

MADS
S12: Kanban
(Bloque 4 - Lean y Kanban)

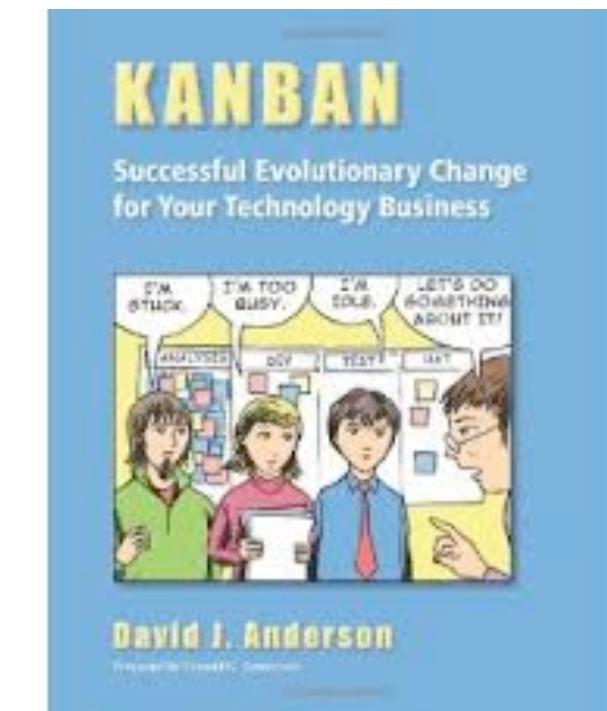
Introducción a Kanban

Historia

- Orígenes en los sistemas de fabricación lean y en los sistemas pull
- Entre 2003 y 2010 trabajo de David J. Anderson en los equipos de desarrollo de Microsoft y Corbis para adaptar estos sistemas al desarrollo de software
- Presentación en sociedad en la conferencia Agile 2007 en Washington
- Crece el interés: 6 presentaciones en Agile 2008 en Toronto
- Publicación del libro de David J. Anderson en 2010
- Aceptación por la comunidad ágil e integración con la metodología más aceptada en este momento: Scrum



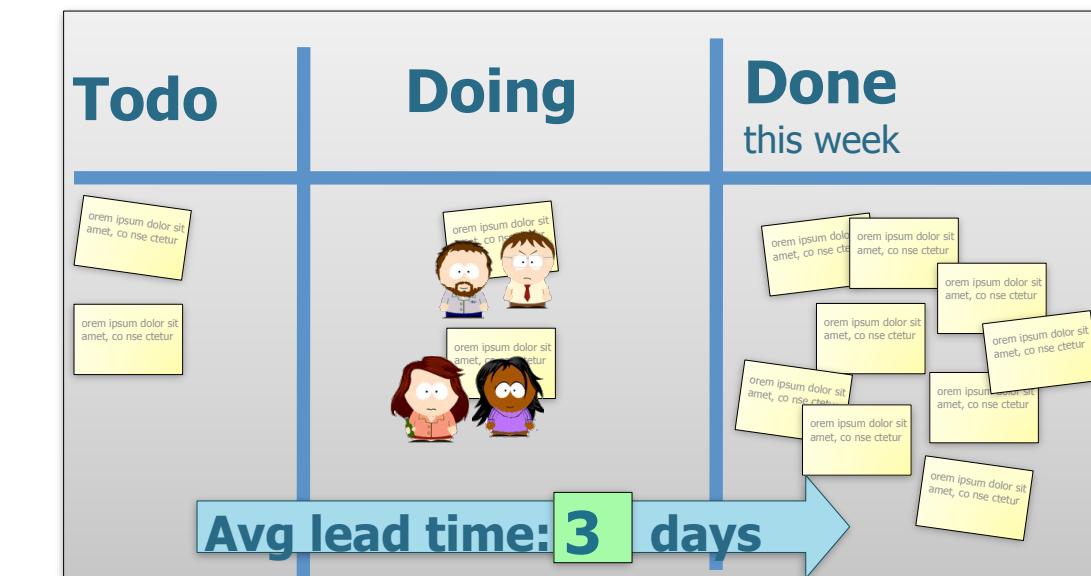
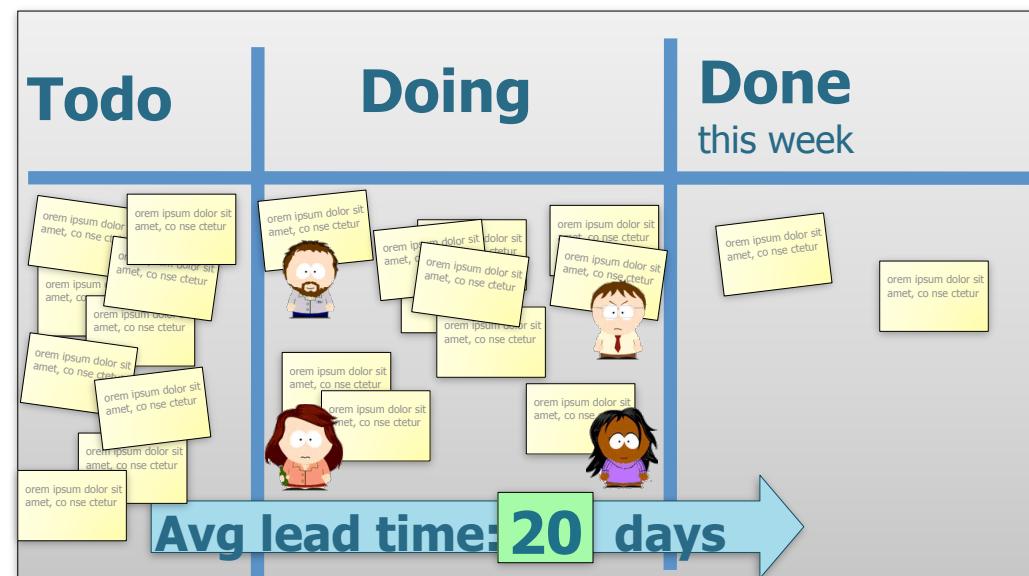
David J. Anderson



Motivaciones

- Conseguir un ritmo de trabajo sostenible en el desarrollo de software
 - Visualizar en todo momento la carga de trabajo del equipo de desarrollo (WIP: *Work In Progress*, trabajo en progreso)
 - Visualizar y estandarizar el flujo de trabajo de las historias de usuario
 - Políticas explícitas (*definition of Done*, límites WIP, etc.)
-
- Kanban nos ayuda a cambiar el prisma de ¿Qué hacen las personas? a **¿Cómo va el trabajo?**

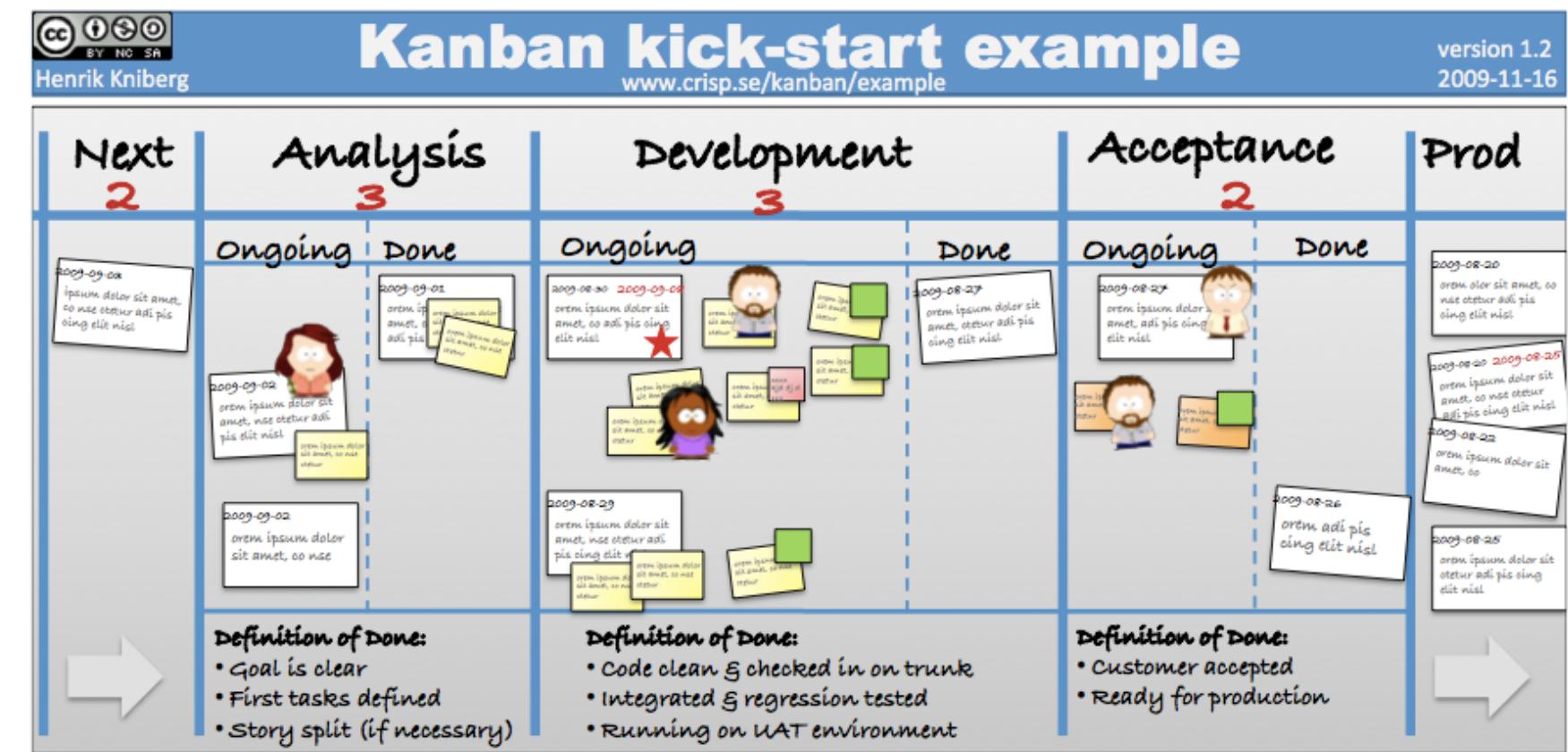
¿Qué equipo necesita mejorar?



© Henrik Kniberg

Kanban in a nutshell

- **Visualizar el flujo de trabajo**
 1. Dividir el trabajo en **pequeñas partes**, escribir cada elemento en una tarjeta y ponerla en un tablero
 2. Crear un **tablero** compartido por el equipo, dividido en columnas que identifican en qué parte del flujo de trabajo se encuentra el elemento



© Henrik Kniberg

- **Limitar el WIP** (*Work In Progress*, trabajo en progreso): asignar un límite al número de elementos que puede haber en cada estado del flujo de trabajo
- Medir el **tiempo medio de terminación de un elemento** (llamado *lead time* o *cycle time*) y optimizar el proceso para hacerlo tan pequeño y predecible como sea posible

More prescriptive



RUP (120+)

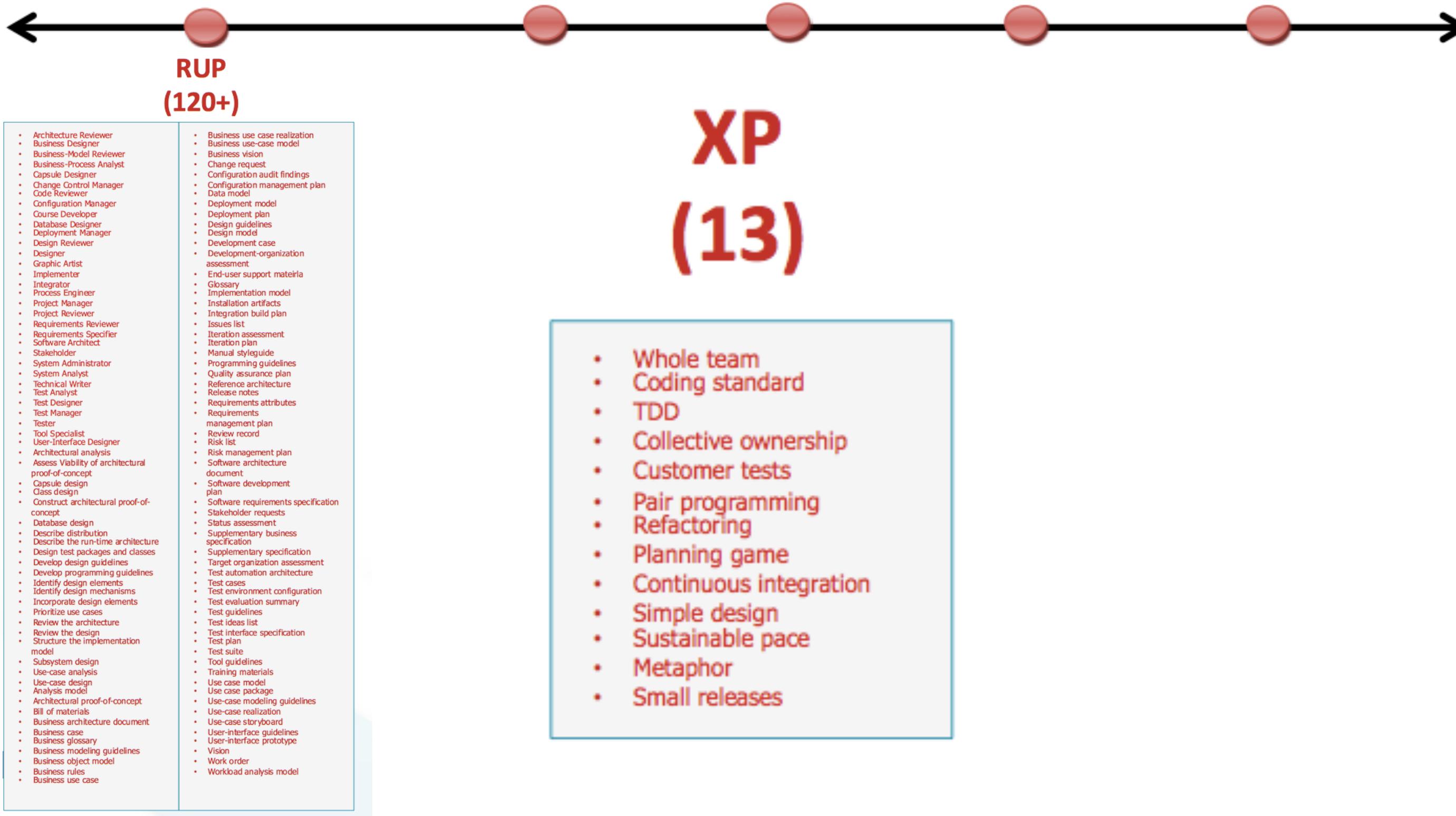
<ul style="list-style-type: none">• Architecture Reviewer• Business Designer• Business-Model Reviewer• Business-Process Analyst• Capsule Designer• Change Control Manager• Code Reviewer• Configuration Manager• Course Developer• Database Designer• Deployment Manager• Design Reviewer• Designer• Graphic Artist• Implementer• Integrator• Process Engineer• Project Manager• Project Reviewer• Requirements Reviewer• Requirements Specifier• Software Architect• Stakeholder• System Administrator• System Analyst• Technical Writer• Test Analyst• Test Designer• Test Manager• Tester• Tool Specialist• User-Interface Designer• Architectural analysis• Assess Viability of architectural proof-of-concept• Capsule design• Class design• Construct architectural proof-of-concept• Database design• Describe distribution• Describe the run-time architecture• Design test packages and classes• Develop design guidelines• Develop programming guidelines• Identify design elements• Identify design mechanisms• Incorporate design elements• Prioritize use cases• Review the architecture• Review the design• Structure the implementation model• Subsystem design• Use-case analysis• Use-case design• Analysis model• Architectural proof-of-concept• Bill of materials• Business architecture document• Business case• Business glossary• Business modeling guidelines• Business object model• Business rules• Business use case	<ul style="list-style-type: none">• Business use case realization• Business use-case model• Business vision• Change request• Configuration audit findings• Configuration management plan• Data model• Deployment model• Deployment plan• Design guidelines• Design model• Development case• Development-organization assessment• End-user support materials• Glossary• Implementation model• Installation artifacts• Integration build plan• Issues list• Iteration assessment• Iteration plan• Manual styleguide• Programming guidelines• Quality assurance plan• Reference architecture• Release notes• Requirements attributes• Requirements management plan• Review record• Risk list• Risk management plan• Software architecture document• Software development plan• Software requirements specification• Stakeholder requests• Status assessment• Supplementary business specification• Supplementary specification• Target organization assessment• Test automation architecture• Test cases• Test environment configuration• Test evaluation summary• Test guidelines• Test ideas list• Test interface specification• Test plan• Test suite• Tool guidelines• Training materials• Use case model• Use case package• Use-case modeling guidelines• Use-case realization• Use-case storyboard• User-interface guidelines• User-interface prototype• Vision• Work order• Workload analysis model
---	--

