

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

# Filosofía

---

# Lean



Judith Meles

# Principios

Eliminar Desperdicios

Amplificar Aprendizaje

Embeber la Integridad  
conceptual

Diferir compromisos

Dar poder al equipo

Ver el todo

Respetar a la Gente

## Principios Lean



# Gastos en producción Lean



PRODUCCIÓN  
EN EXCESO



STOCK



PASOS EXTRA  
EN EL PROCESO



BÚSQUEDA DE  
INFORMACIÓN



DEFECTOS



ESPERAS



TRANSPORTES





# Los siete desperdicios Lean (en software)

---

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

An illustration of a Kanban board. The board is a green chalkboard with a white grid. Several colorful sticky notes (yellow, blue, pink, green) are attached to the board. Hands are shown interacting with the board: one hand is writing on a green note with a red marker, another is pointing at a yellow note, and others are moving blue and yellow notes. The word "KANBAN" is written in large white letters across the center of the board.

# KANBAN

# Kanban en pocas palabras

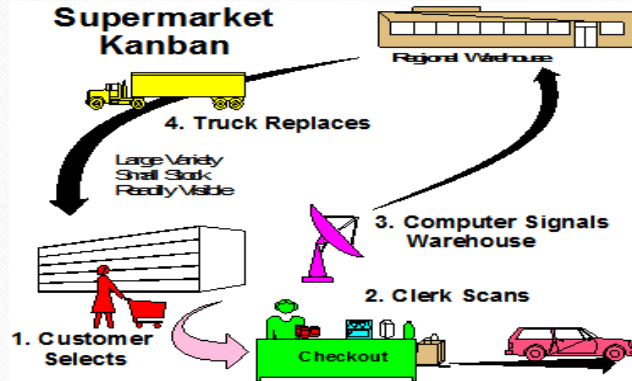
- **kan-ban** (看板) = Signal-card.



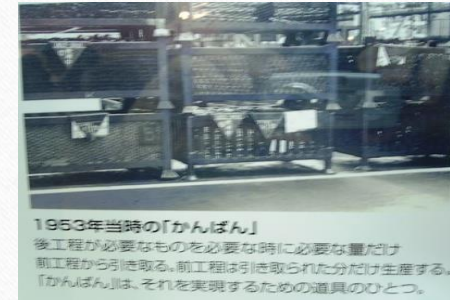


# Kanban en pocas palabras-Just in Time

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



Taiichi Ohno



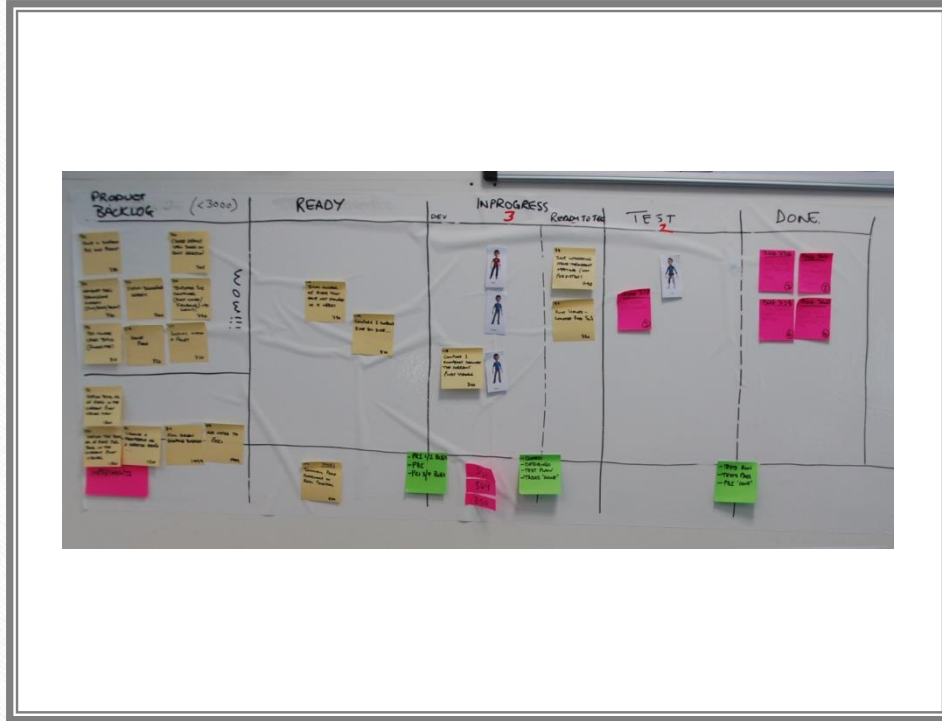
# Kanban en pocas palabras:

