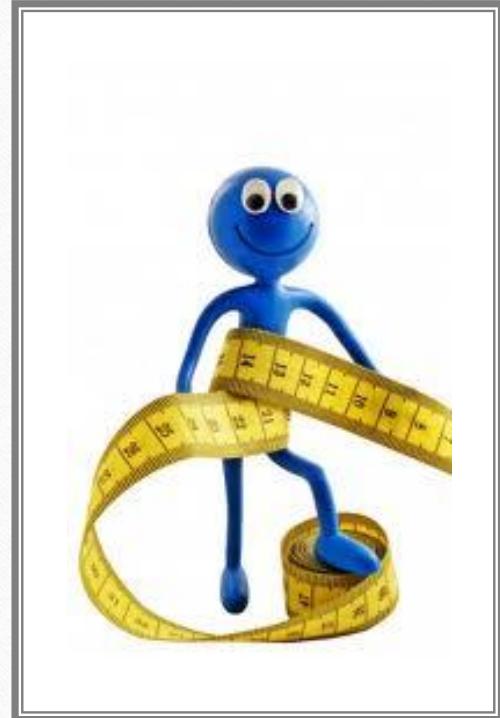


Universidad Tecnológica Nacional
Cátedra de Ingeniería de Software
Docentes: Judith Meles y Laura Covaro

Filosofía

Lean



Judith Meles

Principios Lean

Principios

Eliminar Desperdicios

Amplificar Aprendizaje

Embeber la Integridad conceptual

Diferir compromisos

Dar poder al equipo

Ver el todo

Entregar lo antes posible



Eliminar desperdicios



Amplificar aprendizaje



Tomar decisiones tardías



Entregar lo antes posible



Potenciar el equipo



Crear la integridad



Visualizar todo el conjunto

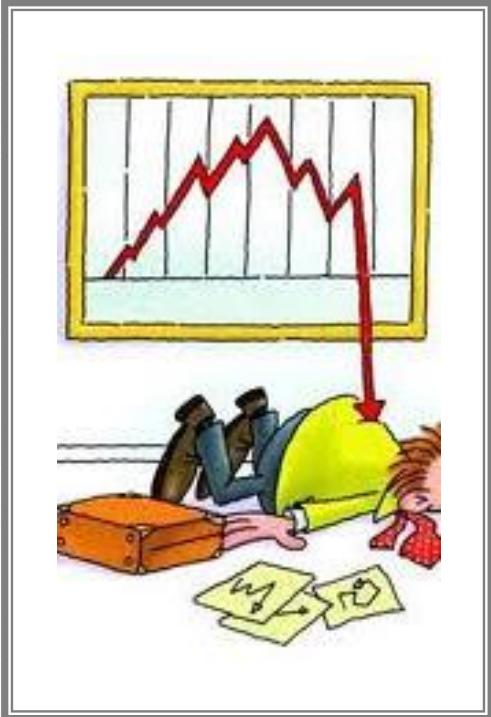
Principios Lean

- Eliminar el desperdicio: evitar que las cosas se pongan viejas antes de terminarlas o evitar re-trabajo.
 - Tiene que ver con el principio ágil de Software funcionando y el de simplicidad (arte de maximizar lo que no hacemos)

Pensamiento Lean: Principios

- Eliminar Desperdicios: reducir el tiempo removiendo lo que no agrega valor.
 - Desperdicio es cualquier cosa que interfiere con darle al cliente lo que el valora en tiempo y lugar donde le provea más valor.
 - En manufactura: el inventario
 - En Software: es el trabajo parcialmente hecho y las características extra!
 - El 20 % del software que entregamos contiene el 80% del valor

Los siete desperdicios Lean (en software)



- Características extra
- Trabajo a medias
- Proceso extra
- Movimiento
- Defectos
- Esperas
- Cambio de Tareas

Pensamiento Lean: Siete Principios

- Amplificar el aprendizaje: crear y mantener una cultura de mejoramiento continuo y solución de problemas.
 - Un proceso focalizado en crear conocimiento esperará que el diseño evolucione durante la codificación y no perderá tiempo definiéndolo en forma completa, prematuramente.
 - Se debe generar nuevo conocimiento y codificarlo de manera tal que sea accesible a toda la organización.
 - Muchas veces los procesos “estándares” hacen difícil introducir en ellos mejoras.

Principios Lean

- Embeber la integridad conceptual. Encastrar todas las partes del producto o servicio, que tenga coherencia y consistencia (tiene que ver con los Requerimientos No Funcionales). La integración entre las personas hace el producto más integro.

PENSAMIENTO LEAN: SIETE PRINCIPIOS

- Embeber la integridad conceptual: se necesita más disciplina no menos!
 - Integridad Percibida: el producto total tiene un balance entre función, uso, confiabilidad y economía que le gusta a la gente.
 - Integridad Conceptual: todos los componentes del sistema trabajan en forma coherente en conjunto.
 - El objetivo es construir con calidad desde el principio, no probar después.
 - Dos clases de inspecciones:
 - Inspecciones luego de que los defectos ocurren.
 - Inspecciones para prevenir defectos.
 - Si se quiere calidad no inspeccione después de los hechos!
 - Si no es posible, inspeccione luego de pasos pequeños.

Principios Lean

- Diferir Compromisos. El último momento responsable para tomar decisiones (en el cual todavía estamos a tiempo). Si nos anticipamos tenemos información parcial.
 - Se relaciona con el principio ágil: decidir lo más tarde posible pero responsablemente. No hacer trabajo que no va a ser utilizado. Enlaza con el principio anterior de aprendizaje continuo, mientras más tarde decidimos más conocimiento tenemos.

