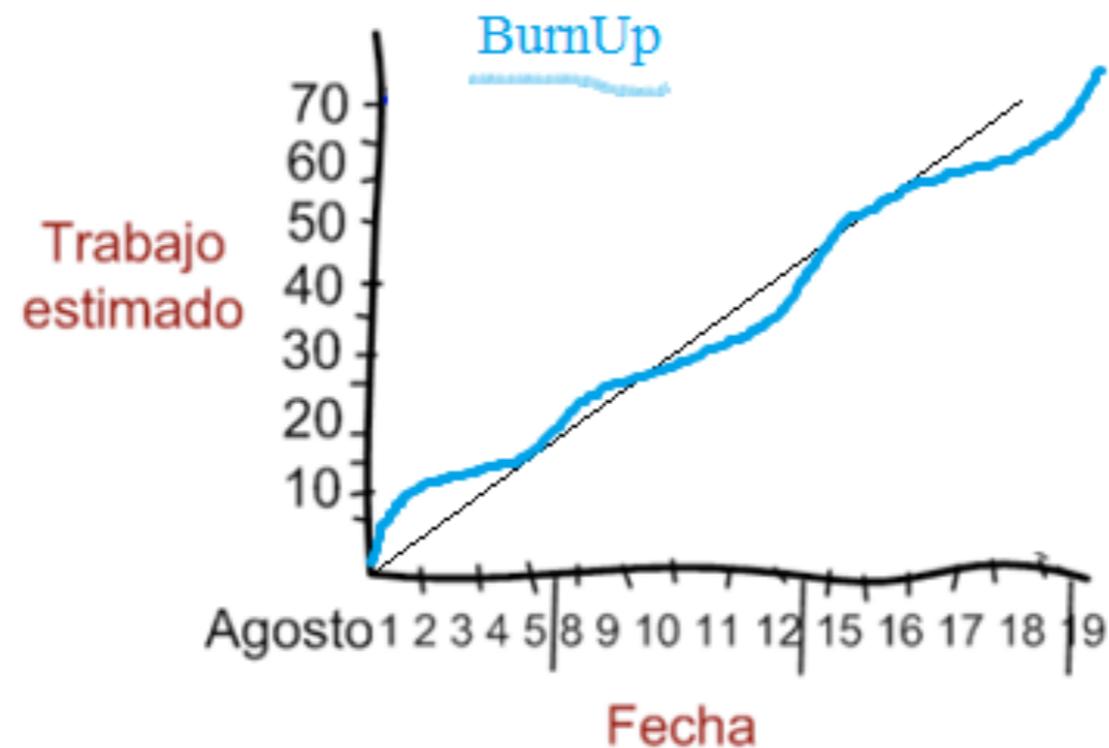


GRÁFICO BURNUP

- ▶ Herramienta seguimiento y control del trabajo planificado.
- ▶ Comparamos la planificación con la evolución real.
- ▶ Se ve la velocidad con la que el equipo realiza el trabajo.



ARTIFACTS

BURNUP CHART

BURNDOWN CHART

KANBAN

GRÁFICO BURNDOWN

- ▶ Representa la evolución del sprint día a día. Podemos ver la previsión futura y los datos reales.
- ▶ En el eje X tenemos el tiempo; mientras que el eje Y representa el trabajo pendiente (tareas, horas, puntos....).

