

Qué es

KANBAN

y cómo usarlo en
el desarrollo de

SOFTWARE

Juan Ignacio Rodriguez de león ~ @jileon
euribates [at] gmail.com



KANBAN

看板

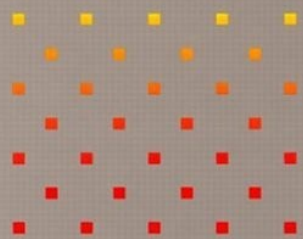


agilecanarias

A Todo Kan



Taiichi
Ohno



TOYOTA PRODUCTION SYSTEM
Beyond Large-Scale Production

Taiichi Ohno
TOYOTA



agilecanarias

A Todo Kanaban!

Kanban Aplicado al software

- David J. Anderson
 - @djaa_dja
 - Adaptación de las técnicas industriales al desarrollo del software



<http://www.djaa.com/>

Principios de Kanban

- Visualizar
- Limitar el **WIP**
- Gestionar y optimizar el flujo

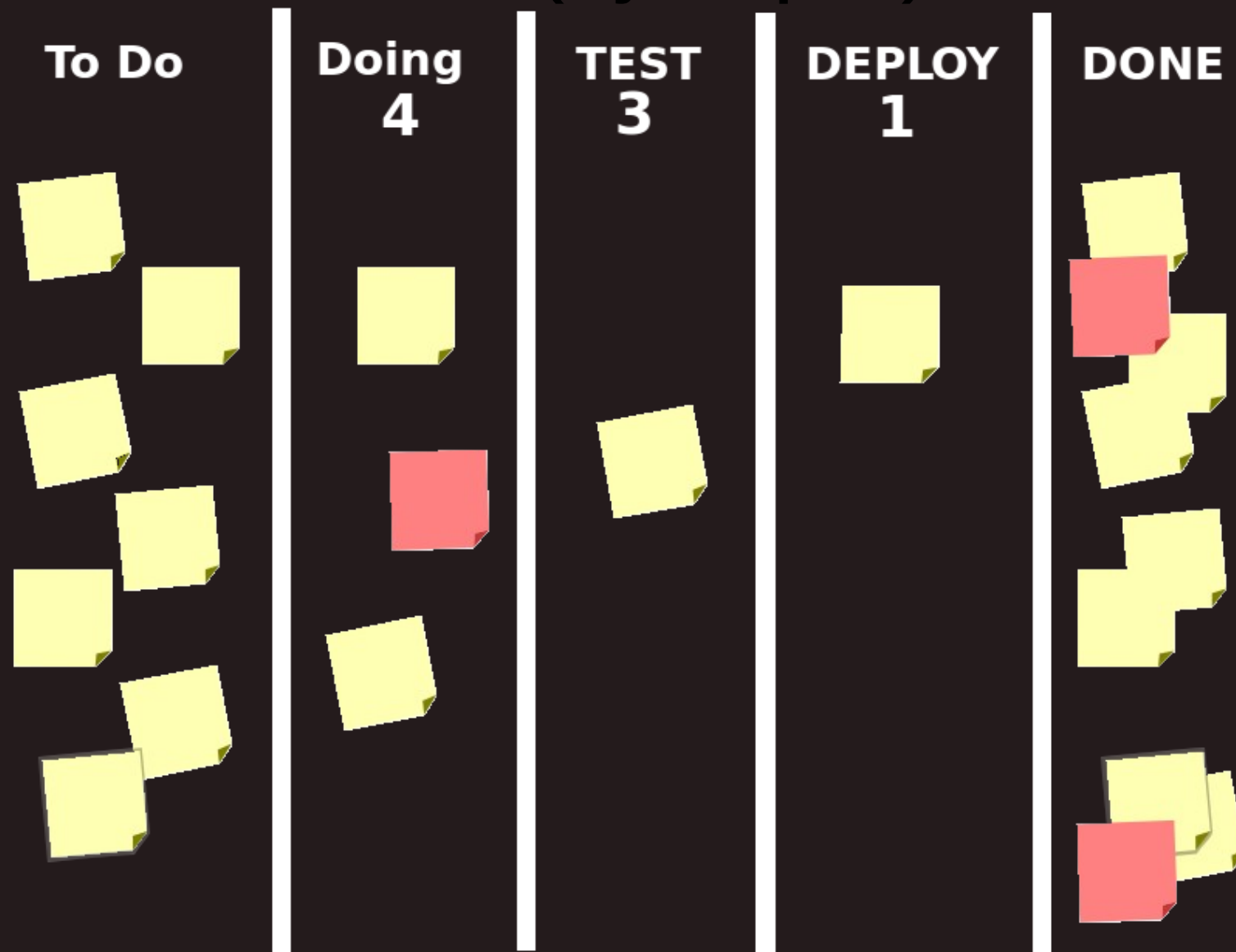
Principios de Kanban

- Visualizar
- Limitar el **WIP**
- Gestionar el flujo

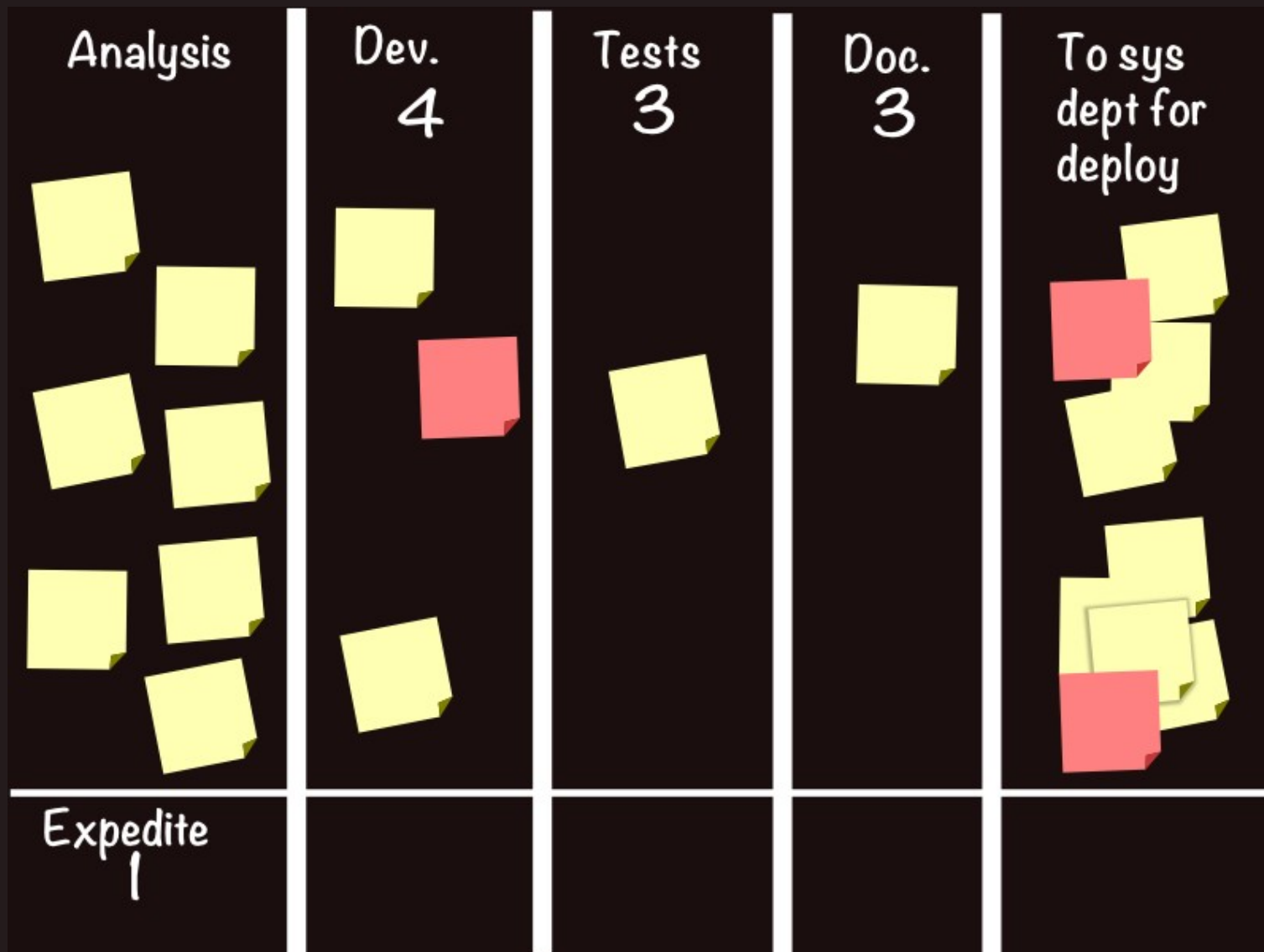
Visualizar significa ...

