

Introducción a Agile y Scrum (repaso para SI)

Ingeniería del Software

José García Fanjul

Rev. J. Tuya 2023-2024

Rev. C. de la Riva 2024-2025



Orígenes



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

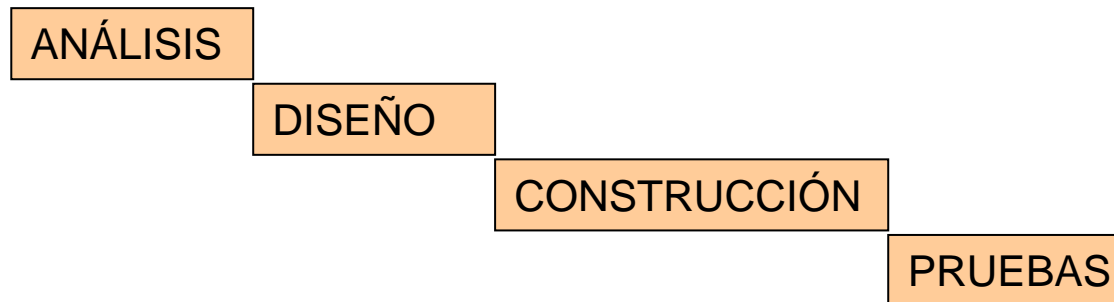
- Toyota Production System: 1948 – 1975. Taiichi Ohno y Eiji Toyoda.
- Just in Time.
- “Waste is a Crime” (desperdiciar, despilfarrar).

Orígenes



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

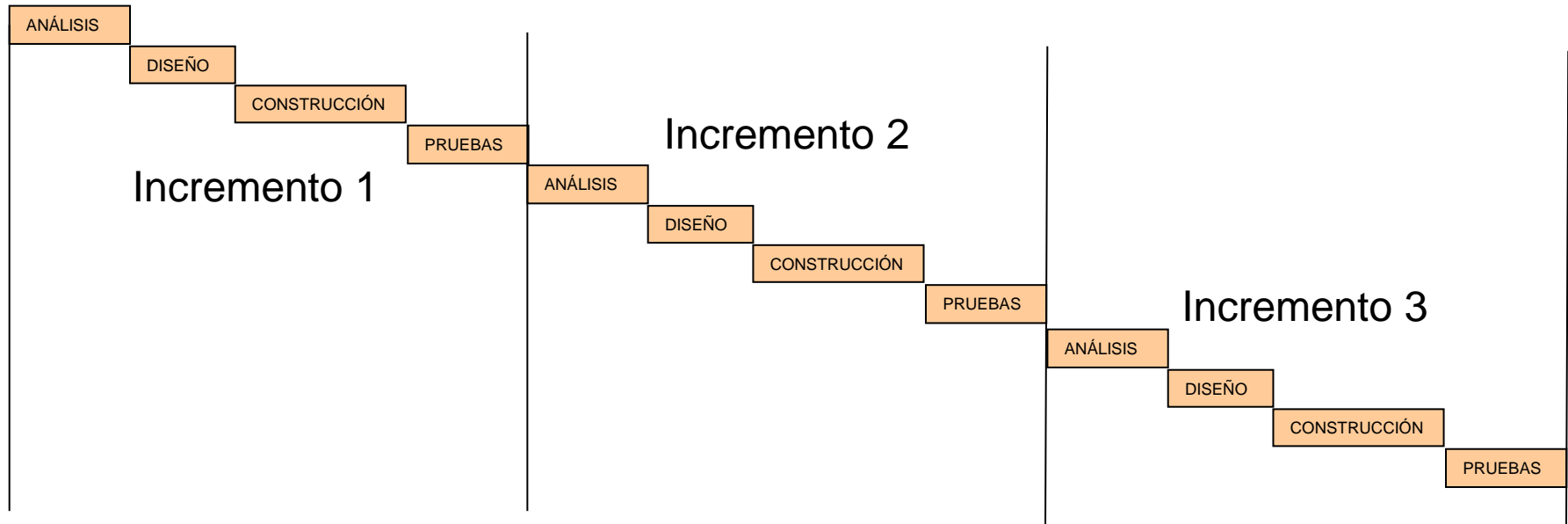
Proceso en “cascada” (BDUF)



- ¿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?