Pensamiento Lean

- Diferir compromisos: las decisiones deben tomarse en el último momento que sea posible.
 - No significa que todas las decisiones deben diferirse.
 - Se debe tratar de tomar decisiones reversibles, de forma tal que pueda ser fácilmente modificable.
 - Vencer la “parálisis del análisis” para obtener algo concreto terminado.
 - Las mejores estrategias de diseño de software están basadas en dejar abiertas opciones de forma tal que las decisiones irreversibles se tomen lo más tarde posible.

Principios Lean

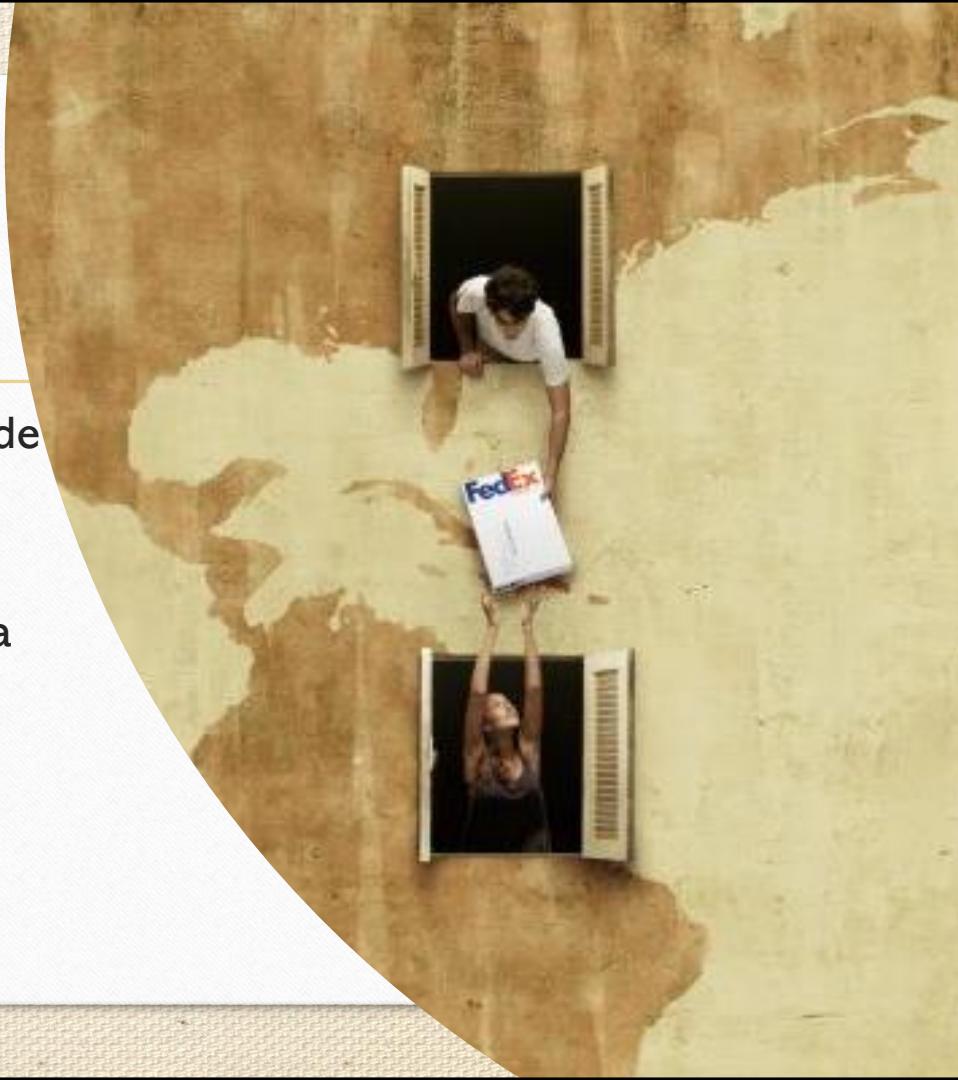
- Dar poder al equipo: ejemplo, vamos a comer a un restaurante y no nos metemos en la cocina del restaurante. Nos fijamos en el precio, pedimos y esperamos. Hay mucho micromanagement, el dueño no decide cuánta sal poner a la comida.
- Respetar a la gente
 - Entrenar líderes
 - Fomentar buena ética laboral
 - Delegar decisiones y responsabilidades del producto en desarrollo al nivel más bajo posible
- Ágil: El propio equipo pueda estimar el trabajo.

Principios Lean

- Ver el todo: tener una visión holística, de conjunto (el producto, el valor agregado que hay detrás, el servicio que tiene los productos como complemento).

Principios Lean

- **Entregar rápido:** estabilizar ambientes de trabajo a su capacidad más eficiente y acotar los ciclos de desarrollo.
- **Entregar rápidamente** esto hace que se vayan transformando “n” veces en cada iteración. Incrementos pequeños de valor. Llegar al producto mínimo que sea valioso. Salir pronto al mercado.
 - Relacionado con el principio Ágil de entrega frecuente.



KANBAN

Cambio Evolutivo Exitoso Para su Negocio de Tecnología



David J. Anderson

Prologo de Donald G. Reinertsen
Traducción de Masa Kevin Maeda

En desarrollo de software el referente principal es...