## 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**



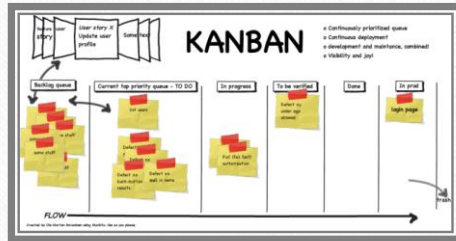




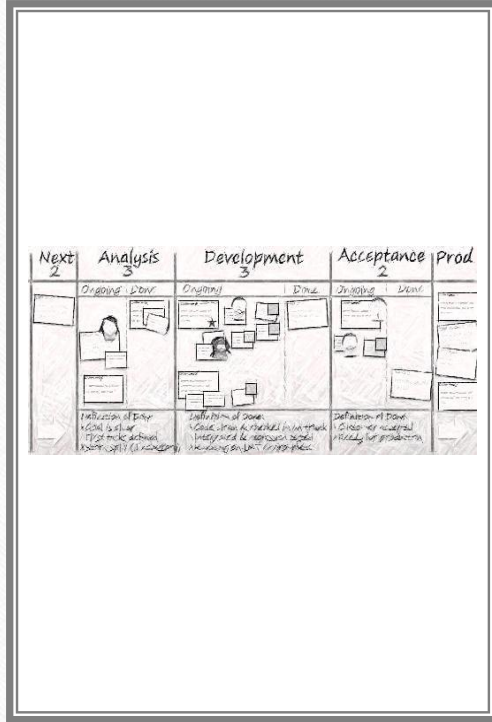
## Kanban en pocas palabras

- 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



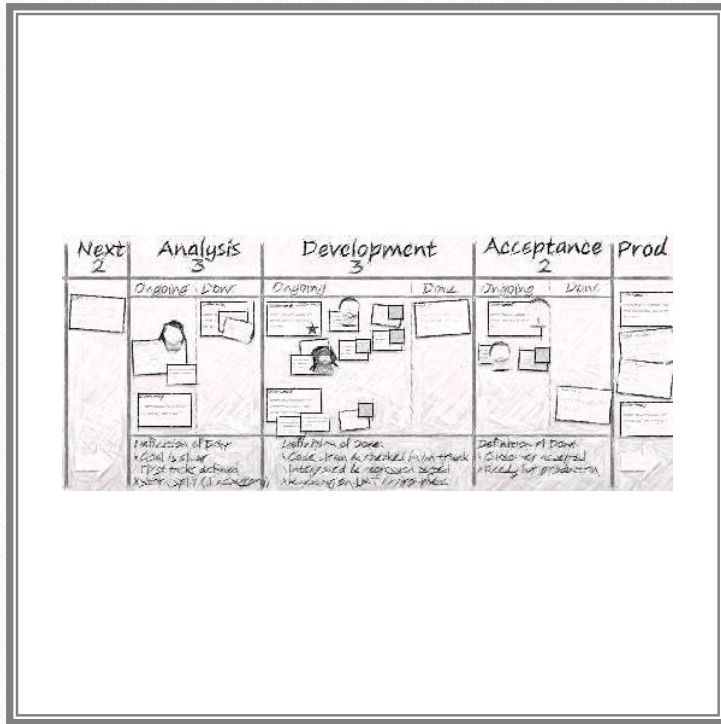
- El método fue formulado por David J. Anderson
- Es un enfoque para gestión de cambio.
- No es un proceso de desarrollo de software o una metodología de administración de proyecto.
- Kanban es un método para introducir cambios en un proceso de desarrollo de software o una metodología de administración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

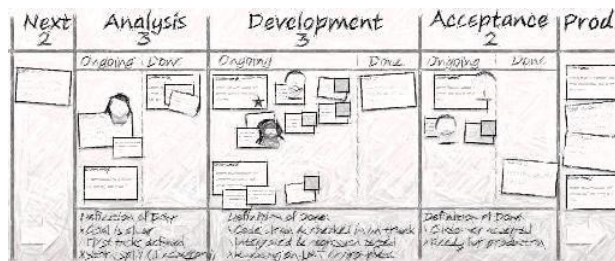




# Kanban en el Desarrollo de Software

- 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?