More adaptive



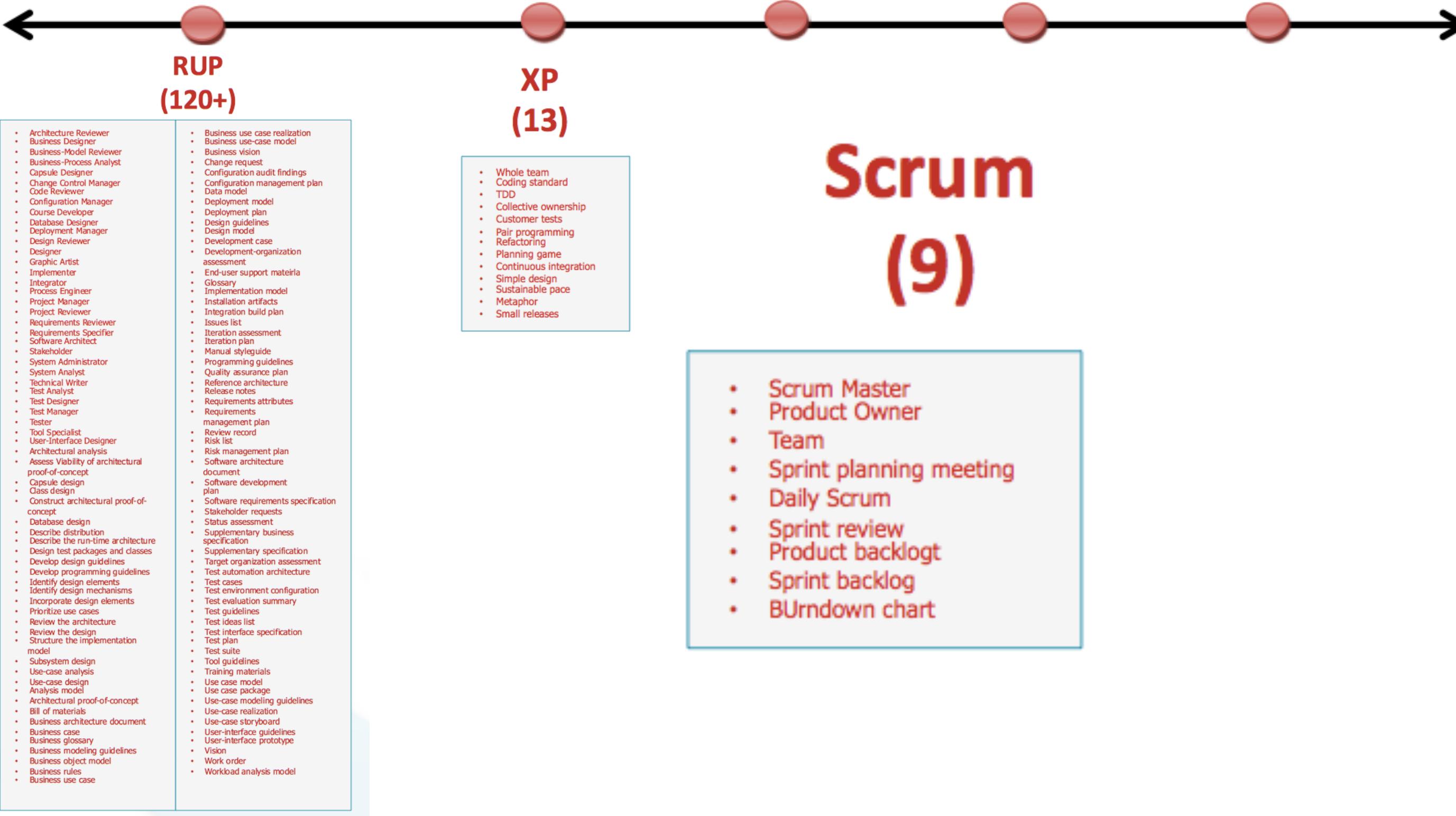
More prescriptive

More adaptive

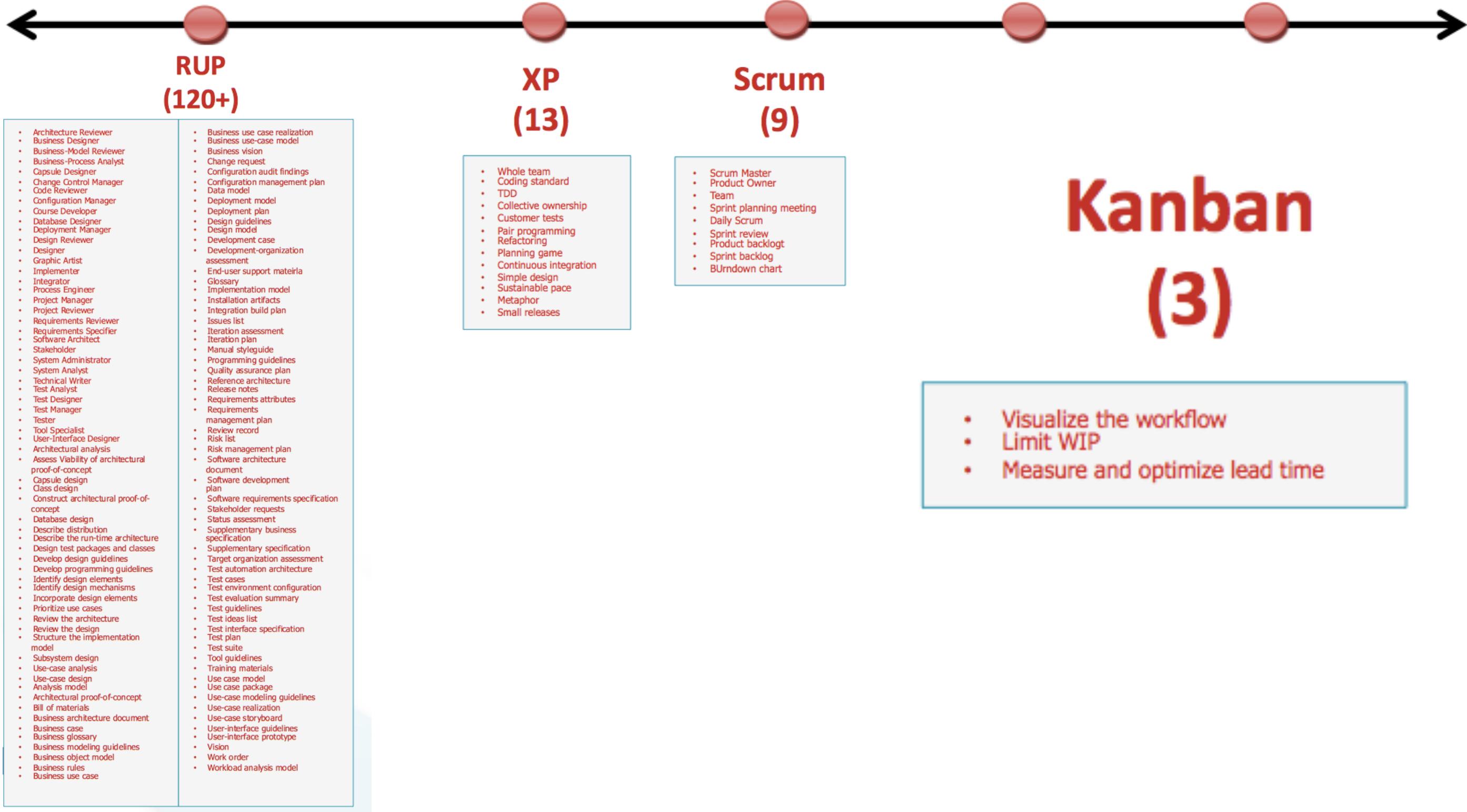


More prescriptive

More adaptive

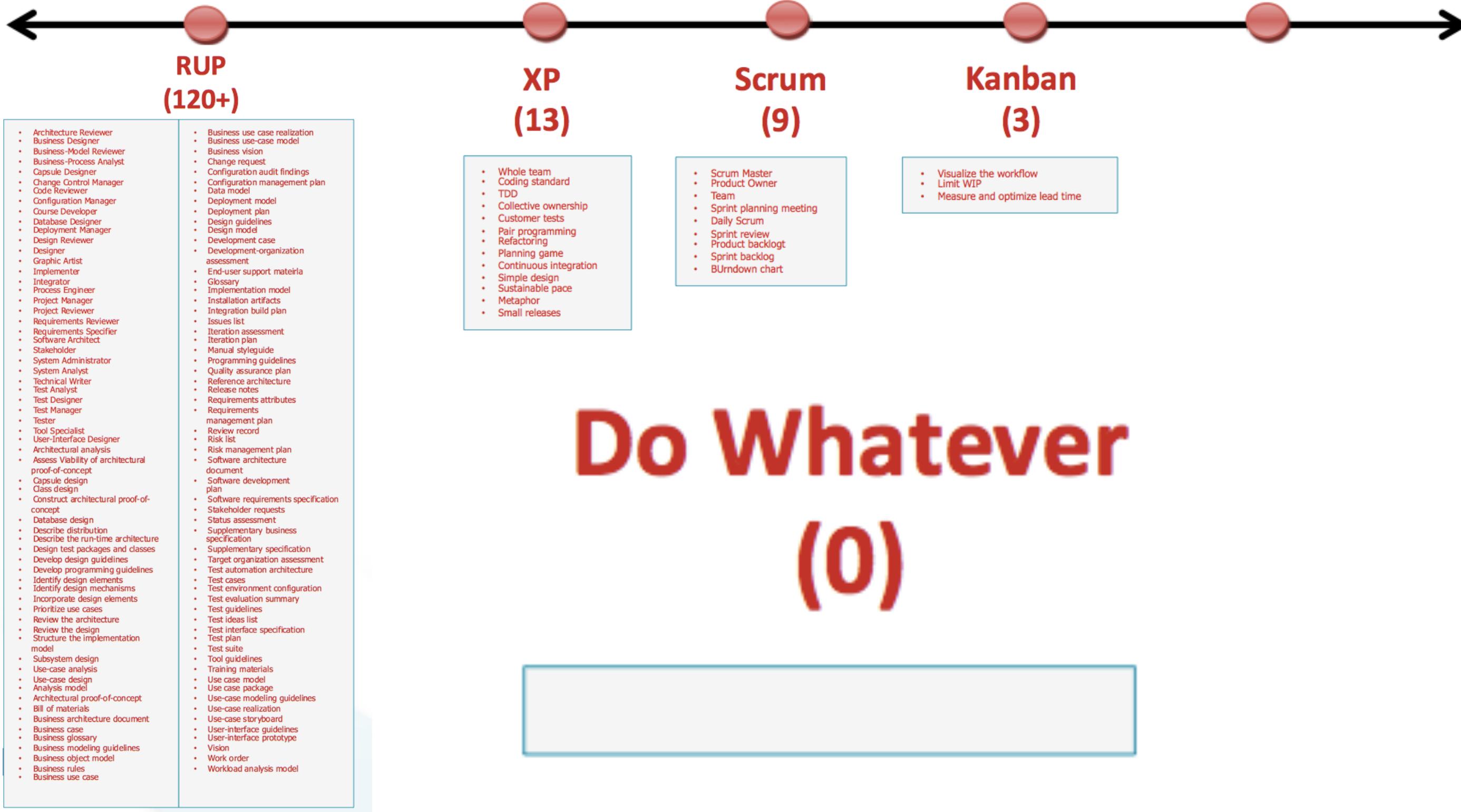


More prescriptive



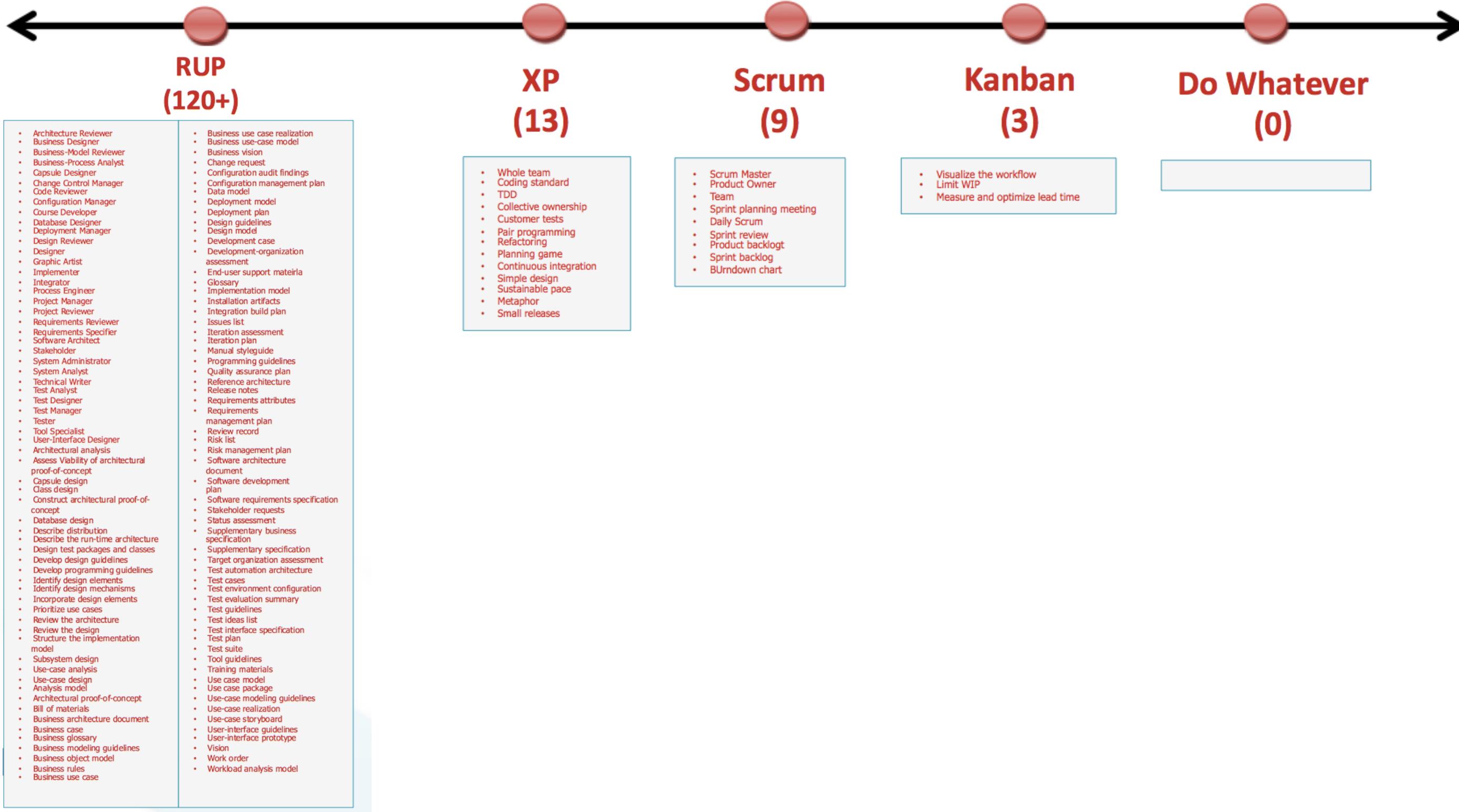
More prescriptive

More adaptive



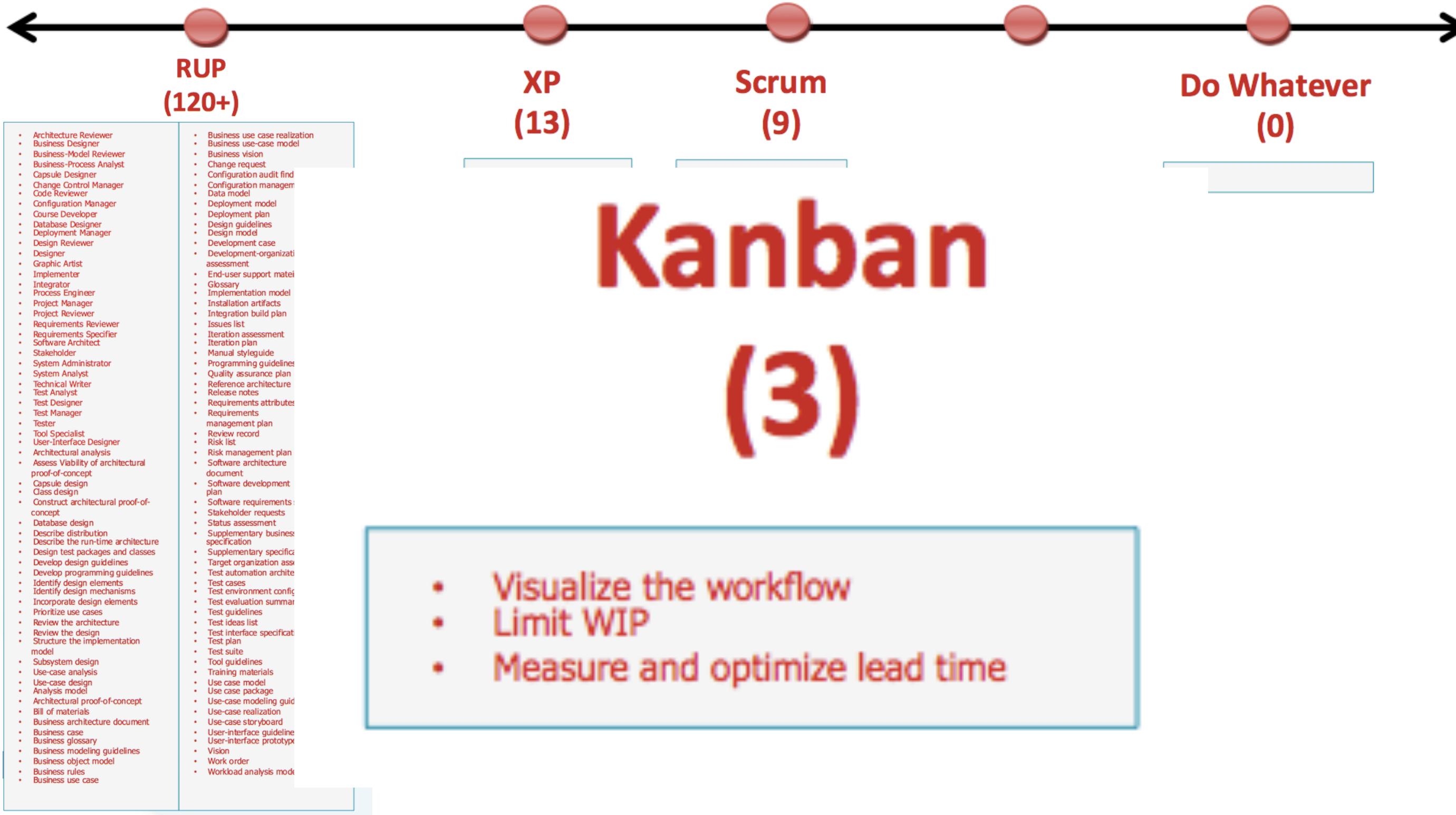
More prescriptive

More adaptive



More prescriptive

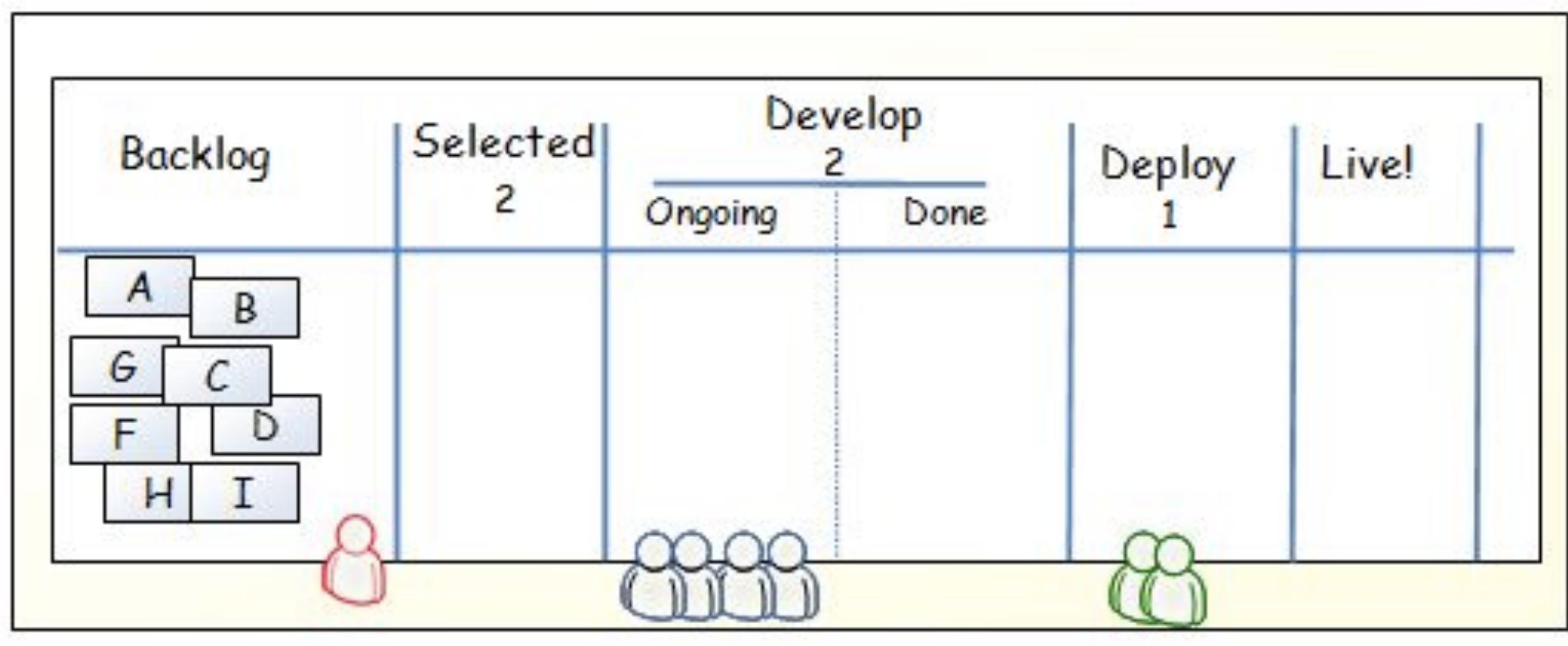
More adaptive



