



CORONEL

DESARROLLA SOFTWARE

DESARROLLA SOFTWARE

METODOLOGIAS AGILES

Eric Gustavo Coronel Castillo

gcoronelc@gmail.com

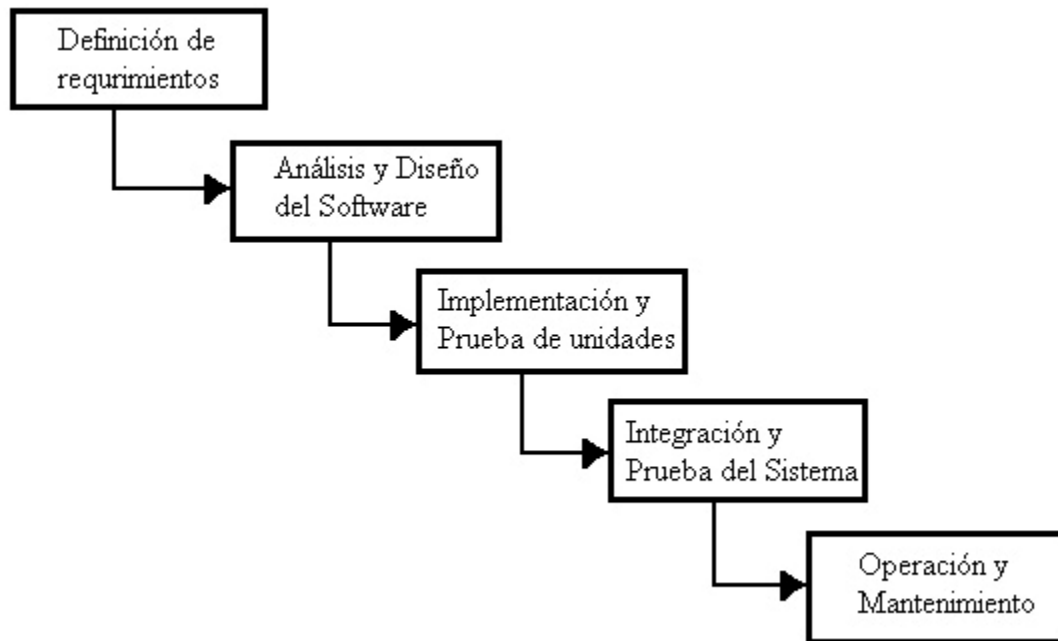
Temas

- Proyecto Ágil
- Componentes Fundamentales
- Scrum
- Lean
- Kanban



PROYECTO AGIL

DESARROLLO TRADICIONAL O DESARROLLO EN CASCADA



PROYECTO AGIL

DEFINICIONES

CASCADA

Las fases del ciclo de vida (requisitos, análisis, diseño, etc.) se realizan (en teoría) de manera lineal, una única vez, y el inicio de una fase no comienza hasta que termina la fase anterior.

Su naturaleza es lineal, típica de la construcción de productos físicos y su principal problema viene de que no deja claro cómo responder cuándo el resultado de una fase no es el esperado.

PROYECTO AGIL

DEFINICIONES

CICLO DE VIDA INCREMENTAL

Cada iteración contiene las fases del cascada estándar, pero cada iteración trabaja sobre un sub conjunto de funcionalidad. La entrega total del proyecto se divide en subsistemas priorizados.

Desarrollar por partes el producto software, para después integrarlas a medida que se completan. Un ejemplo de un desarrollo puramente incremental puede ser la agregación de módulos en diferentes fases. El agregar cada vez más funcionalidad al sistema.

PROYECTO AGIL

DEFINICIONES

CICLO DE VIDA ITERATIVO

En cada ciclo, iteración, se revisa y mejora el producto. Un ejemplo de desarrollo iterativo es aquel basado en refactorizaciones, en el que cada ciclo mejora más la calidad del producto.

Es importante señalar que este ciclo no implica añadir funcionalidades en el producto, pero si la revisión y la mejora.

PROYECTO AGIL

DEFINICIONES

CICLO DE VIDA ITERATIVO E INCREMENTAL

Se va liberando partes del producto (prototipos) periódicamente, en cada iteración, y cada nueva versión, normalmente, aumenta la funcionalidad y mejora en calidad respecto a la anterior.

