

Introducción a Agile y Scrum

Ingeniería del Software

José García Fanjul
Diciembre 2022



Universidad de Oviedo

Orígenes



“Mock-up of Toyota assembly line”. Urbangarden. (CC BY-NC-SA 2.0).

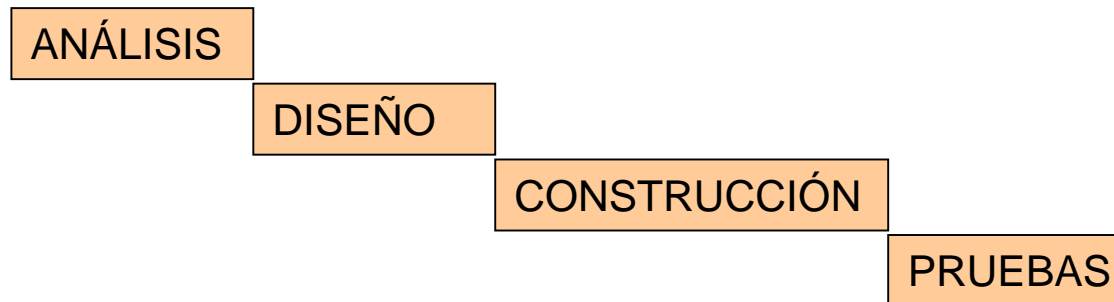
- Toyota Production System: 1948 – 1975. Taiichi Ohno y Eiji Toyoda.
- Just in Time.
- “Waste is a Crime” (desperdiciar, despilfarrar).

Orígenes



- Círculo de Deming (o Deming-Shewart).
- Mejora continua.

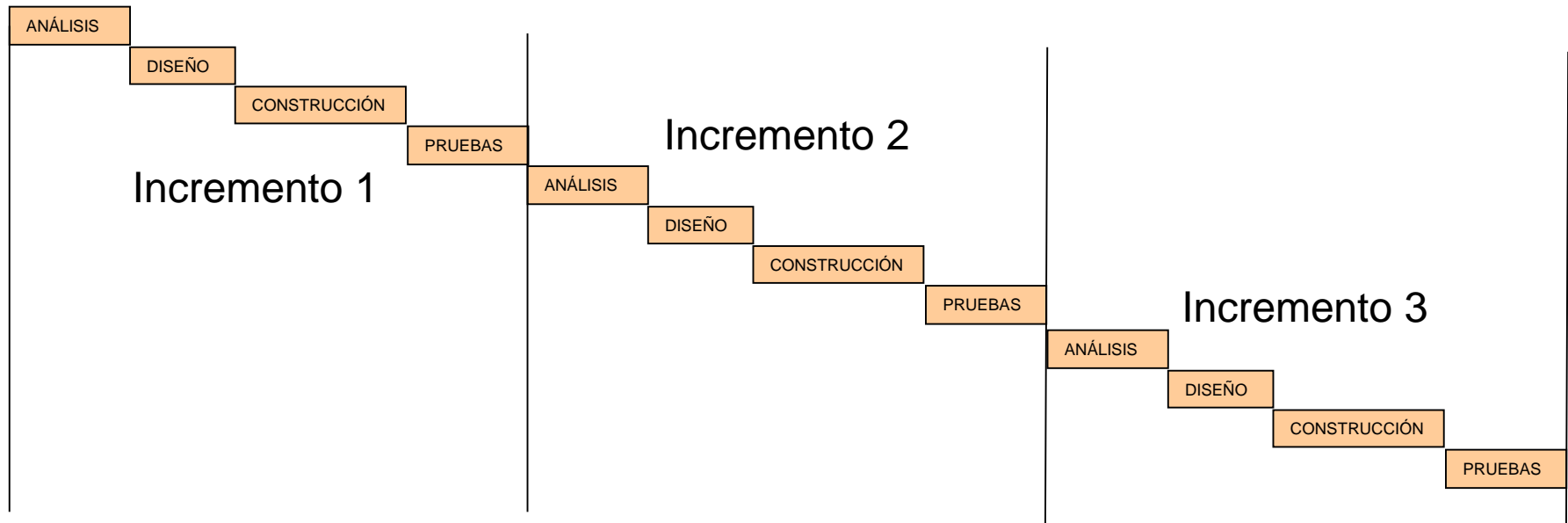
Proceso en “cascada” (BDUF)



- ¿Es necesario definir todos los requisitos al principio? ¿puede hacerse?
- Otros:
 - ¿Cuándo empezamos a construir?
 - ¿Podemos olvidarnos de las pruebas hasta el final?