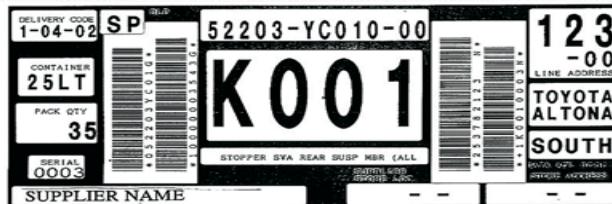
KANBAN

A vibrant, stylized illustration depicting a team working on a Kanban board. Several hands are shown placing and moving colorful sticky notes (yellow, pink, blue, green) on a grid. Some notes have simple icons like a person or a checkmark. The background features abstract wavy lines and a large, faint smiley face. The overall theme is collaboration and productivity.

KANBAN

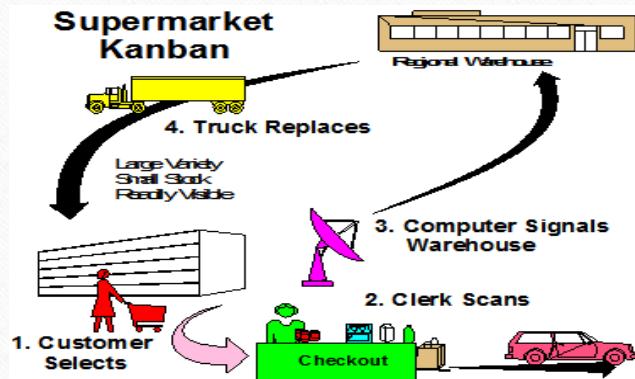
Kanban en pocas palabras

- **kan-ban** (看板) = Signal-card.



Kanban en pocas palabras-Just in Time

- A fines de 1940, Toyota comenzó a estudiar técnicas de almacenamiento y tiempo de stockeo de los supermercados



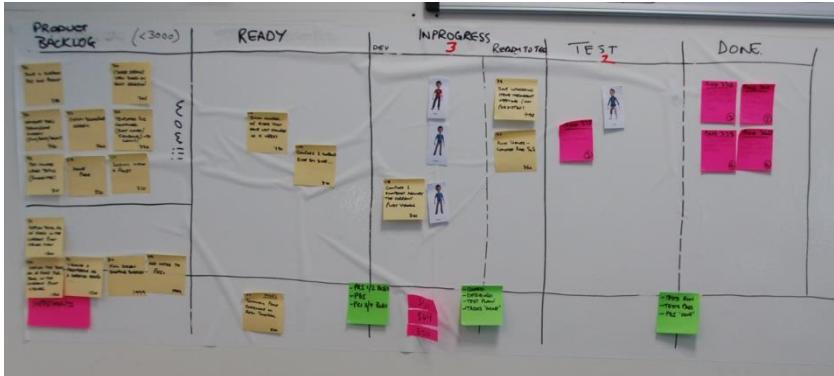
Taiichi Ohno

Kanban en pocas palabras: Administración de Colas

- Los cajeros se focalizan en tomar órdenes.
- El Barista se focaliza en proveer café.
- Separarlos por la cola permite que se absorba la demanda variable.
- Los cajeros se mueven a ayudar al Barista cuando no hay clientes esperando para hacer su pedido.
- **Foco es en Flujo “fin a fin” FLOW = Centrado en el Cliente**



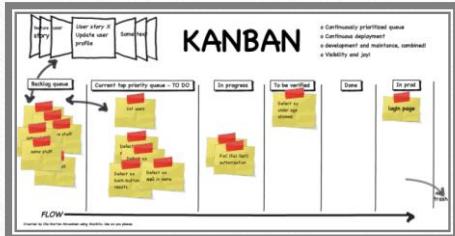
Kanban en pocas palabras



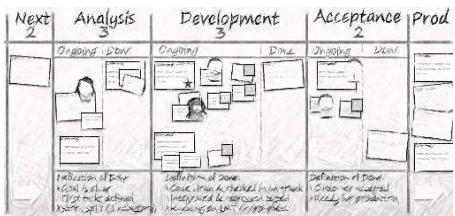
- Principios:
 - Visualizar el Flujo: Hacer el trabajo visible.
 - Limitar el Trabajo en progreso (WIP)
 - Administrar el flujo: Ayudar a que el trabajo fluya
 - Hacer explícitas las políticas.
 - Mejorar colaborativamente.

Kanban en el Desarrollo de Software

- El método fue formulado por David J. Anderson
- Es un enfoque para gestión de cambio.
- No es un proceso de desarrollo de software o una metodología de administración de proyecto.
- Kanban es un método para introducir cambios en un proceso de desarrollo de software o una metodología de administración de proyectos

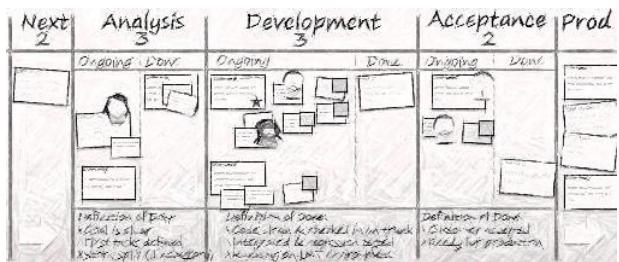


Kanban en el Desarrollo de Software



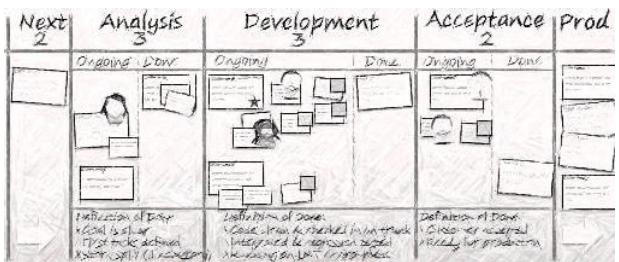
- Kanban aprovecha muchos de los conceptos probados de Lean:
 - Definiendo el Valor desde la perspectiva del Cliente.
 - Limitando el Trabajo en Progreso (WIP).
 - Identificando y Eliminando el Desperdicio.
 - Identificando y removiendo las barreras en el Flujo.
 - Cultura de Mejora Continua.