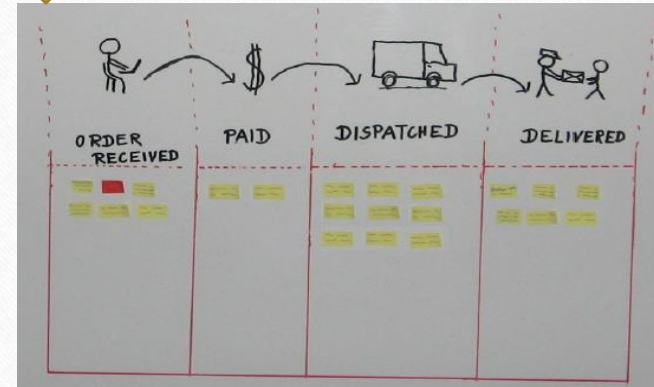
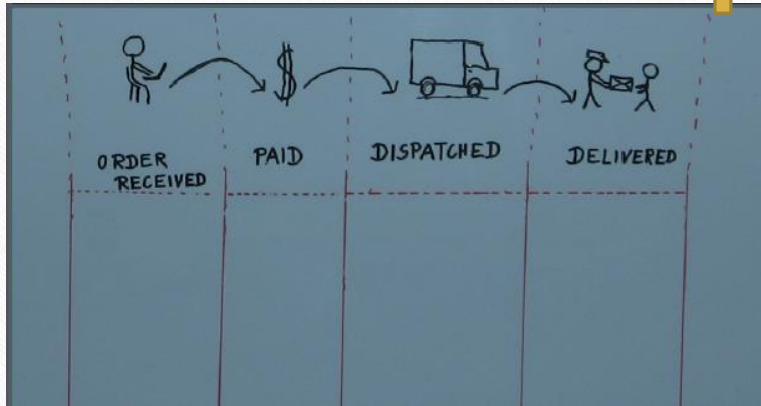
- Visualizar el flujo de trabajo:
  - Dividir el trabajo en piezas, las user stories son buenas para eso.

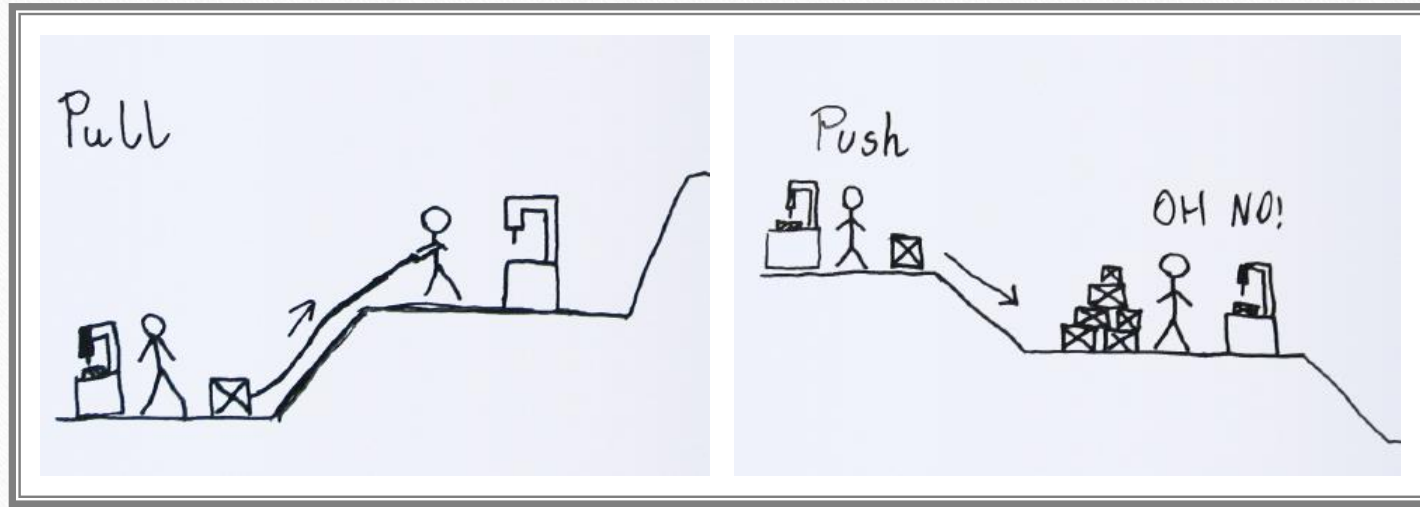




# ¿Cómo aplicar Kanban?

- 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, no push !!!

