

Cómo trabajamos usando Kanban

A photograph of a traditional Japanese garden. In the center, a light-colored gravel path leads towards a low, curved wooden bridge. The bridge is supported by dark wooden posts and railings. To the right of the path, there are large, dark, mossy rocks and dense green foliage. A few small red flowers are visible on some bushes. The overall atmosphere is peaceful and minimalist.

Peopleware
Agile Management
#PAM2016

Diego J. Romero López

diego@intelligenia.com

Índice

Sobre mí

Punto de partida

Kanban

El Kaizen y las buenas prácticas

La evaluación

Los resultados

Mejoras hasta ahora

Conclusión

Sobre mí

**Diego J. Romero López
Ingeniero de Software**

Subdirector técnico y jefe de equipo en intelligenia

Ing. Informática (Ms. Eng.)
MSc. Re. Ingeniería del Software
M. Eng. Dirección y Gestión de Proyectos Software

diegojromerolopez@gmail.com
<https://github.com/diegojromerolopez/>
<https://es.linkedin.com/in/diegojromerolopez>



Punto de partida

- Proyectos en estado zombie.
- Proyectos heterogéneos.
- Clientes poco colaborativos.
- Multitarea a nivel de proyecto.
- Propiedad individual del código.
- Desarrollo y mantenimiento en paralelo.
- Proyectos con alta deuda técnica.
- Proyectos con baja usabilidad.
- Facturación por horas.



- Distinto origen de los desarrolladores.
- Inexistencia de prácticas ágiles.
- Ruido.
- Interrupciones.
- Falta de estándares y de cultura de calidad de software.
- Dificultades para el paso a producción.

Kanban

Las bases [ANDERS10]:

- Comienza el proceso de mejora con los procesos que ya existen.
- Persigue el cambio y la mejora incremental.
- Respeta el proceso, los roles, responsabilidades y títulos actuales.
- Liderazgo constante a todos los niveles.

- Las tareas del tablero son las que han de hacerse. El resto son **desperdicio**.
- Define una serie de estados para cada tarea.
- El flujo de trabajo se representa en un tablero por proyecto.
- Evitar multitarea.
- Dos estados importantes:
 - En desarrollo.
 - Terminado.
- Herramienta electrónica de visualización total.

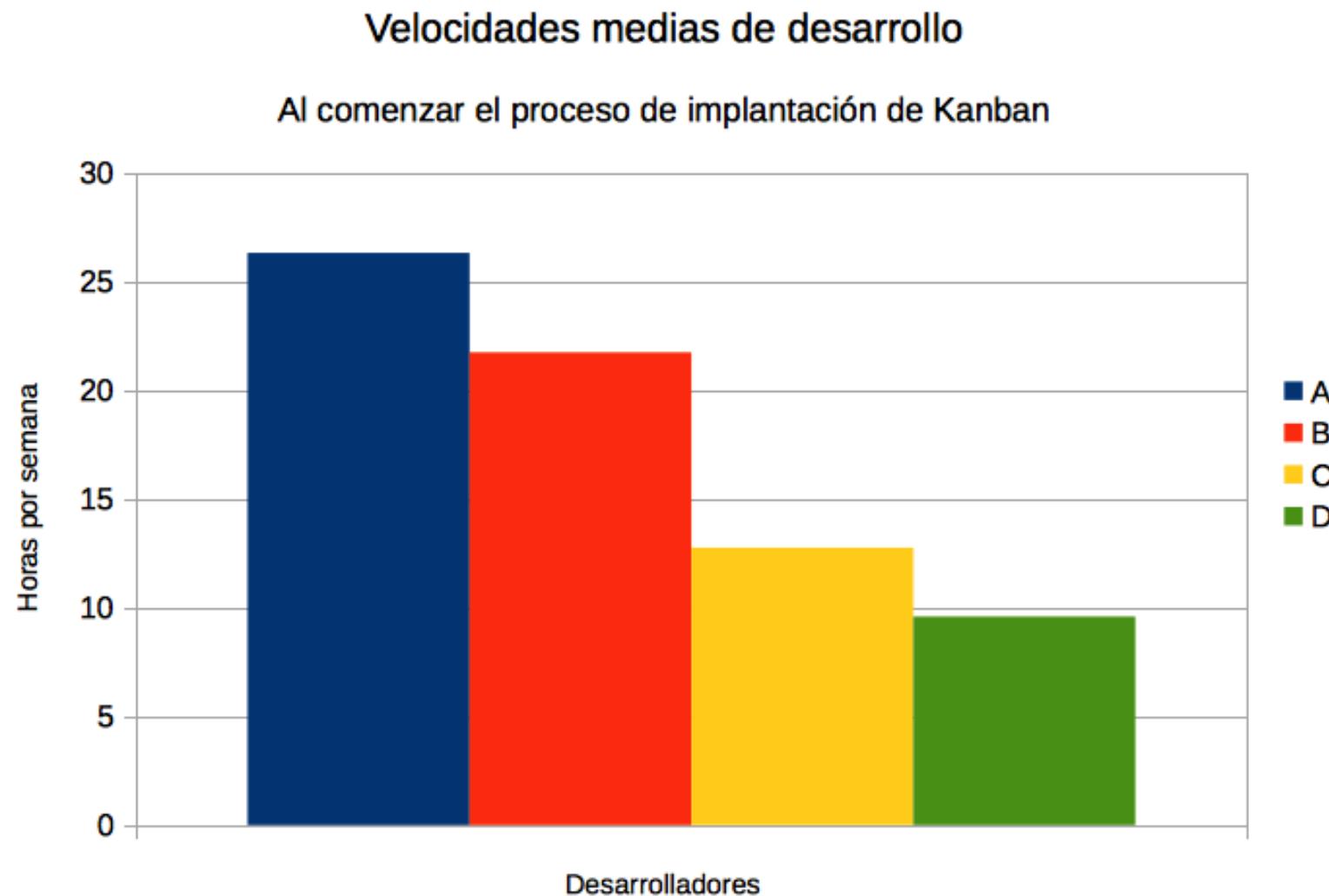
- Trello es un servicio web gratuito de gestión de tableros de tareas.