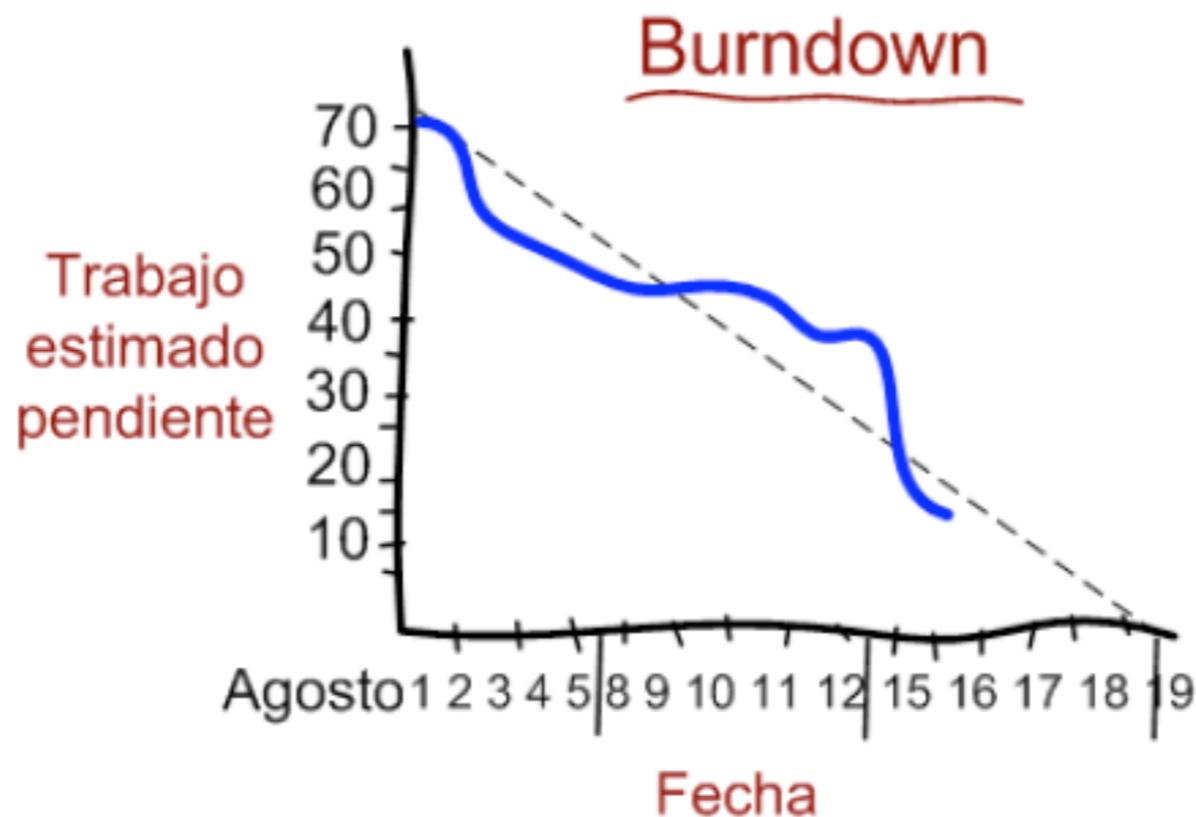
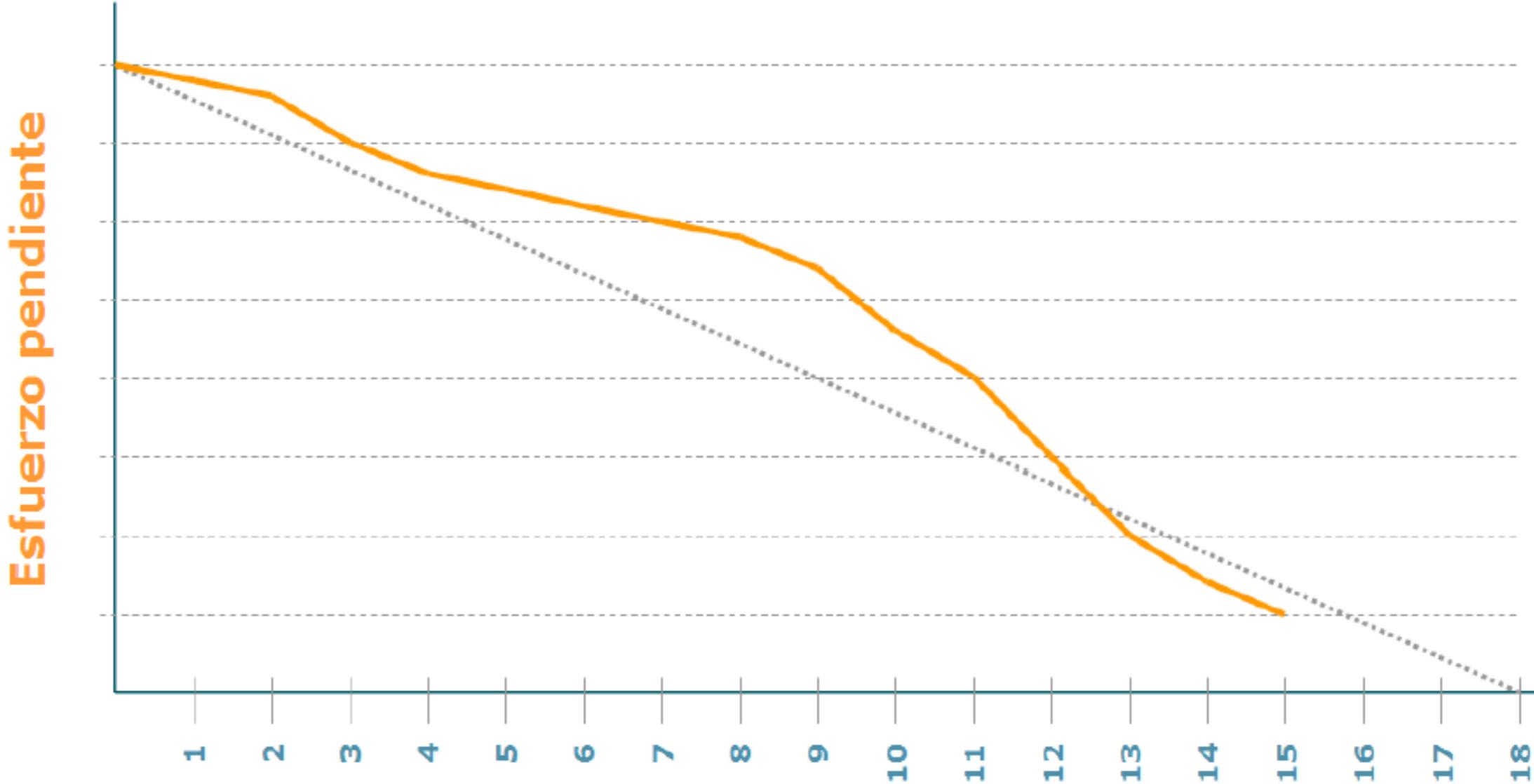
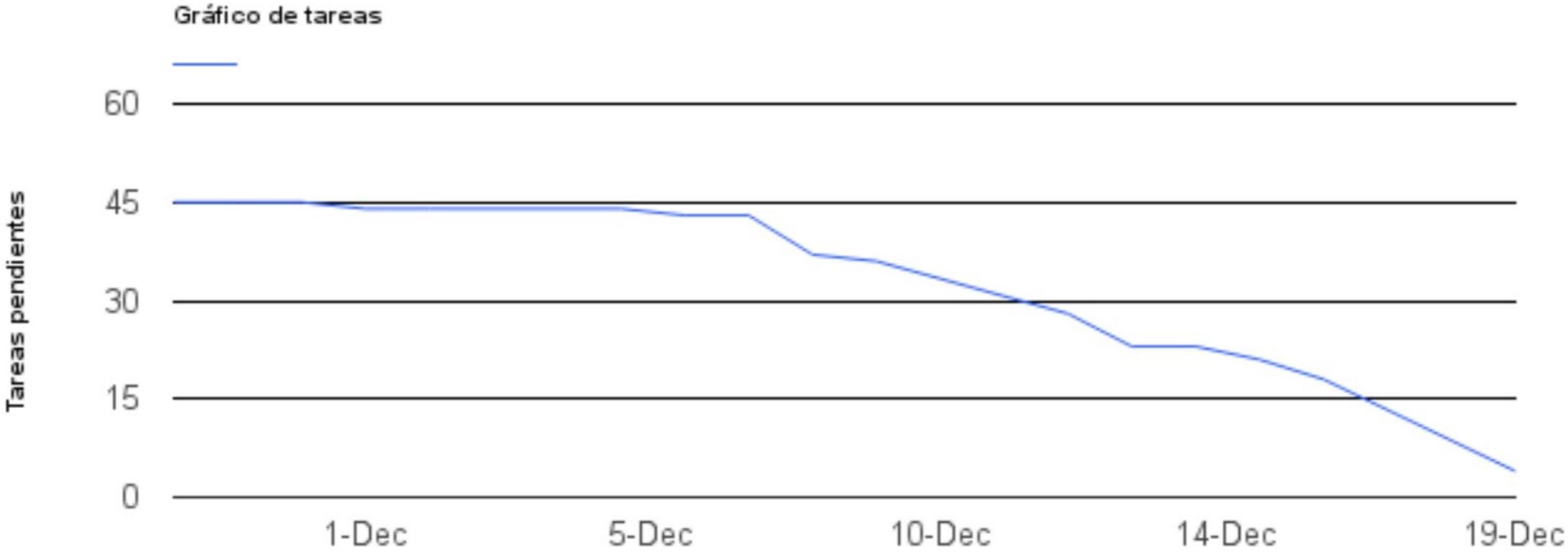