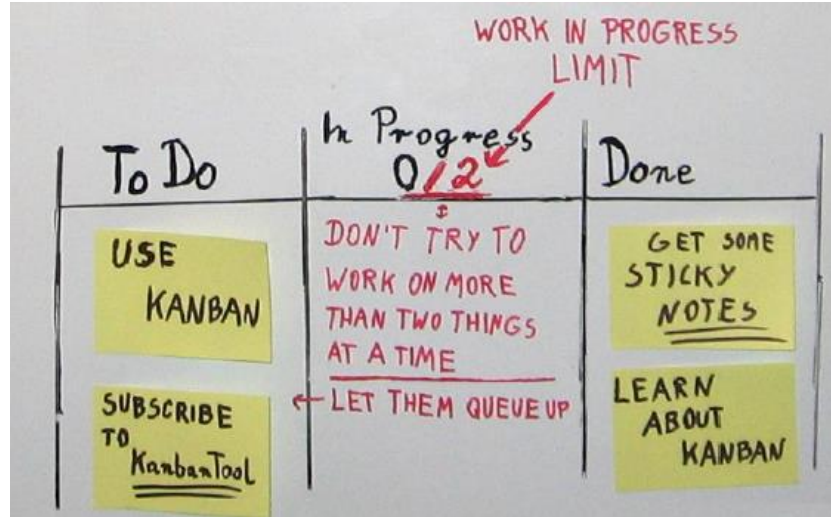
---



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







## ¿Cómo aplicar Kanban?

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



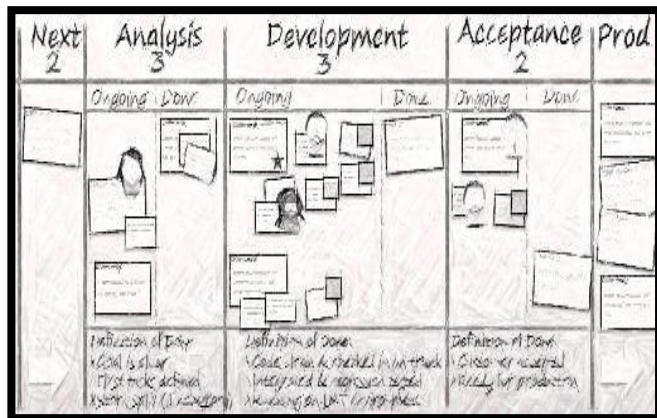
# Ayudar a que el trabajo fluya....

