

Procesos - Scrum

Ingeniería del Proceso Software

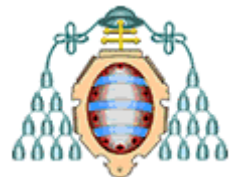
M^a José Suárez Cabal

Javier Tuya (<http://giis.uniovi.es/tuya/>)

Grupo de Investigación en Ingeniería del Software

<http://giis.uniovi.es/>

Curso 2022-2023



Contenido

- Proceso de Scrum
- Roles
- Pila de Producto e historias de usuario
- Planificación del Sprint
- Sprint diario y actualización
- Refinamiento de la pila y cierre del sprint
- Pizarras
- Otras prácticas (Release planning, Story mapping, Pruebas, Variantes), otros métodos

Historia

- 80's: modelo de desarrollo identificado y definido por Ikujiro Nonaka e Hirotaka Takeuchi. Conceptos procedentes de años anteriores
 - Resultado de analizar la forma en la que desarrollaban los nuevos productos las principales empresas de manufactura tecnológica: Fuji-Xerox, Canon, Honda, Nec, Epson, Brother, 3M y Hewlett-Packard.
- 1993: Primer equipo SCRUM para software: Jeff Sutherland (Easel Corporation)
- 1995: Proceso formalizado y publicado (OOPSLA)
- 2001: Manifiesto Ágil (establece los principios, acuña el término métodos ágiles)

Manifiesto ágil

■ Valora:

- ☐ **Individuos e interacciones** más que procesos y herramientas.
- ☐ **Software que funciona** más que documentación exhaustiva.
- ☐ **Colaboración con el cliente** más que negociación de contratos.
- ☐ **Responder ante el cambio** más que seguimiento de un plan.

Manifiesto ágil. Principios (I)

- La prioridad es **satisfacer al cliente** con entregas tempranas y continuas de software que aporte valor.
- Aceptación de **cambios en los requisitos**, incluso si llegan tarde al desarrollo.
- Entrega frecuente de **software que funcione**.
- Gente de negocio y los desarrolladores deben **trabajar juntos**.
- **Individuos motivados**. Darles el entorno y el apoyo que necesitan y confiar en que ellos conseguirán hacer el trabajo.
- **Comunicación** entre los miembros de un equipo de desarrollo mediante conversación cara a cara.
- Medida de progreso: **software que funciona**.

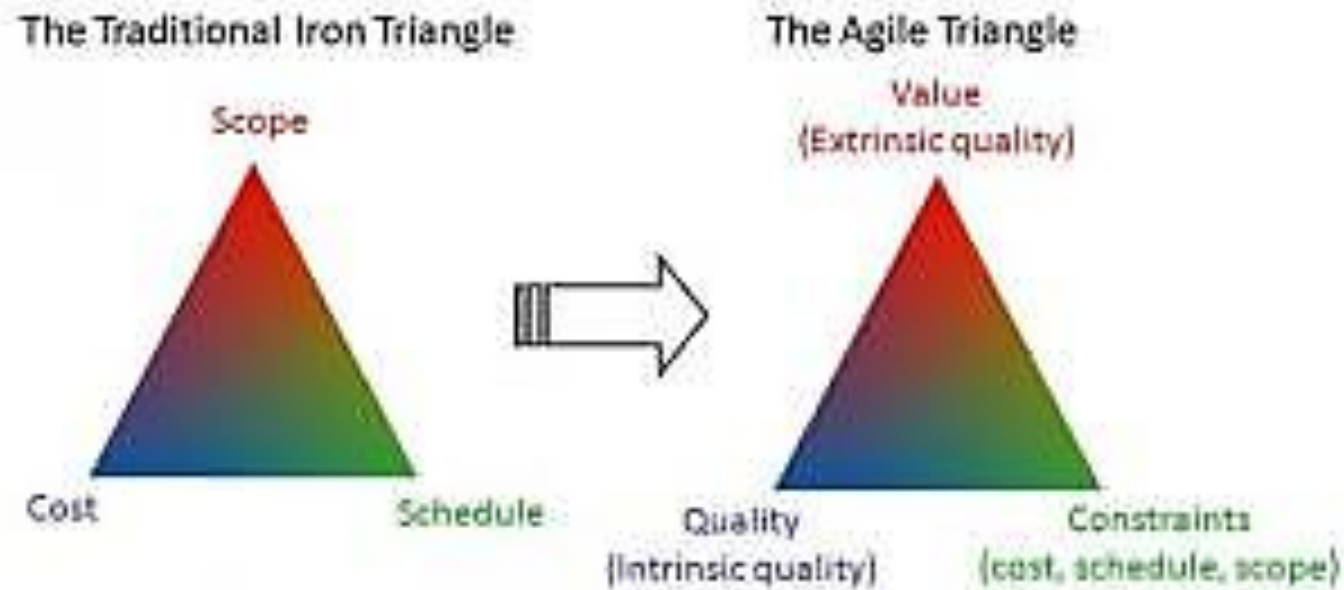
Manifiesto ágil. Principios (II)