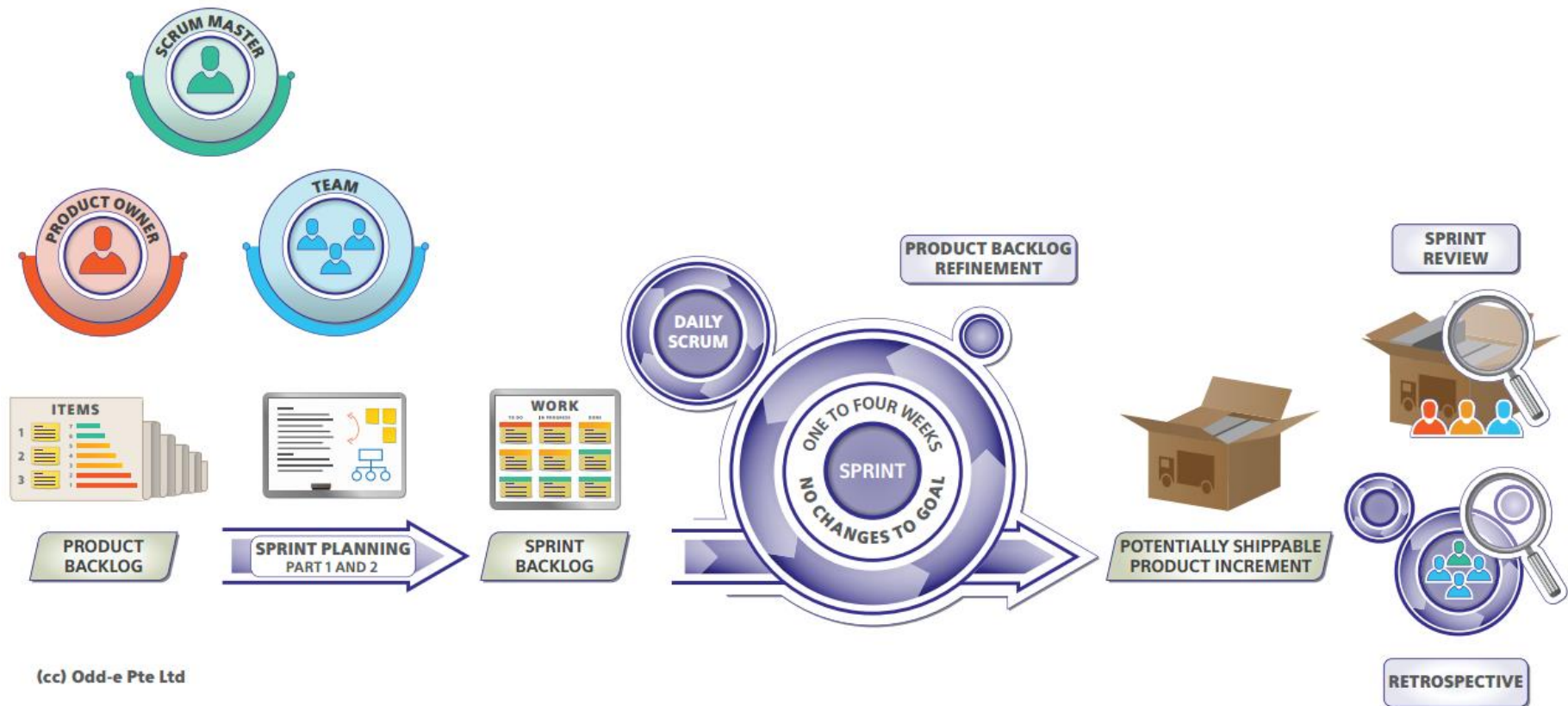
Procesos incrementales



- 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



(cc) Odd-e Pte Ltd

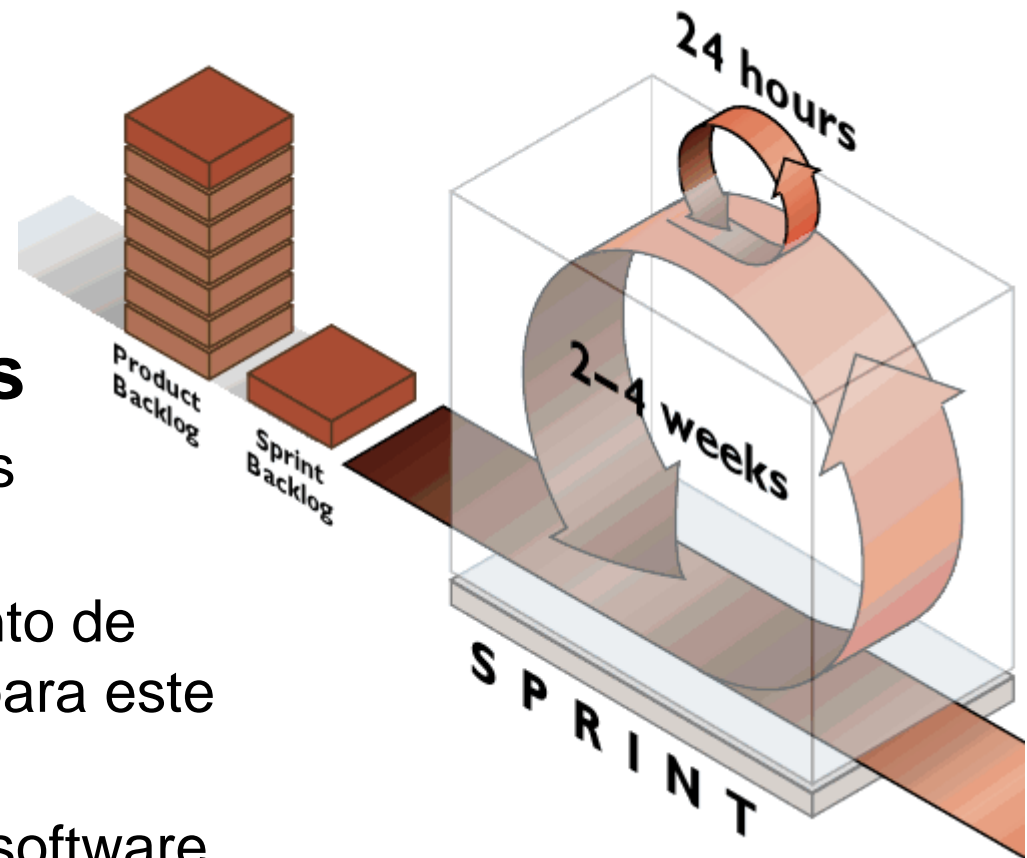


Repaso Scrum - Contenido

- Sprints, Product Backlog, Sprint Backlog
- Historias de usuario. Pizarras, Story Mapping, Burndown
- Proceso (reuniones)
- Equipo (roles)