- Hacer que toda la información necesaria sea visible cuando la gente la necesite.
- Y, además:
 - Representar la información de forma visual, fácil de entender, simbólica
 - Difícil de no-ver
 - Transparencia
 - Compartición de la información

Tablero Kanban (ejemplo)



Otro ejemplo



Características del tablero

- El tablero debe ser bien visible, en una posición central, no obvia
- Toda tarea debe estar reflejada en el tablero
- Fácil acceso y modificación
- Diseñado por el equipo
- ¿Tablero físico o digital? El equipo decide.
(Pero hay muchas ventajas en empezar físico)

Fridge vs Radiator



VS



Hacér las políticas explícitas

- ¿Cómo pasa una tarea de una columna a la otra?
- ¿Hay prioridades en las tareas? (Por ejemplo, escoger siempre la tarea superior)
- ¿Cómo se tratan las urgencias?

En general, las políticas deben permitir **tomar una decisión rápido y sin consultar.**

Información en la nota

#245

12/3/15

Permite al
usuario hacer
login con el
email, aparte de
con su username

@jileon

1/12/14



Nº Jira

#245

12/3/15

Permite al
usuario hacer
login con el
email, aparte de
con su username

@jileon

1/12/14



Nº Jira

#245

12/3/15

Descripción

Permite al
usuario hacer
login con el
email, aparte de
con su username

@jileon

1/12/14



Nº Jira

#245

12/3/15

Descripción

Permite al
usuario hacer
login con el
email, aparte de
con su username

Asignada a

@jileon

1/12/14



Nº Jira

#245

12/3/15

Descripción

Permite al
usuario hacer
login con el
email, aparte de
con su username

Asignada a

@jileon

Fecha
entrada

1/12/14



Nº Jira

#245

Fecha límite
(Si procede)

12/3/15

Descripción

Permite al
usuario hacer
login con el
email, aparte de
con su username

Asignada a

@jileon

Fecha
entrada

1/12/14



Otra información que pueden ir

- ¿Está la tarea bloqueada?
- Tamaño de la tarea
- Fechas límite
- Tipo de trabajo y/o prioridad
- Progreso

Cualquier cosa que **facilite decidir** en qué tarea vamos a trabajar

Principios de Kanban

- Visualizar
- Limitar el **WIP**
- Gestionar el flujo

Pero ¿Qué es WIP?

- Significa “trabajo en curso” (*Work in process*)
- Todo tarea que se ha empezado y no se ha terminado
- Pero hay tareas y factores “invisibles” que suman al WIP:
 - Especificaciones no implementadas
 - Código no testeado o no integrado
 - Deuda técnica
 - Cualquier tarea que requiera *context-switching*: reuniones, papeleo administrativo, etc...

Aun hay más...