GRÁFICO BURNDOWN

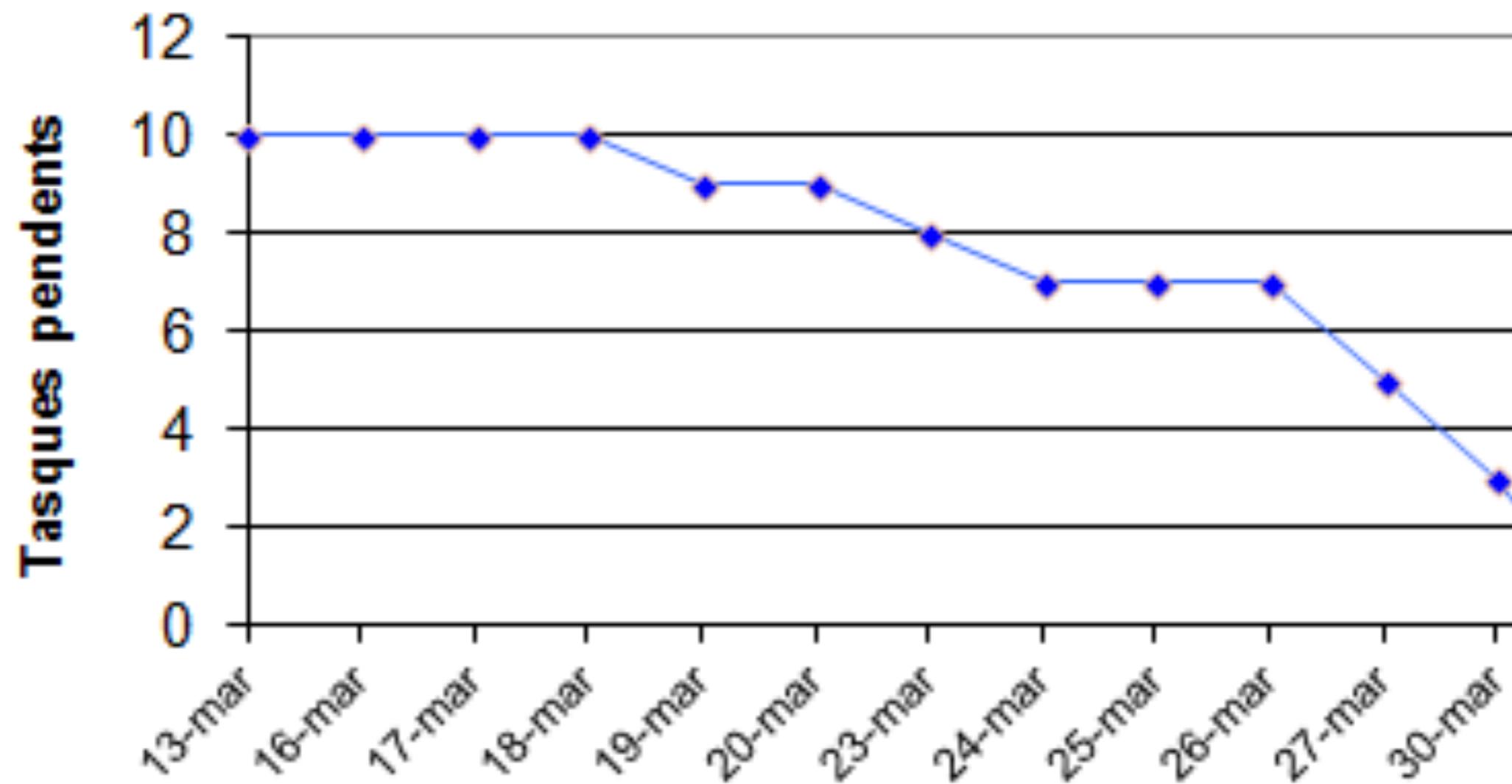


BURNDOWN TAREAS

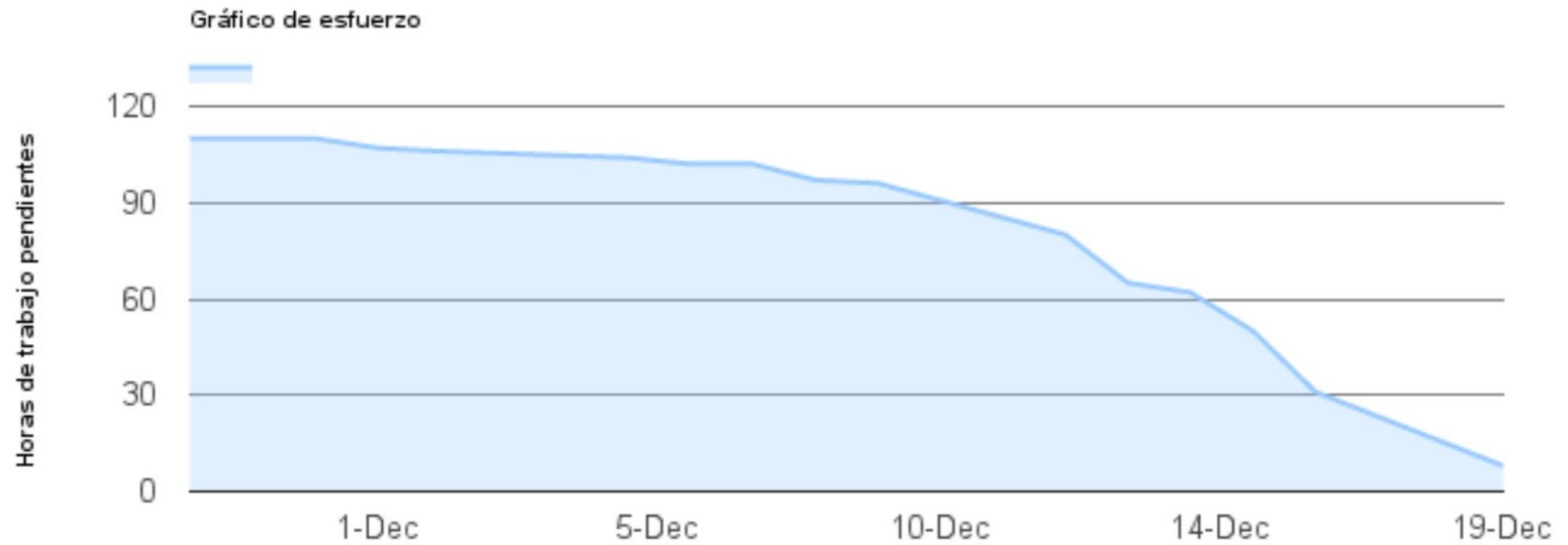


BURNDOWN TAREAS

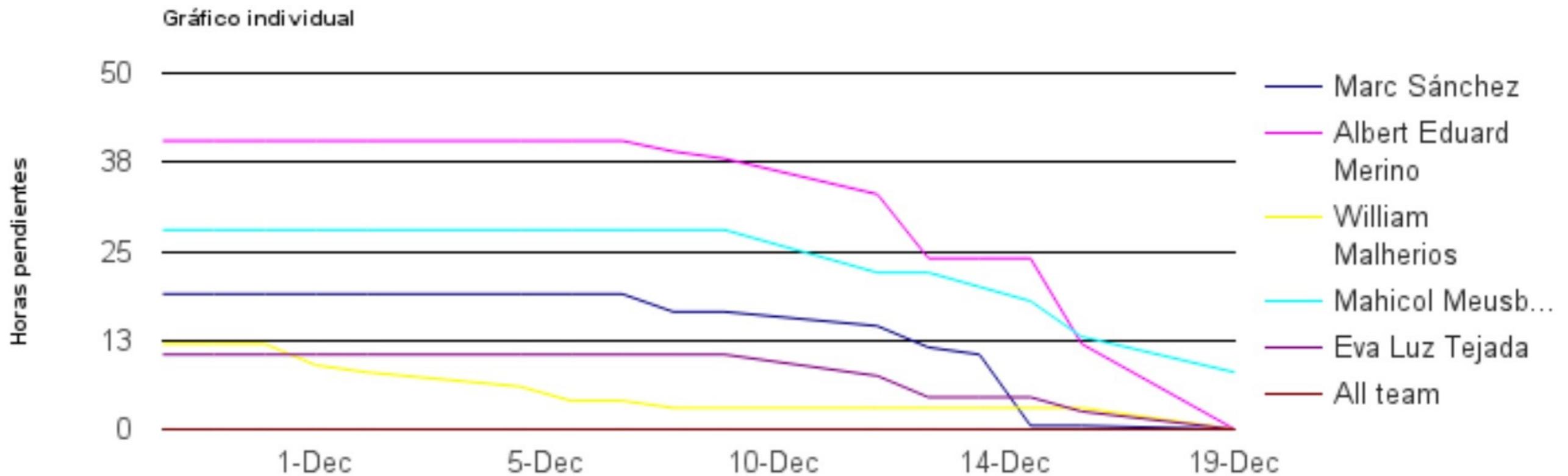
Gràfic de tasques pendents



BURNDOWN HORAS