- Ejemplo introductorio para repaso/prácticas

Scrum: sprints

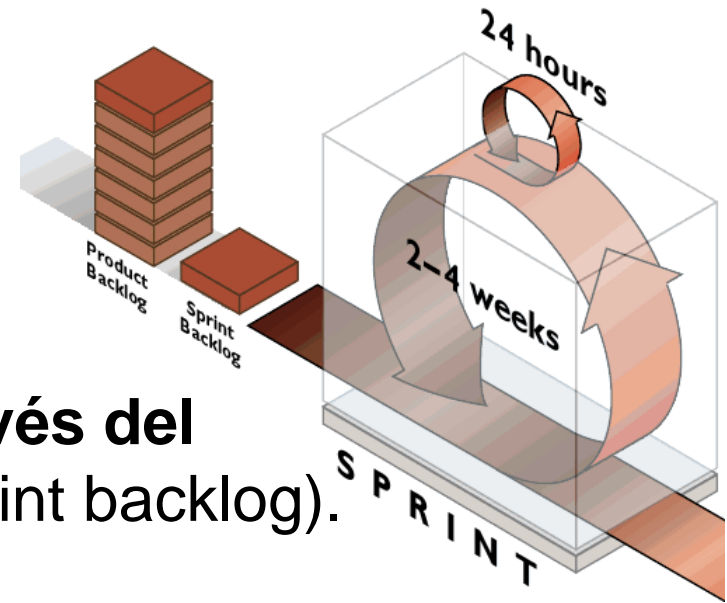
- Metodología ágil.
- Incremental: **sprints**
 - No más de 4 semanas (típico: 2 semanas)
 - Implementar el conjunto de historias (requisitos) para este sprint (sprint backlog)
 - El producto debe ser software que pueda ser entregado al cliente (potentially shippable).