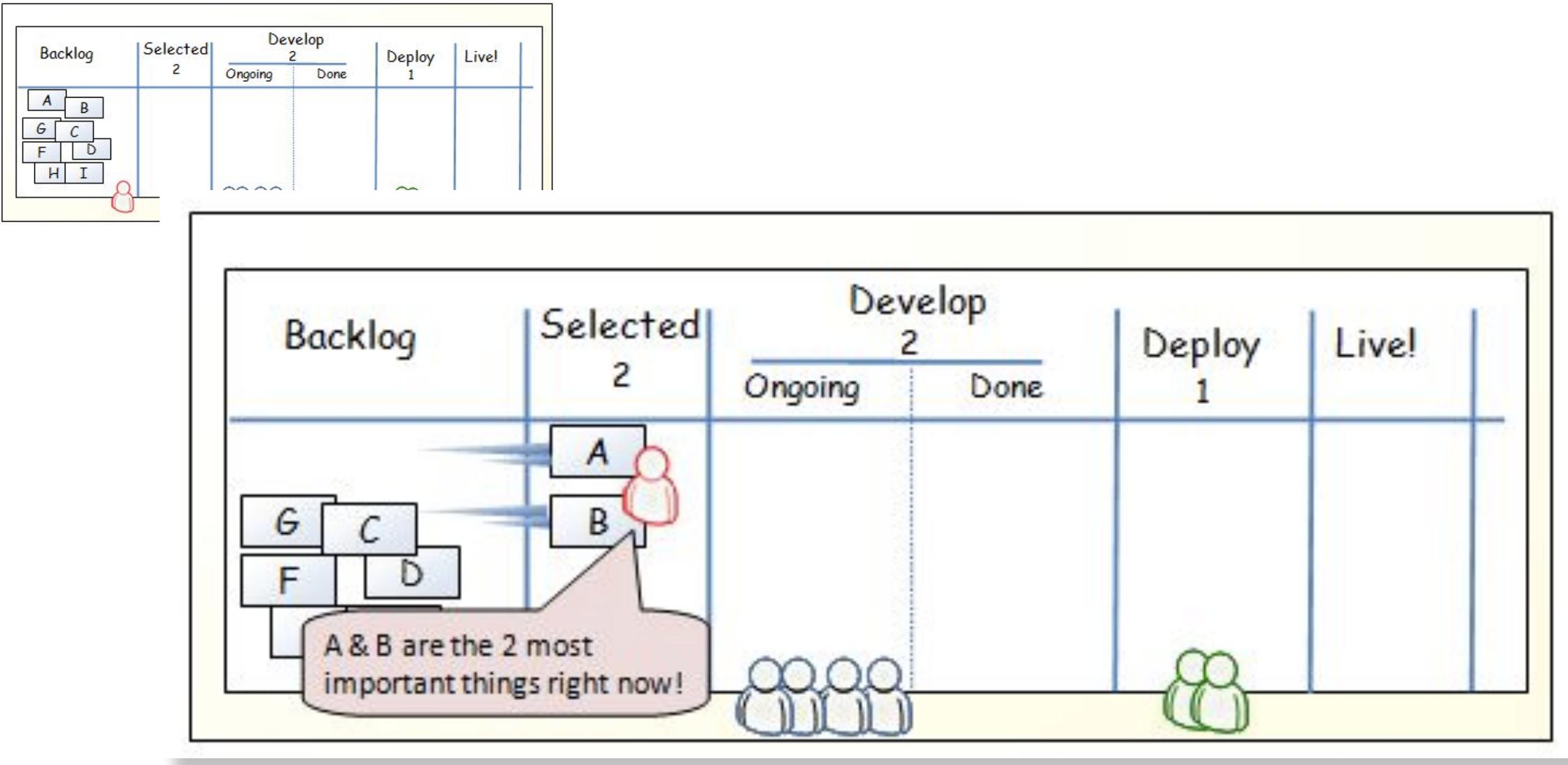
One day in Kanban land

© Henrik Kniberg, [One day in Kanban land](#)

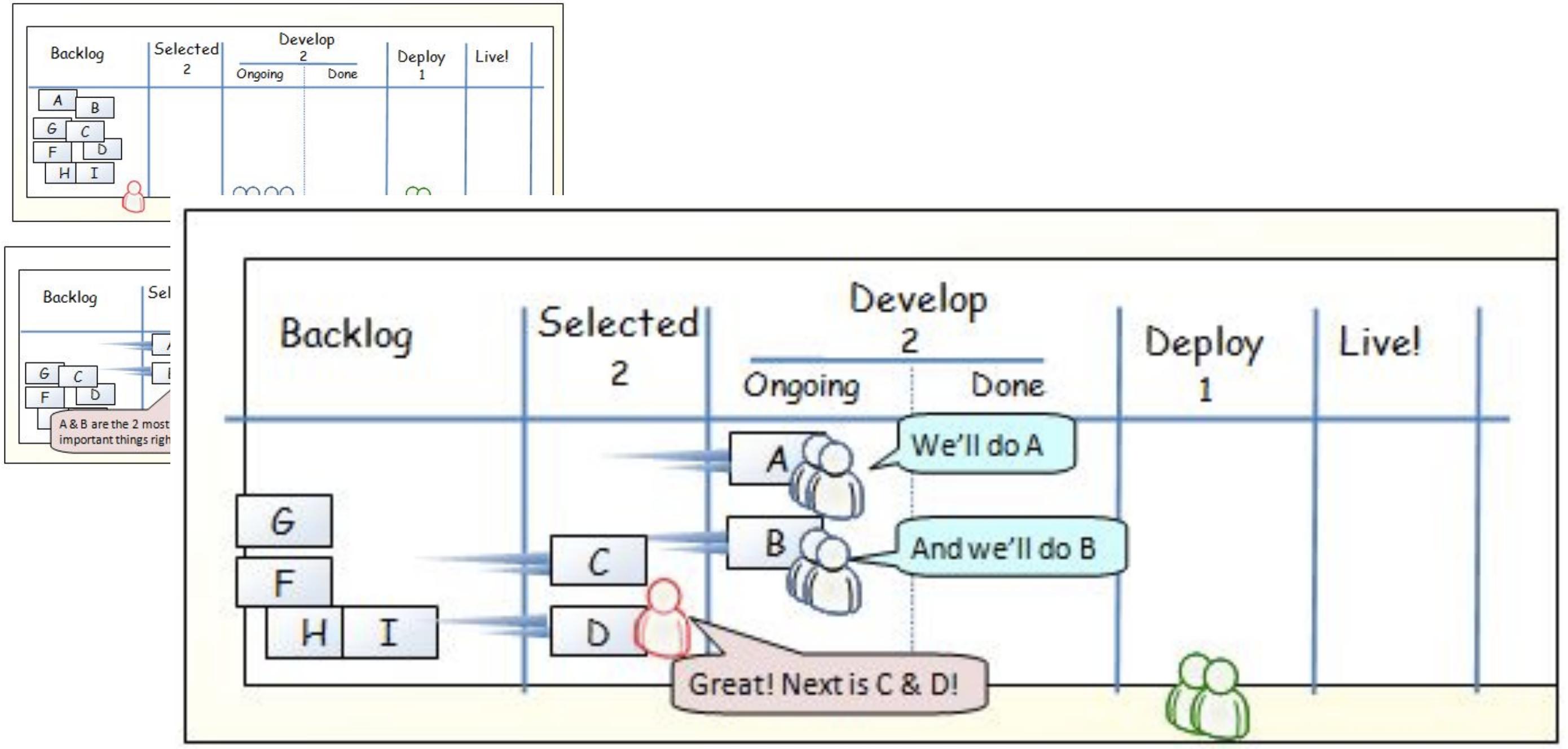
One day in Kanban land (© Henrik Kniberg)



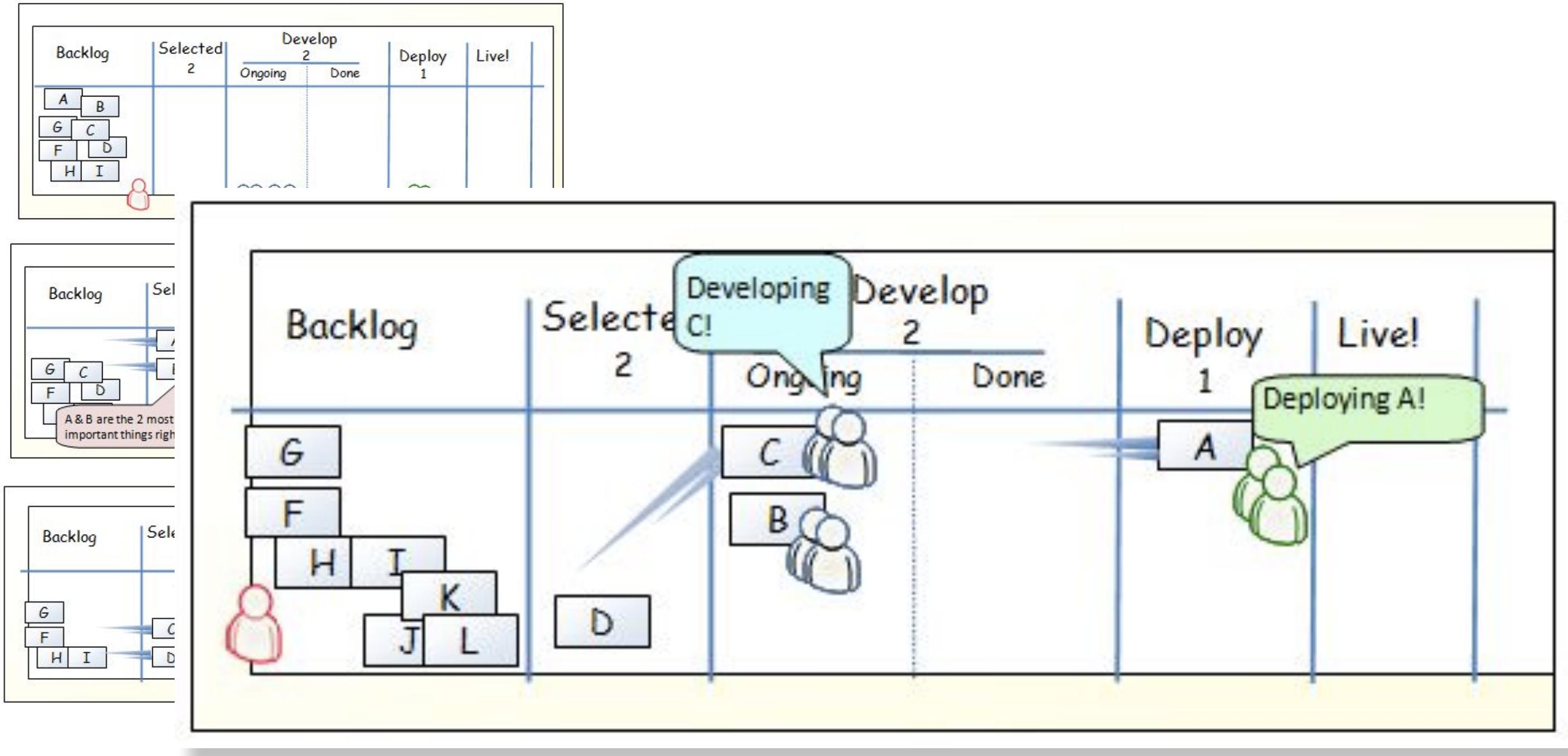
One day in Kanban land (© Henrik Kniberg)



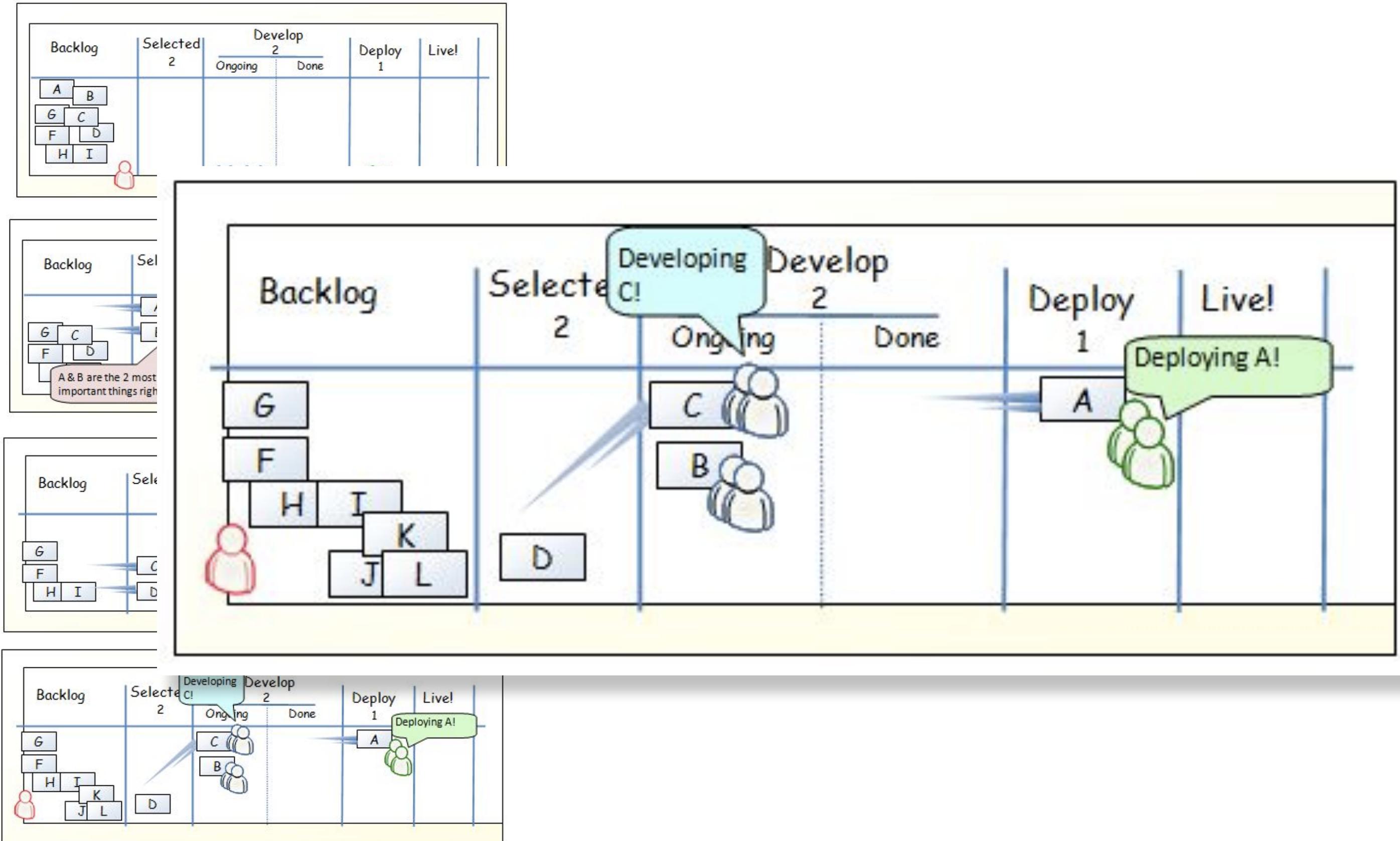
One day in Kanban land (© Henrik Kniberg)



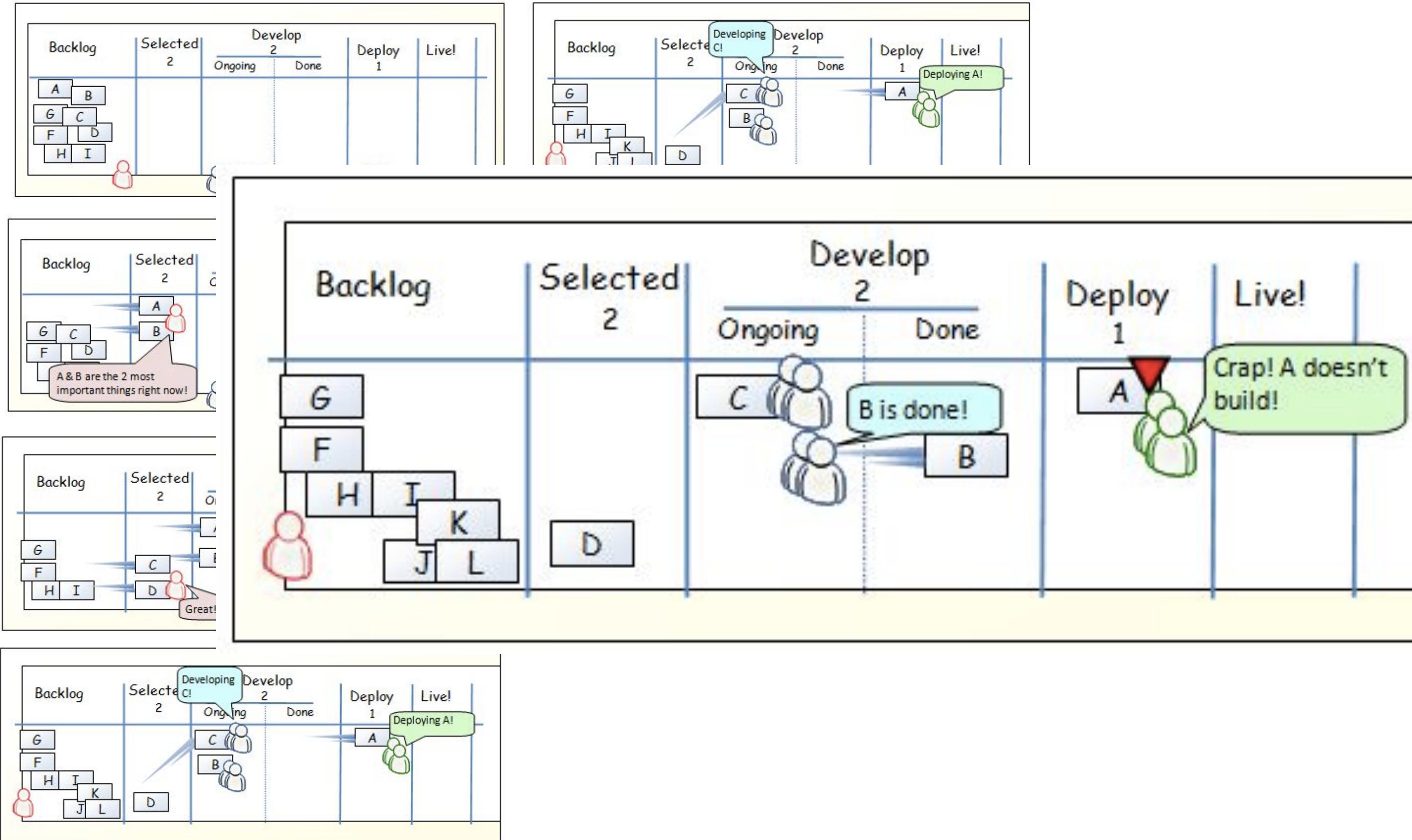
One day in Kanban land (© Henrik Kniberg)