BURNDOWN POR MIEMBROS DEL EQUIPO



BURNDOWN EN ZENHUB

Iteration 1

First iteration focused on the API

Start: Feb 23, 2016 Due: Apr 13, 2016

Labels ▼ Hide Pull Requests

Burn Pipelines ▼



ARTIFACTS

BURNUP CHART

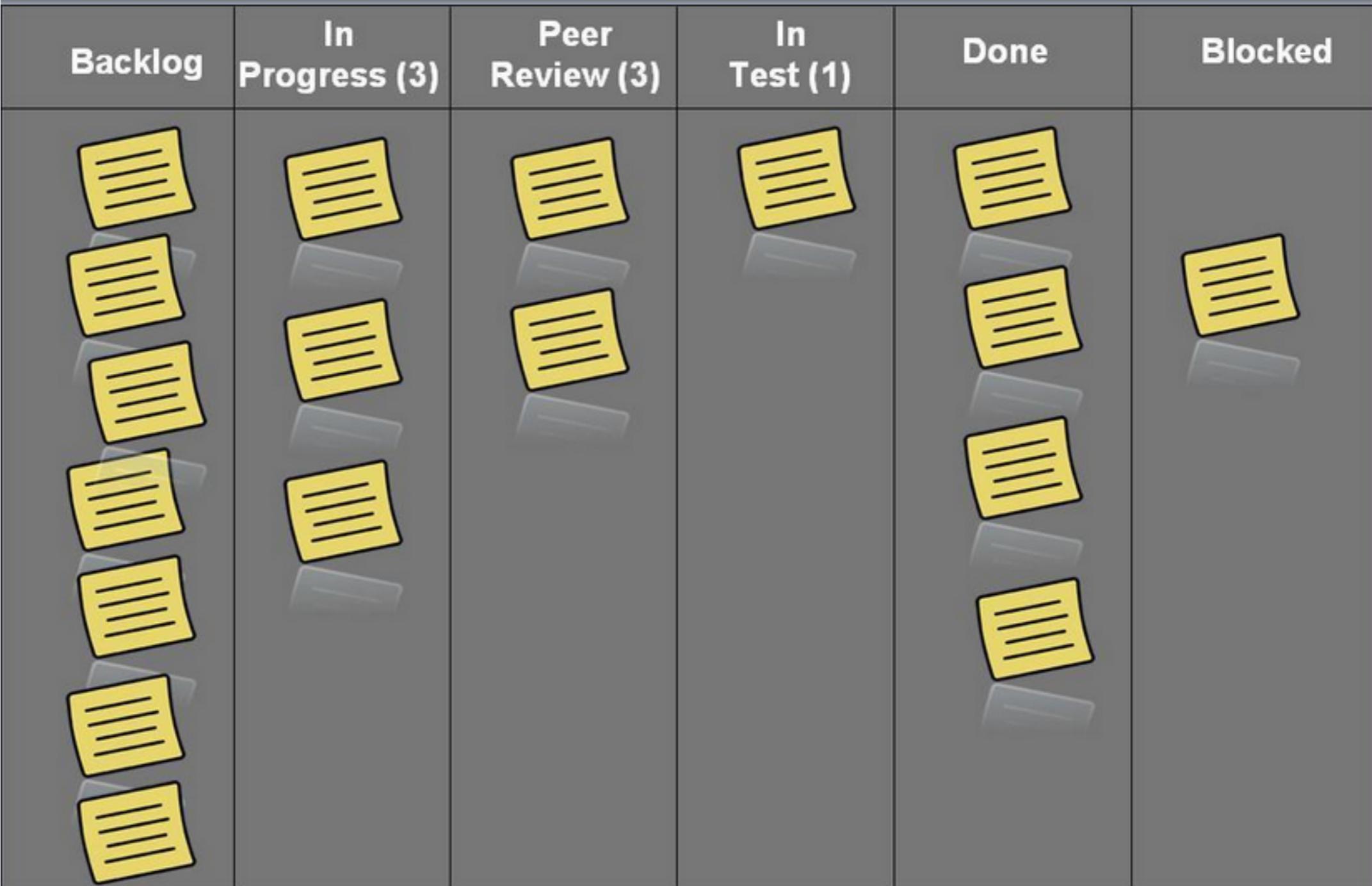