Working software!

Scrum: backlog

- Relación con usuarios a través del “product owner”.
- **Elaboración de requisitos a través del “backlog”** (product backlog y sprint backlog).
 - Necesidades del product owner.
 - Los requisitos cambian de prioridad con frecuencia, según avanza el desarrollo.
 - Las características del sistema se comienzan a elaborar en el **Product Backlog**
 - En cada sprint se eligen las características con más alta prioridad para el **Sprint Backlog**



Product backlog (Pila del Producto)

Elemento	Detalles (URL wiki)	Prioridad	Estimación del Valor	Estimación de Esfuerzo Inicial
Como comprador, quiero poner un libro en el carrito de la compra (ver bocetos en el wiki)	...	1	7	5
Como comprador, quiero quitar un libro del carrito de la compra	...	2	6	2
Mejorar el rendimiento del procesador de transacciones (ver métricas de objetivos de rendimiento en el wiki)	...	3	6	13
Investigar soluciones para acelerar la validación de tarjetas de crédito (ver métricas de objetivos de rendimiento en el wiki)	...	4	6	20
Actualizar todos los servidores a Apache 2.2.3	...	5	5	13
Diagnosticar y arreglar los errores de los scripts de procesamiento de órdenes (ID bugzilla 14823)	...	6	2	3
Como comprador, quiero crear y guardar una lista de regalo	...	7	7	40
Como comprador, quiero añadir o borrar elementos en mi lista de regalo	...	8	4	20

Sprint backlog (Pila del Sprint)

Elemento de la Pila de Producto	Tarea del Sprint	Voluntario	Esfuerzo estimado inicial
Como comprador quiero poner un libro en el carrito de la compra	modificar base de datos		5
	crear página web (interfaz de usuario)		8
	crear página web (lógica Javascript)		13
	escribir pruebas de aceptación automáticas		13
	actualizar la página de ayuda del comprador		3
	...		
Mejorar el rendimiento de procesamiento de transacciones	juntar el código DCP y completar los test del nivel de capa		5
	completar la orden de máquina para pRank		8
	Cambiar el DCP y el lector para usar el API http de pRank		13



