

# ¿Scrum, Kanban, Pikachu o Squirtle? What the agile?

Ander Conal

14.05.20

VIRTUAL COFFEE



# Ander Conal

---

## SOFTWARE DEVELOPER

Cuando no estoy trabajando, lo más posible es que me encuentres haciendo boxeo, paseando con mis 2 perros y el resto de mi familia, jugando un poco a la Switch o quitando el polvo a los vinilos de mi época como promotor de eventos y DJ. En la actualidad, trabajo en Plain Concepts Bilbao donde, además de programar, doy charlas y formaciones.

[@anderconal](#)

# Agenda completa

1. Scrum: Resumen y puntos clave.
2. Kanban: Resumen y puntos clave.
3. Otros Pokémon (XP, Lean Software Development...): Resumen y puntos clave.
4. Diferencias entre Scrum, Kanban y otros Pokémon.
5. Valores ágiles.
6. ¿Cómo elegir la metodología más acorde con nuestro proyecto?

# Scrum and XP from the Trenches

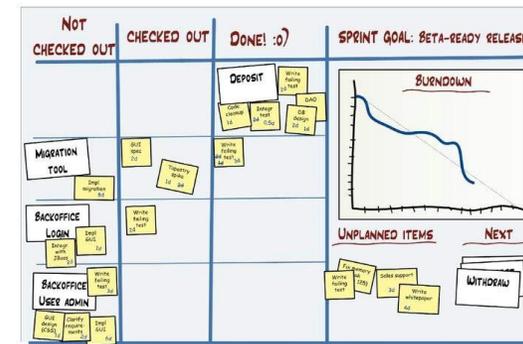
How we do Scrum

- Henrik Kniberg

An agile war story

## Scrum and XP from the Trenches

How we do Scrum



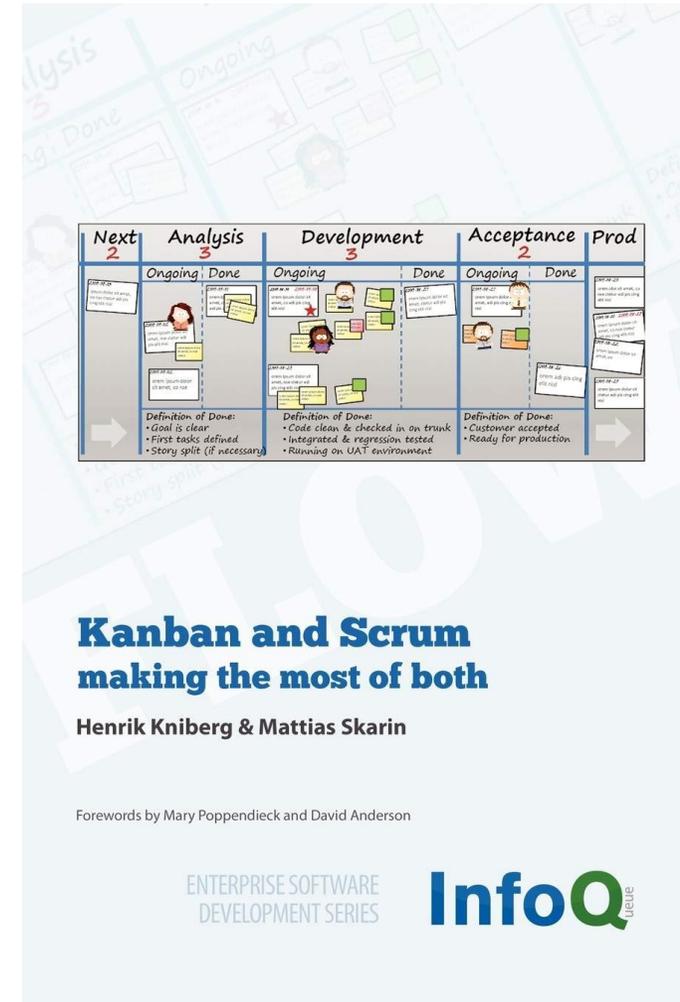
Henrik Kniberg

Forewords by Jeff Sutherland, Mike Cohn

# Kanban and Scrum

Making the most of both

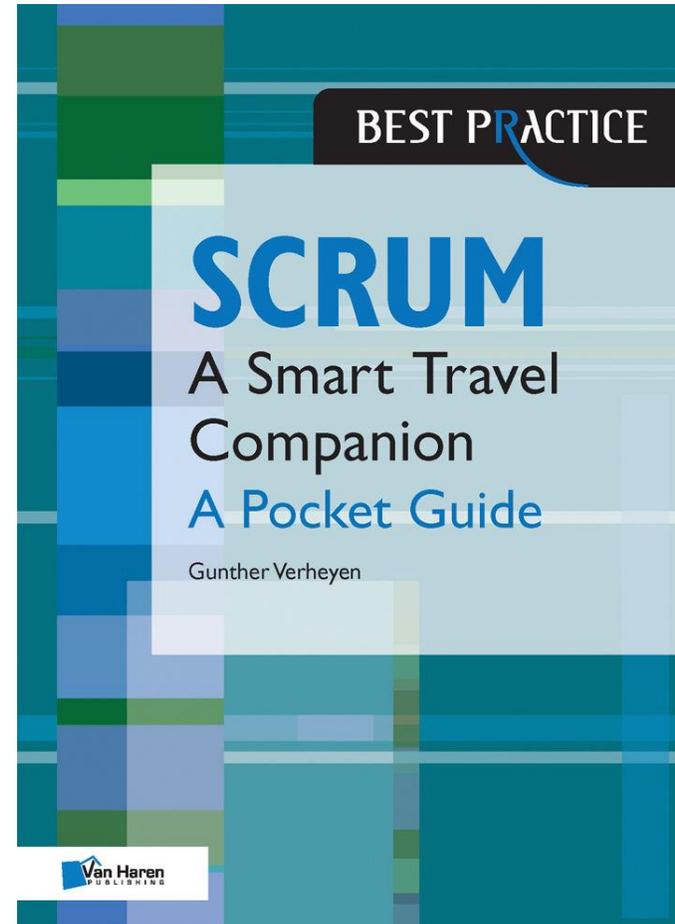
- Henrik Kniberg & Mattias Skarin



# Scrum

A Smart Travel Companion  
A Pocket Guide

- Gunther Verheyen



# Scrum: Resumen y puntos clave

# Scrum

## Resumen

- **Divide tu organización** en equipos pequeños, interdisciplinarios y auto-organizados