Además, el ciclo de vida iterativo e incremental es una de las bases de un proyecto ágil, más concretamente, con iteraciones cortas en tiempo, de pocas semanas, normalmente un mes y raramente más de dos

PROYECTO AGIL

DEFINICIONES

CICLO DE VIDA ÁGIL

Sería un ciclo de vida iterativo e incremental, con iteraciones cortas (semanas) y sin que dentro de cada iteración tenga que haber fases lineales (tipo cascada).

A partir de la anterior, por cada metodología ágil que existe, se tienen diferentes matizaciones, adaptaciones, etc.

PROYECTO AGIL

DEFINICIONES

PROYECTO ÁGIL

Un proyecto ágil se podría definir como una manera de enfocar el desarrollo software mediante un ciclo iterativo e incremental, con equipos que trabajan de manera altamente colaborativa y auto-organizados.

PROYECTO AGIL

EL MANIFIESTO AGIL

VALORES AGILES

1. Valorar a los individuos y las interacciones del equipo de desarrollo sobre el proceso y las herramientas.
2. Desarrollar software que funciona más que conseguir una documentación exhaustiva.
3. La colaboración con el cliente más que la negociación de un contrato.
4. Responder a los cambios más que seguir estrictamente un plan.

PROYECTO AGIL

EL MANIFIESTO AGIL

PRINCIPIOS AGILES

1. La prioridad es satisfacer al cliente mediante entregas tempranas y continuas de software que le aporten valor.
2. Dar la bienvenida a los cambios. Se capturan los cambios para que el cliente tenga una ventaja competitiva.
3. Entregar frecuentemente software que funcione desde un par de semanas a un par de meses, con el menor intervalo de tiempo posible entre entregas.
4. La gente del negocio y los desarrolladores deben trabajar juntos a lo largo del proyecto.

PROYECTO AGIL

EL MANIFIESTO AGIL

PRINCIPIOS AGILES

5. Construir el proyecto en torno a individuos motivados. Darles el entorno y el apoyo que necesitan y confiar en ellos para conseguir finalizar el trabajo.
6. El diálogo cara a cara es el método más eficiente y efectivo para comunicar información dentro de un equipo de desarrollo.
7. El software que funciona es la medida fundamental de progreso.
8. Los procesos ágiles promueven un desarrollo sostenible. Los promotores, desarrolladores y usuarios deberían ser capaces de mantener una paz constante.

PROYECTO AGIL

EL MANIFIESTO AGIL

PRINCIPIOS AGILES

9. La atención continua a la calidad técnica y al buen diseño mejora la agilidad.
10. La simplicidad es esencial.
11. Las mejores arquitecturas, requisitos y diseños surgen de los equipos organizados por sí mismos.
12. En intervalos regulares, el equipo reflexiona respecto a cómo llegar a ser más efectivo, y según esto ajusta su comportamiento.

INTRODUCCIÓN

COMPARATIVA

CARACTERÍSTICA	TRADICIONAL	ÁGIL
PROCESO	Altamente controlado	Ligeramente controlado
CONTRATO	Cerrado	Flexible
INTERACCIÓN CON EL CLIENTE	Reuniones	Forma parte del equipo de desarrollo
TAMAÑO DE LOS GRUPOS	Grandes	Pequeños, menos de 10 integrante
ARQUITECTURA DEL SOFTWARE	Muy importante	Menos importante
DOCUMENTACIÓN	Exhaustiva	Poca, sólo la necesaria para entender el código