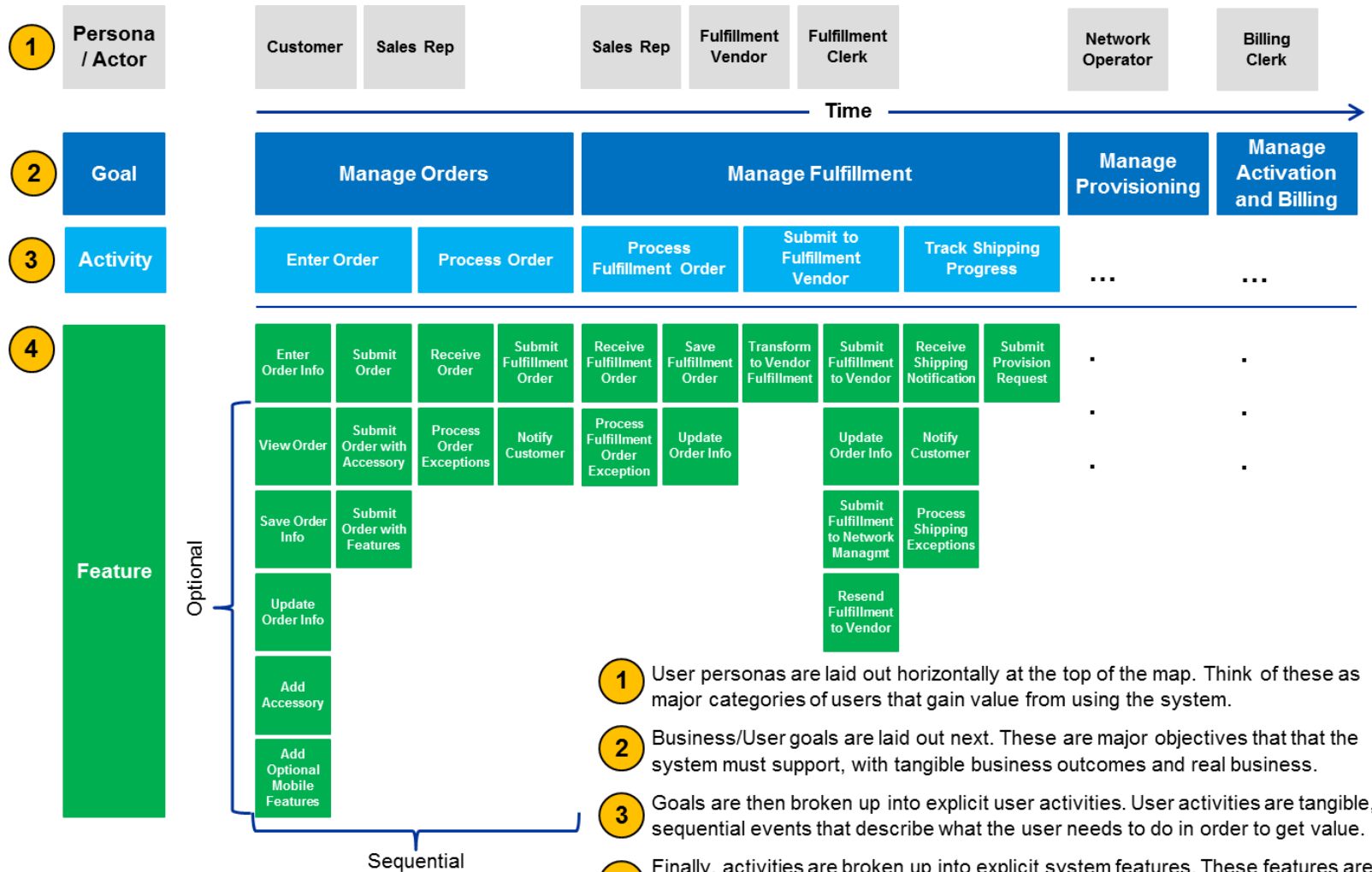
Historias de usuario

- Descripciones claras y concisas de la funcionalidad en términos del valor que aporta al usuario final del producto
- Estructura típica (formato “tarjeta”):
 - Como [rol] quiero [característica] para [valor de negocio]
 - Conversación: detalles adicionales
 - Criterios de aceptación para mostrar cuándo está completa (requisitos)
- Características:
 - Independientes, criterios suficientes para implementar. Pequeñas, que se puedan estimar y probar
- Otros “tipos” de historias:
 - Epic: Solo en el product backlog (funcionalidad general, todavía sin dividir)
 - Técnicas: Aunque no aporta funcionalidad, es necesaria
- Mito: Scrum impide hacer especificaciones detalladas? Depende del dueño de producto, equipo y cada historia.