---

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

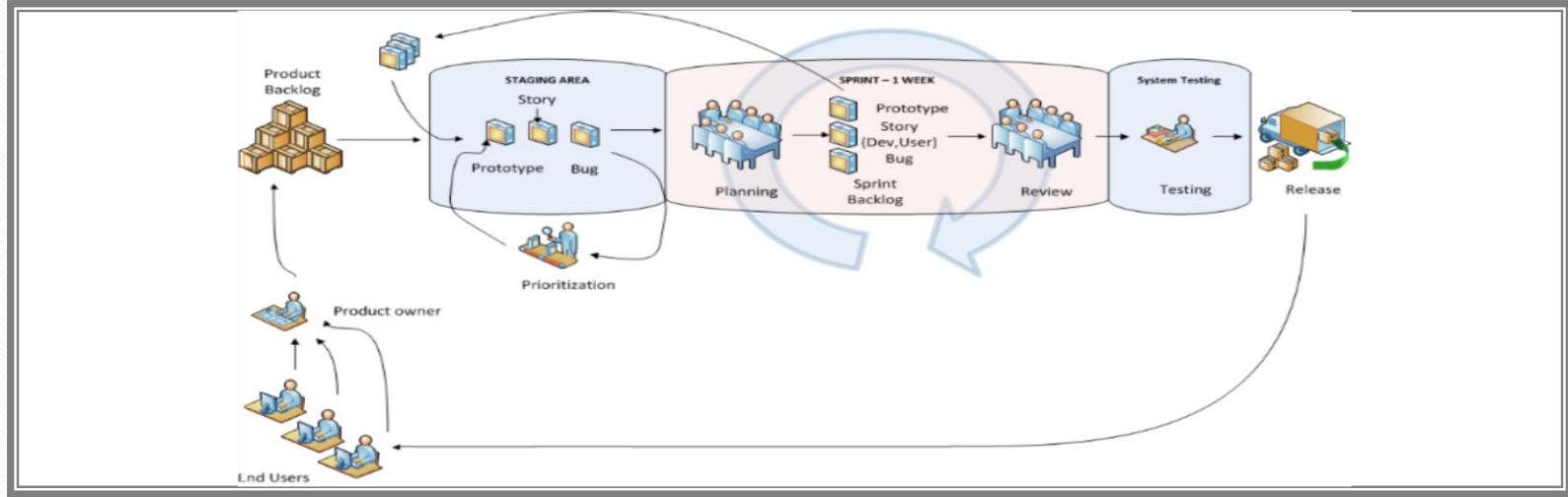


# ¿Cómo aplicar Kanban en nuestro proyecto?



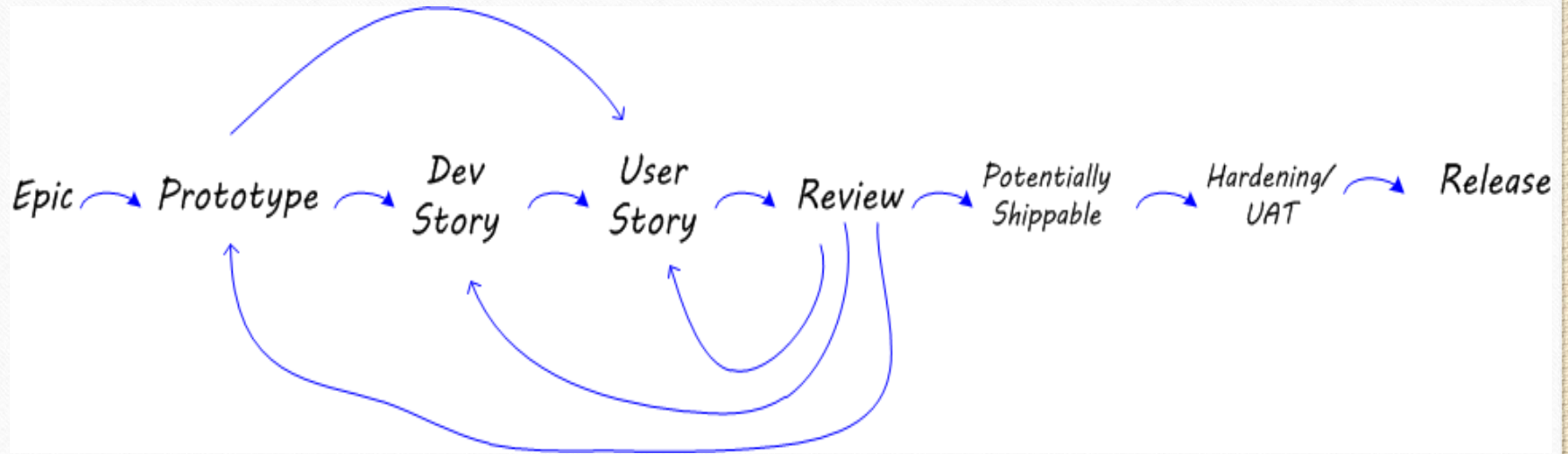
- **Proceso:** modelar nuestro proceso.
- **Trabajo :** decidir la unidad de trabajo.
- **Límites de WIP:** limitar el WIP para ayudar al flujo de trabajo.
- **Política:** definir políticas de calidad.
- **Cuellos de Botella y Flujo:** mover recursos a los cuellos de botella.
- **Clase de Servicio:**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