Kanban en el Desarrollo de Software



- Kanban fomenta la evolución gradual de los procesos existentes.
- Kanban no pide una revolución, sino que fomenta el cambio gradual.
- Kanban está basado en una idea muy simple: Limitar el trabajo en progreso (WIP).
- El Kanban (o tarjeta de señal) implica que una señal visual se produce para indicar que el nuevo trabajo se puede tirar (“pull”) porque el trabajo actual no es igual al límite acordado.

¿Cómo aplicar Kanban?



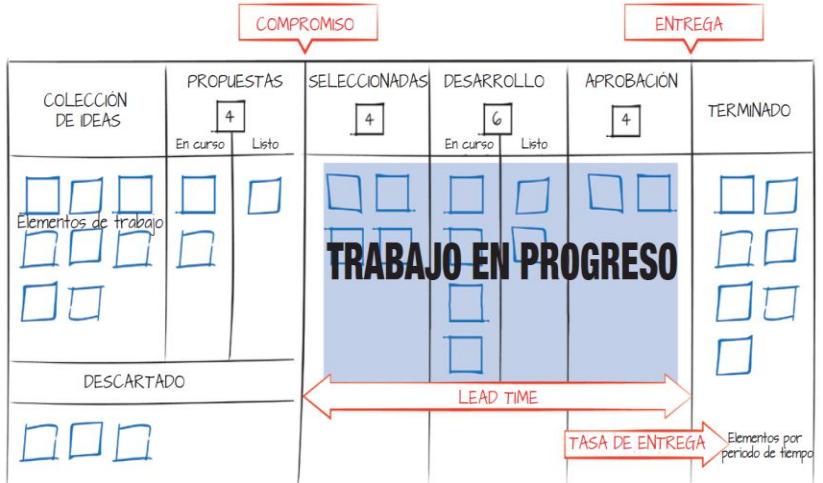
- Empezar con lo que se tiene ahora.
- Entender el proceso actual.
- Acordar los límites de WIP para cada etapa del proceso.
- A continuación, comienza a fluir el trabajo a través del sistema tirando de él, en presencia de señales Kanban.

¿Cómo aplicar Kanban?

- Visualizar el flujo de trabajo:
 - Dividir el trabajo en piezas, las user stories son buenas para eso.



¿Cómo aplicar Kanban?

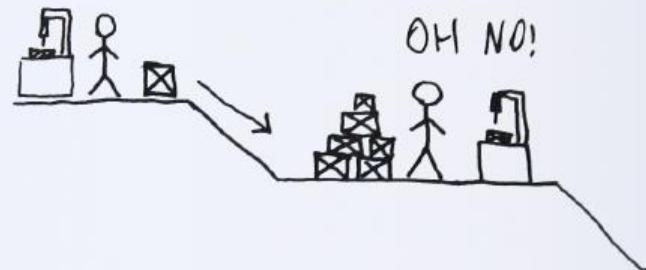


- Visualizar el flujo de trabajo:
 - Utilizar nombres en las columnas para ilustrar donde está cada ítem en el flujo de trabajo.
 - Distribuir el trabajo en las columnas: el trabajo fluirá de izquierda a derecha en las columnas.

Pull



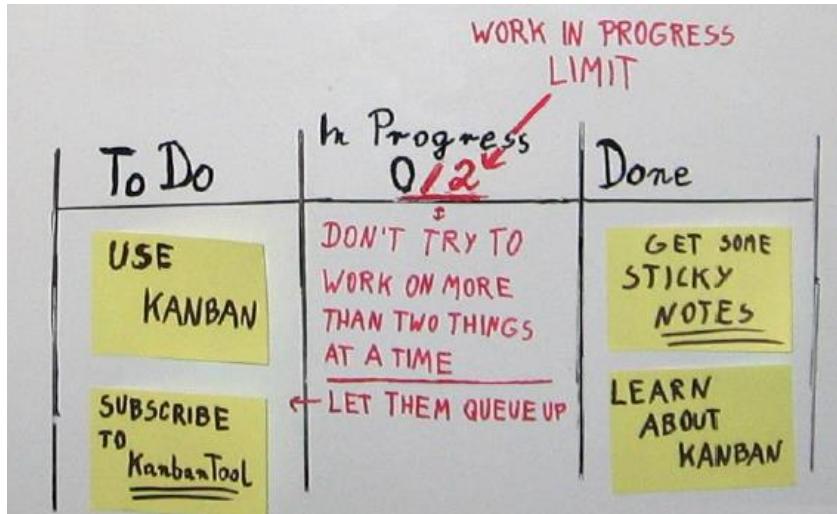
Push



Pull, no push !!!

¿Cómo aplicar Kanban?

- Limitar WIP – Asignar límites explícitos de cuántos ítems puede haber en progreso en cada estado del flujo de trabajo.