COMPONENTES FUNDAMENTALES

EL PRODUCT OWNER

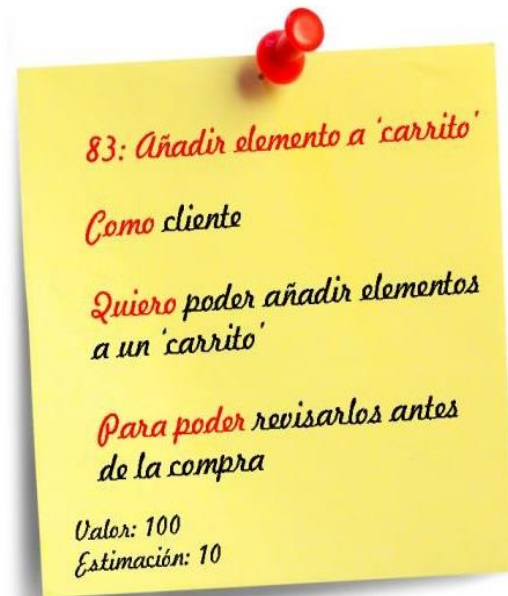
El “product owner” (o propietario del producto) es aquella persona con una visión muy clara del producto que se quiere desarrollar, que es capaz de transmitir esa visión al equipo de desarrollo y, además, está altamente disponible para transmitirla.

El product owner también es el responsable de la comunicación entre equipo y usuarios, y de gestionar qué trabajo se tiene que desarrollar, en qué orden y qué valor se va entregando.

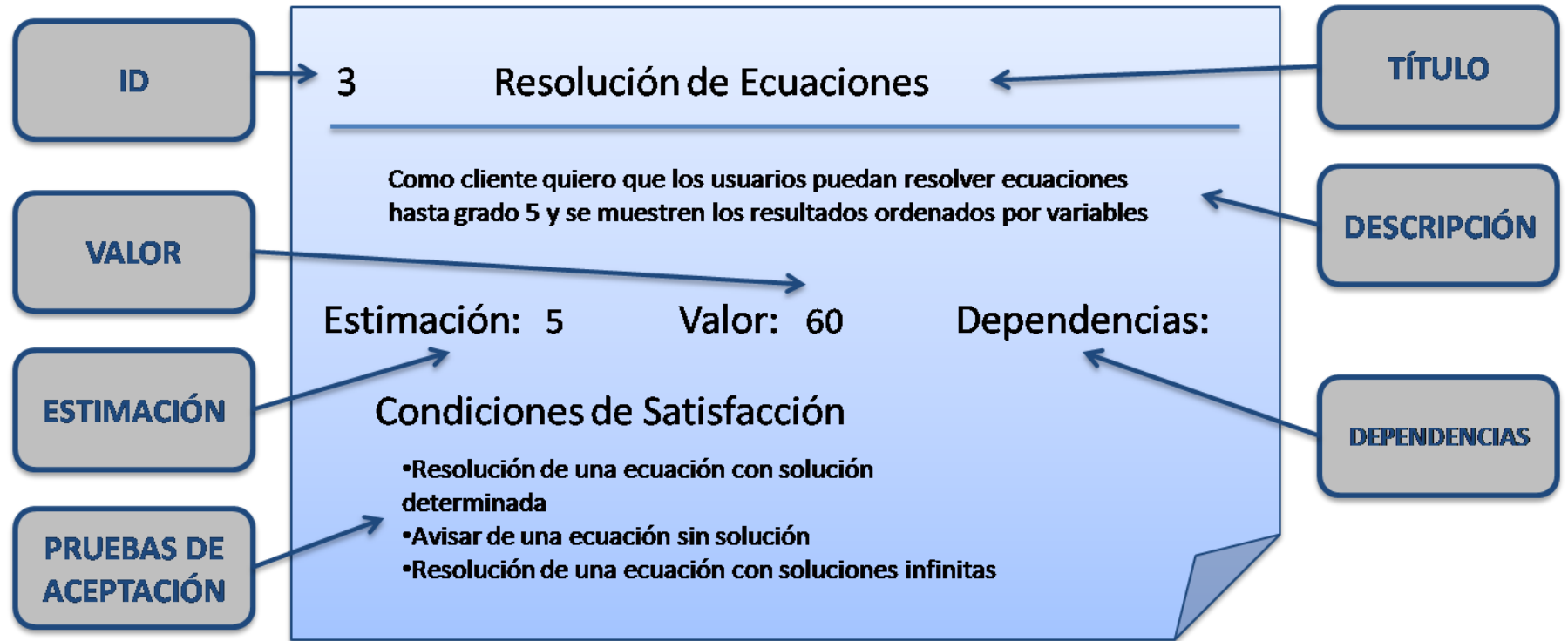
COMPONENTES FUNDAMENTALES

HISTORIAS DE USUARIO

En las metodologías ágiles la descripción de estas necesidades se realiza a partir de las historias de usuario (user story) que son, principalmente, lo que el cliente o el usuario quiere que se implemente; es decir, son una descripción breve, de una funcionalidad software tal y como la percibe el usuario (M. Cohn, 2004) .