Pizarras (Task Board)

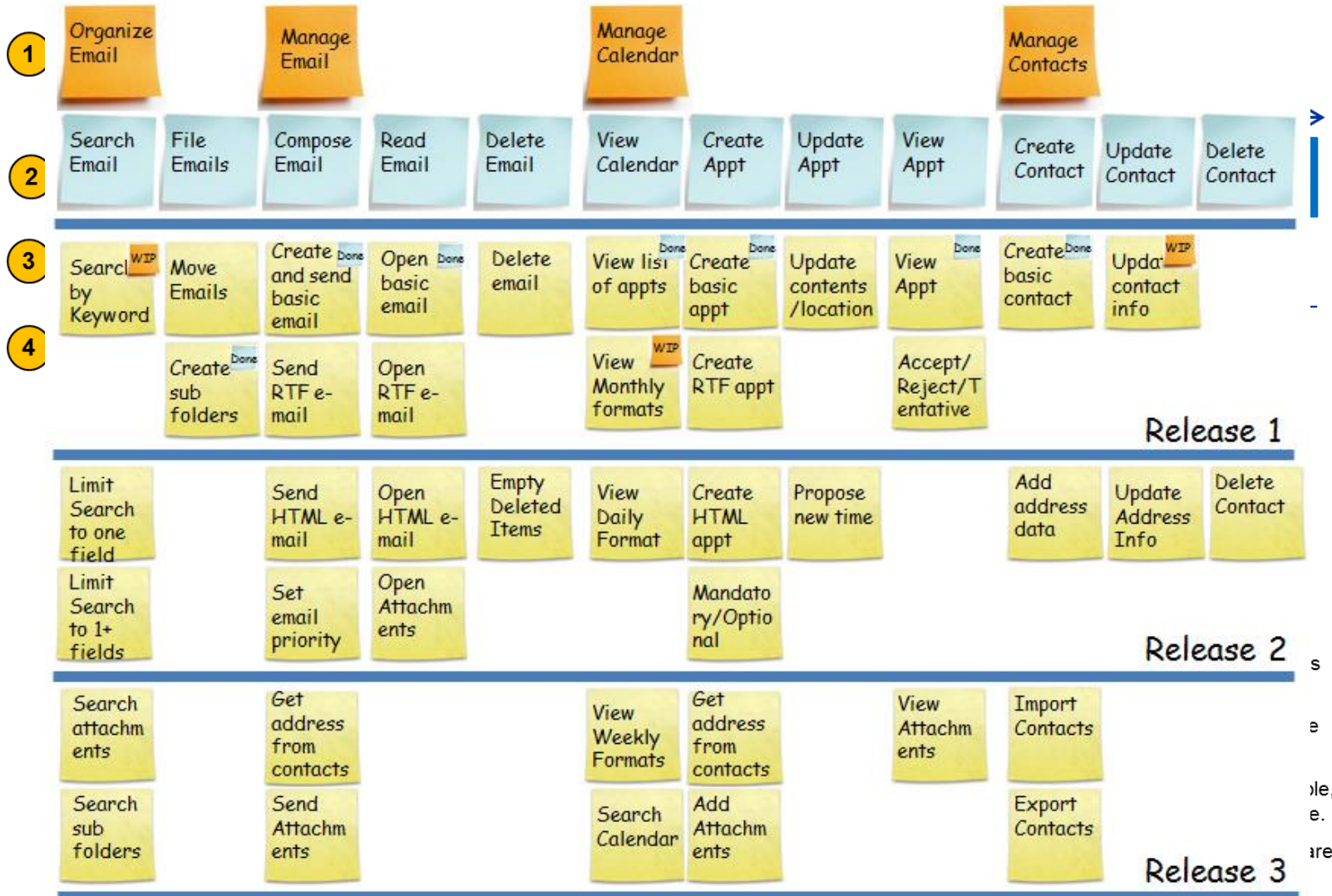


Story Mapping



- 1 User personas are laid out horizontally at the top of the map. Think of these as major categories of users that gain value from using the system.
- 2 Business/User goals are laid out next. These are major objectives that the system must support, with tangible business outcomes and real business.
- 3 Goals are then broken up into explicit user activities. User activities are tangible, sequential events that describe what the user needs to do in order to get value.
- 4 Finally, activities are broken up into explicit system features. These features are laid out directly underneath the user activity that the feature realizes.