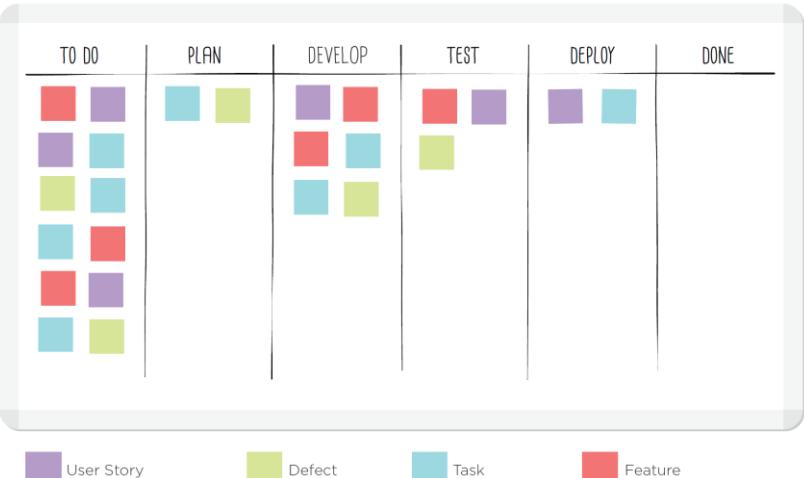


Ayudar a que el trabajo fluya....

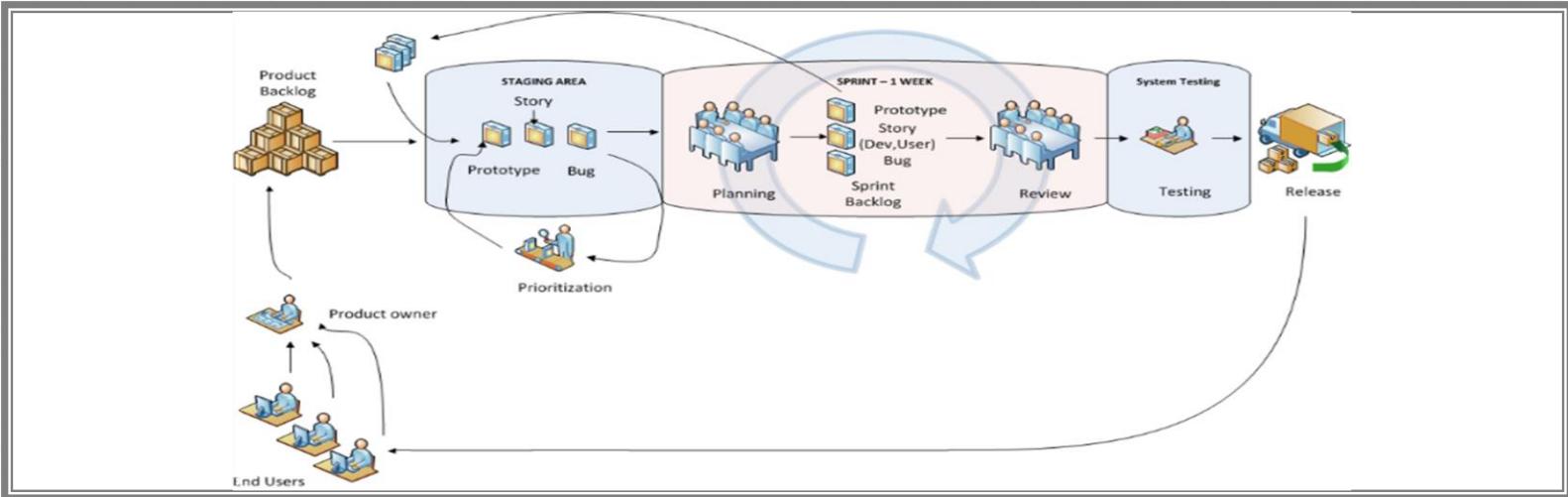
Al 100 % de capacidad se tiene un rendimiento mínimo...

Por último, la auto asignación de tareas se ve reflejada con un avatar personalizado... :))

¿Cómo aplicar Kanban en nuestro proyecto?