- Desarrollo sostenido: promotores, desarrolladores y usuarios deben 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**, el arte de maximizar la cantidad de trabajo que no se hace, es esencial.
- Las mejores arquitecturas, requisitos y diseños emergen de **equipos autoorganizados**.
- En intervalos regulares, **el equipo reflexiona** en cómo ser más efectivo, **se afina y ajusta** su comportamiento de acuerdo con esto.

Proceso de SCRUM

- Es un Framework (marco de trabajo y gestión) más que una metodología
- Ciclos cortos (Sprints): típicamente 1-4 semanas. Entregas tras cada ciclo
- Equipo multifuncional:
 - Autoorganizado
 - Autogestionado
 - Altamente colaborativo
- Bastante utilizado en industria
 - Especialmente con requisitos no prefijados/volátiles
 - Requiere confianza y colaboración cliente-equipo de desarrollo

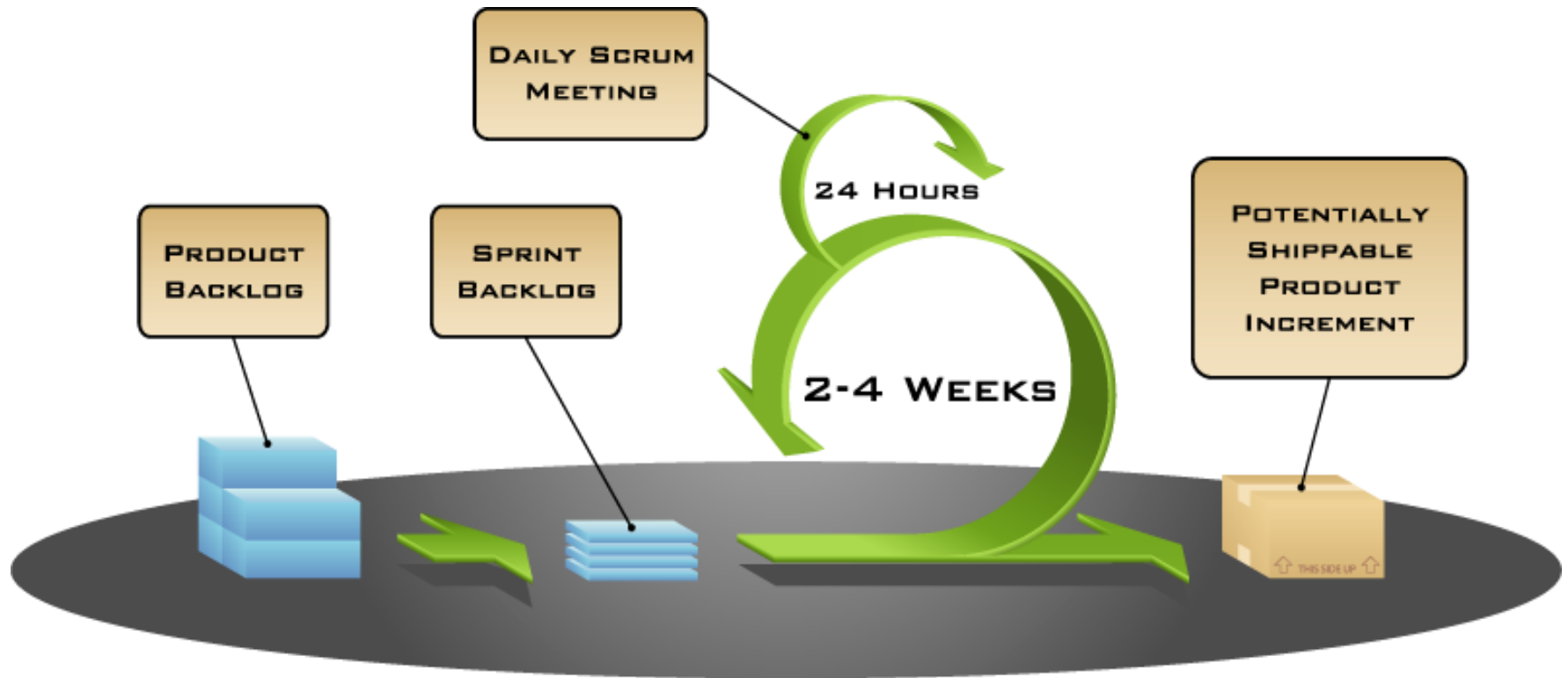
Proceso de SCRUM



J.Highsmith (2011) https://dfwscrum.files.wordpress.com/2011/07/dfwscrum_20110726_jimhighsmith.pdf

