

lean IT y kanban

concepto

“lean IT involucra a las personas usando un framework formado por principios, sistemas y herramientas para integrar, alinear y sincronizar las IT de una organización con el negocio”

- apoyando la mejora continua de los procesos de negocio
- mejorando la efectividad de los propios servicios y procesos IT

principios

constancia en el propósito (Deming): fijar como objetivo lo más importante

respeto por las **personas**: liberar su excelencia y potencial creativo

mejora continua: si no mejoramos, nos quedamosatrás

comportamiento proactivo: tomar la iniciativa y asumir responsabilidad por la calidad de nuestro trabajo

la voz del **cliente**: involucrarnos y entender sus necesidades

principios

calidad por defecto: obligación de parar y corregir los problemas

pensar como sistemas: comprender la cadena de valor global, no centrarse en lo local

flow/pull/jit: eliminar retrasos e interrupciones en toda la cadena de valor

cultura: se crea cultura con el tiempo, cambiando la percepción del trabajo y los roles

valor y desperdicio

“el enfoque principal de lean IT es resolver problemas para entregar valor al cliente, eliminando sistemáticamente cualquier desperdicio a lo largo de la cadena de valor”

- cadena de valor: todo el ciclo de vida de los distintos procesos
- valor: aquello que el cliente quiere y por lo que está dispuesto a pagar
- trabajo: VA, NVA, NNVA

las tres Ms

mura (variación): inconsistencia en el flujo de trabajo causada por cambios en volumen, variación y calidad

muri (sobrecarga): someter a recursos humanos y materiales a cargas de trabajo no realistas

muda (desperdicio): trabajo que no aporta valor

**¿qué desperdicios se dan en
el desarrollo de software?**

trabajo realizado parcialmente: complicado de integrar
características extra: no se suelen usar
reaprendizaje: redescubrimiento de soluciones a problemas que ya resolvimos en su momento

"de mano en mano": el conocimiento no se transmite sólo con documentos y diagramas

pausas: esperas derivadas de contratación, revisión o aprobación de procesos...

cambio de tarea

defectos

técnicas

pensamiento en A3

framework mental para usar el método científico de resolución de
problemas: PDCA

atacar a la raíz y no a los síntomas

técnicas

- mapeo de la cadena de valor: similar al mapa de procesos pero representando flujos de información
- kaizen: mejora continua de sistema en forma de pequeños incrementos (flow, cadena de valor global) y proceso (áreas concretas)
- trabajo estandarizado: define y documenta el método más efectivo de llevar a cabo una tarea, consolidando el conocimiento en posesión de unos pocos

lean software development

- Mary y Tom Poppendieck, pioneros
- adaptación de los métodos de Toyota

no sólo se centra en el desarrollo del software, también en las cadenas de valor (donde opera el software)

“identificar el 20% del código que proporciona el 80% de la funcionalidad y entregarlo just-in-time”

requisitos

asumir **respetuosamente** que el cliente podría no saber qué quiere:

- fuertes opiniones basadas en una comprensión incompleta de la situación
- modelo mental basado en su familiaridad con el proceso o sistema actual
- distintos clientes piensan de forma distinta... ¡ojo con los clientes que no vayan a usar finalmente el sistema!
- los clientes no se dan cuenta de lo que saben

el equipo debe ir a **gemba**

gestión de demanda

cada iteración tiene un límite de número de puntos a asignar,
establecido por el **equipo** (no por un jefe de proyecto)

conseguir una cadencia sostenible en el tiempo: **ritmo**

investigar la **causa de cada interrupcion** para prevenir su recurrencia
priorizar para mantener el nivel de demanda

los ciclos cortos hacen que el trabajo entre según se vaya necesitando

resto del proceso

kanban para ejecución y pruebas

incluir al service desk en la cadena de valor global, no sólo enfocarse en desarrollo

métricas: indicadores simples frente a cálculos complejos

la mayor parte de los problemas surgen en las stand-ups

los defectos se previenen

gráficos burndown, lead time, velocidad de desarrollo...