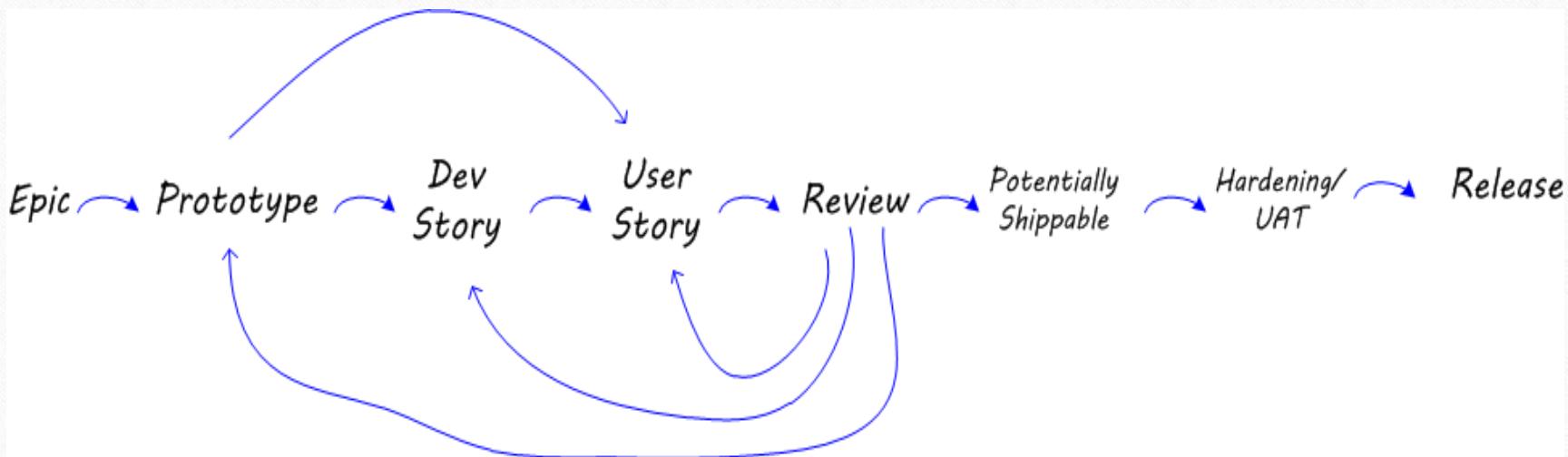


- **Proceso:** modelar nuestro proceso.
- **Trabajo :** decidir la unidad de trabajo.
- **Límites de WIP:** limitar el WIP para ayudar al flujo de trabajo.
- **Política:** definir políticas de calidad.
- **Cuellos de Botella y Flujo:** mover recursos a los cuellos de botella.
- **Clase de Servicio:** diferentes trabajos tienen diferentes políticas – definición de hecho (“done”), para cada estado.
- **Cadencia:** Releases, planificaciones, revisiones



Modelar el proceso

Cadena de Valor



Definir el proceso...

Definir tipos de trabajo...

Asignando
capacidad en
función de
la demanda

Requerimientos

- Caso de uso
- Historias de Usuario
- Porciones de Casos de Uso
- Características

Defectos

- Defectos en Producción
- Defectos

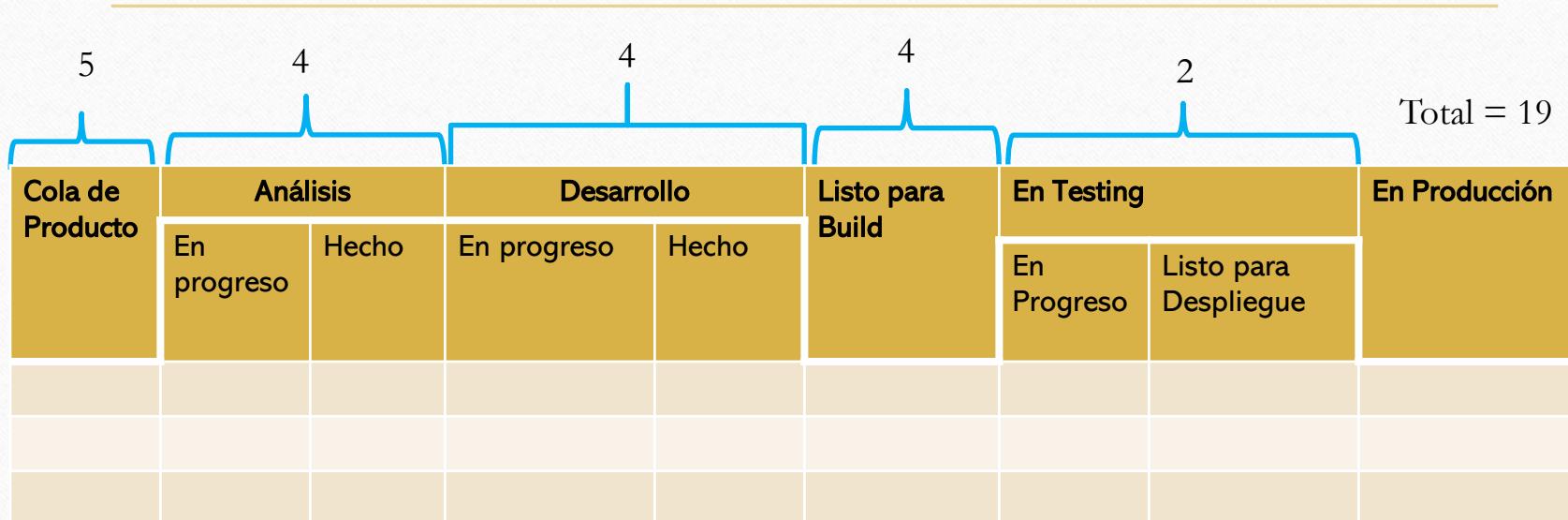
Desarrollo

- Mantenimiento
- Refactorización
- Actualización de Infraestructura

Solicitudes

- Solicitud de Cambio
- Sugerencias de Mejora

Definir el WIP...



Definir tipos de trabajo...