COMPONENTES FUNDAMENTALES



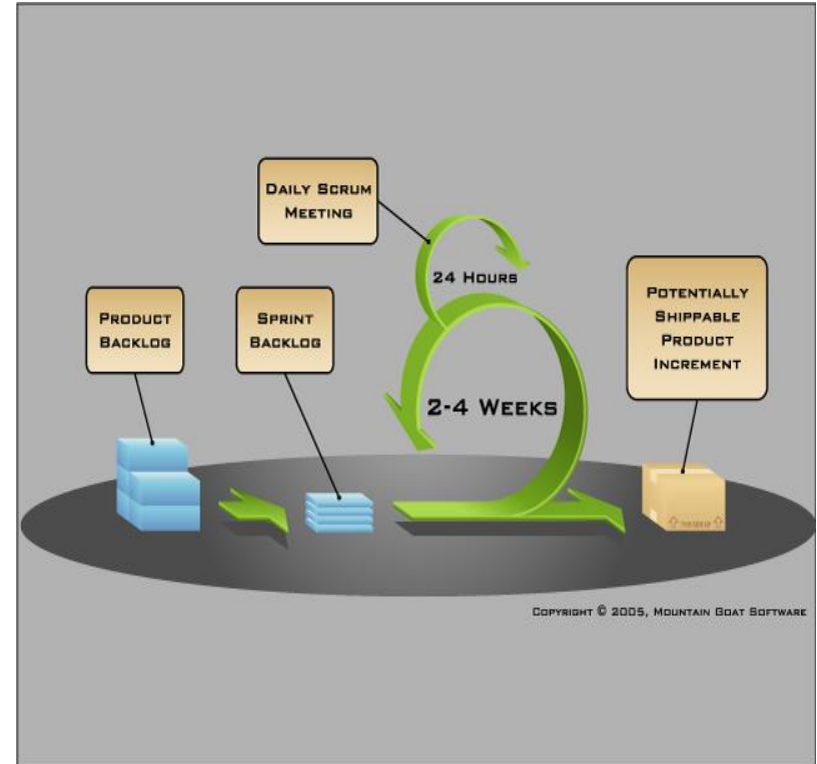
SCRUM

¿QUÉ ES?