# Modelar el proceso

# Cádena de Valor



# Definir el proceso...

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



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

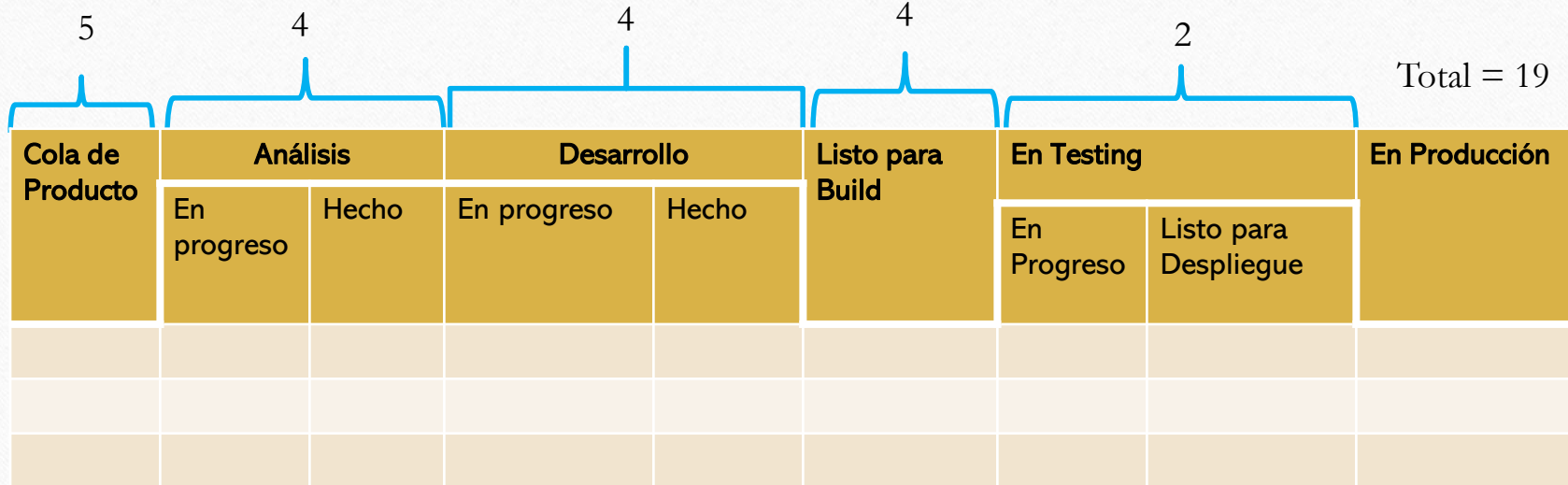
## Desarrollo

- 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	
 Casos de Uso 60 %								
 Mantenimiento 30 %								
 Defectos 10%								



# Políticas Explícitas para cada clase de servicio

## Políticas para la clase de servicio expreso, un ejemplo



**Expreso**

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  
produc-  
ción

# 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

**Fecha  
Fija**

# 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

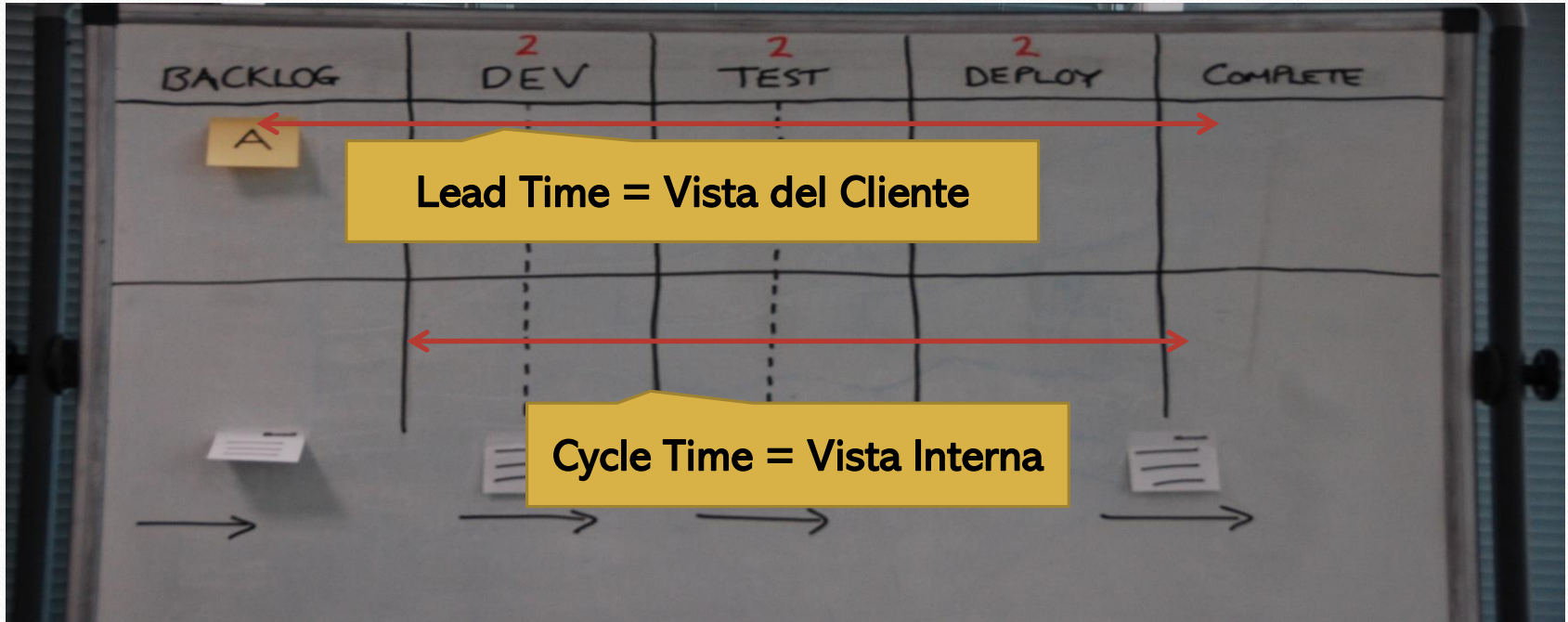
Usan la técnica FIFO, si no hay Expresos o con Fecha Fija