The screenshot shows a Trello board with the following structure:

- Pendiente:**
 - Activar cron en trello-stats.intelligenia.com con los tiempos para desarrolladores
 - Añadir una tarjeta...
- En desarrollo:**
 - Añadir una tarjeta...
- En revisión:**
 - Añadir una tarjeta...
- Pendiente de subida:**
 - ¿Hace falta realmente petición por tarjeta? Esas pueden traer todas la tabla con más dato
 - 1 comment
 - ¿Por qué no funciona los tiempos en trello-stats.intelligenia.com
 - 1 comment
 - Comprobar si efectivamente la primera fecha de las tarjetas es el comentario de las tarjetas más nueva
 - Indicar cuándo se ha obtenido los datos
 - 1 comment
 - Bloquear fetch por tanto evitar que se "pise" una tarjeta con otra de la misma tabla
 - 1 comment

Por qué Kanban

- Evita multitarea.
- Integración de técnica Pomodoro / Timeboxing



Por qué Kanban

- Tareas de mantenimiento.
- Facturación por horas.
- Tareas esporádicas.
- Flexibilidad máxima. No hay prácticas ágiles *obligatorias*.

Kanban es fácil

El kaizen y las buenas prácticas

Kanban no es fácil, kanban es fácil.

Kanban debe estar precedido por un cambio de mentalidad hacia el *kaizen*.

Cambiar la mentalidad

Suscitar responsabilidad.

Mejora continua.

Hacer las cosas *bien*.

Máxima transparencia.

Cambiar la forma de trabajar

Retroalimentación constante.

El cliente no es el enemigo.



Proceso de formación a nivel técnico

- Minimización del estado global. [MOSELEY06].
- Orientación a Objetos (Naked Objects). [PAWSON04].
- Principio de única responsabilidad. [MARTIN08].
- Nomenclatura estándar. [MARTIN08].
- Reducción de complejidad ciclomática [MCCABE83].
- Refactorización. [FOWL99].
- Revisiones de código. [BECK04].
- Programación por parejas. [BECK04].



Otras buenas prácticas

- Documentación (Usando C4 [BROWN14])
- Pruebas e integración continua.
- Usabilidad.
- Darle valor al software.
- Silencio en la oficina.
- Evitación de las interrupciones.



La evaluación

Evaluación subjetiva

Buscamos una forma de medir la satisfacción de los implicados con el proceso.