¿Cómo se reacciona ante un adelanto de la entrega?

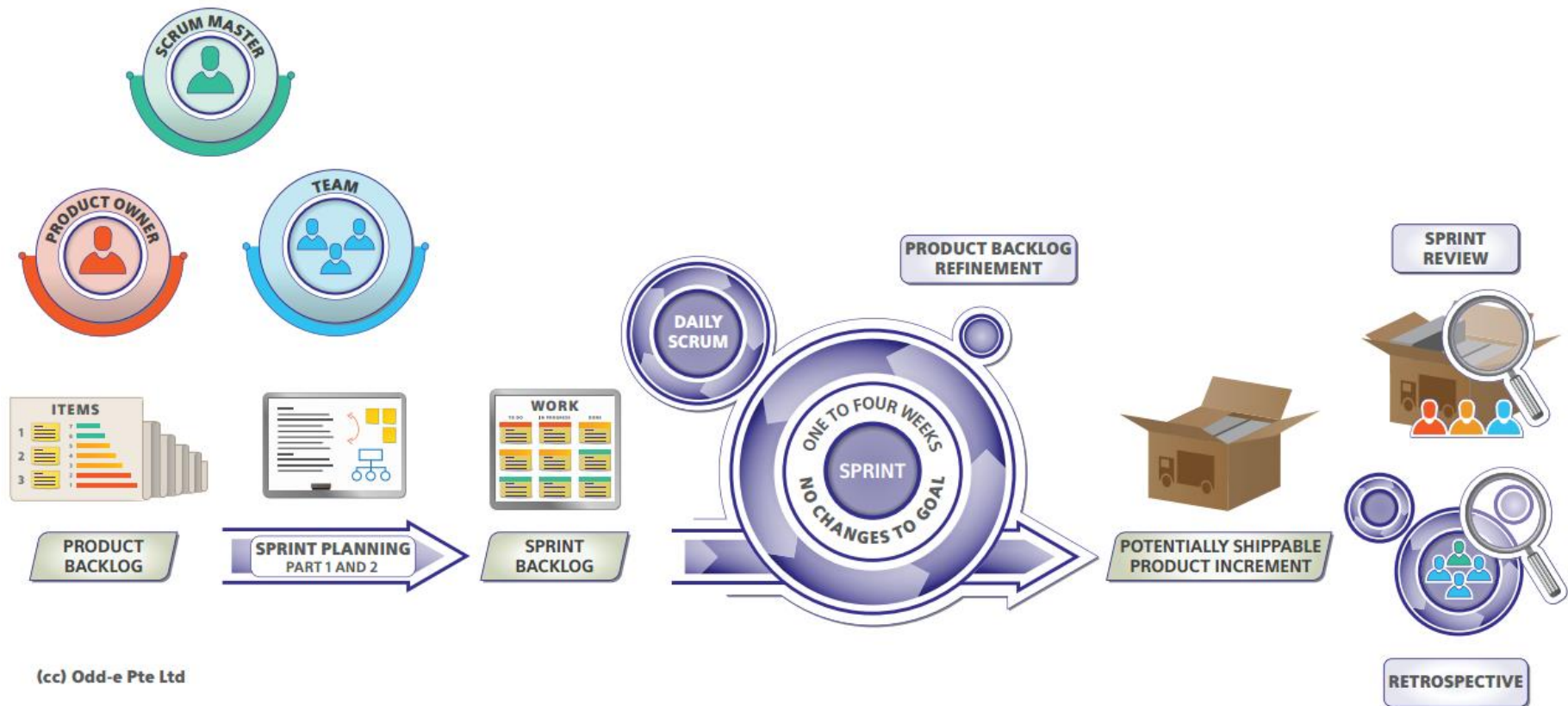
Procesos incrementales



- Dividir desarrollo en incrementos.
- Cada incremento es una versión funcional del sistema.
- **Priorizar los requisitos.**
 - Recursos finitos.
 - Negociación.