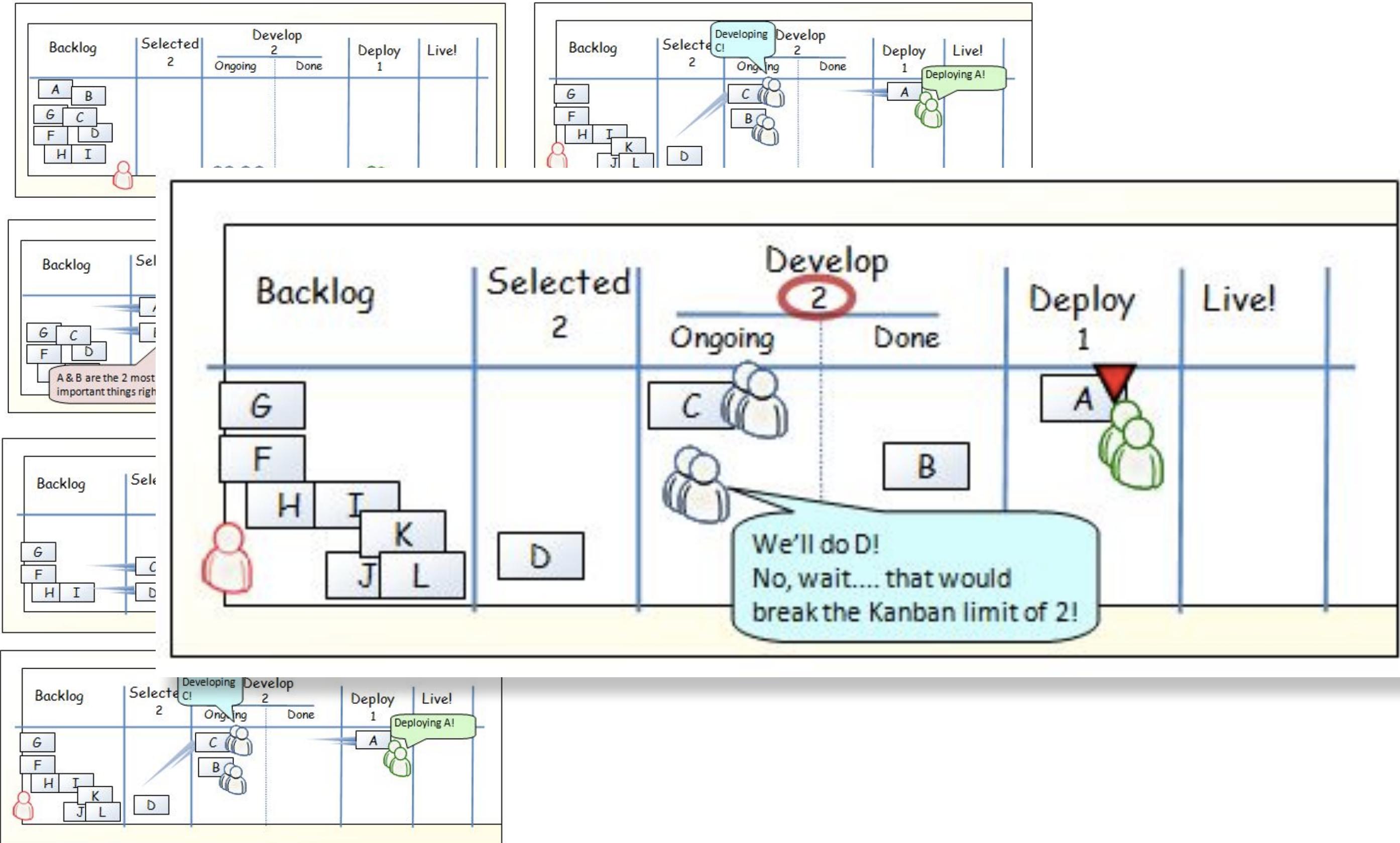
One day in Kanban land (© Henrik Kniberg)



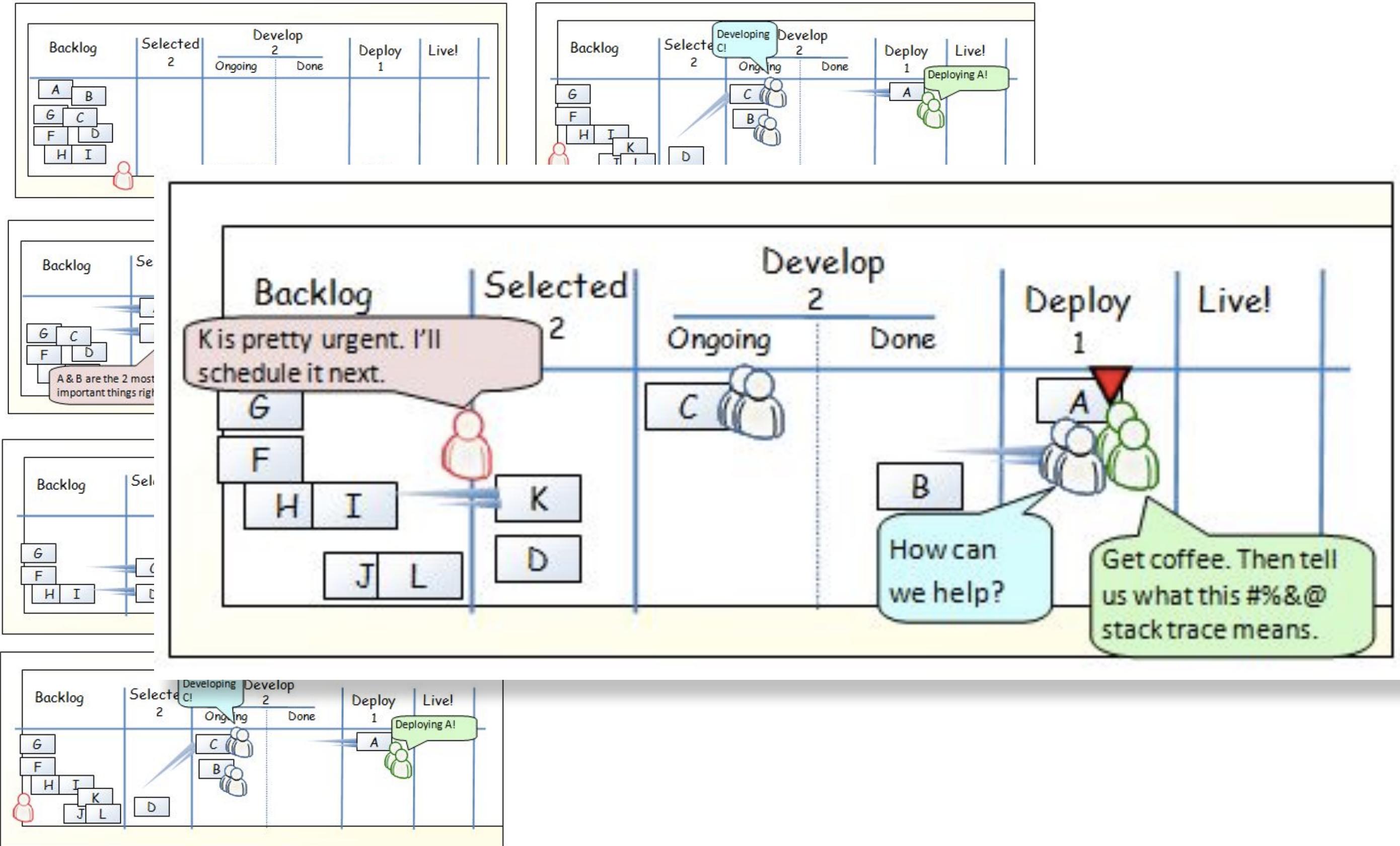
One day in Kanban land (© Henrik Kniberg)



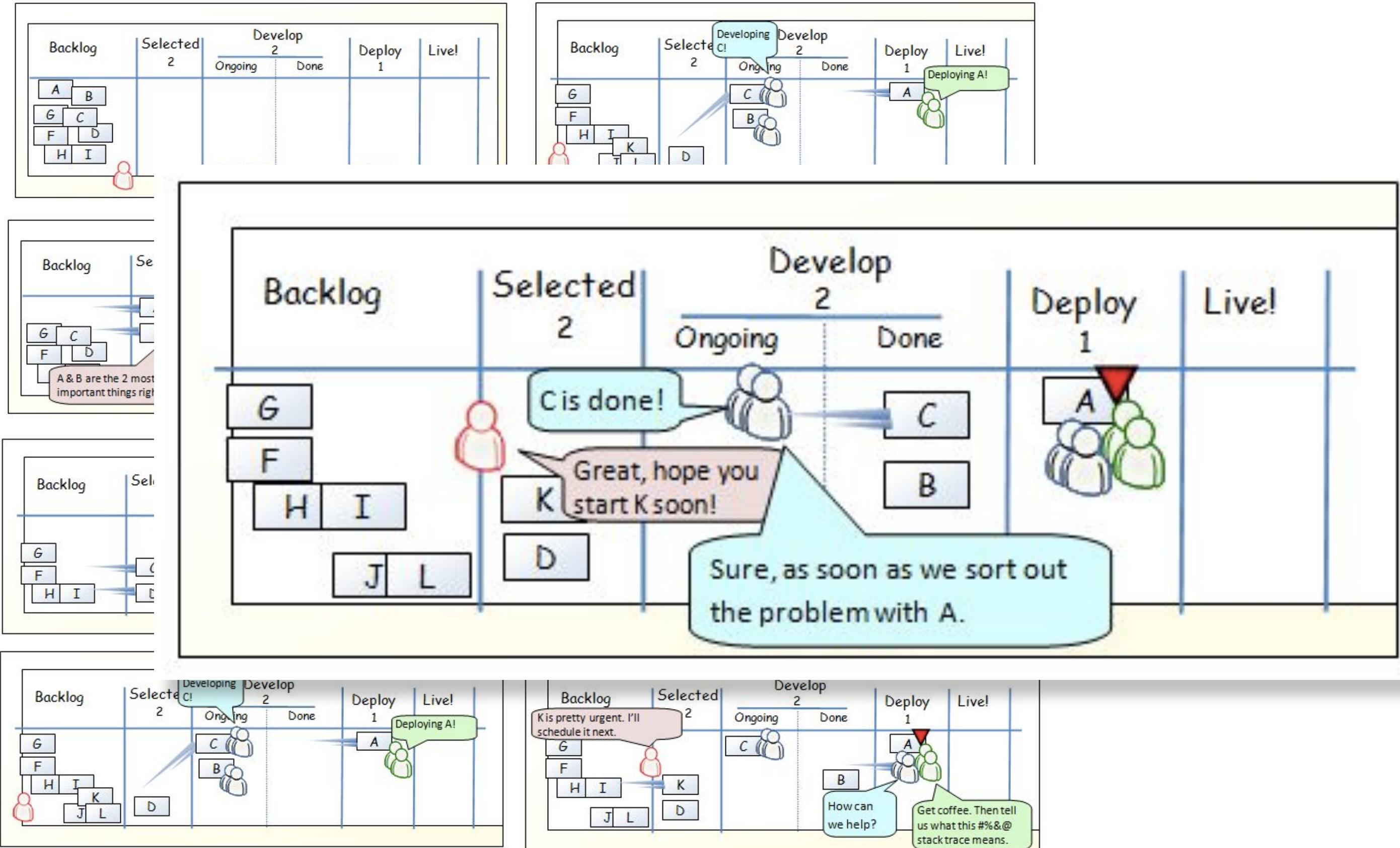
One day in Kanban land (© Henrik Kniberg)



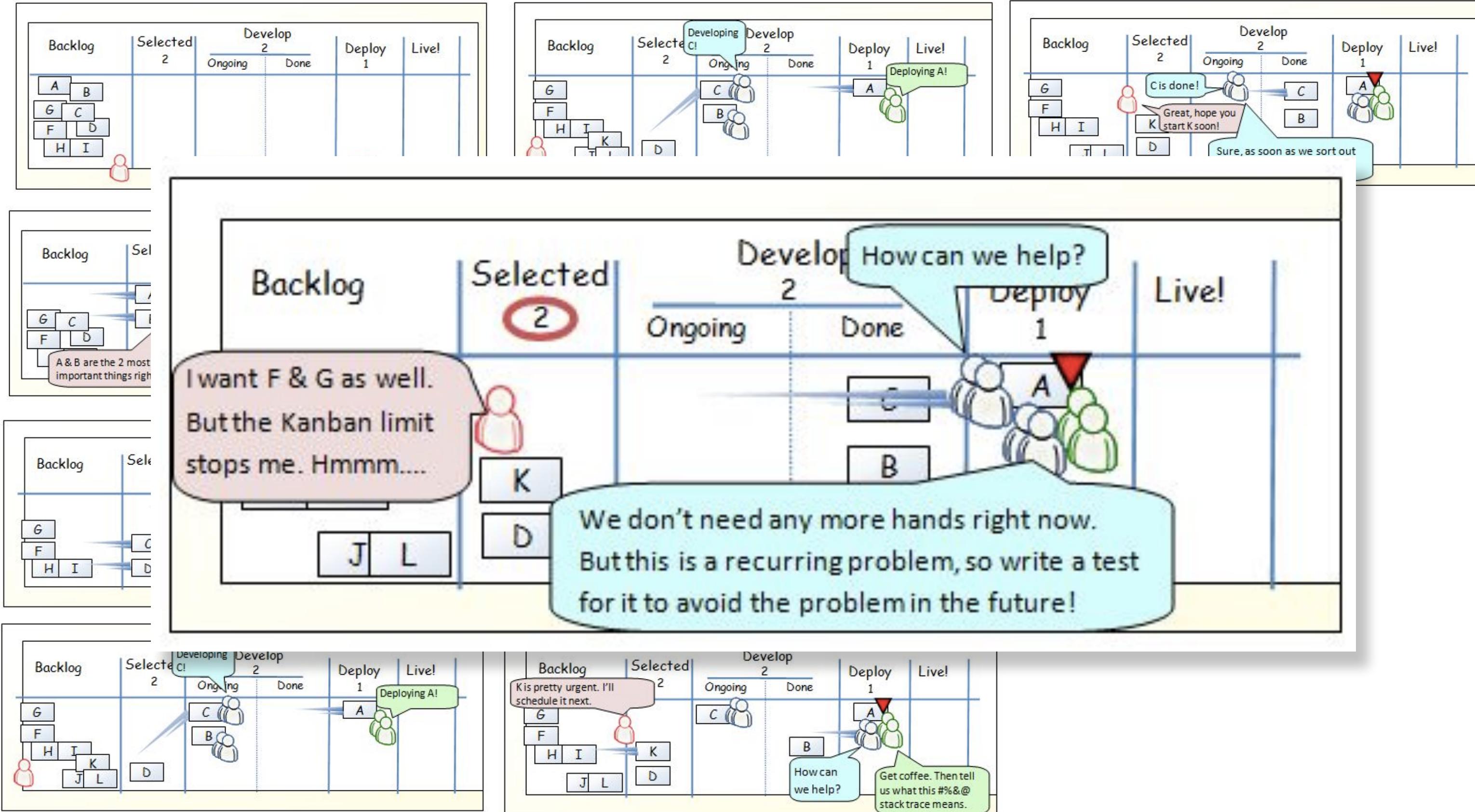
One day in Kanban land (© Henrik Kniberg)



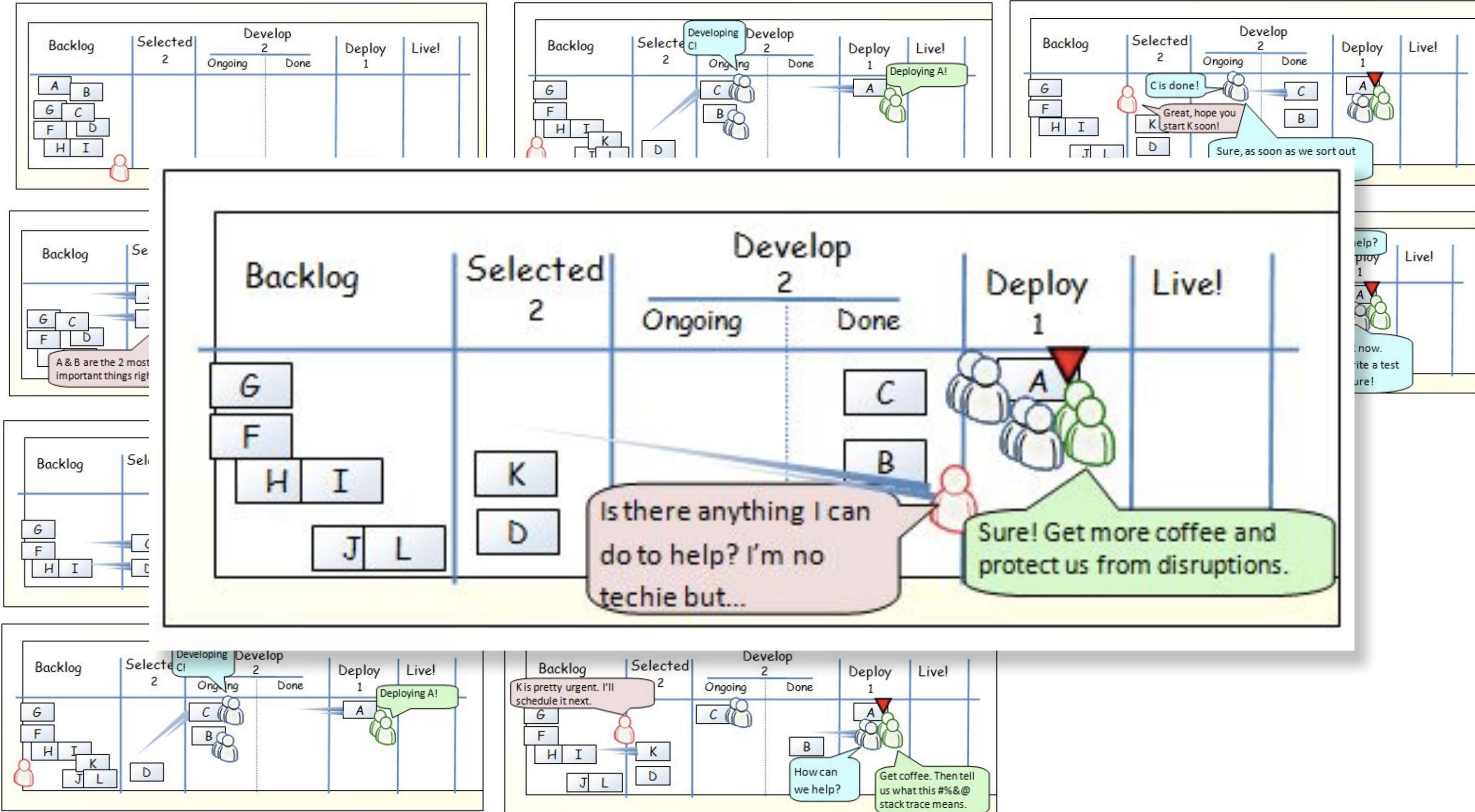
One day in Kanban land (© Henrik Kniberg)



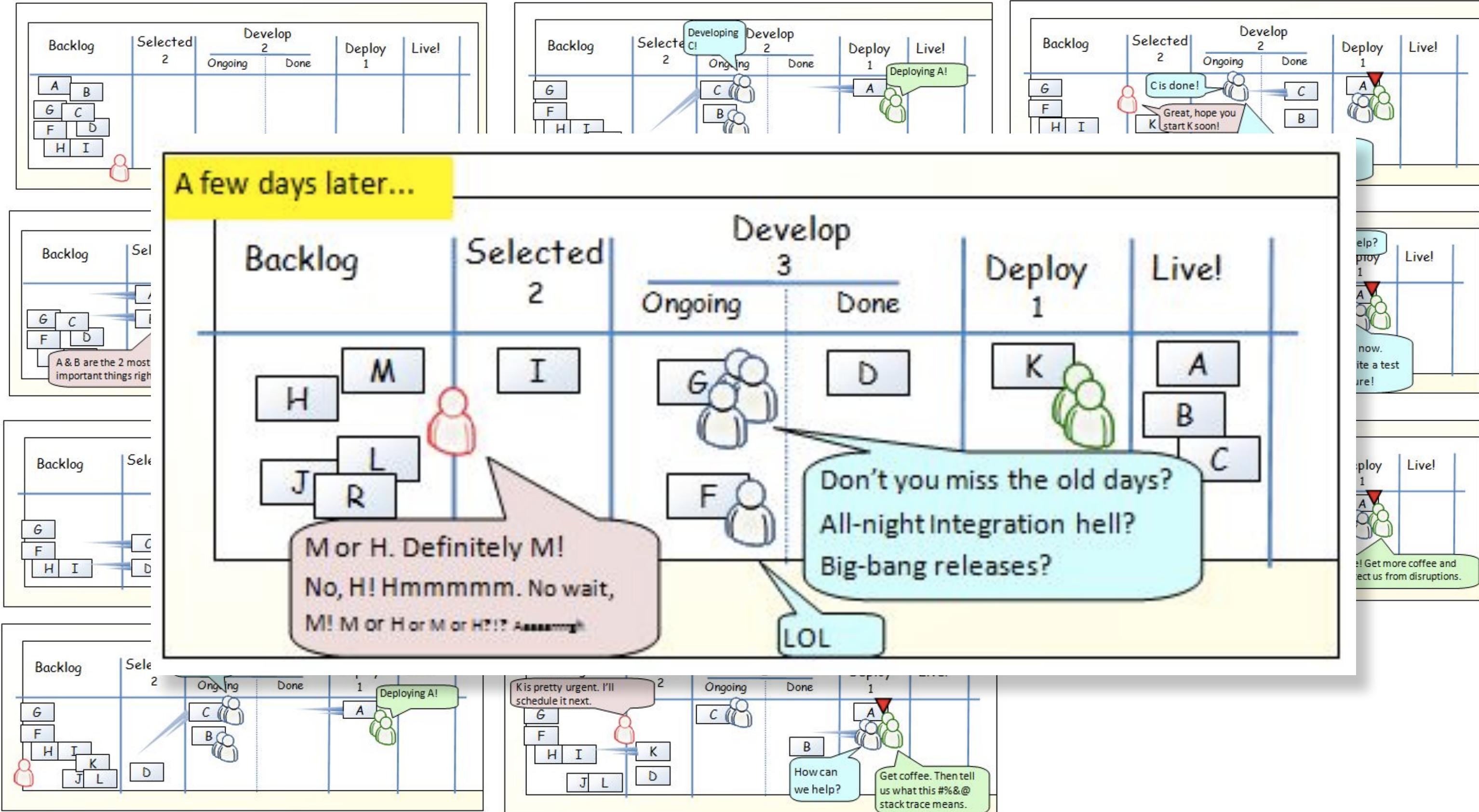
One day in Kanban land (© Henrik Kniberg)