Pueden analizarse por tamaño, en orden de magnitud

Son entregados en entregas programadas

**Intangible**

# KANBAN: Métricas Clave



# 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**



# KANBAN: Métricas Clave

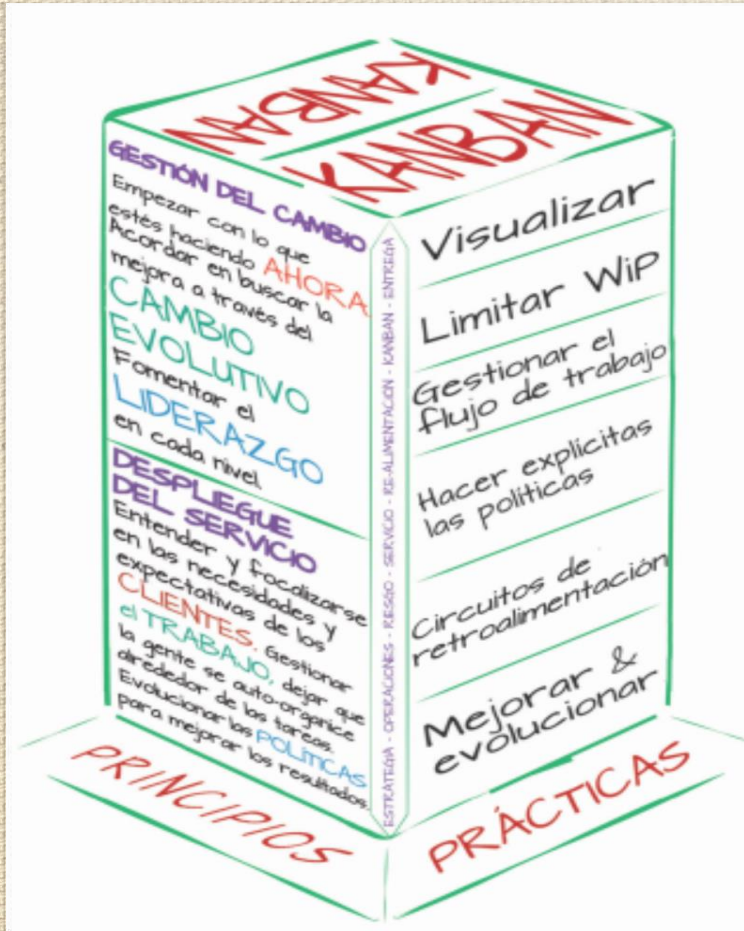
## Touch Time (Tiempo de Tocado)

- 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.

$$\textit{Touch Time} \leq \textit{Cycle Time} \leq \textit{Lead Time}$$

## Eficiencia del Ciclo de Proceso

$$\% \text{ Eficiencia ciclo proceso} = \text{Touch Time} / \text{Elapsed Time}.$$









# Scrum – Kanban: Similitudes

36

Ambos son Lean y Ágiles

Emplean sistemas de planificación Lean.