- Retrasos → Trabajo extra → Más WIP
- Un WIP grande produce por si mismo una mayor coordinación, es decir, más WIP
- Código de mala calidad, que producirá *bugs*, que añade más WIP

Limitar el WIP

- Limitar el WIP normalmente aumenta el flujo de trabajo
- Limitar el WIP no significa hacer menos cosas, significa: Hacer menos cosas **a la vez**
- A mayor WIP, menor *Lead-time* o plazo de entrega
- Normalmente, con sólo bajar o limitar el WIP se produce una **mejora notable**.

$$\text{LEAD TIME} = \frac{\text{WIP}}{\text{THROUGHPUT}}$$

Nº DE TAREAS QUE
SE ATIENDEN
SIMULTANEAMENTE

WIP

$$\text{LEAD TIME} = \frac{\text{WIP}}{\text{THROUGHPUT}}$$

TIEMPO MEDIO
QUE TARDA UNA
TAREA DESDE QUE
ENTRA HASTA QUE
SE TERMINA

TIEMPO MEDIO
QUE LLEVA
RESOLVER UNA
TAREA

Ley de little

- WIP = 16
- Throughput = 2

$$\frac{16}{2} = 8 \text{ días}$$

Ley de little

- WIP = 8
- Throughput = 2

$$\frac{8}{2} = 4 \text{ días}$$

Sólo se ha cambiado el WIP, todo lo demás: capacidad del equipo, tamaño del mismo, carga de trabajo, etc... sigue igual

El objetivo

- El objetivo final **no** es establecer el WIP
- El objetivo final es **reducir el *lead-time***
- Limitar el WIP Reduce el *lead-time* por varias razones:
 - Ley de Little
 - Menos tiempo gastado en *context-switching*
 - Menos retrasos
 - Mejor flujo

Límites de WIP



Estrategias elegir WIP global

- Entre 1 y 3 tareas por miembro del equipo
- 2/3 del tamaño del equipo (Fuerza la cooperación)
- Sube y baja 20: Empieza con un número grande (p.e. 2 x la carga actual) y baja un 20% cada vez
- Técnica AODBC (A Ojo de Buen Cubero): elige un número y tira pa'lante

Tendrás que ir ajustando hasta llegar al WIP adecuado al equipo.
No se debe perder tiempo intentado acertar a la primera.