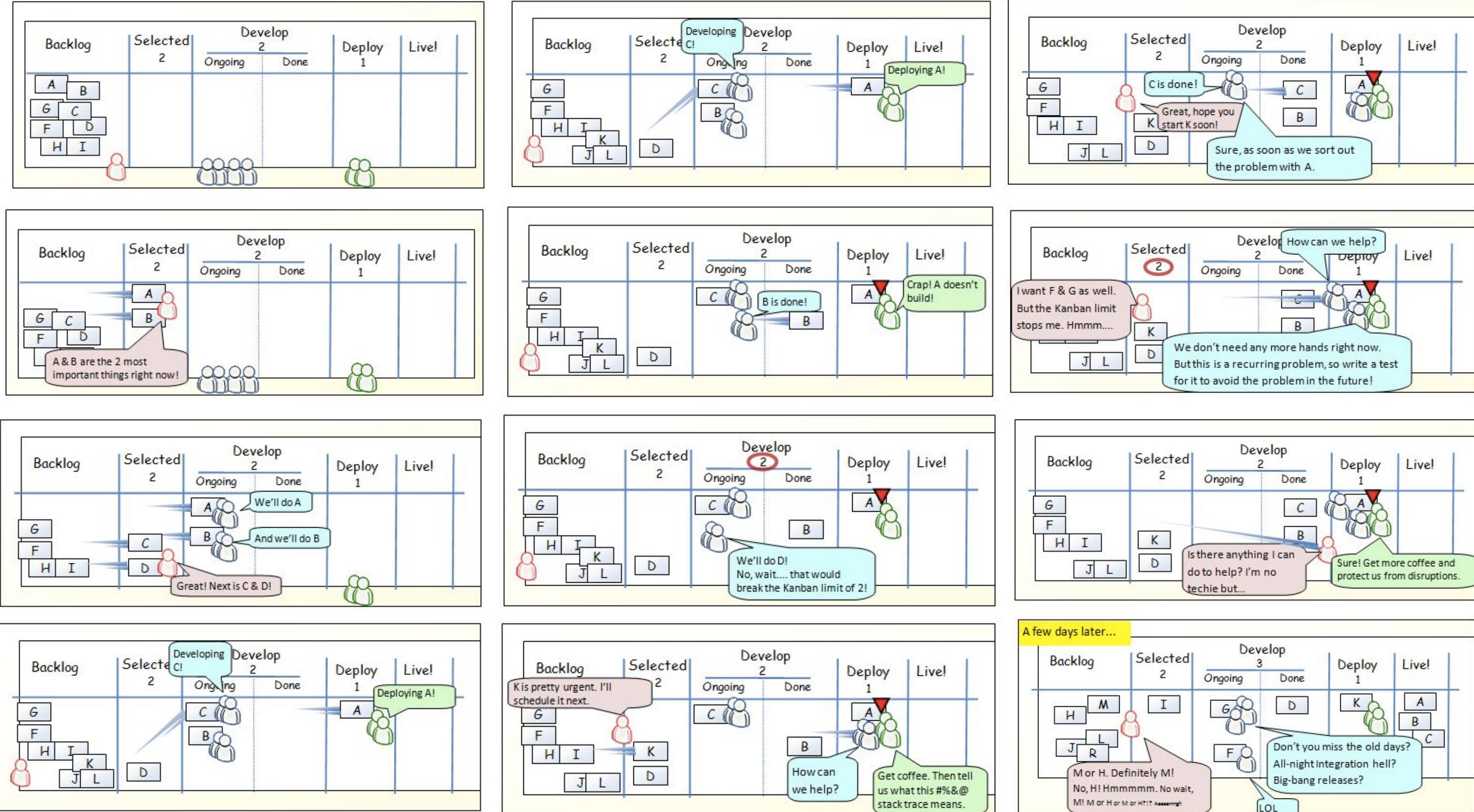
One day in Kanban land (© Henrik Kniberg)



One day in Kanban land (© Henrik Kniberg)



One day in Kanban land (© Henrik Kniberg)



1. Visualizar el flujo de trabajo

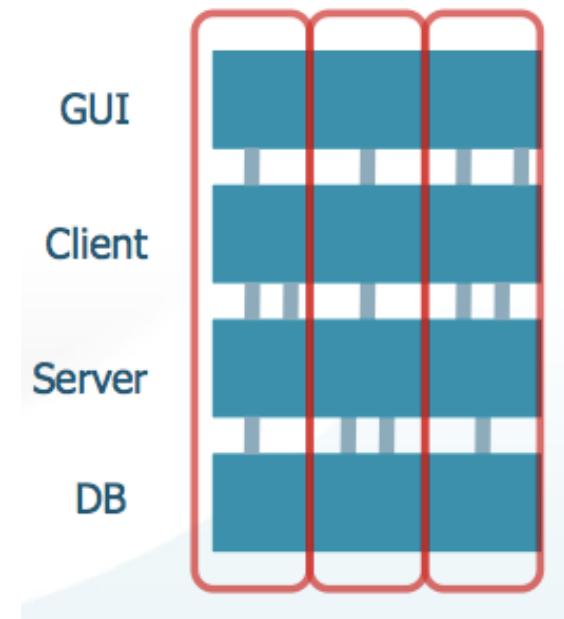
Flujo de trabajo

- Cada empresa de desarrollo de software tiene su propio flujo de trabajo
- Para aplicar Kanban debemos empezar identificando el flujo de trabajo
 - ¿Qué son los ítems de trabajo (*work items*)?
 - ¿Por qué fases pasan?
 - ¿Existen distintos tipos de ítems?
 - ¿Su tamaño tiene mucha variabilidad?

Historias de usuario e ítems de trabajo

- Si las historias de usuario tienen un tamaño demasiado grande, las debemos dividir en tareas más pequeñas
- Todas las historias deben ir acompañadas de criterios de aceptación (lo primero que debemos hacer si desarrollamos con TDD)
- Dividiremos las tareas en ítems de trabajo que representaremos como etiquetas que pegaremos en el tablero Kanban
- Los ítems de trabajo sí que pueden ser partes “horizontales” del proyecto
- Los ítems de trabajo tampoco pueden ser demasiado pequeños

Una historia no se resuelve con un método de un API. Debemos implementar todos sus aspectos: interfaz de usuario, código de cliente, servidor, base de datos, etc.



Independent
Negotiable
Valuable
Estimable
Small
Testable

Acrónimo creado por Bill Wake
www.xp123.com

Post-it con el ítem de trabajo

- Descripción del ítem de trabajo.
Concisa, precisa y entendible por todos los miembros del equipo.
- Fecha de comienzo
- Quién está trabajando en el ítem. Lo ideal sería un avatar pegado encima de la etiqueta.
- Plazo de finalización
- Código de identificación con el que podemos encontrar más información sobre el ítem en otro lugar (Google Docs, etc.)
- Indicador de progreso (cuánto se ha avanzado en el ítem)



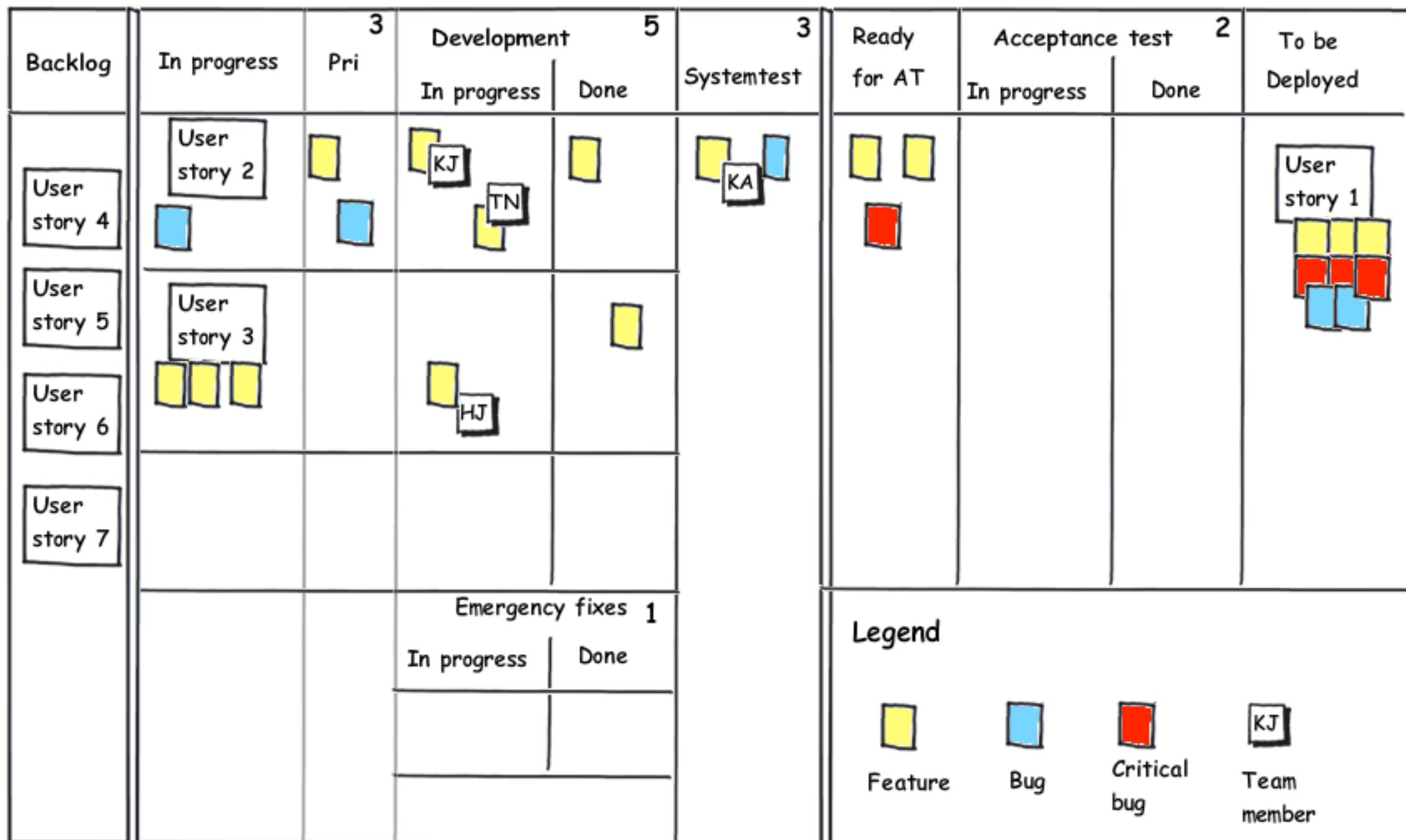
@ Avatars: [nitsnets](#)

El tablero Kanban

- El objetivo principal del tablero Kanban es mostrar visualmente en qué está trabajando el equipo en un momento dado
- Principales utilidades:
 - Elemento fundamental para la reflexión, la comunicación y la discusión en las reuniones de pie diarias
 - Medir distintas métricas relacionadas con el flujo: número de ítems terminados por semana, WIP
 - Definir políticas estrictas de flujo de trabajo y de terminación (*Definition of Done*)

Tablero Kanban

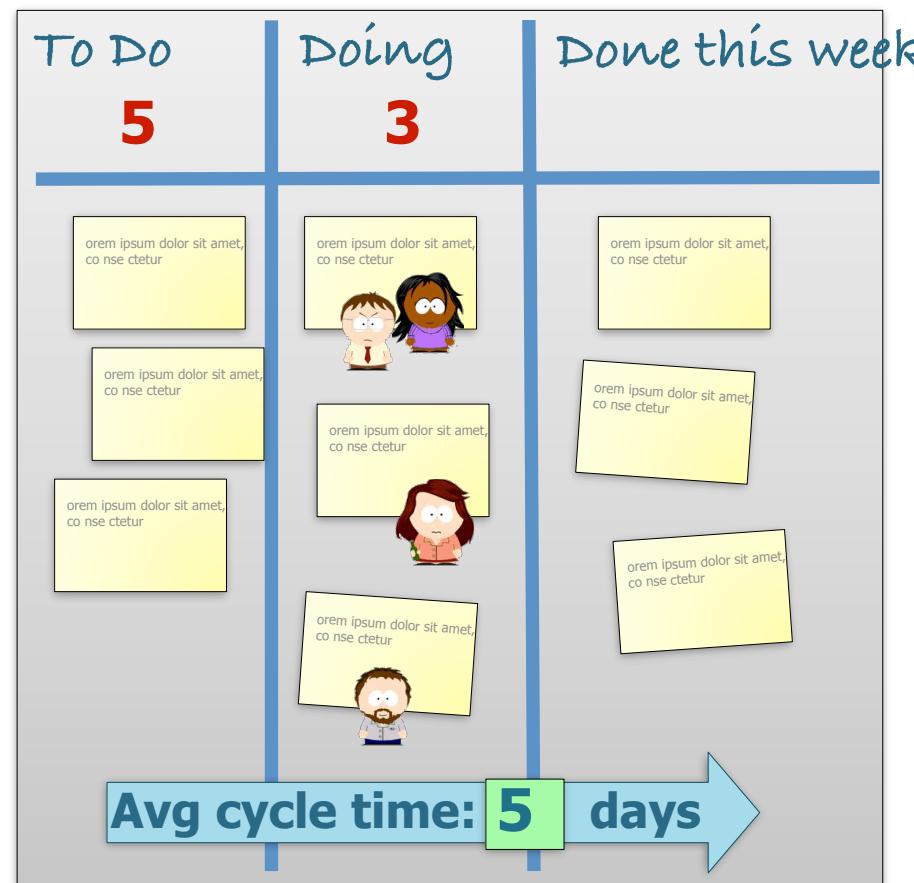
Kanban board



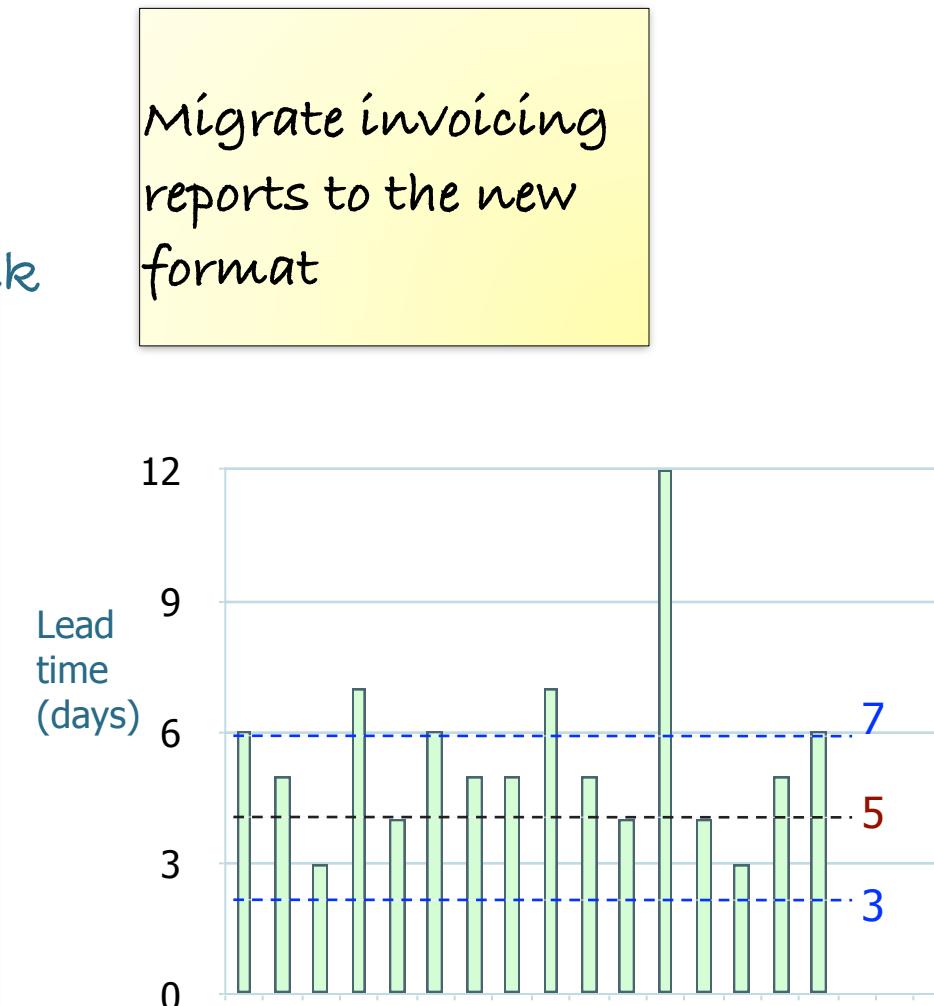
created with Balsamiq Mockups - www.balsamiq.com

<http://ketiljensen.wordpress.com/2009/10/31/kanban-the-next-step-in-the-agile-evolution/>