- Retroalimentación del equipo de desarrollo de forma mensual.
- Encuestas cada 6 meses:
 - Encuesta para la dirección, equipo de desarrollo y equipo comercial.
 - Encuesta sobre el proceso de desarrollo de los proyectos para el equipo de trabajo basada en [ANDERS12].

ASSESSMENT IN PROGRESS
PLEASE KNOCK AND WAIT

Mejora (10)

1. Sabemos cómo hacer las cosas bien.
2. Hay retrospectivas frecuentes.
3. El equipo sabe cuál es su rendimiento.
4. El equipo sabe cuáles son sus debilidades.
5. El equipo sabe cuál es el estado del proyecto.
6. El equipo conoce lo que falta para terminarlo.
7. El equipo conoce los obstáculos del proyecto.
8. El equipo sabe en qué obstáculos se están trabajando.
9. El equipo sabe los pasos para resolver obstáculos.
10. El equipo sabe lo que ha aprendido en cada momento.

Retroalimentación (7)

1. Reuniones diarias.
2. A los implicados de cada proyecto (cliente, equipo comercial, etc.) se les informa con frecuencia.
3. Dirección conoce el estado del equipo y de los proyectos.
4. Conversaciones con otros miembros del equipo sobre problemas del proyecto.
5. Conocimiento de la situación de facturación del proyecto.
6. Presentaciones y discusiones sobre KPI (ratio de defectos, satisfacción del cliente).
7. La frecuencia de este tipo de presentaciones y discusiones es al menos una vez al mes.

Políticas de trabajo claras y en uso (12)

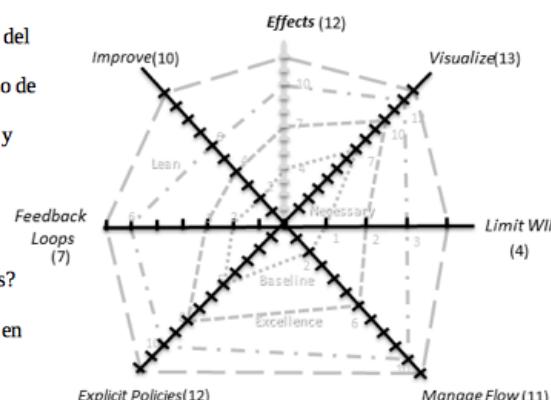
1. ¿Hay distintos tipos de tareas?
2. ¿Cómo se indica la prioridad de las tareas?
3. ¿Quién y cuándo gestiona las colas de tareas?
4. ¿Hay un proceso de asignación de trabajo?
5. Hay una definición de preparada para entrar en desarrollo.
6. Hay un proceso de estimación definido.
7. Se limita el tamaño de las tareas.
8. Está claro cómo seleccionar tareas de un estado para moverlas a otra.
9. Hay una estrategia para compartir conocimiento.
10. Definición de "Terminado" en cada estado.
11. Hay clases de servicio (¿se gestionan de forma distinta las tareas según su naturaleza y riesgo?).
12. Se sabe cuál es la capacidad máxima de trabajo por cada clase de servicio.

Efectos (evidencias de que esto funciona)

1. Los miembros del equipo ven y entienden la panorámica de cada proyecto.
2. Hay sentimiento de equipo (ayuda y respeto).
3. El foco está en eliminar las tareas que bloquean.
4. Se terminan las tareas actuales antes de comenzar otras.
5. El equipo tiene prioridades y las cumple.
6. No hay estrés (el equipo trabaja a un buen ritmo pero no excesivo).
7. Hay motivación para buscar mejoras en el software y el trabajo.
8. Hay evolución del proceso (cambios en WIP, flujo de trabajo, visualización, etc.) Para adaptarse mejor al proceso.
9. Ha habido un incremento de la profundidad de la implantación Kanban.
10. El proceso de evolución sigue el modelo ágil Kanban.
11. Hay mentorización y tiene efecto en el proceso de mejora.
12. El comportamiento de los flujos de trabajo hace que el proceso evolucione. Nos adaptamos al trabajo (no al revés).