Principios de Kanban

- Visualizar
- Limitar el **WIP**
- Gestionar el flujo

Flujo



Para aumentar el flujo, a veces hay que reducir el tráfico



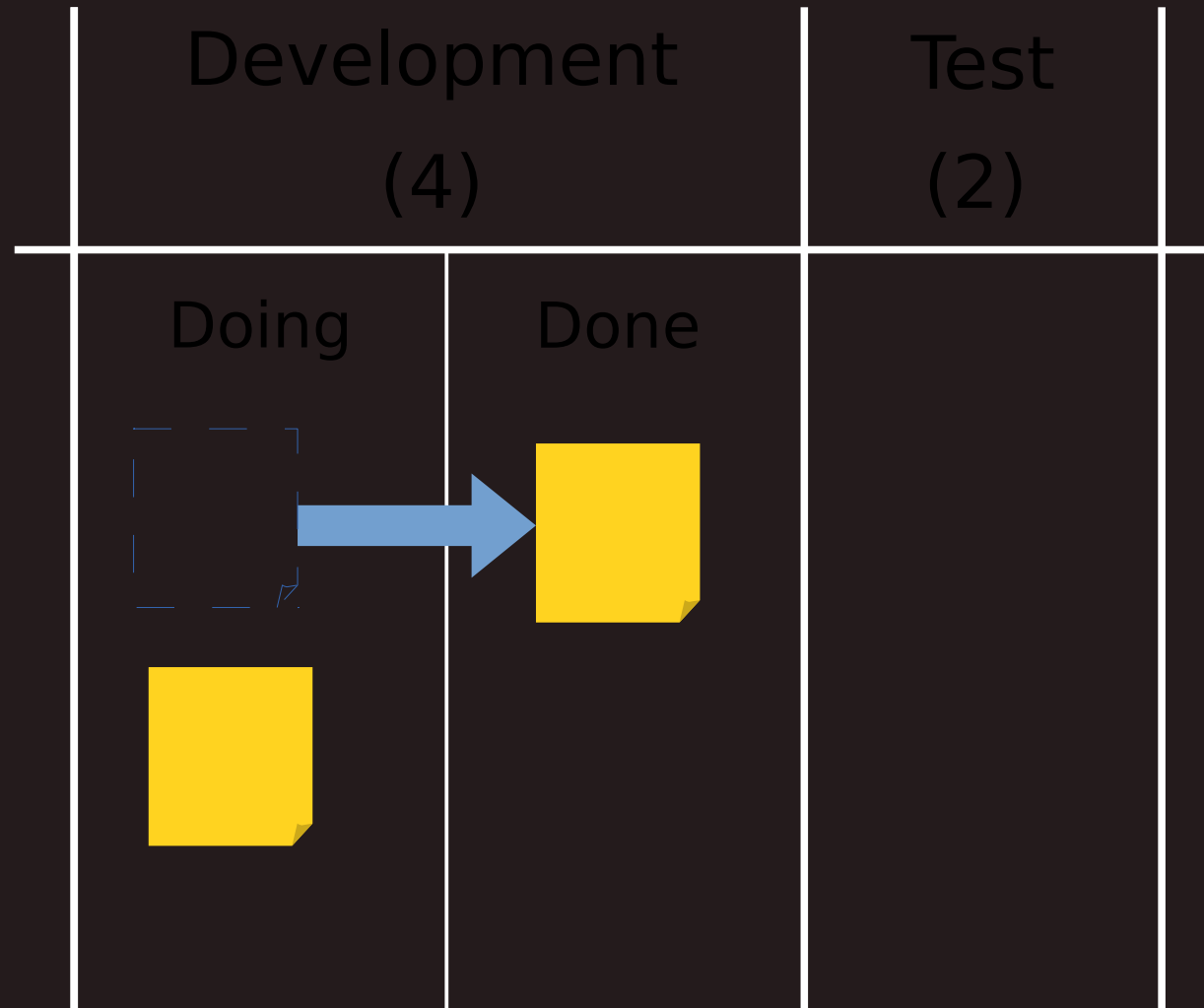
Colas

- Facilitan el paso de tareas y suavizan el flujo del trabajo
- Señal visual de que un trabajo está listo para continuar
- Divisor una columna, por ejemplo Development, en dos, *doing* y *done*.
- Las tareas en la cola cuentan para el límite del WIP de la columna