BURNDOWN CHART

KANBAN

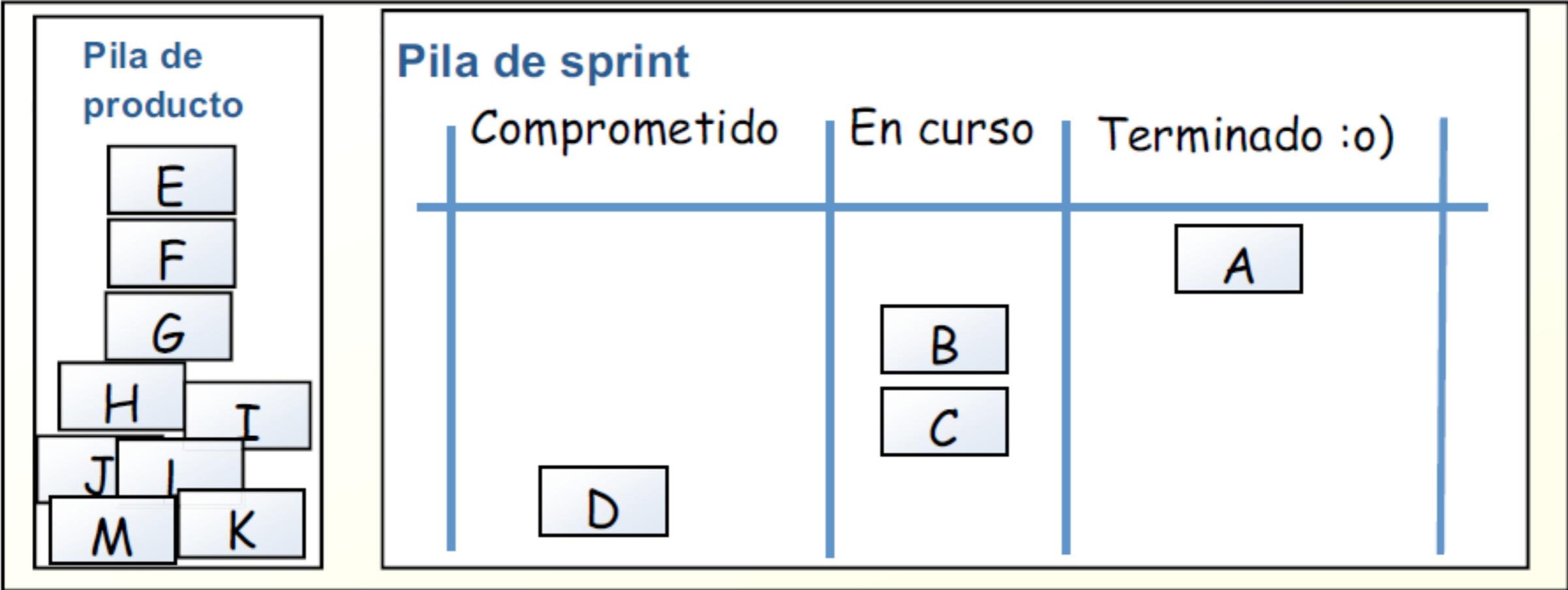
KANBAN

- ▶ Nombre que proviene de japonés:
 - ▶ Kan = Visual
 - ▶ Ban = tarjeta o tablero
- ▶ Funciones:
 - ▶ Flujo de trabajo
 - ▶ Limita el trabajo en curso