Un sistema Kanban sencillo

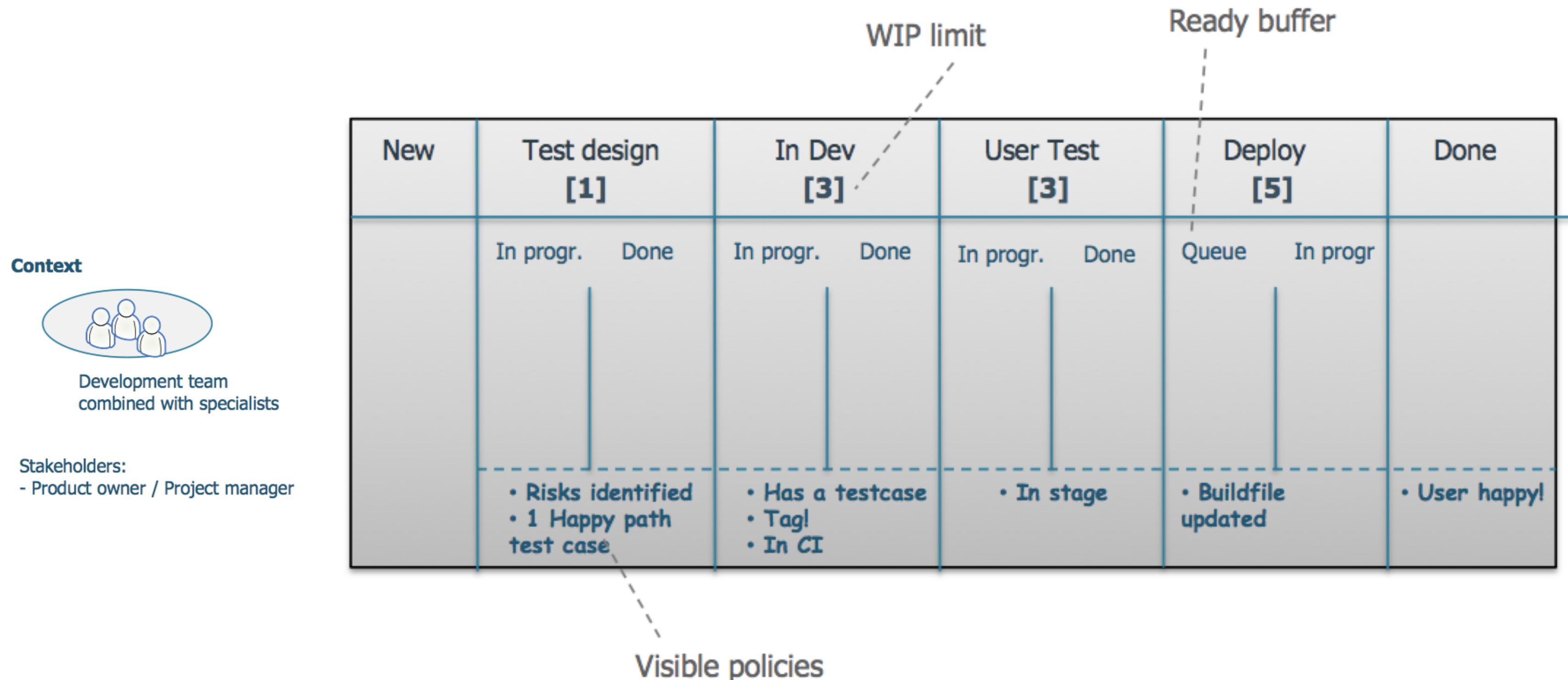


w1	w2	w3	w4	w5	w6		
8	10	7	7	9			



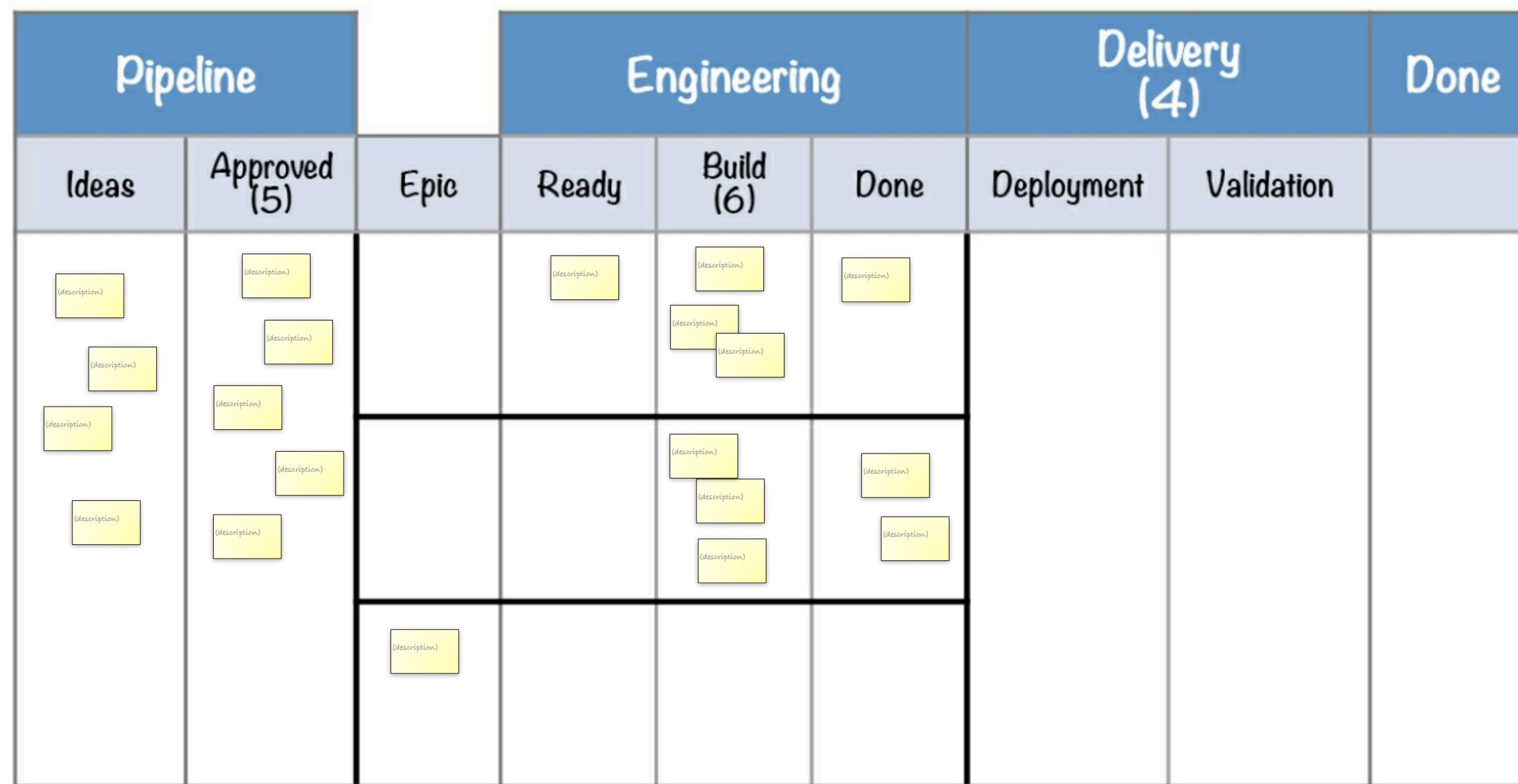
© Henrik Kniberg

Ejemplos de tableros Kanban



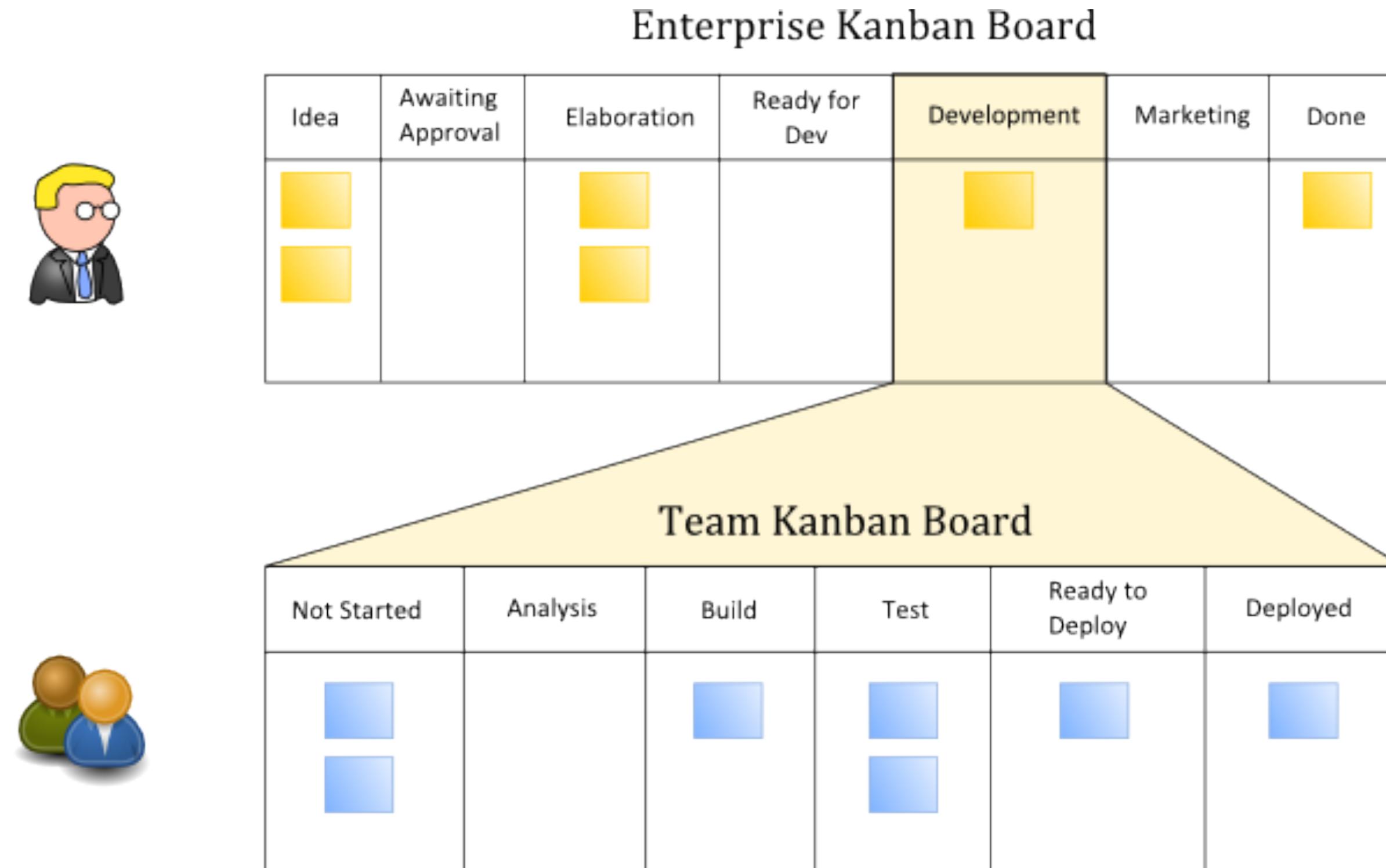
Tableros para features y tareas

- ¿Cómo representar en un mismo tablero features (historias de usuario) y las tareas en las que se descompone?
- Una solución es la propuesta que hace Mike Burrows en su libro "Kanban from the Inside" de tablero multi-nivel que usa el patrón expandir-colapsar. Llama "Epic" a lo que nosotros denominamos feature o historia de usuario.



Ejemplos de tableros Kanban

- Tableros Kanban **multi-nivel**



Estados en paralelo

- En todos los ejemplos que hemos mostrado los ítems cambian de estado de forma secuencial.
- Es posible incorporar cambios de estado paralelos mostrando los estados como tickets dentro del ítem de trabajo.
- También se puede mostrar un estado (por ejemplo: defectuoso) con un pos-it sobre el ítem de trabajo.
 - De esta forma un ítem con un defecto lo podemos dejar visible en su columna actual, sin volverlo a llevar a columnas retrasadas.
 - También se puede al mismo tiempo crear un nuevo ítem de trabajo de tipo *bug* para resolver el error.

- Performance tested**
- User approved**

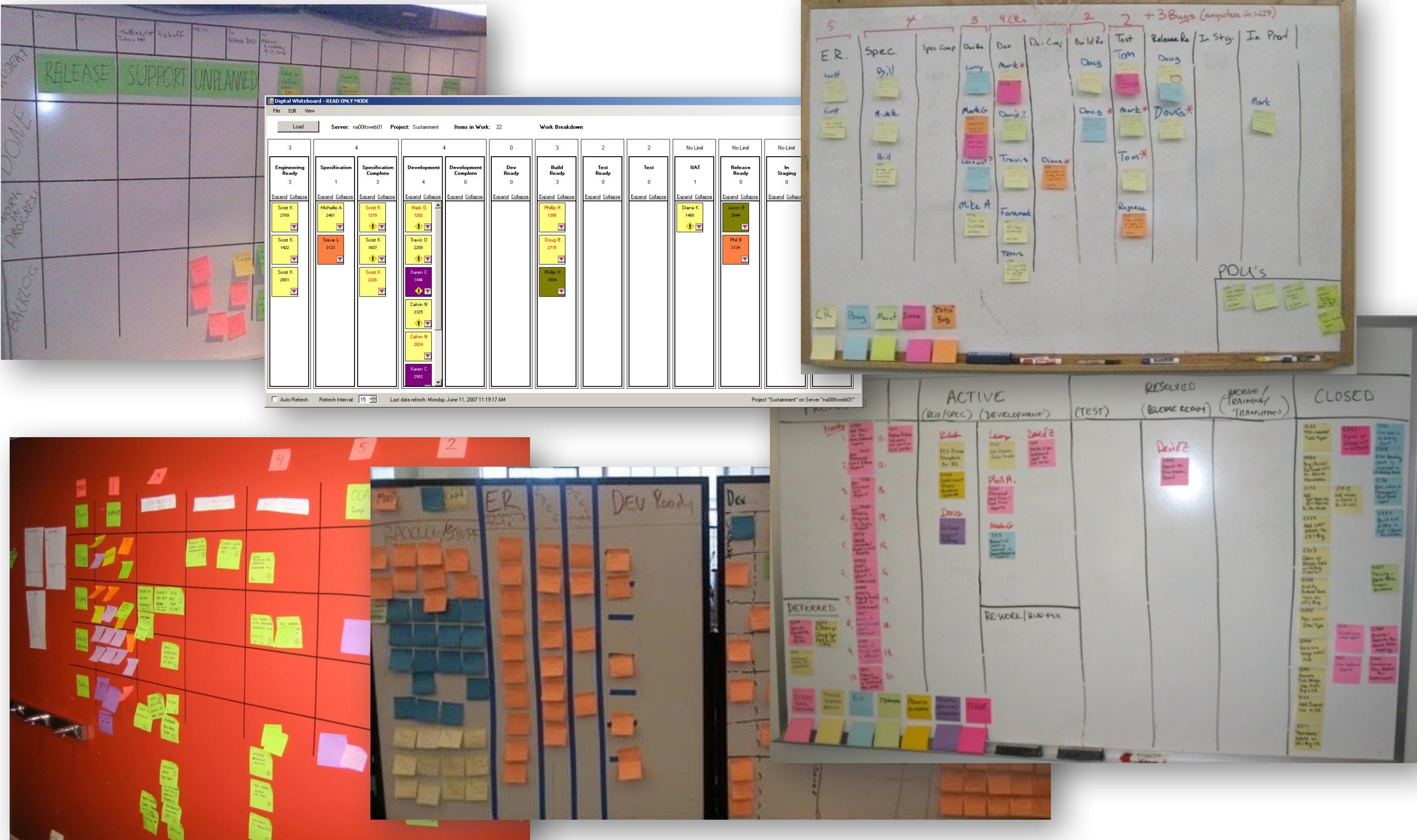
268: Curve auto-close

- Performance
- User approve

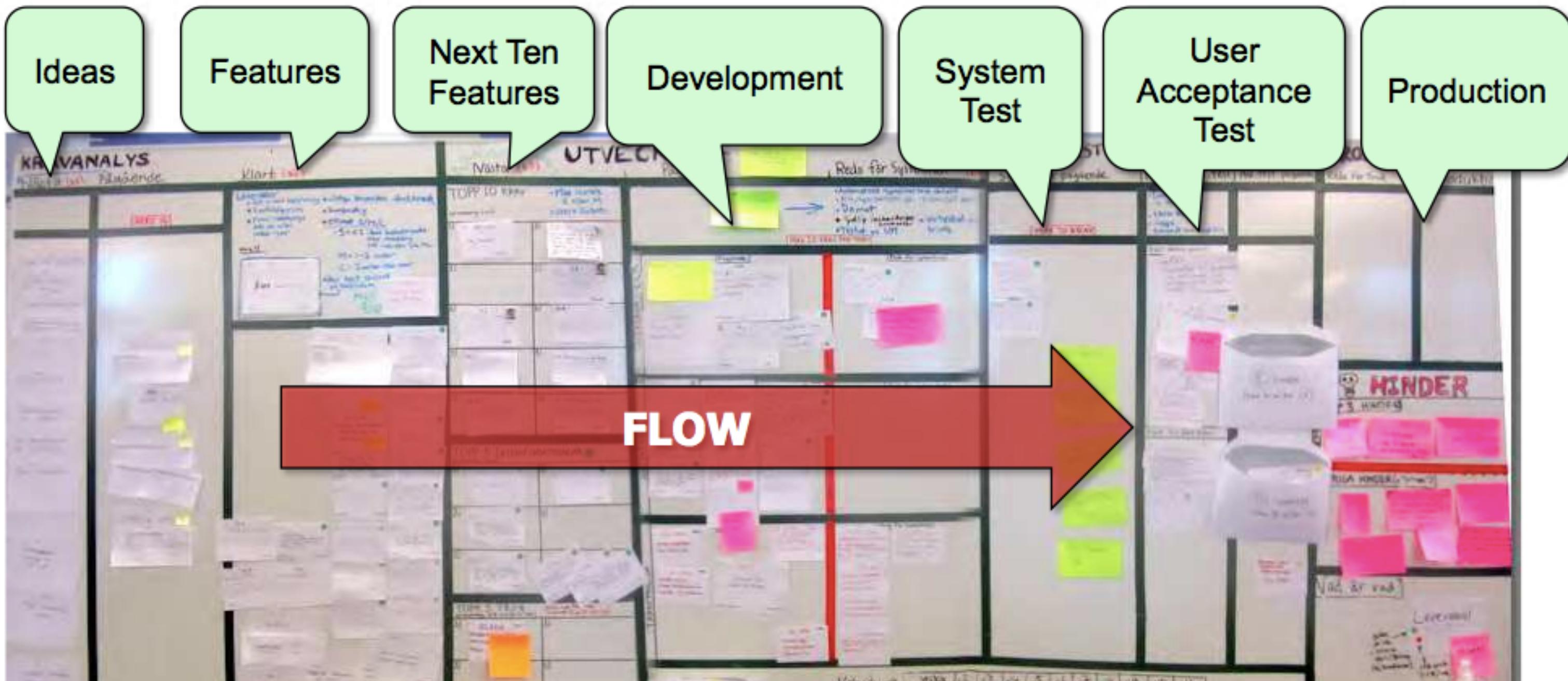
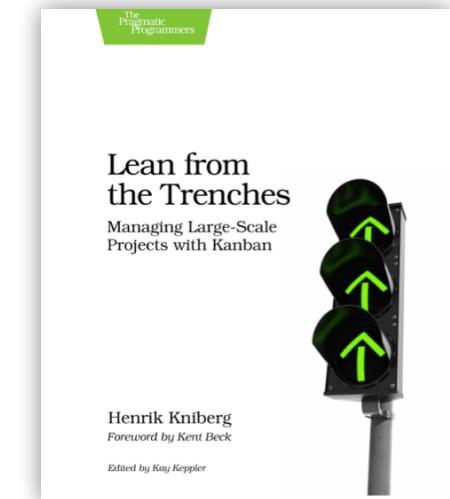
MB 23/1

Permissions model!