Visualización (13)

1. Se ven las tareas sobre las que se está trabajando.
2. Se ven los tipos de tareas de forma diferenciada.
3. El flujo de trabajo es visible.
4. Se ve las tareas siguientes y cuáles están terminadas.
5. Se ve sobre lo que está trabajando cada miembro.
6. Se indican los bloqueos de alguna forma.
7. Las políticas de trabajo están visibles en algún repositorio.
8. Hay un estado "Listo y pendiente de desplegar".
9. Las métricas (cycle, lead, S/E, etc.) son visibles.
10. Los límites WIP están visibles para cada estado.
11. Las dependencias entre tareas (jerarquía de tareas) se muestran.
12. Se muestran las dependencias entre flujos de trabajo.
13. Se muestra el riesgo de cada tarea y característica (fecha límite de tarea, coste de retraso, compromiso con el cliente, riesgos técnicos o de mercado).



Adaptado de
<http://leanagileprojects.blogspot.com.es/2013/03/depth-of-kanban-good-coaching-tool.html>
(c) Christophe Achouantz

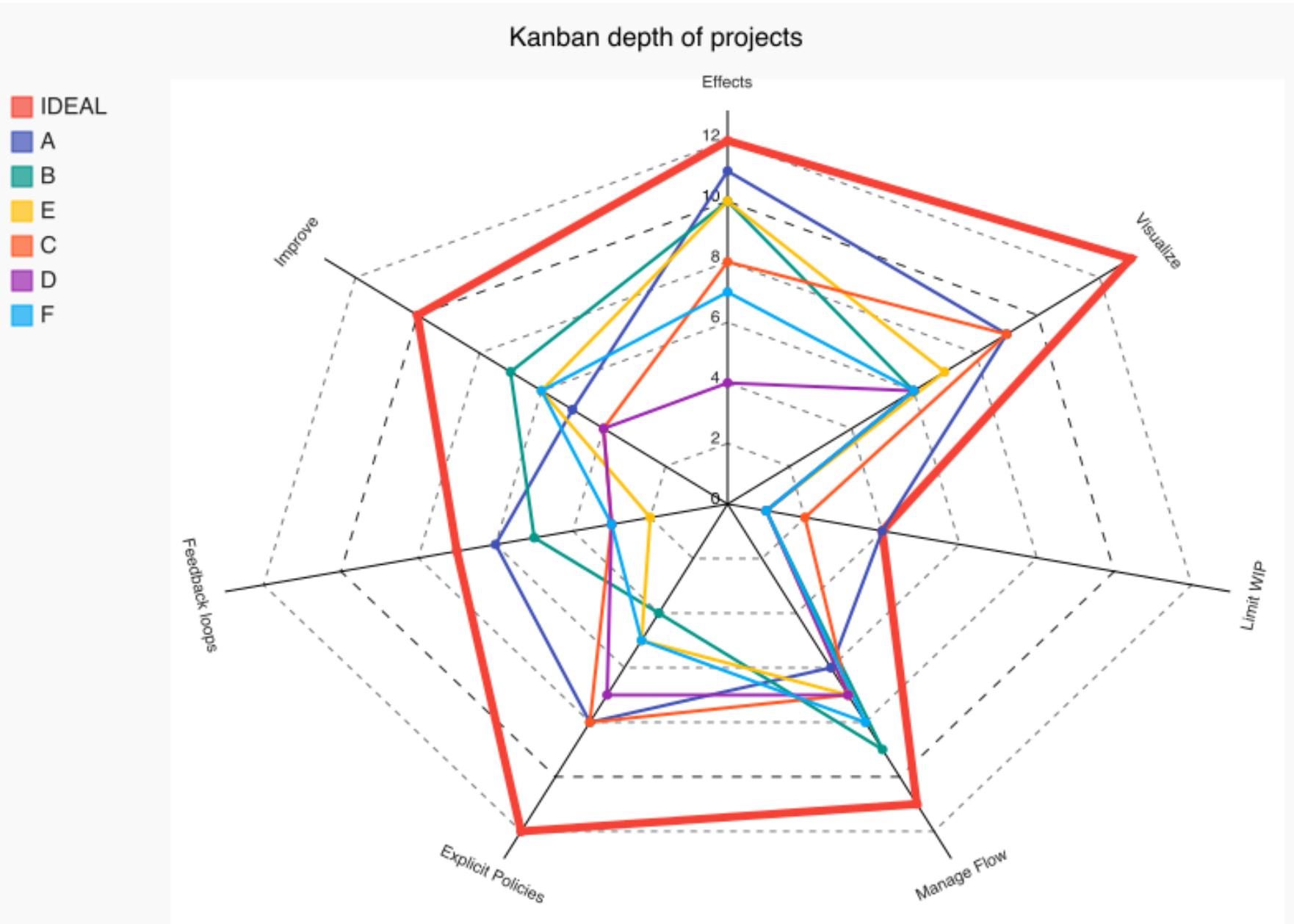
Límite de WIP (4)