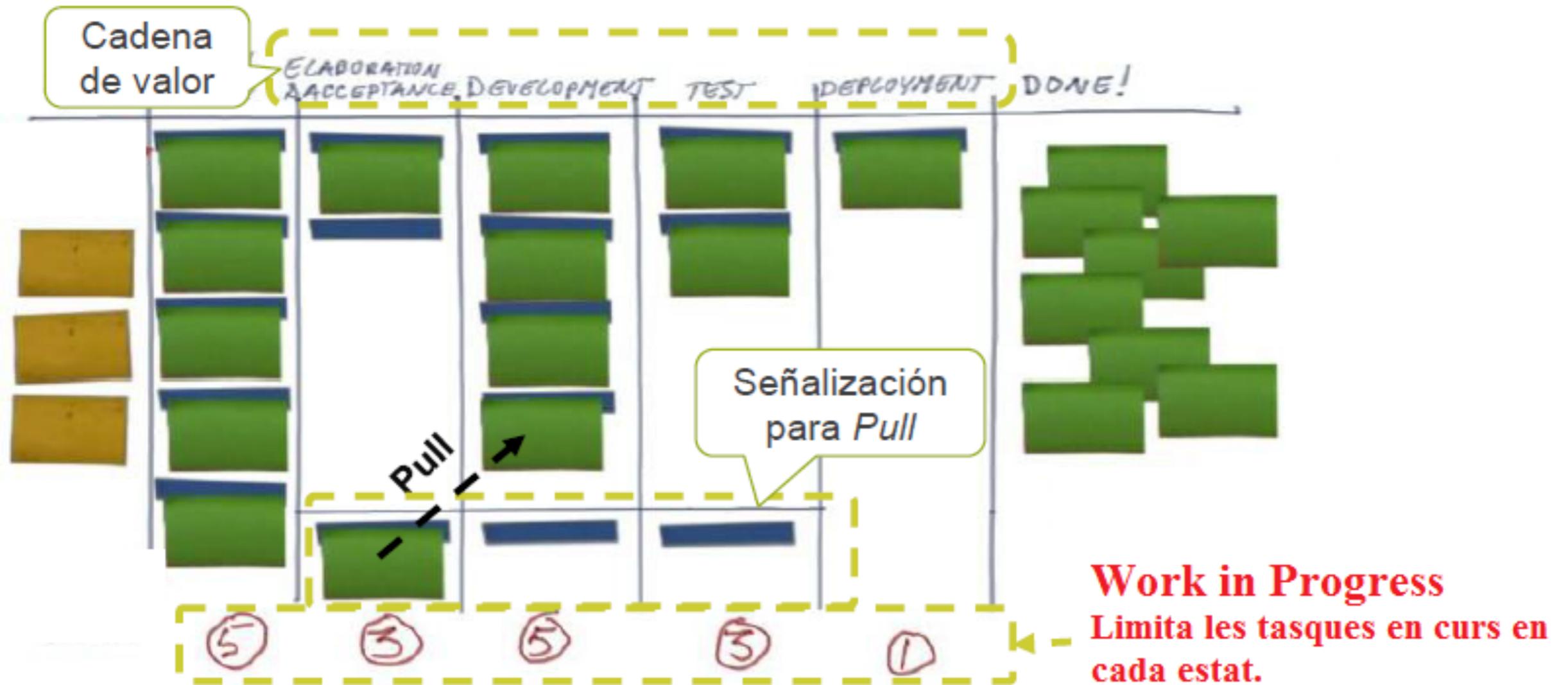
KANBAN



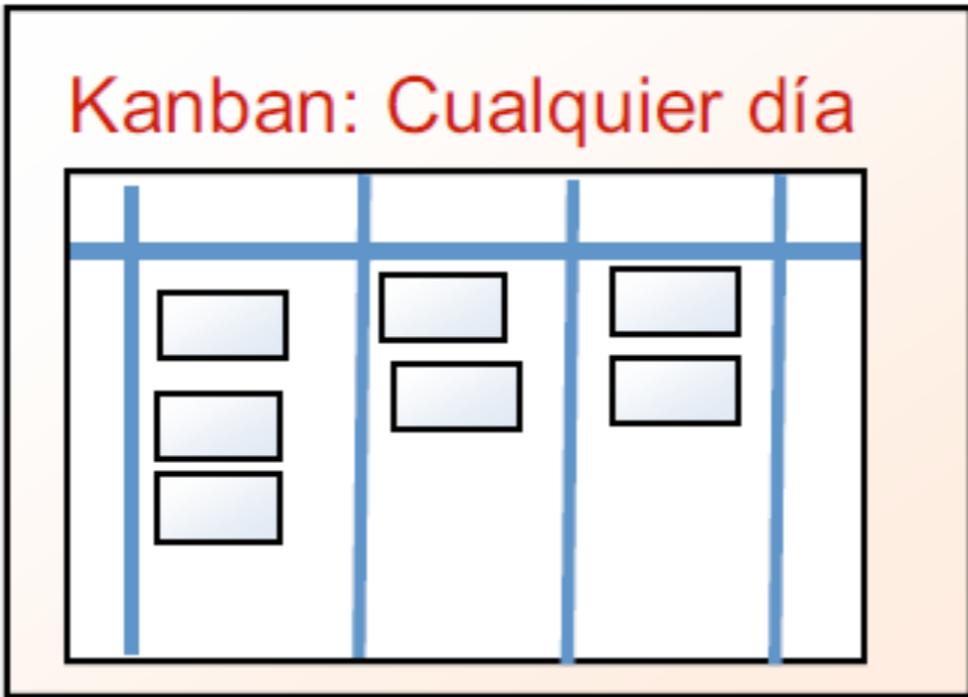