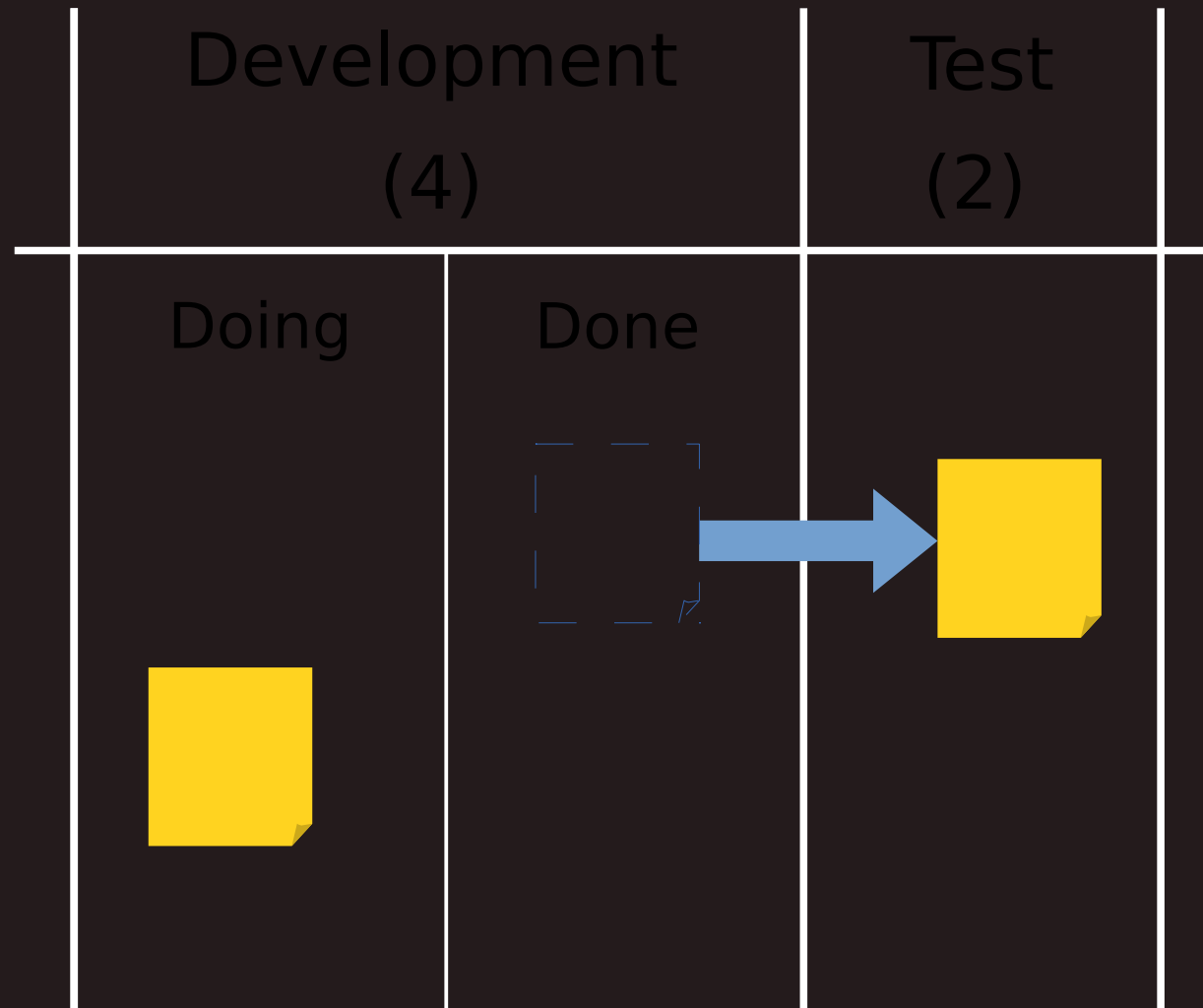
El desarrollador termina

Development (4)		Test (2)
Doing	Done	
		
		

Pasa la tarea a la cola



El *tester* acepta la tarea (pull)



Si no podemos continuar ...

- No nos quedamos de brazos cruzados:
 - Ayudamos a otro desarrollador a terminar su tarea, si es posible
 - Ayudamos a desbloquear la siguiente columna, para poder hacer un hueco y seguir con otra tarea
 - Cualquier otra cosa que contribuya a solucionar el atasco.
- NO podemos continuar hasta que el problema se resuelve

¿Qué conseguimos con esto?

- Enterarnos antes de los problemas
- Detectar los cuellos de botella
- Fomentar el trabajo cooperativo
- Enterarnos antes de los problemas (Si, está repetido, porque es muy importante)

Reunión diaria

Parecida a la de **Scrum**, con algunas diferencias

- Frente al tablero
- Se comentan las tareas empezando por la derecha (Las que más cerca están de terminar)
- Hora fija (A determinar por el equipo)
- Cortas. 15 min máximo

Métricas

- Concentrarse en las más fáciles de obtener
- Fundamentales:
 - Lead Time
 - Througput
- Anotar la fecha de entrada (*in*) y salida (*out*) de cada tarea