Asignando capacidad en función de la demanda

Cola de Producto	Análisis		Desarrollo		Listo para Build	En Testing		En Producción
	En progreso	Hecho	En progreso	Hecho		En Progreso	Listo para Despliegue	

Políticas Explícitas para cada clase de servicio

Políticas para la clase de servicio expreso, un ejemplo



Color de tarjeta:
blanco

WIP = 1

Los demás trabajos se ponen en espera

Se puede exceder el límite de WIP para procesar este trabajo

La capacidad no se reserva

De ser necesario se hace una entrega especial, para ponerla en producción

Políticas Explícitas para cada clase de servicio

Políticas para la clase de servicio “Fecha Fija”, un ejemplo

Color de tarjeta:
Rosa

Deben
adherirse al
WIP
definido

Fecha de
entrega en
la parte
superior

Permanecen
la cola hasta
que sea
conveniente
que ingresen

Si se retrasa
y la fecha de
entrega está
en riesgo
puede
promoverse a
la clase de
servicio
“expreso”

Son
entregados en
entregas
programadas
cuidando la
fecha de
entrega



Políticas Explícitas para cada clase de servicio

Políticas para la clase de servicio “Estándar”, un ejemplo

Color de tarjeta:
Amarillo

Deben
adherirse
al WIP
definido

Son
priorizados
y puestos
en la cola
con un
mecanismo
definido
basado en
valor de
negocio

Usan la
técnica FIFO,
si no hay
Expresos o
con Fecha
Fija

Pueden
analizarse
por tamaño,
en orden de
magnitud

Son
entregados
en entregas
programadas

Estándar

Políticas Explícitas para cada clase de servicio

Políticas para la clase de servicio “Intangible”, un ejemplo

Color de tarjeta: verde

Deben adherirse al WIP definido

Son priorizados y puestos en la cola con un mecanismo definido basado en valor de negocio

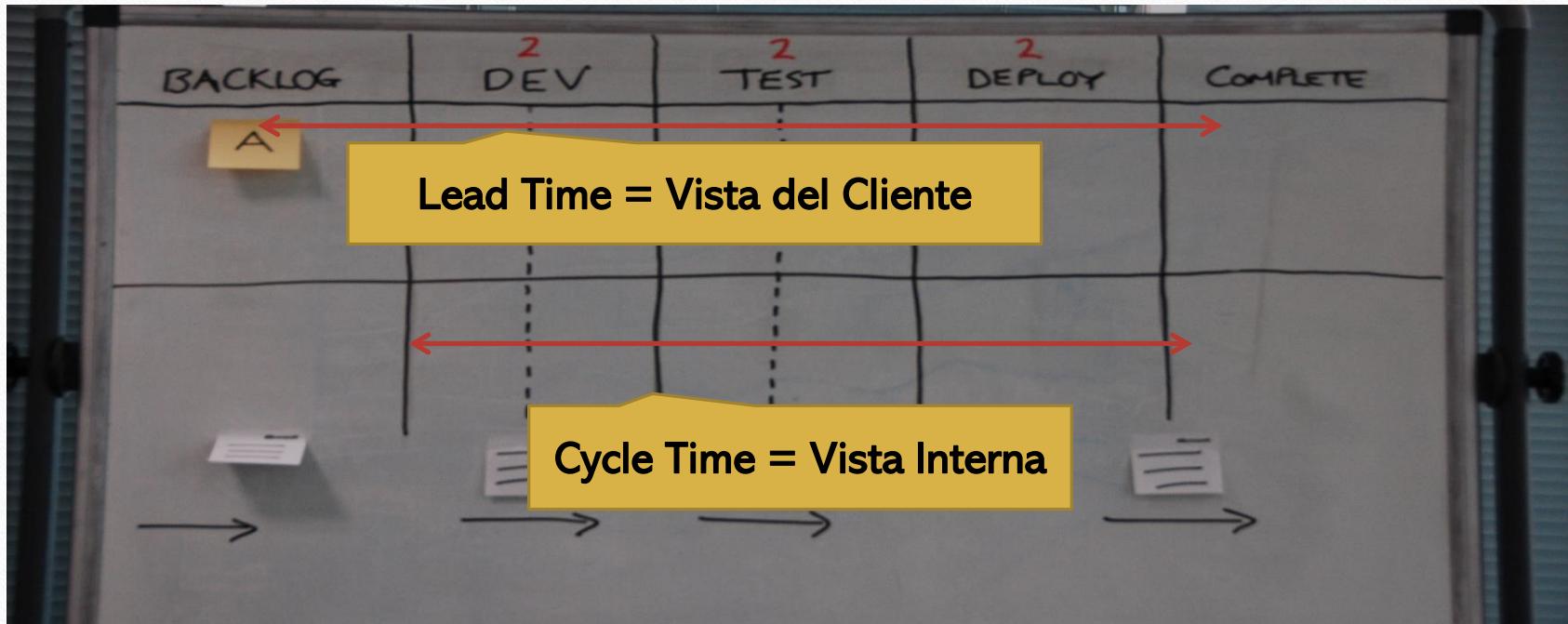
Usan la técnica FIFO, si no hay Expresos o con Fecha Fija

Pueden analizarse por tamaño, en orden de magnitud

Son entregados en entregas programadas



KANBAN: Métricas Clave



Cycle Time (Tiempo de ciclo)

- Es la métrica que registra el tiempo que sucede entre el inicio y el final del proceso, para un ítem de trabajo dado. Se suele medir en días de trabajo o esfuerzo.
- Medición más mecánica de la capacidad del proceso
- **Ritmo de Terminación**

Lead Time (Tiempo de entrega)

- Es la métrica que registra el tiempo que sucede entre el momento en el cual se está pidiendo un ítem de trabajo y el momento de su entrega (el final del proceso). Se suele medir en días de trabajo.
- **Ritmo de entrega**

Touch Time (Tiempo de Tocado)

- El tiempo en el cual un ítem de trabajo fue realmente trabajado (o "tocado") por el equipo.
- Cuántos días hábiles pasó este ítem en columnas de "trabajo en curso", en oposición con columnas de cola / buffer y estado bloqueado o sin trabajo del equipo sobre el mismo.