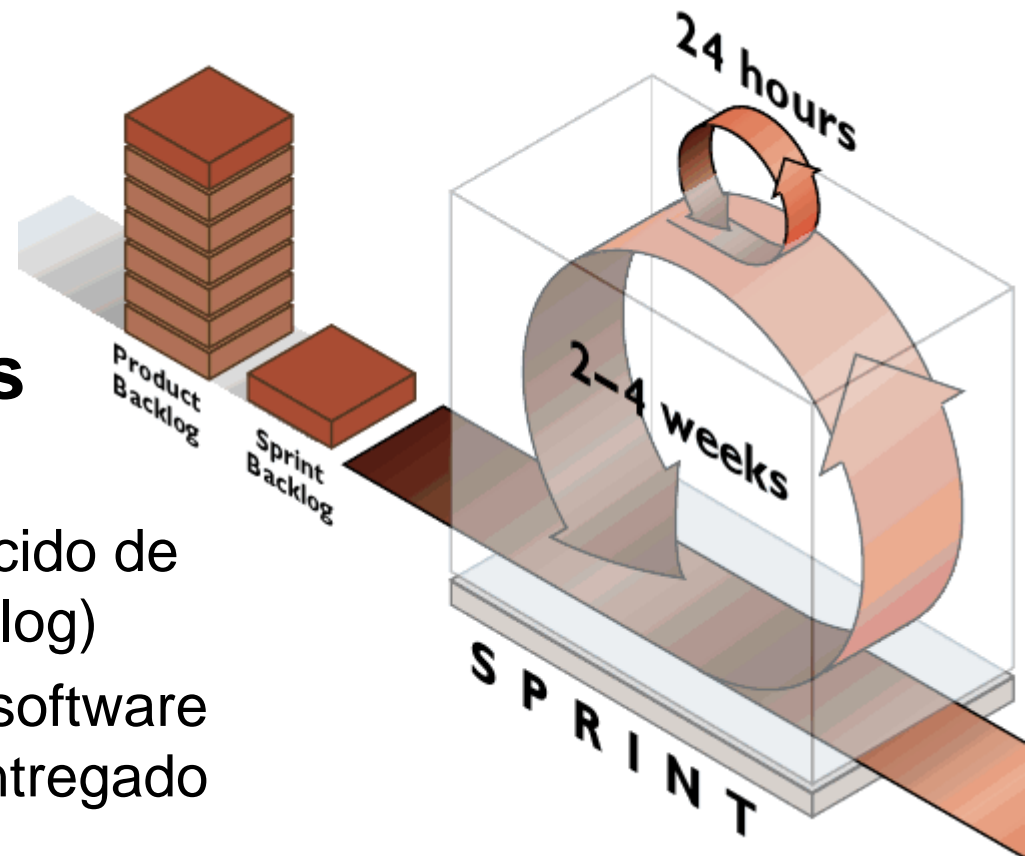
¿Qué sucede si no se construye el último incremento?