kanban

un dispositivo de señalización que introduce el flujo de trabajo a lo largo de un proceso a un ritmo manejable

sólo envía trabajo cuando lo ordena el cliente
indica específicamente qué trabajo debe hacerse
controla la cantidad de trabajo en progreso
regula las interrupciones
orquesta el ritmo de trabajo

principios

visualizar: tablero y tarjetas o postits

limitar trabajo en progreso: establecer un límite de lo que se puede hacer simultáneamente

gestionar flujo: buscar cuellos de botella

hacer explícitas las políticas del proceso, para poder discutirlas

implementar bucles de realimentación, a modo de **retrospectiva**
mejorar colaboración y experimentar

radiadores de información

tablero electrónico:

- accesible desde cualquier punto
- sin pérdida de información
- cálculo automático de métricas

tablero físico:

- más grande
- lugar natural para reunión
- más fácil de poner en marcha y cambiar

dejar la decisión al equipo

elementos de trabajo

- postits o tarjetas
- etiquetas impresas

¡ojo con cómo despegamos los postits del mazo!

¿quién es responsable del postit?

- avatares
- franjas para cada miembro del equipo

estimar cada postit en puntos historia de manera homogénea

work in progress

representa todo el trabajo en marcha en un instante de tiempo dado

- lo que estoy haciendo
- lo que está esperando para ser verificado
- lo que aún no he empezado y está en mi bandeja de entrada

**limitar el WIP implica hacer menos cosas
simultáneamente**

limitar wip

si es muy bajo: gente ociosa

si es muy alto: se empiezan muchas tareas que no se acaban

no hay reglas: debe ajustarse en función del equipo

- una regla: $2 \times N - 1$
- otra: $1.5 \times N$

indicadores

lead time: tiempo que pasa desde que un ítem entra en el tablero kanban hasta que llega a la última fase

cycle time: tiempo que pasa desde que se empieza a trabajar con el ítem hasta que llega a la última fase

columnas

básicas: por hacer, haciendo, hecho

de proceso: análisis, diseño, implementación, pruebas

buffer: asegura la fluidez entre dos flujos consecutivos del proceso

**cada una de ellas puede tener su propio wip y su propio
buffer**

cuellos de botella

se llega al máximo wip y no se pueden introducir nuevas tareas en el tablero

¿hay alguna etapa que tenga demasiados postits?

alternativa directa: repartir tareas entre otros miembros del equipo

CoS

diferentes clases de servicio pueden señalarse con diferentes colores:

- verde: deseable, pero no urgente ni importante
- rojo: urgente
- azul: con fecha límite de finalización

para las tareas urgentes, podría utilizarse una **fila de aceleración** (no contaría para el wip, pero deberían ser excepciones)

otra técnica para clasificar tareas: columnas como **filtros de prioridad**

¿y si tengo varios proyectos a la vez?

varios tableros: uno para cada uno de ellos

- más claro
- más sitios a donde mirar...

un tablero: todos al mismo

- saturación de información
- distinción con colores o pegatinas

¿por qué tienes varios proyectos a la vez??

caso práctico: kanban en el ICANE

TO-DO	WIP	DONE

TO-DO	WIP	DONE
	A	
	C	
	G	
	M	
	S	
	X	

TO-DO	WIP	SB	DONE
	A		
	C		
	G		
	M		
	S		
	X		

TO-DO	WIP	SB	DONE											
	A													
	C													
	G													
CARGAS <table border="1"><tr><td>L</td><td>M</td><td>X</td><td>J</td><td>V</td><td>?</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>	L	M	X	J	V	?							M	
L	M	X	J	V	?									
	S													
	X													

TO-DO	WIP	SB	DONE				
	A						
	C						
	G						
CARGAS	L	M	X	J	V	?	
							M
							S
							X

TO-DO		WIP	SB	DONE
2	1	A		
		C		
CARGAS				
L	M	X	J	V
		M		
		S		
		X		

más sobre kanban: [Kanban in action](#), Marcus Hammarberg and Joakim Sundén