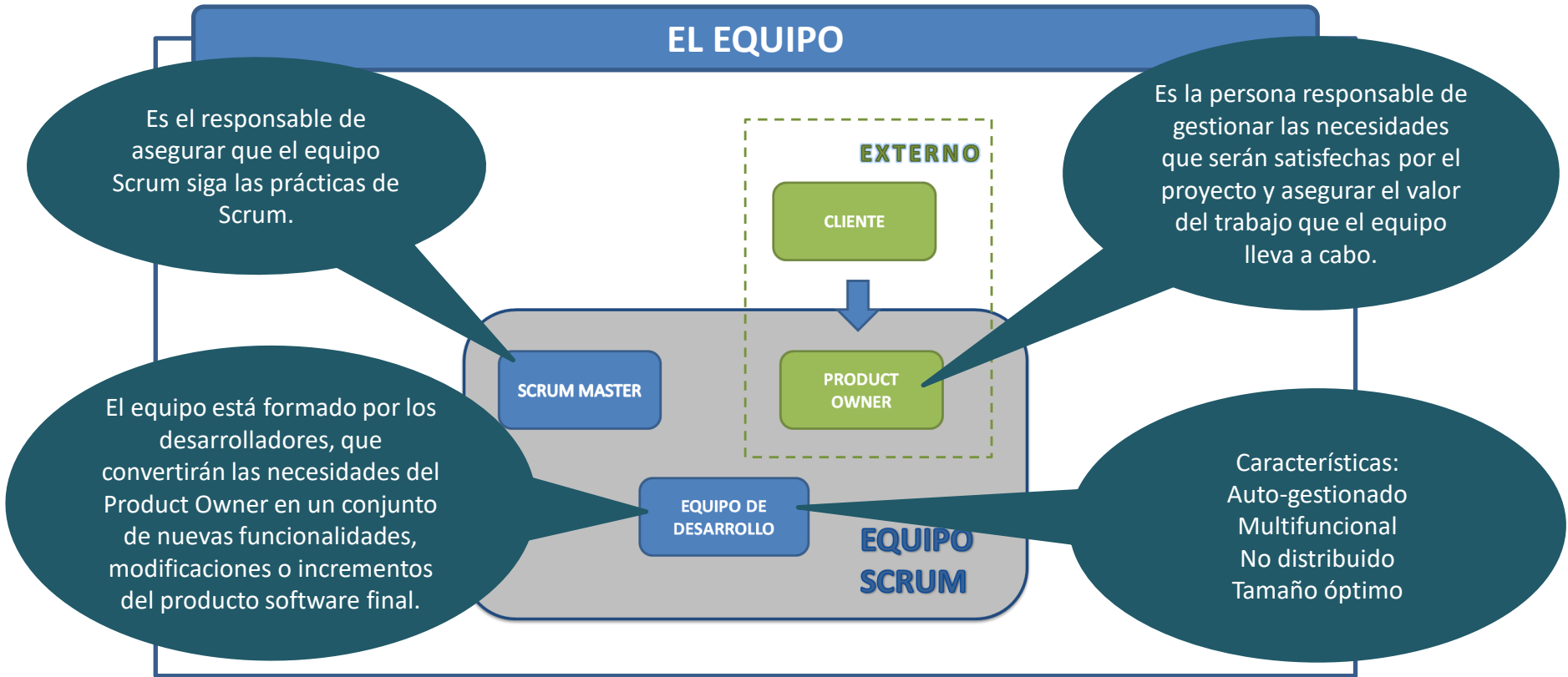
Scrum proporciona un marco para la gestión de proyectos. Podríamos decir que hoy en día es la metodología ágil más popular, y, de hecho, se ha utilizado para desarrollar productos software desde principios de la década de los 90.

El desarrollo software mediante iteraciones incrementales.

Las reuniones a lo largo del proyecto.



SCRUM



SCRUM

EL SPRING

El Product Backlog consiste en un listado de historias del usuario que se incorporarán al producto software a partir de incrementos sucesivos.



Product Backlog



Sprint Backlog



Sprint



Incremento

Historias de usuario del Spring.

En Scrum a cada iteración se le denomina Sprint. Scrum recomienda iteraciones cortas, por lo que cada Sprint durará entre 1 y 4 semanas.

SCRUM



BASES

Estrategia de fabricación aplicada con mucho éxito en Japón y ahora muy famosa en el mundo del software, muchas veces bajo el término de Lean Software Development.

Bases de lean:

- Construir sólo lo necesario.
- Eliminar todo aquello que no añade valor.
- Parar si algo no va bien (lo que está relacionado con el principio de cero defectos).

PRINCIPIOS

1. Eliminar desperdicios (eliminating waste)
2. Amplificar el aprendizaje (amplifying learning)
3. Decidir lo más tarde posible (decide as late as possible)
4. Entregar lo más rápido posible (delivering as fast as possible)
5. Capacitar y potenciar al equipo (empowering the team)
6. Construir con calidad (build quality in)
7. Ver el todo (seeing the hole)

DESPERDICIOS LEAN

Hay 7 típicos desperdicios del lean software (según los clasificaron Mary and Tom Poppendieck):

1. Desperdicios del lean software 1: Trabajo realizado parcialmente
2. Desperdicios del lean software 2: Características extra
3. Desperdicios del lean software 3: Reaprendizaje
4. Desperdicios del lean software 4: De mano en mano
5. Desperdicios del lean software 5: Las pausas
6. Desperdicios del lean software 6: Cambio de Tarea
7. Desperdicios del lean software 7: Defectos