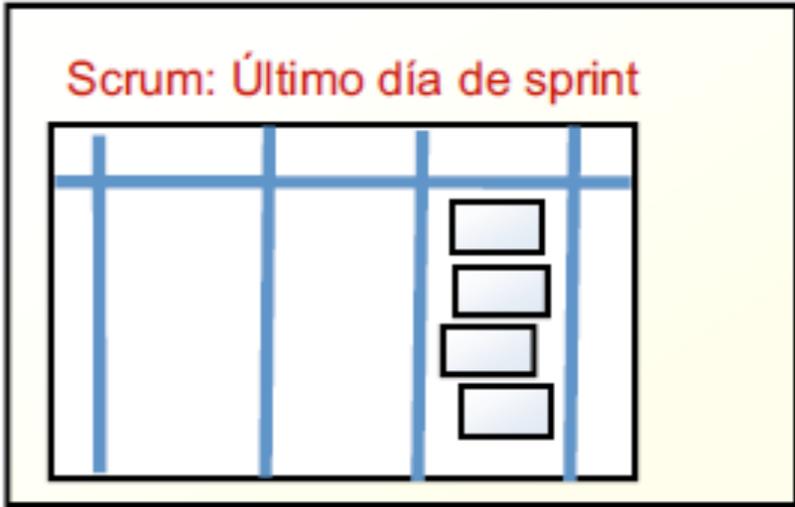
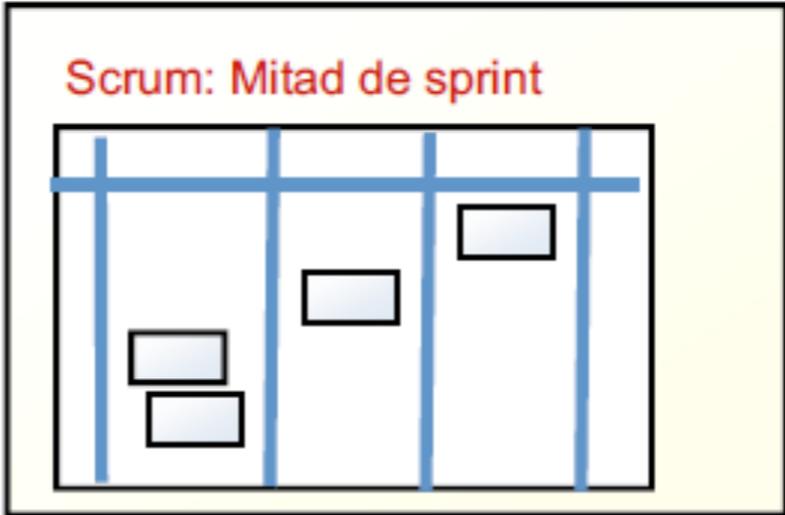
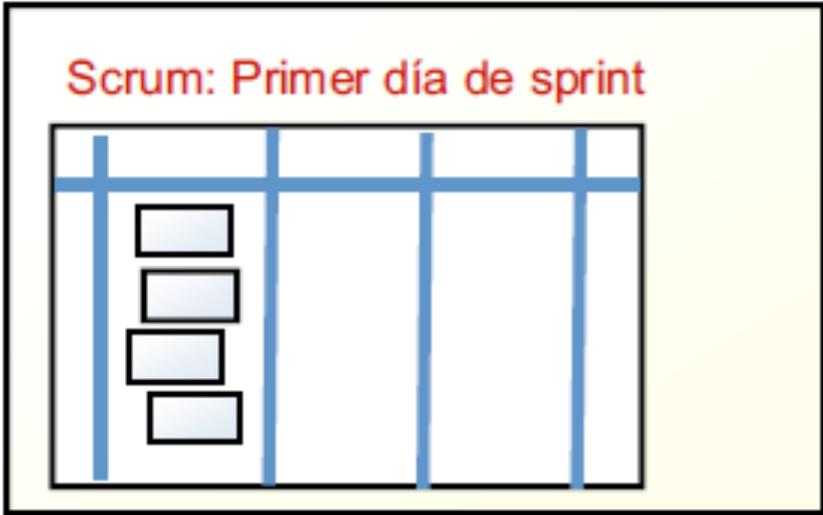
KANBAN