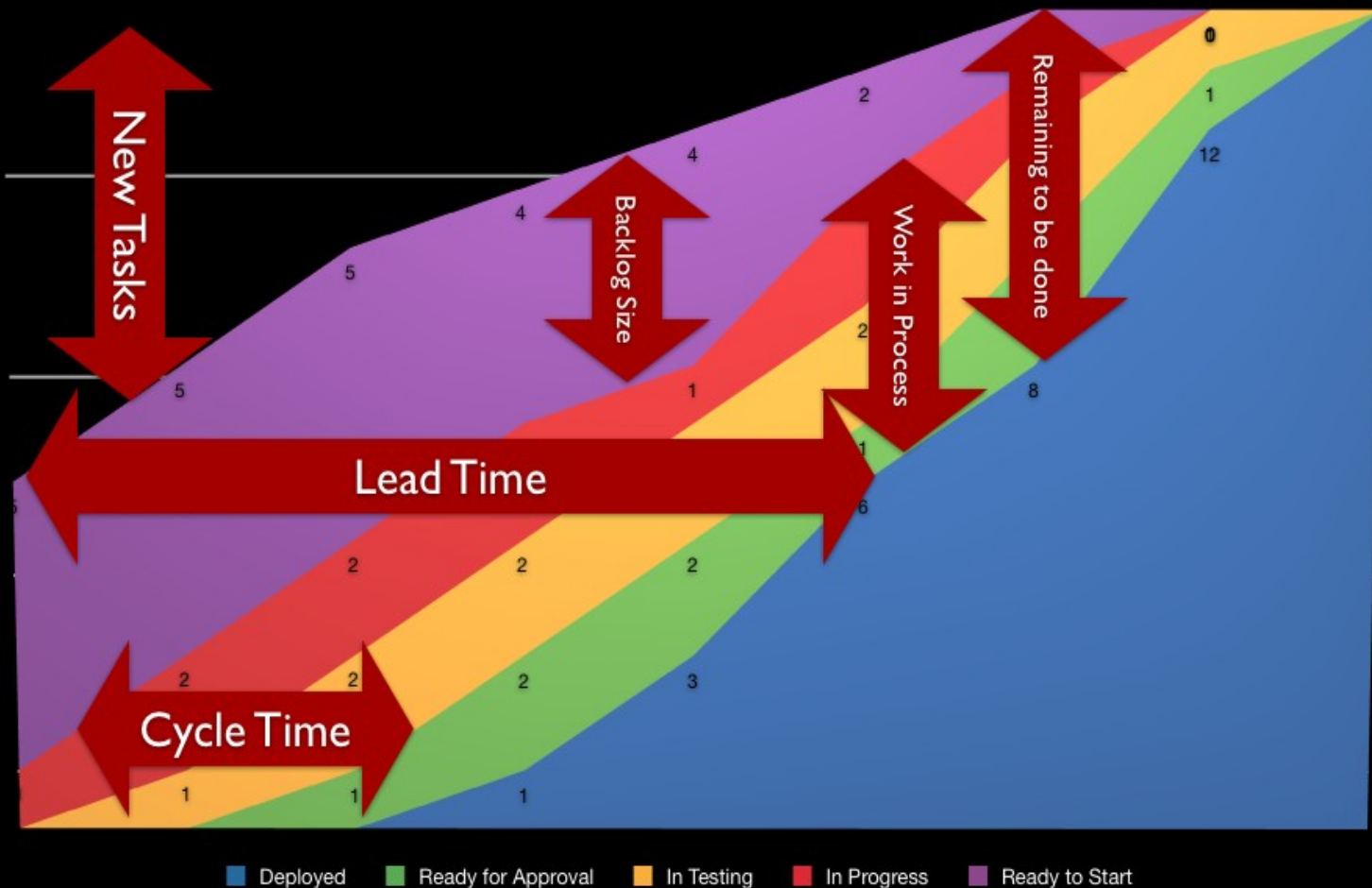


Predictibilidad

Otras métricas

- Due Date Performance
 - Para tareas de fecha tope, cuantas han terminado en la fecha acordada y cuantas no
- Tareas bloqueadas
 - Días perdidos por tarea bloqueada (Histograma)
- Diagrama de Flujo Acumulado (Cumulative Flow Diagram – CFD)
 - Se anota cada día el nº de tareas en cada columna