Kanban evoluciona en cada equipo

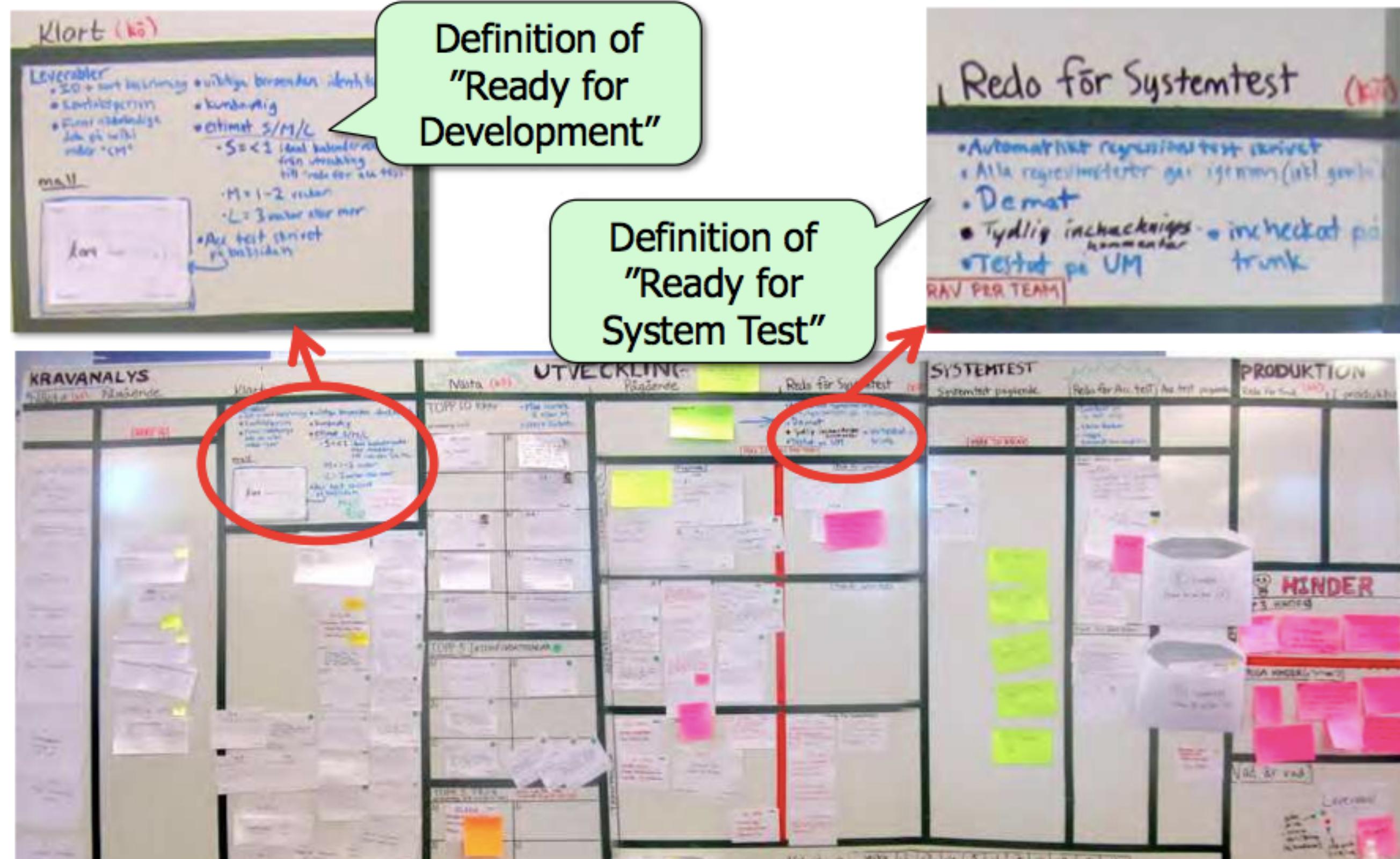


Un ejemplo de tablero - Lean from the Trenches

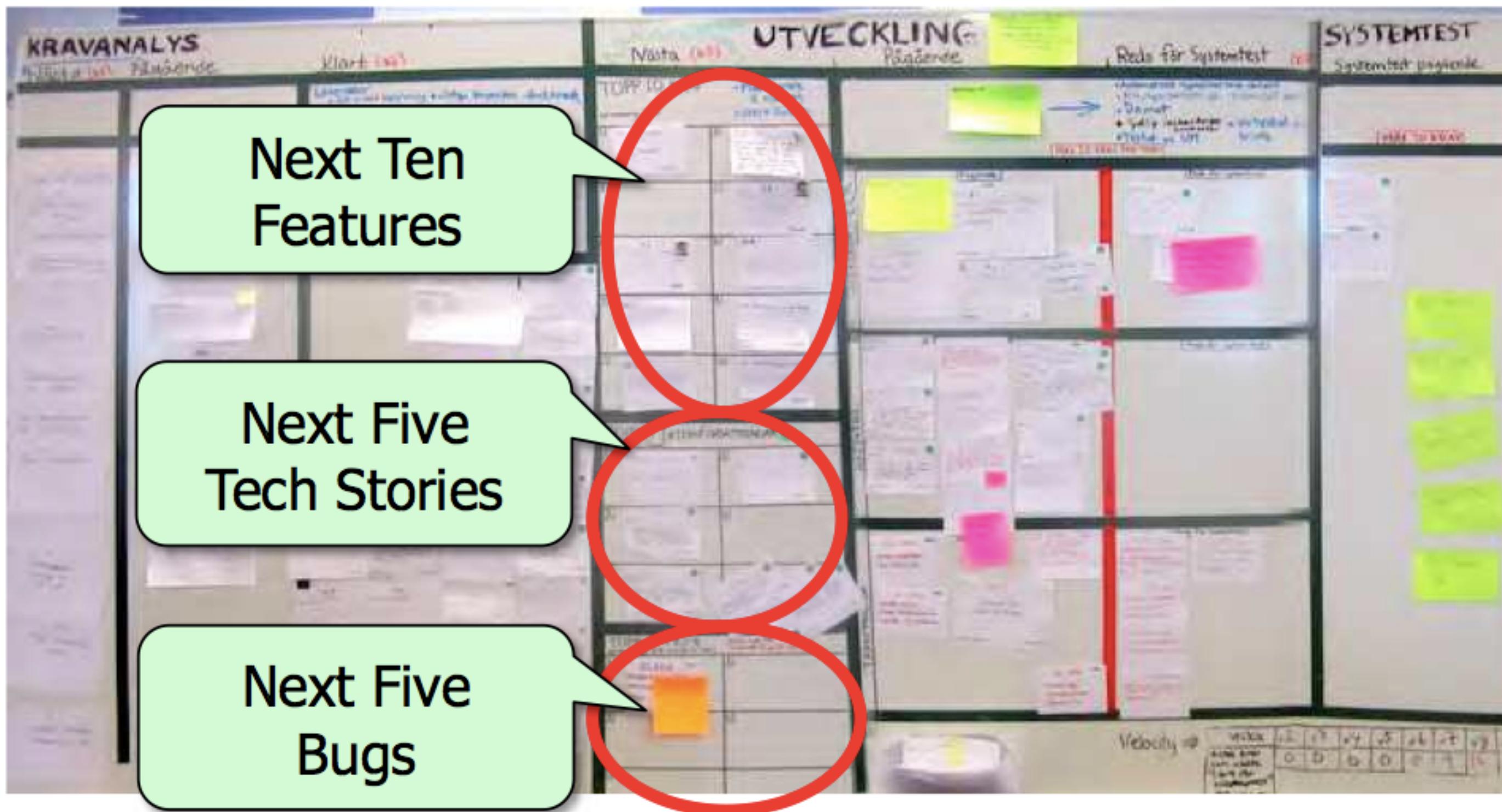


Henrik Kniberg, [Lean From the Trenches](#)

Políticas en el tablero

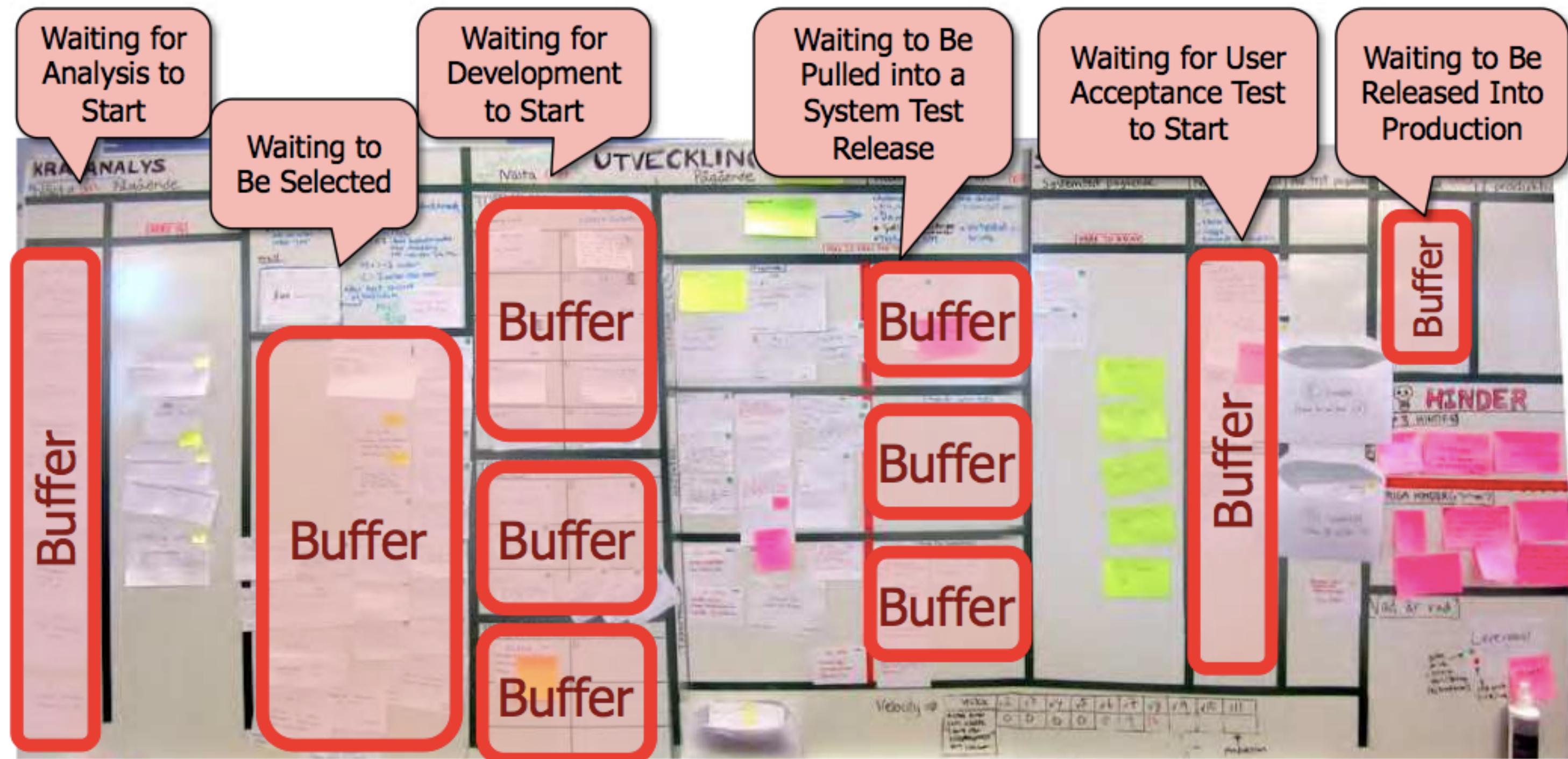


Diferentes tipos de trabajo



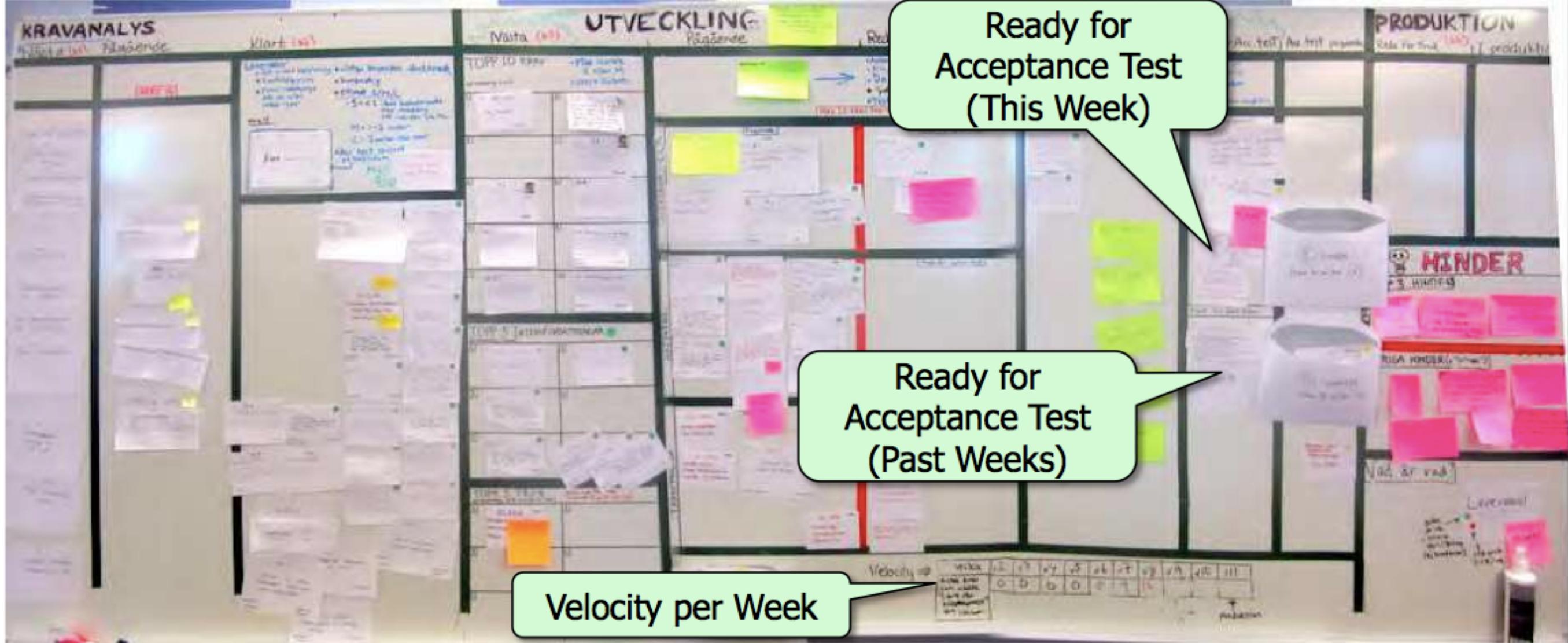
Henrik Kniberg, [Lean From the Trenches](#)

Buffers



Henrik Kniberg, [Lean From the Trenches](#)

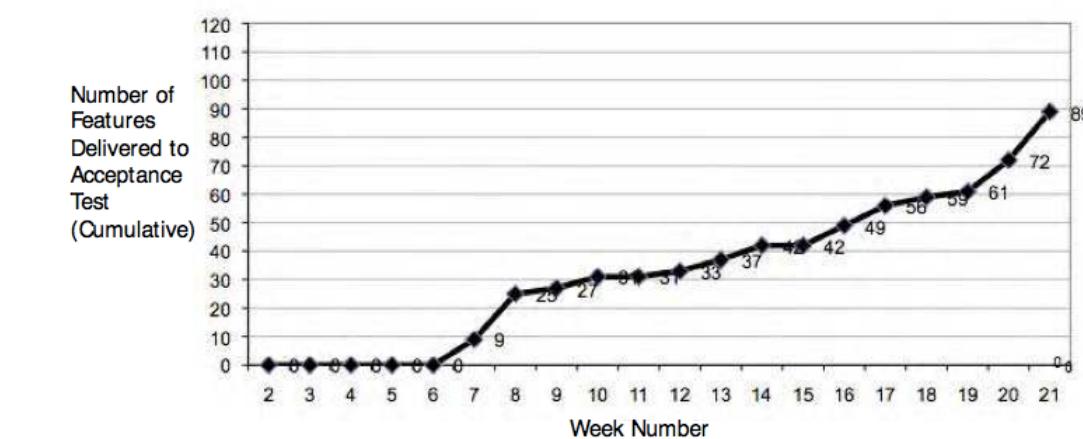
Estadísticas



Vecka	v10	v11	v12	v13	v14	v15	v16	v17	v18
Antal nya funktioner som nått till 'Redo för AcTest'	4	0	2	4	5	0			

↑
prodsätt.