Story Mapping

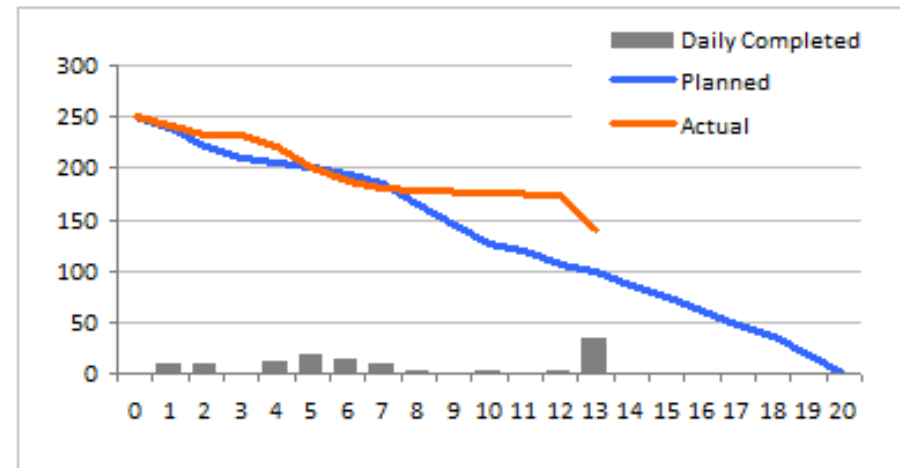


Scrum: proceso

■ Proceso:

- Evolución continua del Product Backlog
- Sprints

- Al iniciar el sprint: Sprint planning meeting -> sprint backlog.
- Ejecución y seguimiento del progreso
 - Scrum diario: Stand-up meeting
 - **Historias de usuario HECHAS** (incremento de producto “potencialmente entregable”)
 - Velocidad (progreso): Burn-down charts.
- Al finalizar el sprint: Review & Retrospective: demostración y revisión.



Planificación de cada Sprint (Sprint Planning Meeting)

- Dos reuniones (un día en total)
- Reunión 1 (medio día), requisitos:
 - Definir QUÉ se va a hacer, entender qué quiere el DP
 - DP+Equipo revisan/discuten/refinan product backlog (elementos más prioritarios no finalizados), Incluyendo historias y criterios de aceptación
 - Scrum master presta asistencia a todos
 - Compromiso de lo que se realizará en el sprint
 - Dueño de producto decide las prioridades
 - Equipo decide el volumen de tarea a realizar (empezando por el principio de la pila)

Planificación de cada Sprint (Sprint Planning Meeting)

- Reunión 2 (medio día), planificación:
 - Definir CÓMO se va a hacer (planificación detallada de tareas):
 - Ej. obtener información adicional, diseñar pantallas, cambio en BD, programación de partes del entregable, hacer pruebas (ver siguiente)
 - Selección final de elementos para el sprint backlog
 - En función de la estimación final de cada una (p.e. horas)
 - DP puede no asistir, pero debe estar disponible (p.e. si hay que reajustar/negociar el compromiso del trabajo a realizar)



Scrum diario (Stand-up meeting)

- Reunión corta (ej. 15 minutos), todo el equipo
- Cada miembro informa
 - Qué ha hecho desde la última reunión
 - Qué tiene planificado hacer antes de la siguiente reunión
 - Cualquier bloqueo o impedimento que tenga
- No hay discusiones, si acaso en reunión de seguimiento posterior
- Aconsejable que no acudan jefes



Cierre del Sprint (Review & Retrospective)