1. No hay límite WIP pero sí un compromiso en terminar las tareas empezadas antes de empezar con las nuevas. Trabajando así con un máximo de tareas cómodo.
2. Hay límites WIP explícitos: por persona y en algunas columnas.
3. Hay límite WIP a nivel de flujo de trabajo. Esto es, a nivel de características del sistema desarrollando.
4. Hay muchos flujos de trabajo independientes en función de quién solicita la tarea y si viene del cliente o es una necesidad interna.

Gestión del flujo (11)

1. Reuniones diarias de planificación.
2. Los problemas externos se paran y se escalan (al líder de equipo, luego a la dirección, etc.)
3. La siguiente tarea se reprioriza continuamente en función de las necesidades del cliente.
4. Se actualiza el documento con la cuenta de horas al terminar cada tarea.
5. El equipo sabe el reto/problema del proyecto en el que está trabajando.
6. ¿Cuál es el objetivo del proyecto y cuándo ha sido completado?
7. El equipo sabe el obstáculo sobre el que se está trabajando.
8. El equipo conoce los obstáculos que impiden llegar al objetivo.
9. El equipo sabe los pasos para resolver cada obstáculo.
10. El equipo aprende en cada obstáculo.

Resultados de la encuesta para todos los proyectos



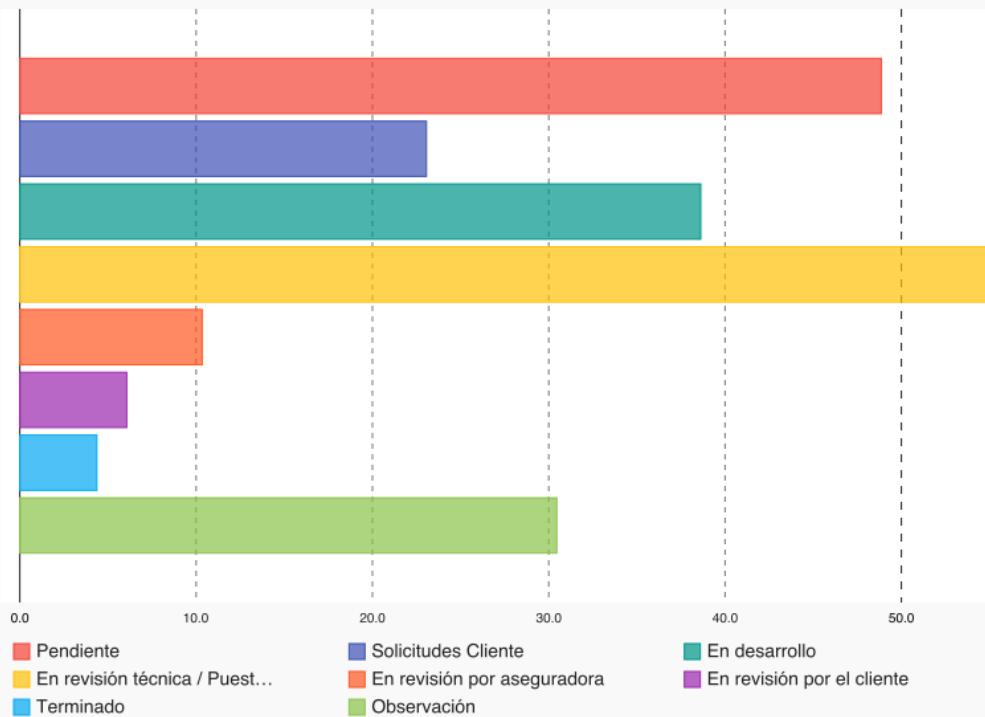
Evaluación

Evaluación objetiva

Medidas de tiempos de vida y desarrollo de las tareas en cada tablero:

- Lead.
- Cycle.
- Tiempo medio de desarrollo de tarea.
- Tiempo medio que están las tareas en cada estado.
- Número de veces que una tarea vuelve.
- ...