VEL

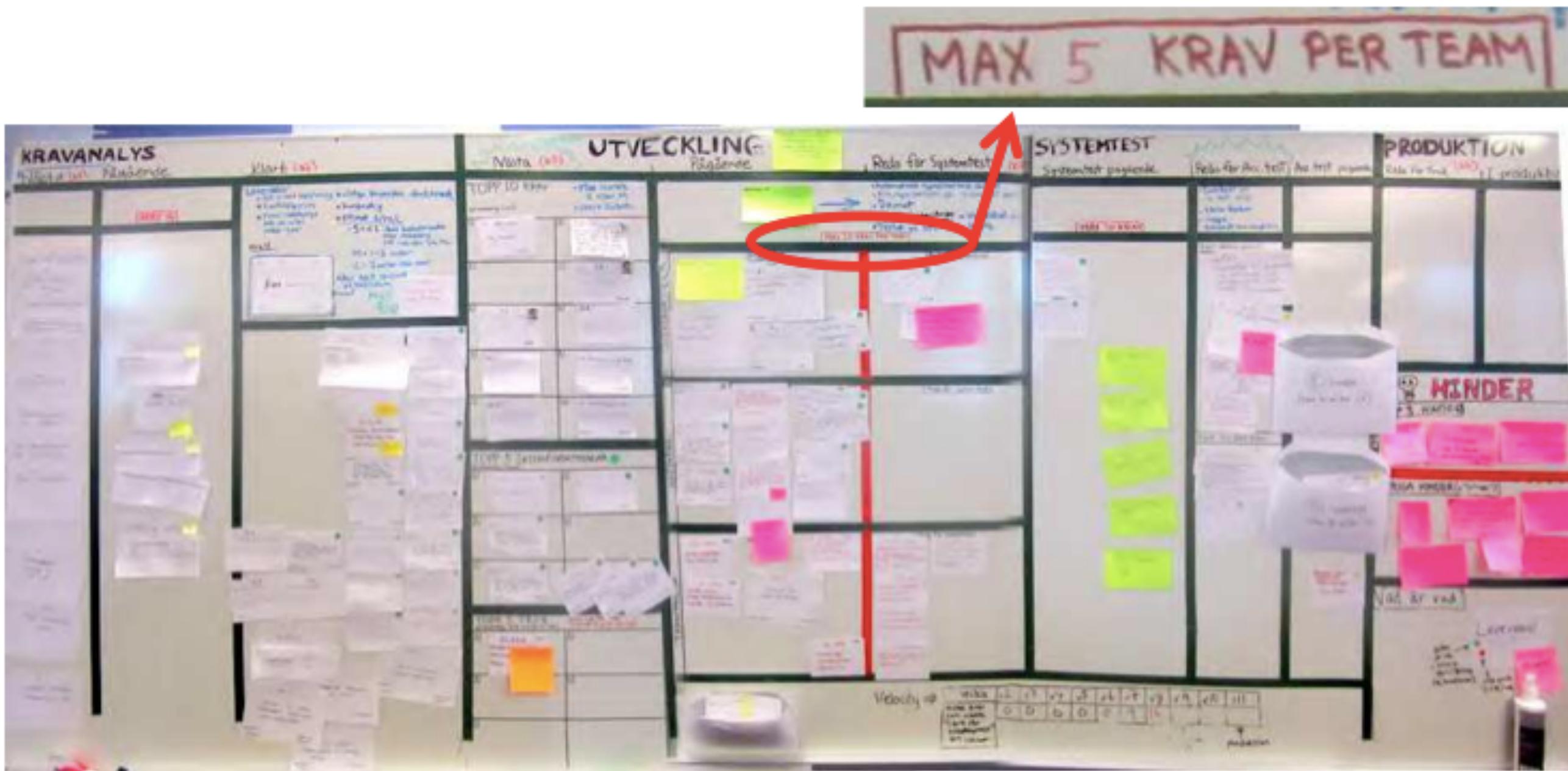


Escalar el tablero: tres equipos trabajando en el mismo proyecto

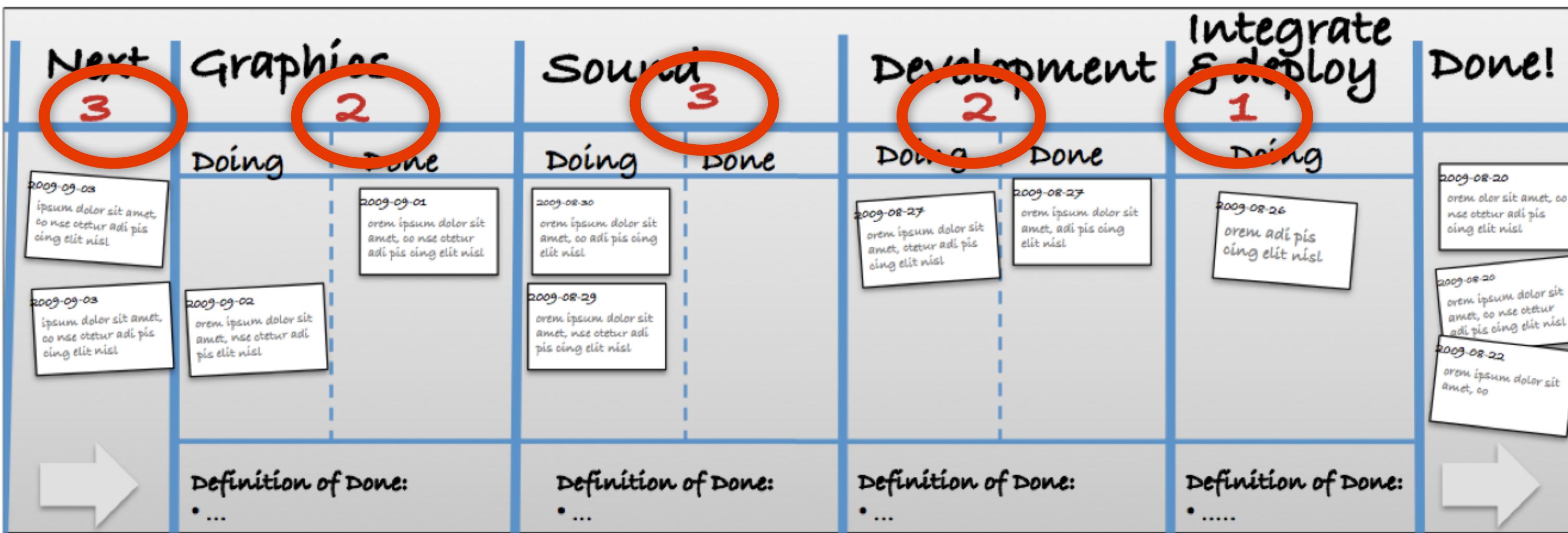


2. Limitar el WIP

Límite de trabajos en progreso



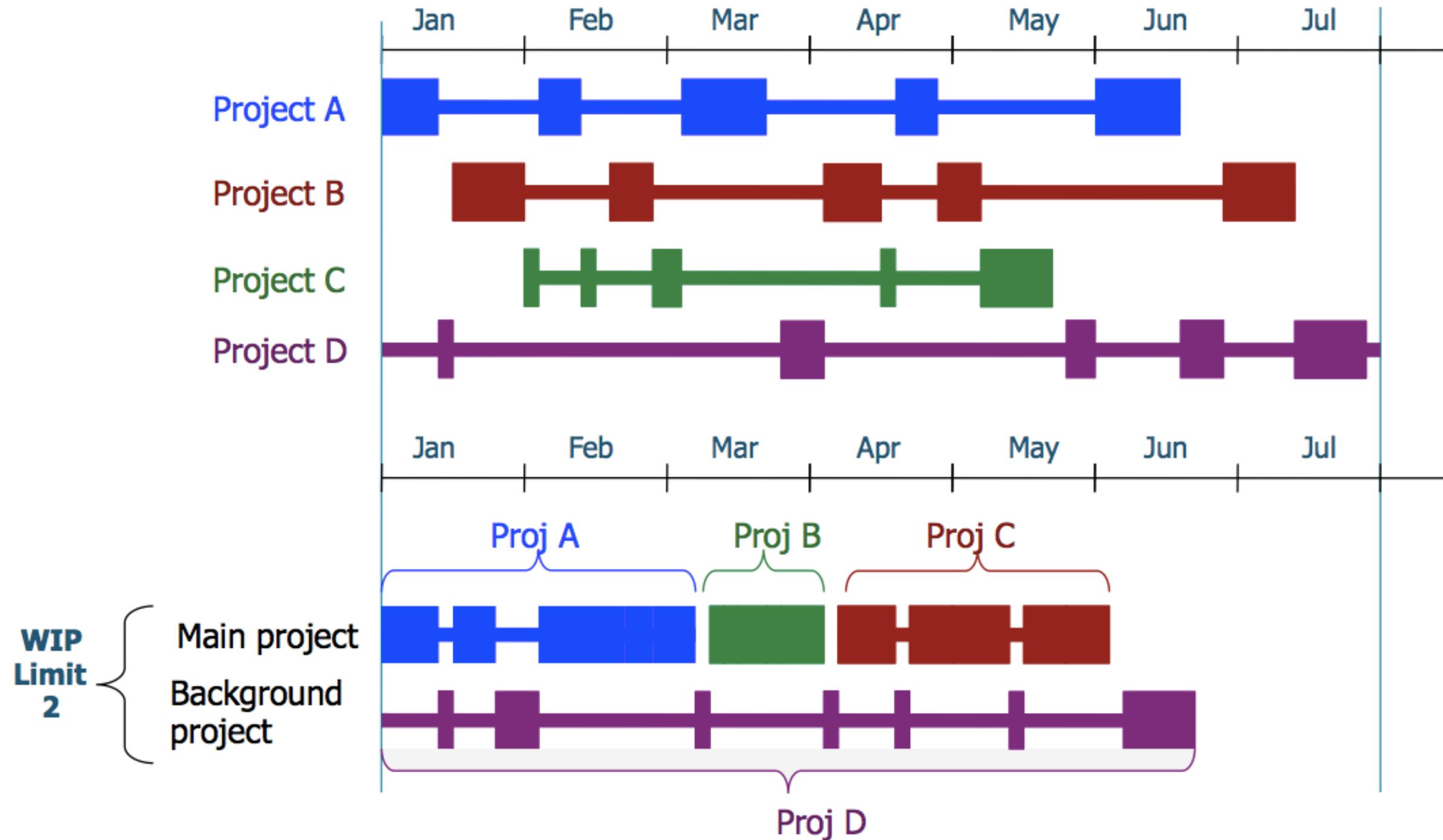
Un límite por estado del flujo



¿Por qué limitar el WIP?

- Evitar exceso de multitarea
- Evitar sobrecargas en las siguientes partes de la cadena de proceso (*downstream*)
- El límite del WIP debe establecerse por consenso entre todos los implicados en el proyecto
- La tensión creada por establecer un WIP obliga a discusiones y análisis beneficiosos para el equipo y el proyecto

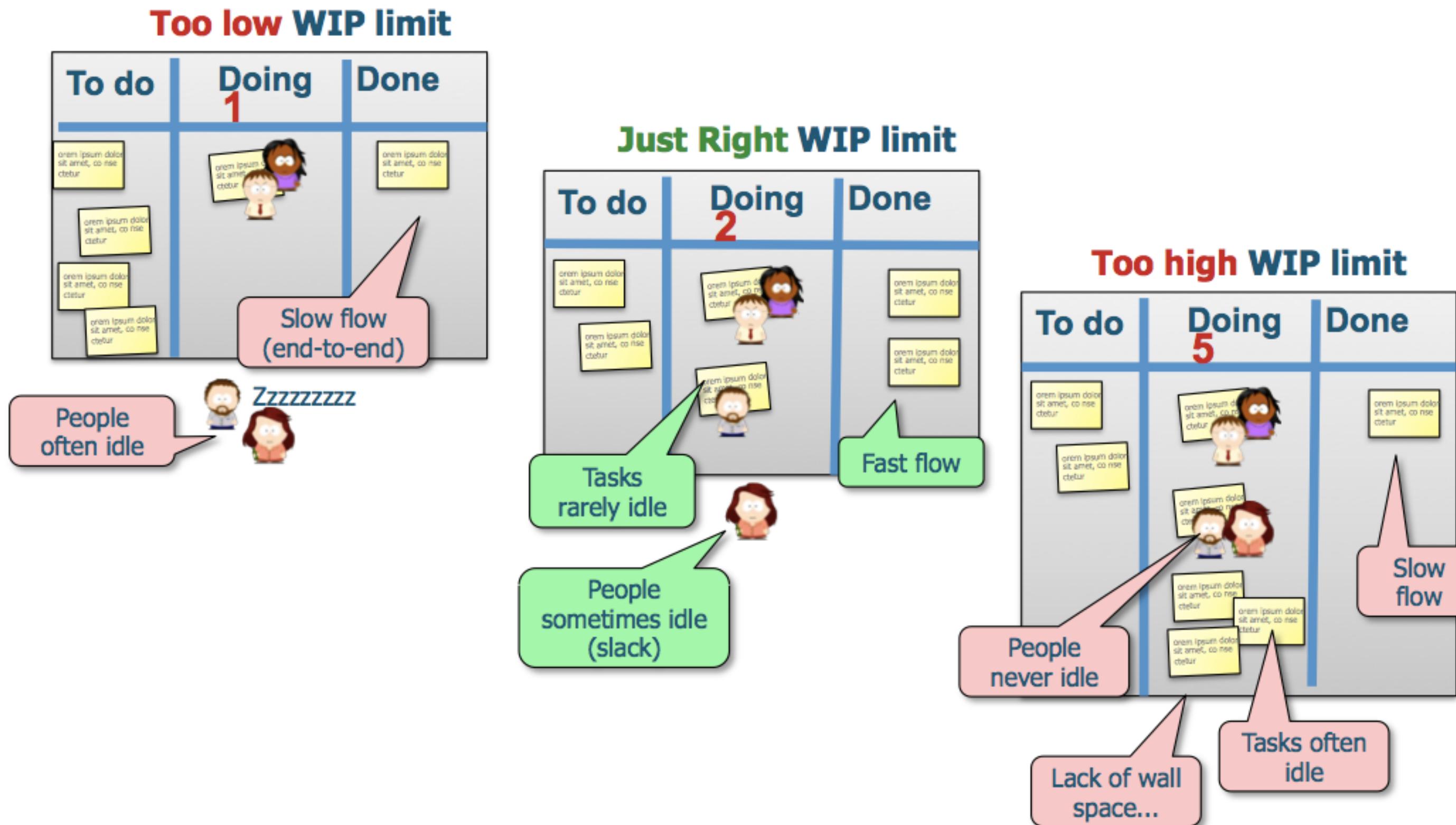
Limitando el WIP se aumenta el *throughput*



¿Cuál es el límite óptimo?

- Se obtiene midiendo, experimentando y mejorando
- Empezar usando alguna regla sencilla: $2n-1$, siendo n el número de personas trabajando en esa fase
- Estudiar el flujo y optimizar el límite (Kanban es un proceso empírico)

Optimizar el límite de WIP



El límite de WIP obliga a terminar

- Frase importante en Kanban



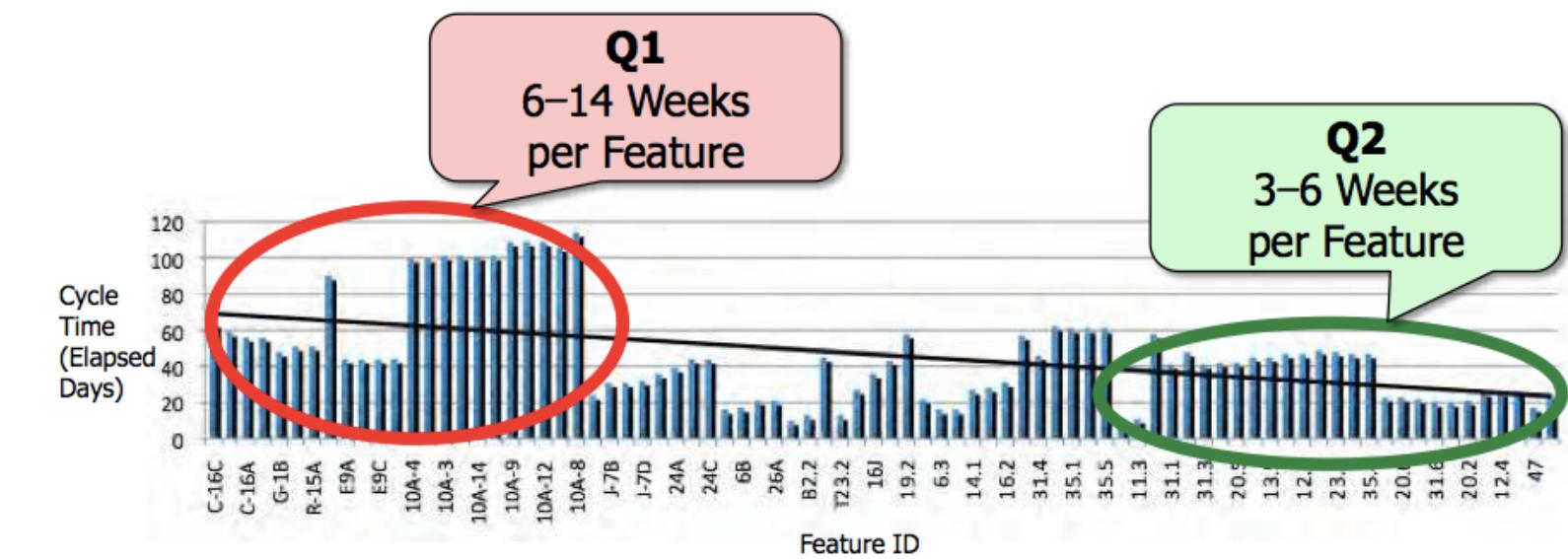
Stop starting, start finishing

- El límite de WIP obliga al equipo a centrarse en terminar tareas antes de poder acometer nuevas

3. Medir y optimizar el flujo (tiempo medio de terminación)

Tiempo medio de terminación (cycle time o lead time)

- Tiempo medio que tarda un ítem de trabajo en ser procesado en cada paso
- Equivalente al número de ítems terminados por unidad de tiempo
- Se suele analizar su evolución a lo largo del proyecto usando lo que se denomina diagrama de flujo acumulativo
- Su cálculo es muy sencillo: basta una hoja de cálculo (o incluso una pizarra)



Henrik Kniberg, *Lean from the Trenches*

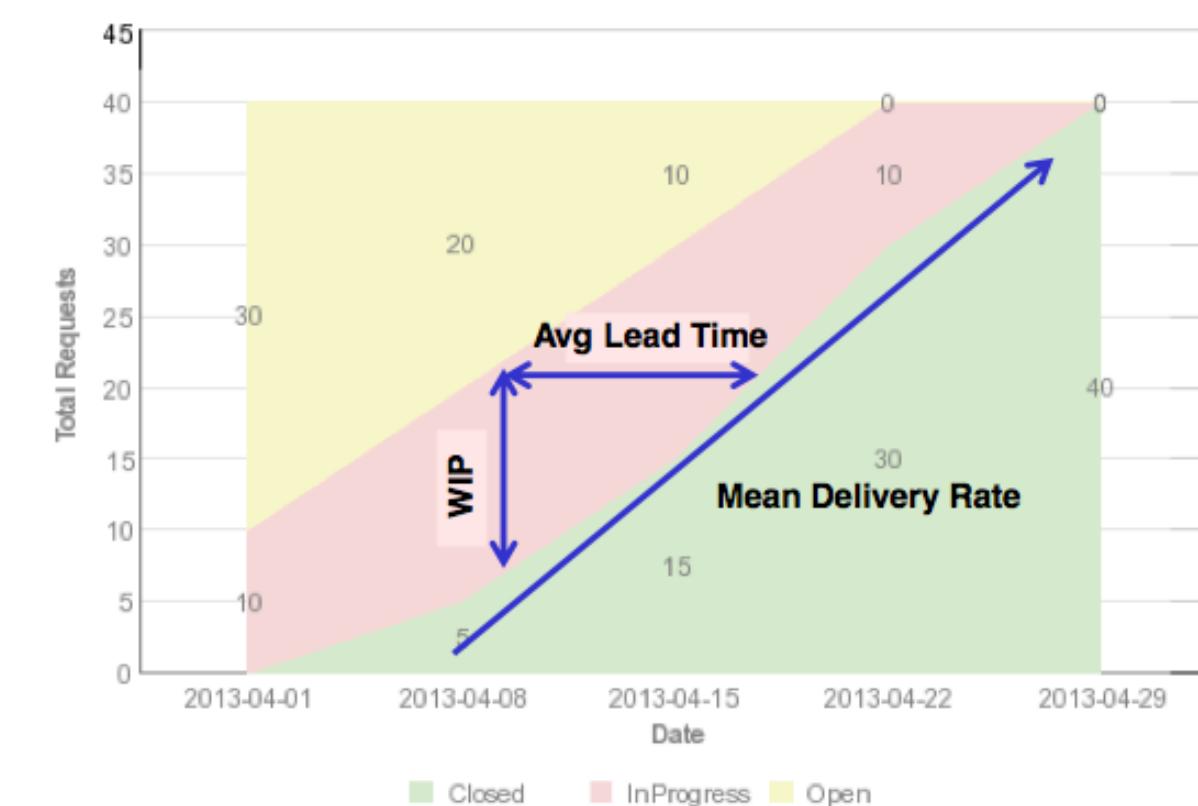
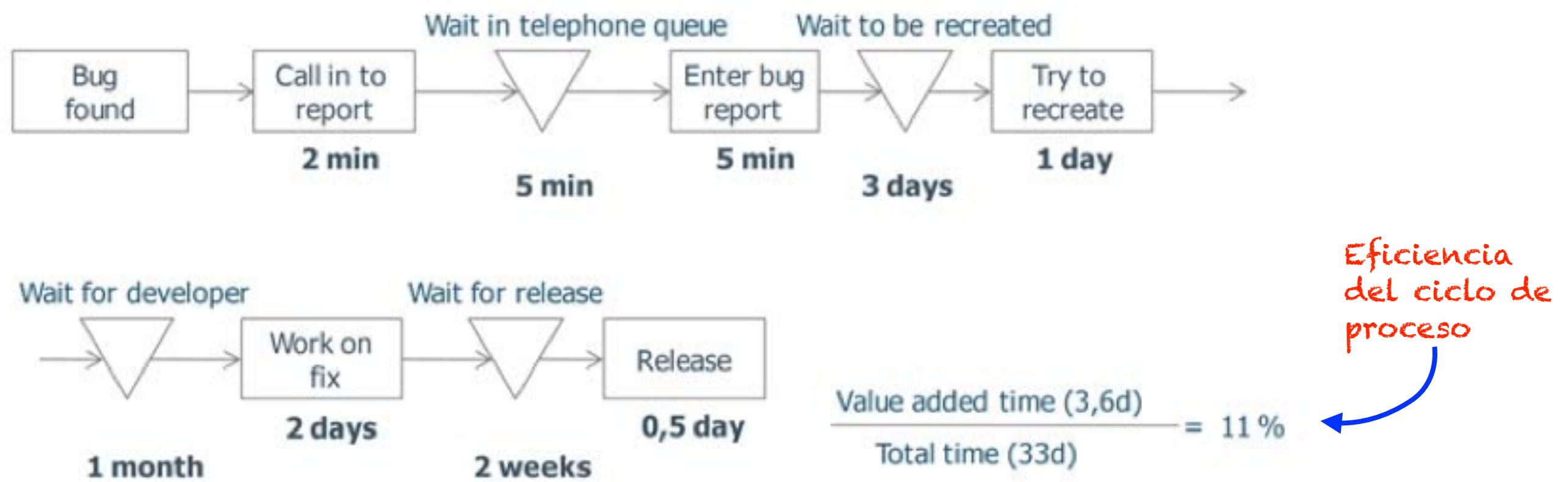


Diagrama de flujo acumulativo

Cadena de valor

- La definición de la cadena de valor (*value chain*) por la que pasan los ítems de trabajo es importante tanto para analizar el flujo y detectar cuellos de botella



Henrik Kniberg & Mattias Skarin, Kanban and Scrum, InfoQ

Resumen prácticas Kanban

1. Visualizar el flujo de trabajo
2. Limitar el *Work In Progress*
3. Medir y optimizar el flujo (el tiempo de ciclo o *lead time*)
4. Hacer explícitas las políticas
 - *Definition of Done* (Definición de Hecho)
 - *Classes of Service* (Clases de servicios)
 - *Service Level Agreement* (Acuerdos de nivel de servicios)
5. Retroalimentación y mejora continua
 - Ritmo y cadencia
 - *Daily Standups* (Reuniones diarias)
 - Retrospectivas

Bibliografía

- Libros básicos
 - Marcus Hammarberg, Joakim Sundén, *Kanban in Action*
- Libros avanzados
 - Mike Burrows, *Kanban from the inside*
 - David J. Anderson, *Kanban*