- **Divide el trabajo** en una lista de entregables pequeños y concretos. Ordena la lista por orden de prioridad y estima el esfuerzo relativo de cada elemento
- **Divide el tiempo** en iteraciones cortas de longitud fija, con código potencialmente entregable y demostrado después de cada iteración
- **Optimiza el plan de entregas** y actualiza las prioridades en colaboración con el cliente
- **Optimiza el proceso** teniendo una retrospectiva después de cada iteración

# Scrum

## Puntos clave

- Prescribe
  - Roles
  - Iteraciones de tiempo fijo
  - Equipos multi-funcionales
  - La estimación y la velocidad

# Scrum

## Puntos clave

- Limita el WIP por iteración
- Es empírico
- Se resiste a los cambios durante la iteración
- El tablero sprint se limpia entre iteraciones
- Los elementos de la pila de producto deben caber en un sprint
- Permite trabajar en múltiples proyectos simultáneamente
- Es Lean y Ágil

# Kanban: Resumen y puntos clave

# Kanban

## Resumen

- Visualiza el flujo de trabajo
- Limita el WIP
- Mide el lead time

# Kanban

## Puntos clave

- Limita el WIP por estado en flujo de trabajo
- Es empírico
- Permite trabajar en múltiples productos simultáneamente
- Es Lean y Ágil

# Otros Pokémon (XP, Lean Software Development...): Resumen y puntos clave

# XP (Extreme Programming)

## Resumen

- Es un framework de desarrollo de software que nos empuja a producir software de alta calidad y a obtener mayor calidad de vida para el equipo de desarrollo. Es el framework más específico en cuanto a buenas prácticas de ingeniería.