Average time for all board cards by list for 56fd264f839ead21ab012b53



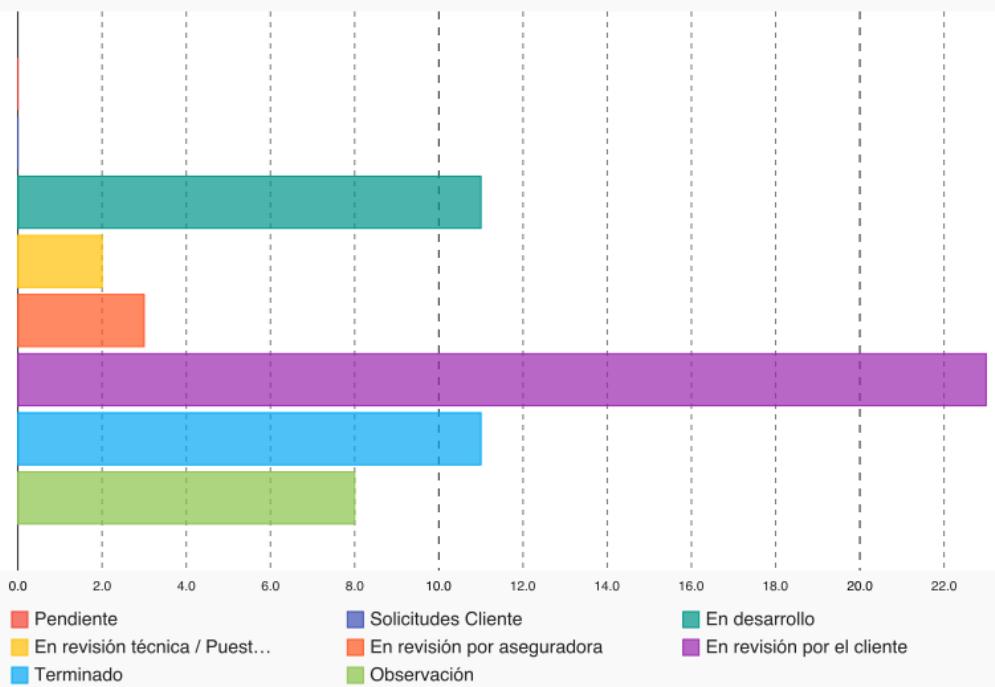
Detección de cuellos de botella

Detección de

- Desarrollos de baja calidad.
- Requisitos que no estaban claros.
- Tareas de demasiado tamaño.

Number of times a list is the source of a card movement to backwards in

56fd264f839ead21ab012b53



Mejoras hasta ahora



Trello Stats

- Panel de control.
- Máxima transparencia.
- Plataforma Software disponible de forma abierta y gratuita (es software libre) en
<https://github.com/diegojromerolopez/django-trello-stats>

Conclusiones

Lo bueno

- Terminamos dos proyectos *zombies*.
- Un cliente ha adoptado kanban en su equipo de trabajo.
- El equipo valora positivamente el cambio con respecto a lo que había antes.

Conclusiones (y 2)

Lo no tan bueno

- Resistencia al cambio.
- Fricción con otros equipos.
- Ruido e interrupciones.
- La calidad del software es “accesoria”.
- No hay proceso de pruebas.
- Ha sido un proceso difícil.

Bibliografía

- [ANDERS10] *David J. Anderson*. Kanban.
- [ANDERS12] David J. Anderson y otros. How deep is your Kanban?
- [BECK04] Kent Beck. Extreme Programming Explained.
- [BROWN14] Simon Brown. Software Architecture for Developers.
- [FOWL99] Martin Fowler. Refactoring: Improving the Design of Existing Code.
- [MACCABE83] Thomas J. MacCabe. Structured testing.
- [MARTIN08] Robert C. Martin. Clean Code: A handbook of Agile Software Craftsmanship.
- [MOSELEY06] Ben Moeseley y Peter Marks. Out of the Tar Pit.
- [PAWSON04] Richard Pawson. Naked Objects. Tesis para la obtención del Doctorado en Filosofía en la Universidad Trinity College de Dublín (Irlanda). URL.
- [ROM2016] Diego J. Romero López. Trabajo Fin de Máster:
<https://github.com/diegojromerolopez/kanban-research/blob/master/master-thesis/master-thesis-diegojromerolopez-es.pdf>

¡Muchas gracias!