KANBAN

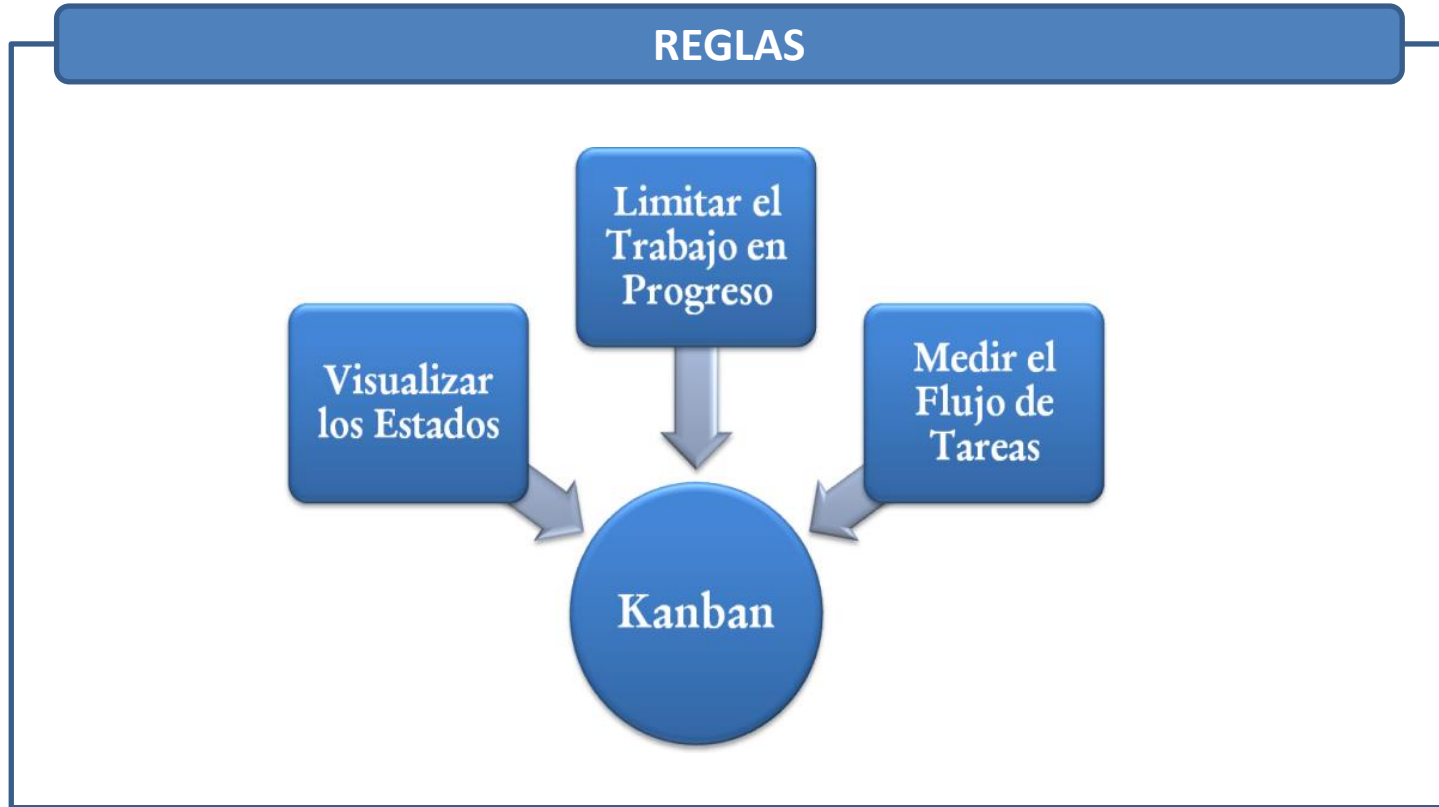
¿QUÉ ES?

Kanban es una palabra japonesa que significa "tarjetas visuales" (kan significa visual, y ban tarjeta).

Esta técnica se creó en Toyota, y se utiliza para controlar el avance del trabajo, en el contexto de una línea de producción.

Actualmente está siendo aplicado en la gestión de proyectos software.