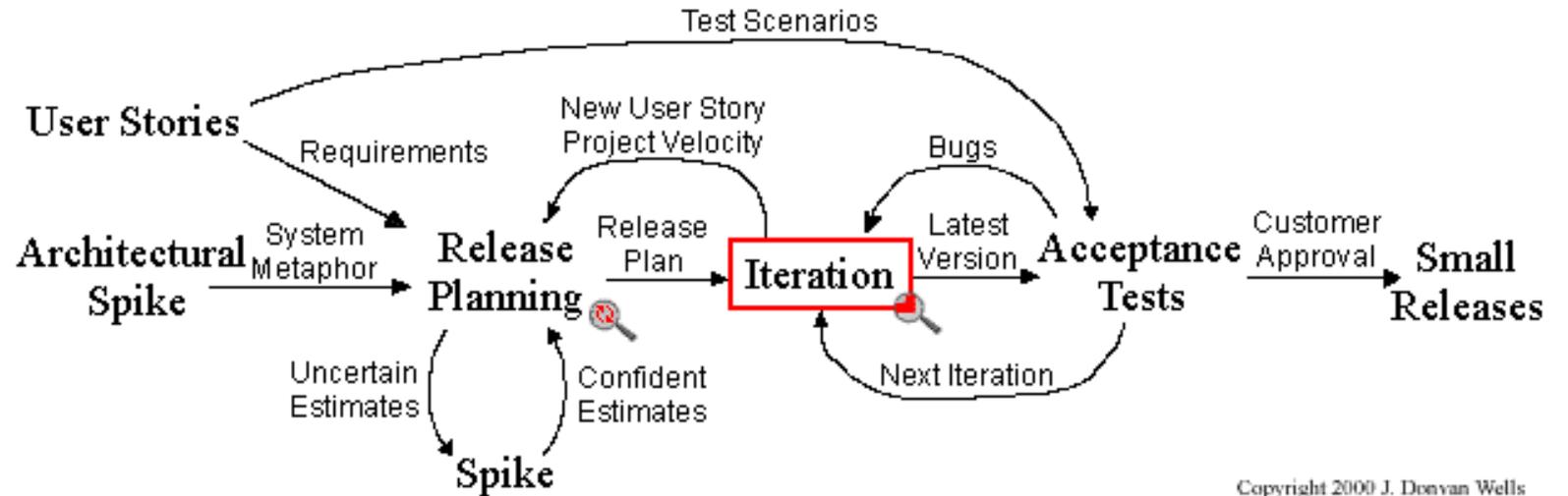
# XP

Puntos clave

- Planning
- Managing
- Coding
- Designing
- Testing



## Extreme Programming Project



# XP

## Puntos clave

- Entregas pequeñas y frecuentes
- El cliente siempre está disponible
- TDD
- Pair programming
- Integración continua (CI)
- Refactorizar cuando y donde sea posible
- Cuando se encuentre un bug, se crean tests
- Spikes para reducir riesgo

# Otros Pokémon

## Resumen

- Lean software development
- Feature-Driven Development (FDD)
- Dynamic Systems Development Method (DSDM)
- Crystal methods

# Diferencias entre Scrum, Kanban y otros Pokémon.

# Diferencias

Entre Scrum y Kanban

- **Scrum**

- Prescribe roles
  - Product Owner, Scrum Master, Equipo de desarrollo

- **Kanban**

- No establece ningún rol. Puedes añadirlo si lo deseas

# Diferencias

Entre Scrum y Kanban

- **Scrum**

- Prescribe iteraciones de tiempo fijo

- **Kanban**

- No se prescriben iteraciones de tiempo fijo
  - Puedes elegir el momento de hacer la planificación, la mejora de procesos, y la entrega

# Diferencias

Entre Scrum y Kanban

- **Scrum**

- Limita el WIP por iteración

- **Kanban**

- Limita el WIP por estado en flujo de trabajo

# Diferencias

Entre Scrum y Kanban

- **Scrum**

- Se resiste a los cambios durante la iteración

- **Kanban**

- Admite cambios tan pronto como se libere capacidad y no se exceda el límite de WIP del estado al que pasaría dicha tarea

# Diferencias

Entre Scrum y Kanban

- **Scrum**

- El tablero sprint se limpia entre iteraciones

- **Kanban**

- El tablero es persistente, no se necesita limpiarlo y volver a empezar

# Diferencias

Entre Scrum y Kanban

- **Scrum**

- Prescribe equipos multi-funcionales

- **Kanban**

- Los equipos multi-funcionales son opcionales

# Diferencias

## Entre Scrum y Kanban

- **Scrum**

- Los elementos de la pila de product deben caber en un sprint

- **Kanban**

- No hay ninguna norma explícita que indique que los elementos deben ser lo suficientemente pequeños como para caber en un intervalo de tiempo específico
- En el mismo tablero podría haber un elemento que necesita un mes para terminarse y otro que necesita un día