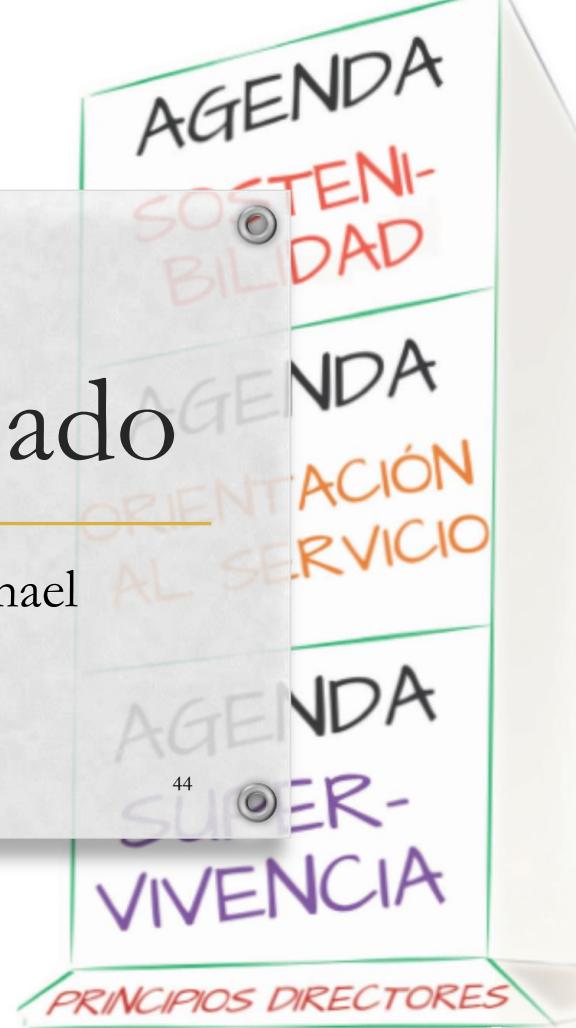
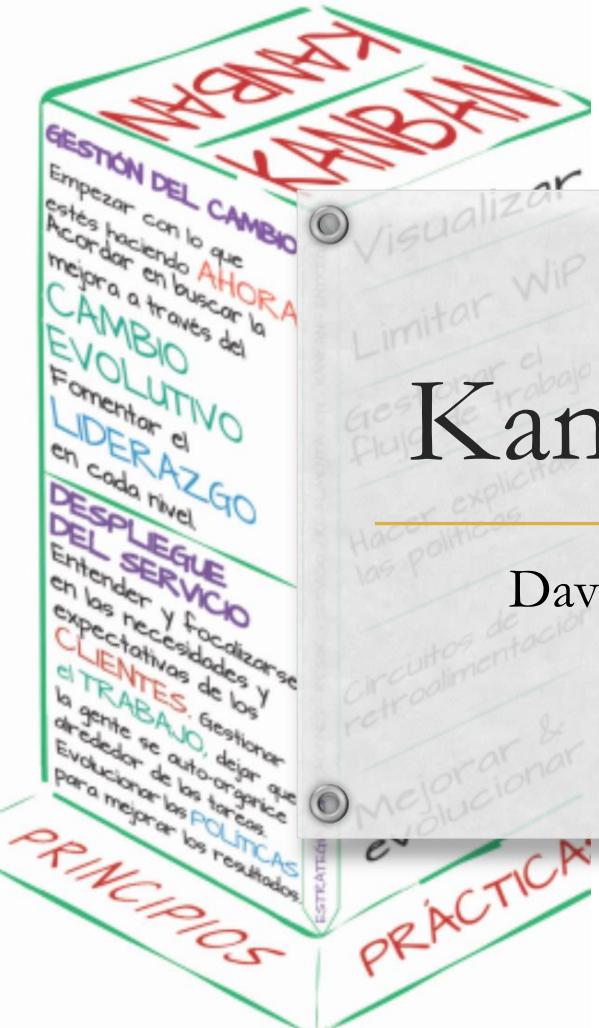
Touch Time ≤ Cycle Time ≤ Lead Time

Eficiencia del Ciclo de Proceso

% Eficiencia ciclo proceso = Touch Time / Elapsed Time.

Kanban Condensado

David Anderson & Andy Carmichael



Kanban Principios Rectores

David Anderson &
Andy Carmichael



Sostenibilidad



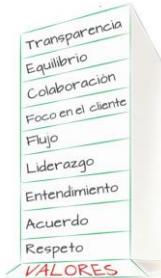
Orientación al
Servicio



Supervivencia

Kanban Valores

David Anderson &
Andy Carmichael



Transparencia



Colaboración



Equilibrio



Foco en el Cliente



Flujo



Liderazgo



Entendimiento



Respeto

Kanban Prácticas

David Anderson &
Andy Carmichael



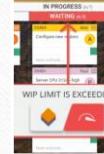
Visualizar



Gestionar el flujo
de trabajo



Circuitos de
retroalimentación



Limitar WIP



Hacer explícitas
las políticas



Mejorar &
Evolucionar

Kanban Principios

David Anderson &
Andy Carmichael



Gestión del Cambio

- Empezar con lo que tengas AHORA
- Buscar mejora a través del Cambio Evolutivo
- Fomentar el Liderazgo en cada nivel

Despliegue del Servicio

- Entender y focalizarse en las necesidades y expectativas de los clientes.
- Gestionar el trabajo, dejar que la gente se autoorganice alrededor de las tareas
- Evolucionar las políticas para mejorar los resultados