Cumulative Flow Diagram



Kaizen (Mejora continua)

- Usando la retroalimentación que nos dan:
 - El flujo del trabajo tal y como se ve en el tablero
 - Las métricas
 - Las reuniones diarias
- Intentamos hacer mejoras pequeñas
 - Medimos antes y después, para ver si la hipótesis es correcta
 - Eliminamos un cuello de botella (Automáticamente aparecerá otro)

Para terminar

- Ventajas de Kanban
- Temas no cubiertos
- Bibliografía
- Herramientas software

Ventajas de Kanban

- Empieza donde estás. No especifica roles
- Fácil de implementar, *low-tech & low-cost*
- Fácil de cambiar
 - (Pero resiste el impulso de cambiar al crear el tablero: pon “lo que hay ahora”)
- Permite obtener métricas útiles sin demasiado esfuerzo
- Mejora continua

Temas no cubiertos

- Clases de servicio
- Planificación y estimación
- Scrum + Kanban = Scrumban
- Análisis raíz-causa
- Retrospectivas
- Personal Kanban
- Seguramente un montón de cosas más...

Bibliografía

- Kanban. Successful Evolutionary Change for Your Technology Business

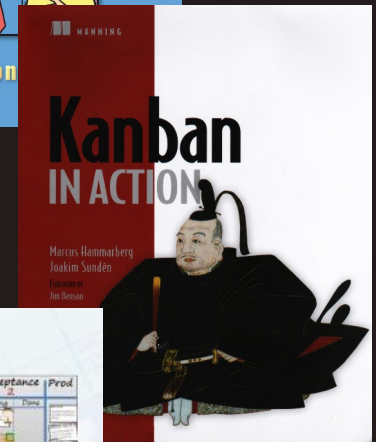
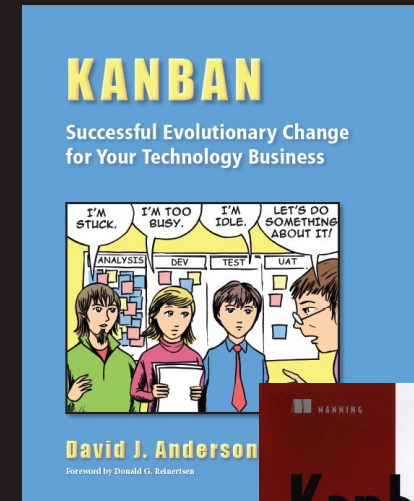
David J. Anderson

- Kanban in Action

Marcus Hammarberg and Joakim Sundén

- Kanban and Scrum - making the most of both

Henrik Kniberg and Mattias Skarin



Herramientas software

- LeanKit Kanban (<http://leankit.com>)
- AgileZen (<http://www.agilezen.com>)
- Trello (<https://trello.com>)
- KanbanFlow (<https://kanbanflow.com/>)
- Kanbanize (<http://kanbanize.com/>)
- Kanbanery (<https://kanbanery.com/>)

Añadidos a otras herramientas

- **JIRA Agile** es la herramienta Kanban de Atlassian para su sistema JIRA
 - <http://www.atlassian.com/software/jira/agile>
- **Team Foundation Service (TFS)** de MS incluye Kanban
 - <http://mng.bz/4vd0>
- **HuBoard** es un sistema Kanban añadido al sistema de gestión de tareas de GitHub
 - <http://huboard.com/>

Gracias por su tiempo :-)



<http://www.meetup.com/Agile-Canarias/>
@Agile-Canarias