# Diferencias

Entre Scrum y Kanban

- **Scrum**

- Prescribe la estimación y la velocidad

- **Kanban**

- No se prescribe la estimación, pero es común que se utilicen esa u otras técnicas

# Diferencias

## Entre Scrum y Kanban

- **Scrum**

- Prescribe priorización y se hace ordenando la pila de producto
- Los cambios en las prioridades tienen efecto en el siguiente sprint

- **Kanban**

- Puedes elegir cualquier esquema de priorización (o ninguno)
- Los cambios tienen efecto tan pronto como la capacidad esté disponible
- Puede que exista o no una pila de producto y puede estar priorizada o no

# Diferencias

## Entre Scrum y Kanban

- **Scrum**

- Prescribe reuniones diarias
- Reunión diaria orientado a la persona, cada miembro del equipo hace su reporte, 1 a 1

- **Kanban**

- No se utilizan las reuniones diarias, pero la mayoría de equipos Kanban las hacen de todos modos
- Formato orientado más al panel, poniendo el foco en cuellos de botella y otros problemas visibles

# Diferencias

Entre Scrum y Kanban

- **Scrum**

- Se utilizan diagramas de Burndown para seguir el progreso de una iteración

- **Kanban**

- No se prescriben gráficos burndown, pero se permite utilizar cualquier tipo de gráfico

# Valores ágiles

# Valores ágiles

- Individuos e interacciones sobre procesos y herramientas
- Software funcionando sobre documentación extensiva
- Colaboración con el cliente sobre negociación contractual
- Respuesta ante el cambio sobre seguir un plan

# Los 12 Principios del Manifiesto Ágil

- Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.
- Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.
- Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.
- Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.
- Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.
- El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.

# Los 12 Principios del Manifiesto Ágil

- El software funcionando es la medida principal de progreso.
- Los procesos Ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.
- La atención continua a la excelencia técnica y al buen diseño mejora la Agilidad.
- La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
- Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.
- A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.

**¿Cómo elegir la  
metodología más  
acorde con nuestro  
proyecto?**

# Proceso empírico

Prueba, falla, modifica, prueba, falla, itera, falla, ajusta, prueba...

- Analiza cada opción y elige la que consideres que se adapta mejor a tu tipología de proyecto
- A la hora de elegir, ten en cuenta el estado en el que se encuentra el proyecto (desarrollo, a punto de salir a producción, en producción...)
- Prueba sin miedo. Si hay algo que no ha ido bien, piensa en cómo mejorarlo, modifica el proceso, y vuelta a probar
- No hay opción perfecta. Debe ser un proceso empírico sin fin. Siempre habrá algo nuevo que probar, o algo viejo que suprimir o modificar

# Equipo Pokémon

Compuesto de lo mejor de cada tipo de Pokémon

- ¿Qué tal si probamos a hacer Scrum incorporando el límite de WIP por estado de Kanban?
- ¿Y si probamos Kanban pero estimamos las tareas, como en Scrum?
- ¿Por qué no hacemos Scrum pero priorizamos hacer TDD y pair programming como en XP?
- ¿Y qué hay de probar a hacer Kanban puro siguiendo las buenas prácticas de ingeniería de XP?
- ¿Qué pasaría si probamos a hacer Scrum, dejando que los cambios en la prioridad entren en el presente sprint en cuanto se libere la capacidad como ocurre en Kanban y, además, nos obligamos a trabajar con CI / CD como en XP?



# Thank you

[@anderconal](#)

[@plainconcepts](#)

[www.plainconcepts.com](http://www.plainconcepts.com)

# plain concepts

Rediscover  
the meaning of technology



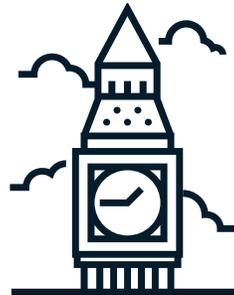
SPAIN



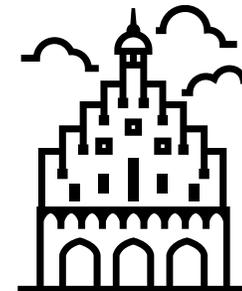
USA



UAE



UNITED KINGDOM



GERMANY



NETHERLANDS

[www.plainconcepts.com](http://www.plainconcepts.com)

For further information

[info@plainconcepts.com](mailto:info@plainconcepts.com)