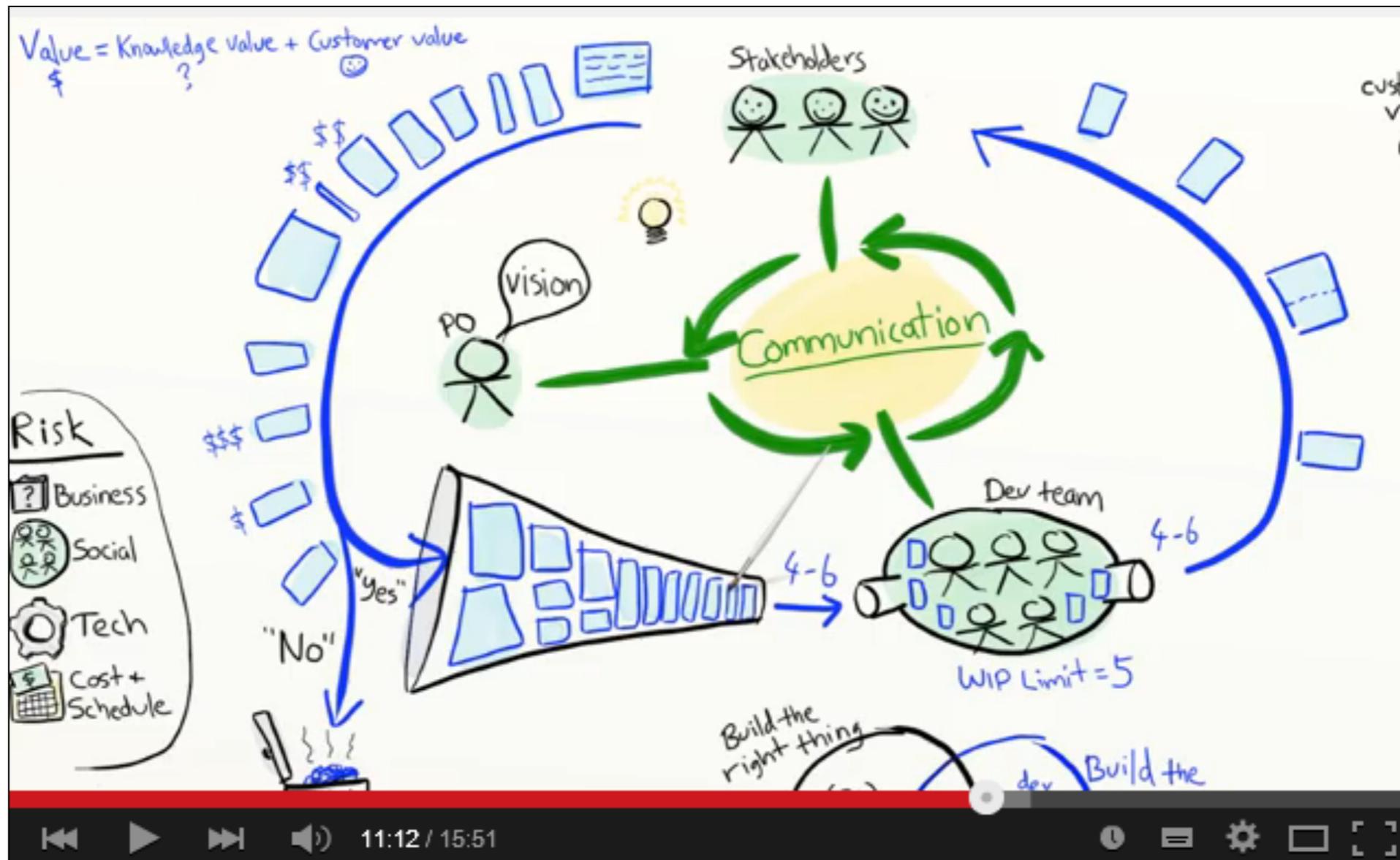
KANBAN



KANBAN



SCRUM



SCRUM





Universitat
de Lleida

¡ MUCHAS GRACIAS !

jtorrecillas2@gmail.com

 Jordi Torrecillas Garcia

 @jtorrecillas2