KANBAN



KANBAN



KANBAN



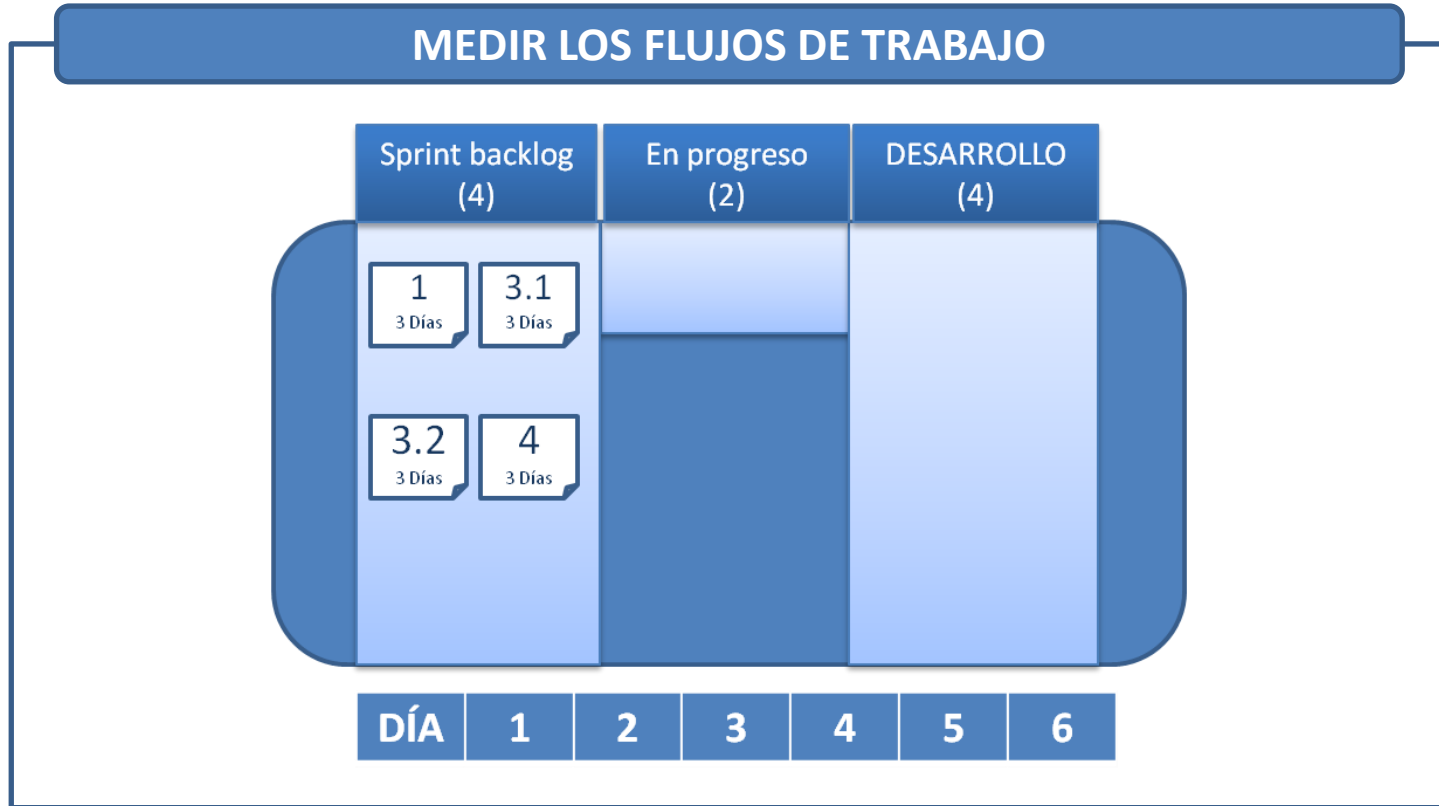
KANBAN

LIMITAR EL TRABAJO EN PROGRESO

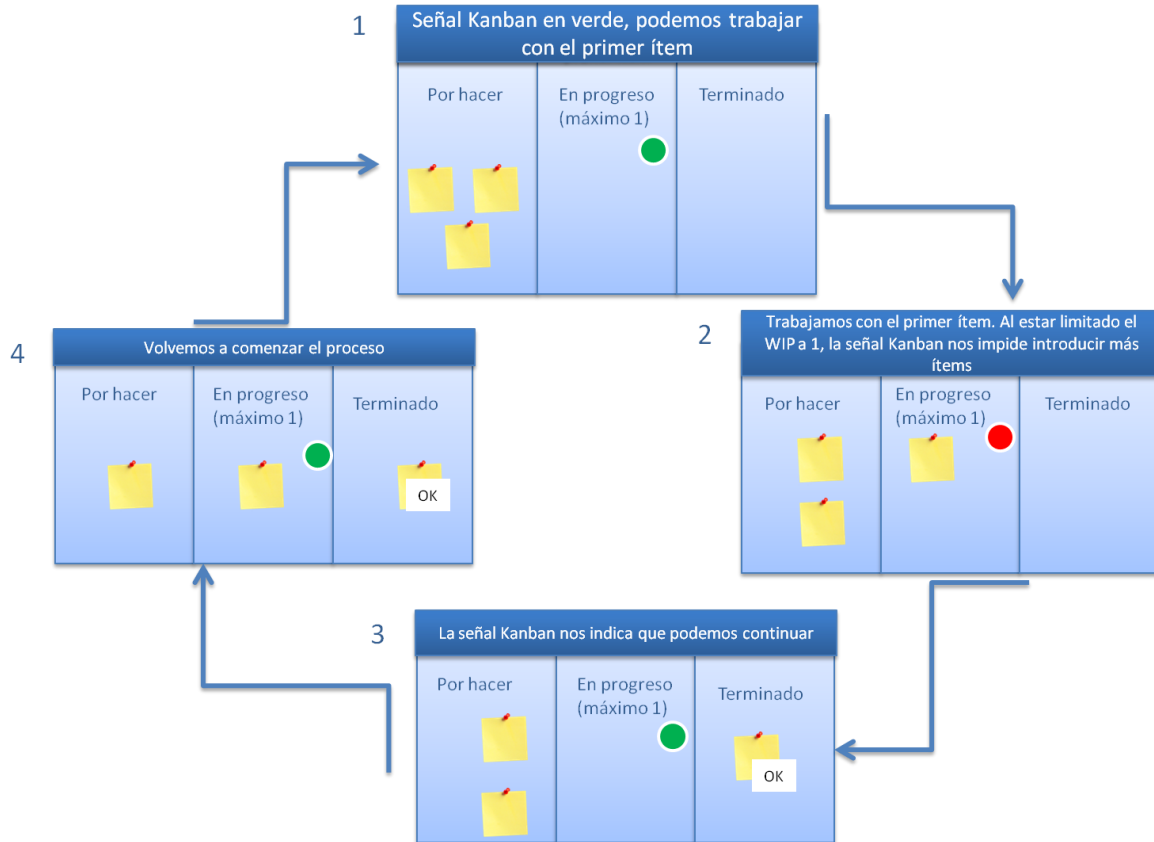
Una vez que tenemos identificados las posibles fases de las tareas de desarrollo software en el proyecto es necesario limitar el trabajo que puede realizarse en cada fase. Para ello, Kanban indica que sólo se debe empezar con algo nuevo cuando un bloque de trabajo o ítem anterior haya sido terminado.

En Kanban se limita el trabajo en progreso mediante el "WIP" (Work in Progress). Al incluir un número de WIP en alguna columna de nuestro tablero, estaremos indicando que en esa columna no se pueden añadir (y por lo tanto trabajar) más tareas de las indicadas por él.

KANBAN



KANBAN - EJEMPLO 1: FLUJO KANBAN



KANBAN - EJEMPLO 2: SPRING BACKLOG



KANBAN - OTRAS PIZARRAS KANBAN

PETICIÓN DE CORRECCIONES	SELECCIÓN DE PETICIONES	DESARROLLO	PRUEBA	INSTALACIÓN	PRODUCCIÓN

KANBAN - OTRAS PIZARRAS KANBAN

PETICIÓN DE PARCHES	SELECCIÓN DE PETICIONES	DESARROLLO		PRUEBA		INSTALACIÓN	PRODUCCIÓN
		ACTUAL	FINALIZADOS	ACTUAL	FINALIZADOS		





CORONEL

DESARROLLA SOFTWARE

DESARROLLA SOFTWARE

Gracias

Eric Gustavo Coronel Castillo

gcoronelc@gmail.com