SCRUM



Scrum: sprints

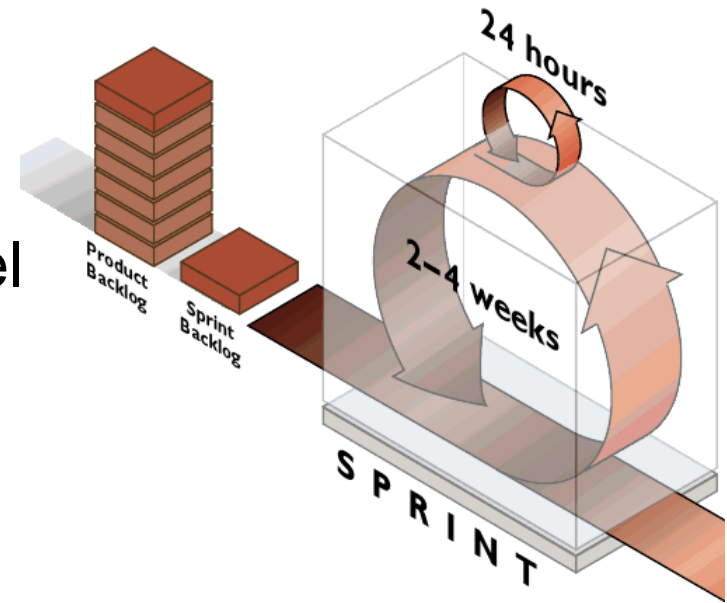
- Metodología ágil.
- Incremental: **sprints**
 - De 15 a 30 días
 - Con un conjunto reducido de requisitos (sprint backlog)
 - El producto debe ser software preparado para ser entregado al cliente (deliverable).



Working software!

Scrum: backlog

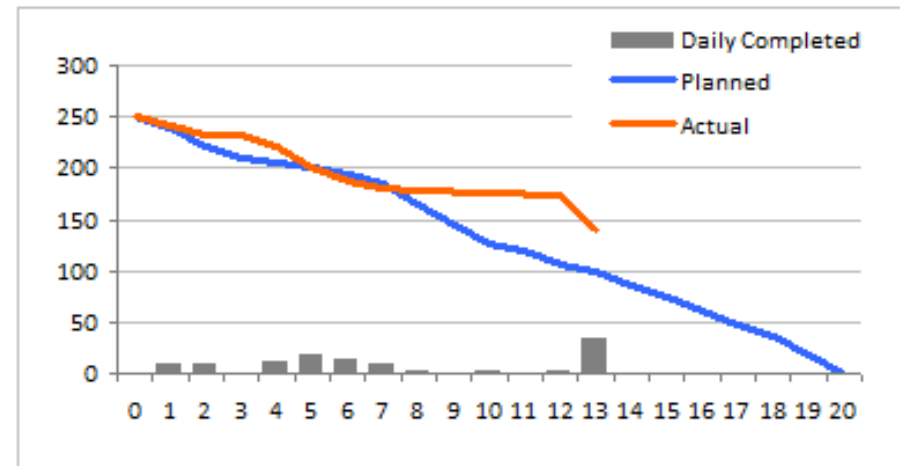
- Relación con usuarios a través del “product owner”.
- **Elaboración de requisitos a través del “backlog”** (product backlog y sprint backlog).
 - Recopilado por el product owner.
 - Los requisitos cambian de prioridad con frecuencia, según avanza el desarrollo.
 - En cada sprint se eligen las características con más alta prioridad.



Scrum: proceso

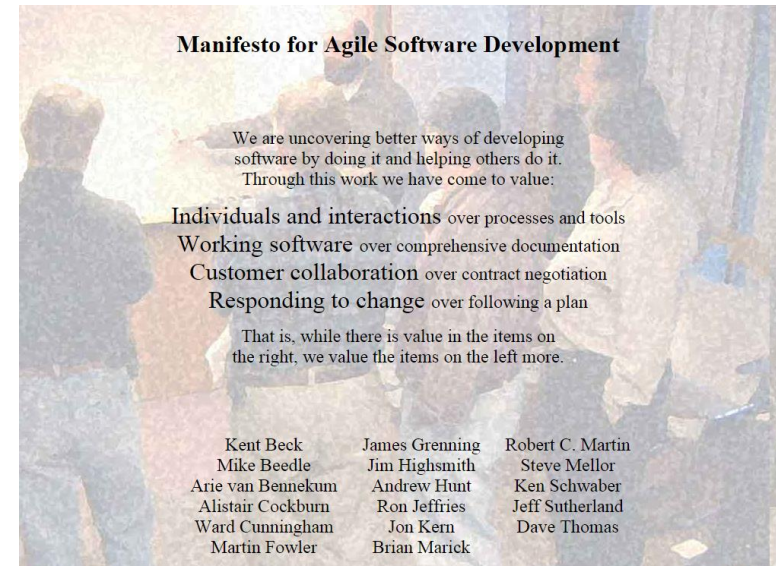
■ Proceso:

- Crear el backlog inicial.
- Sprints
 - Al iniciar el sprint: sprint backlog.
 - Burn-down charts.
 - Al finalizar los sprints: demostración y evaluación.



Scrum: equipo

- Auto-organizado, interdisciplinar.
- Interacción diaria: daily scrum.
 - Qué hemos hecho.
 - Qué vamos a hacer.
 - Detectar obstáculos.
- Retrospectivas: detectar oportunidades de mejorar.



“Accountability belongs to the development team as a whole”.

Manifesto for Agile Software Development

2001

We are uncovering better ways of developing software by doing it and helping others do it.
Through this work we have come to value:

Through this work we have come to value:

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Kent Beck
Mike Beedle
Arie van Bennekum
Alistair Cockburn
Ward Cunningham
Martin Fowler

James Grenning
Jim Highsmith
Andrew Hunt
Ron Jeffries
Jon Kern
Brian Marick

Robert C. Martin
Steve Mellor
Ken Schwaber
Jeff Sutherland
Dave Thomas

Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it.

Through this work we have come to value:

and not

Individuals and interactions ~~over~~ processes and tools

Working software ~~over~~ comprehensive documentation

Customer collaboration ~~over~~ contract negotiation

Responding to change ~~over~~ following a plan

That is, ~~while there is value in the items on the right,~~ we value the items on the left ~~more.~~

Kent Beck
Mike Beedle
Arie van Bennekum
Alistair Cockburn
Ward Cunningham
Martin Fowler

James Grenning
Jim Highsmith
Andrew Hunt
Ron Jeffries
Jon Kern
Brian Marick

Robert C. Martin
Steve Yegorin
Ken Schwaber
Jeff Sutherland
Dave Thomas

!NO!

Buenas prácticas ágiles

- Adaptarse a los cambios.
- Micro-incrementos con versiones funcionales cada pocas semanas.
- Interacción frecuente con el cliente y en el propio equipo.
- Pruebas continuas → Integración continua → Continuos despliegues.

Resumen e invitación a la discusión

- Las metodologías ágiles se enfocan en **adaptarse a los cambios**. Siempre han existido cambios **pero no siempre** tuvimos métodos ágil o Scrum.
- Las personas son más importantes que los procesos... pero **Scrum es un proceso**.
- El desarrollo de software requiere la **colaboración del cliente**. ¿Qué % de clientes está dispuesto a hacer contratos ágiles?
- En cualquier proyecto importante la documentación también lo será, y aún más una **versión funcional del software**.
- Las prácticas ágiles, y Scrum, también pueden utilizarse en **otros ámbitos**.

No silver bullet! 😊

Referencias

- Deemer P, Benefield G, Larman C y Vodde B. “The Scrum Primer”.
<http://www.scrumprimer.org/>
- Larman C y Basili VR. “Iterative and Incremental Development: A Brief History”. IEEE Computer. Junio 2003.
- SoftHouse. “Scrum in five minutes”.
- Sutherland J y Schwaber K. “The Scrum Guide”.
<http://www.scrumguides.org/>
- Sutherland J. “Scrum: A revolutionary approach to building teams, beating deadlines and boosting productivity”. Ed Random House. 2014.
- Schwaber K. “SCRUM Development Process”. Workshop: Business Object Design and Implementation. OOPSLA. 1995.
<http://jeffsutherland.org/oopsla/schwapub.pdf>
- VVAA. Manifesto for Agile Software Development. <http://agilemanifesto.org/>