Establecen límites WIP.

La visibilidad del proceso es la base de su mejora.

Objetivo: entrega temprana y frecuente de software.

Equipos auto-organizados.

División del trabajo en partes.

Revisión y mejora continua del plan del producto, basado en datos empíricos.

# Scrum – Kanban: Diferencias

Scrum	Kanban
Iteraciones de tiempo fijo.	Tiempo fijo es opcional. La cadencia puede variar. Pueden estar marcadas por la previsión de los eventos en lugar de tener un tiempo prefijado.
Equipo asume un compromiso de trabajo por iteración.	El compromiso es opcional.
Métrica para planificación y mejora: <b>Velocidad</b> .	Métrica por defecto es <b>Lead Time</b> (Tiempo de Entrega o tiempo medio)
Equipos Multifuncionales.	Equipos Multifuncionales o especializados.
Funcionalidad divididas para poder completarse en un Sprint.	No hay prescripción respecto del tamaño de la funcionalidad.
Deben emplearse gráficos <b>Burndown chart</b> .	No se prescriben diagramas de seguimiento.

# Scrum – Kanban: Diferencias

Scrum	Kanban
Limitación <b>WIP</b> indirecta (por Sprint).	Limitación <b>WIP</b> directa (marcada por el estado del trabajo)
Se deben realizar estimaciones.	Las estimaciones son opcionales.
No se puede agregar alcance en medio de una iteración.	Siempre que haya capacidad disponible se puede agregar trabajo.
Sprint Backlog pertenece a un equipo determinado.	Varios equipos pueden compartir pizarra Kanban.
Se prescriben tres roles (PO / SM/ Equipo).	No hay roles prescritos.
En cada sprint se limpia el tablero de seguimiento.	El tablero Kanban es persistente.
Product Backlog Priorizado.	La priorización es opcional.



# SCRUM

VS

# KANBAN

## PLANNING → REGULAR

occurs at the beginning of sprint



## ESTIMATIONS of TIME



**BEFORE** start of sprint

items should be small to finish within sprint



## CHANGES TO WORK SCOPE

should wait for next sprint



## ROLES



## MEETINGS

**SPRINT PLANNING**  
1-4 hour collaborative session

**SPRINT REVIEW**  
0.5-2 hours review the results

**DAILY SCRUM**  
10-15 min everyday everybody talks about achievements/issues

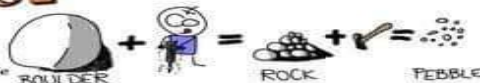
**RETROSPECTIVE**  
0.5-2 hours what went well and what did not

## OWNERSHIP

Product Owner

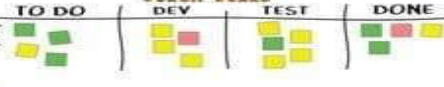
## WHEN TO USE

- small items - small value
- adding increments possible
- requirements in a good shape
- roadmap is clear
- more cross-dependent teams

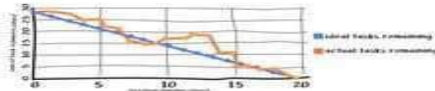


## BOARDS / ARTIFACTS

**PRODUCT BACKLOG**



**BURNDOWN CHART**



## PLANNING

NOT PRECISE planning routine

PLAN WHEN they FINISH items

**CONTINUOUS FLOW**



**demand planning**



## ESTIMATIONS of TIME

**optional** when items are completed



**LIMIT**

how many items can be in working columns at the same time

teams simply **PULL** next item from backlog and implement it

## CHANGES TO WORK SCOPE

added **AS NEEDED**



## ROLES AS NEEDED



## MEETINGS

**NONE REQUIRED**



## OWNERSHIP

DEPENDS on defined roles and necessities



## WHEN TO USE

Changes are **too fast**

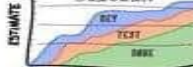
**support/maintenance** work (operational level)



## BOARDS / ARTIFACTS



**CUMULATIVE FLOW DIAGRAM**



**LEAD and CYCLE TIME DIAGRAM**

presents the average amount of time it takes for a task to be processed from the specific start to the finish point