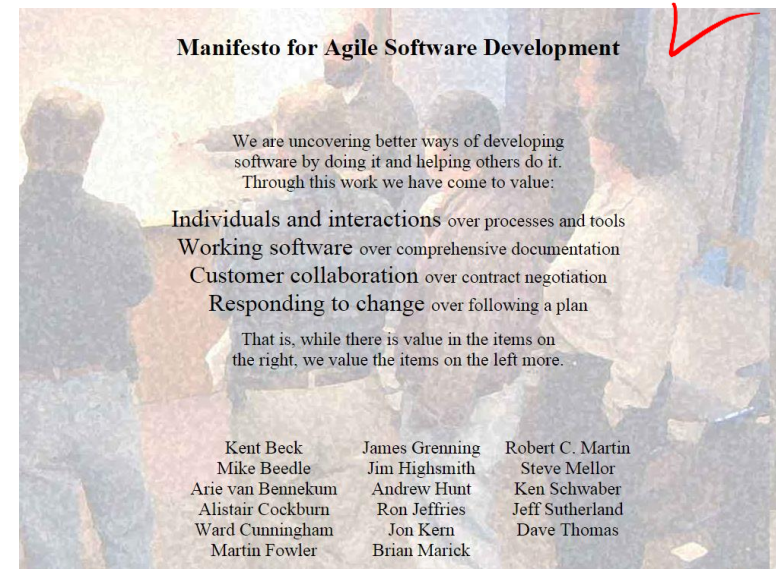
- Reunión 1 (medio día): Revisión del sprint (Sprint Review)
 - Demo y Conversación en profundidad Equipo+DP (posiblemente otras partes interesadas)
 - Inspeccionar el producto para adaptarlo en posteriores sprints y obtener feedback
- Reunión 2 (<medio día): Retrospectiva (Retrospective)
 - Inspeccionar y adaptar el proceso (no hace falta DP)
 - Qué fue bien, qué no, qué puede mejorar
- Actualización
 - Product Backlog
 - Gráfico de trabajo restante de TODO el producto (Backlog Burndown Chart)

Refinamiento de pila de producto (backlog grooming)

- Facilita la planificación de sprints posteriores
- Se hace durante un sprint, en referencia a los elementos futuros (que no están incluidos en él)
 - Analizar los requisitos
 - Separar elementos grandes en pequeños
 - Estimar nuevos elementos
 - Reestimar existentes
- Típicamente 5-10% del tiempo de un sprint, cerca del final, en forma de taller (otros recomiendan semanalmente, no al final)

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”.



Scrum: equipo y roles

- Team =
 - Product Owner + Scrum Master + Scrum Team
- Dueño del Producto (Product Owner)
 - Realiza el papel de cliente (describe y prioriza, soluciona preguntas)
 - Representa al negocio y a los clientes
 - Interactúa con el resto de partes interesadas (p.e. otros usuarios y clientes finales)
 - Total autoridad en toma de decisiones (producto, alcance, presupuesto, prioridades)
 - Interactúa activa y frecuentemente con equipo



Scrum: equipo y roles

■ Gestor de Scrum (Scrum Master):

- ☐ Ayuda a aplicar Scrum (coach): entrenador, facilitador, experto en Scrum
- ☐ Elimina impedimentos (p.e. bloqueos en espera de algún recurso)
- ☐ También puede realizar trabajos como el equipo (working scrum master). Algunos lo desaconsejan
- ☐ No es el jefe del equipo (no existe jefe)
- ☐ Se adapta a la experiencia del equipo



Scrum: equipo y roles

- Equipo de SCRUM (Scrum Team):
 - Autogestionado (alto grado de autonomía, responsabilidad y compromiso)
 - Estimación de tiempos a emplear
 - Decide cómo realizar el trabajo (p.e. herramientas)
 - Se encarga de todos los procesos del desarrollo
 - Multifuncional, aunque tenga expertos en temas concretos, no debe ser una limitación
 - 7 ± 2 personas, estable, experiencia

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 Ostry
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/>

Ejemplo introductorio para repaso y prácticas ✓

Dadas las siguientes historias de usuario

1. Como Atleta quiero solicitar mi inscripción en una competición dentro de los plazos establecidos

Criterios de aceptación: Se registra la inscripción del atleta siempre y cuando existan plazas disponibles en la competición y el plazo de solicitud esté abierto. No se admiten inscripciones duplicadas de un mismo atleta a la competición.

2. Como Atleta quiero que el cálculo de mi categoría en la inscripción se realice de acuerdo a mi fecha de nacimiento

Criterios de aceptación: A partir de la fecha de nacimiento del atleta se asigna una categoría en función del sexo (Masculino, Femenino) y el rango de edad. Las categorías son específicas de cada competición, aunque en algunos casos se comparten por varias competiciones

Elaborar modelo de datos de dominio para comenzar desarrollo