- Métodos “Clásicos”: alcance fijo
- Métodos Ágiles: valor para el cliente

Proceso de SCRUM



COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE

<http://www.scrum.org/>

Roles

- SCRUM Team =
 - Product Owner + Scrum Master + Development Team
- Dueño del Producto (Product Owner)
 - Realiza el papel de cliente (describe y prioriza, refina, soluciona preguntas, revisa)
 - 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

Rol:

Gestor de Scrum (Scrum Master)

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

Rol: Development 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 (cross-functional)
- Aunque tenga expertos en temas concretos, no debe ser una limitación (aprendizaje múltiple)
- 7 ± 2 personas, estable, experiencia

Pila de Producto (los requisitos)

■ Pila de Producto (Product Backlog)

- ☐ Lista priorizada de funcionalidades (ordenada)
- ☐ Evoluciona a lo largo del proyecto
- ☐ Funcionalidades, objetivos de mejora, trabajo exploratorio, correcciones
- ☐ Las estimaciones las suministra el equipo de desarrollo. Prácticas típicas (ej. Planning Pocker)
- ☐ El dueño de producto asigna un valor de negocio
- ☐ Scrum no indica
 - Cómo estimar (puntos, horas, días-persona)
 - Cómo detallar pila de producto (aconsejado: historias usuario)

Pila de Producto. Características

■ DEEP

- ☐ D: detallado apropiadamente
- ☐ E: estimado
- ☐ E: emergente, refinado regularmente
- ☐ P: priorizado, ordenado

Pila de 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

Historias de usuario

- Descripciones claras y concisas de la funcionalidad en términos del valor que aporta al usuario final del producto
- Objetivo: definir tareas a realizar (backlog) y detalles del sistema a implementar
- INVEST
- Estructura típica (formato “tarjeta”): Como [rol] quiero [característica] para [valor de negocio]
 - Conversación: detalles adicionales
 - Criterios de aceptación para mostrar cuándo está completa
- Mito: Scrum impide hacer especificaciones detalladas? Depende del dueño de producto, equipo de desarrollo y cada historia.

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 de desarrollo revisan/discuten Pila (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 de desarrollo decide el volumen de tarea a realizar (empezando por el principio de la pila)

Planificación de cada Sprint

- 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 de la pila para el Sprint
 - Tras haber medido el coste de cada tarea (p.e. horas)
 - DP puede no asistir, pero debe estar disponible (p.e. si hay que reajustar/negociar el compromiso del trabajo a realizar)

Planificación de cada 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

Planificación de cada Sprint

- Reparto de elementos de la pila
 - Hasta acabar la duración del Sprint
 - Ocupando todo el tiempo disponible del equipo de desarrollo (descontando refinamiento de pila de producto, reuniones y ausencias)
- Sprint backlog
 - Lista de todas las historias acordadas para el sprint
 - Tareas asociadas
 - Otras tareas necesarias (no asociadas a historias)
 - P.e. mejoras, refactorizaciones, historias “de investigación” o “técnicas”

Planificación de cada Sprint

■ Cambios durante el sprint

- ☐ Duración fijada a priori (no cambiar duración)
- ☐ DP puede cambiar pila del producto (del sprint) antes de empezar el sprint
- ☐ Cualquier cambio se deja para otro sprint
- ☐ Las tareas sí pueden evolucionar durante el sprint
- ☐ Causas excepcionales de cambios de prioridades: el DP puede terminar el sprint

■ Claves:

- ☐ Equipo de desarrollo protegido durante el sprint
- ☐ Disciplina del DP para pensar la priorización

Planificación de cada Sprint

- Definir qué significa Hecho
 - ☐ En general: incremento potencialmente entregable
- Ejemplo:
 - ☐ codificado
 - ☐ según los estándares
 - ☐ revisado
 - ☐ implementado con desarrollo orientado a pruebas (TDD)
 - ☐ probado 100% con pruebas automáticas en su caso
 - ☐ integrado (integración continua)
 - ☐ documentado

Scrum diario (Stand-up meeting)

- Reunión corta (ej. 15 minutos), todo el equipo de desarrollo
- 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

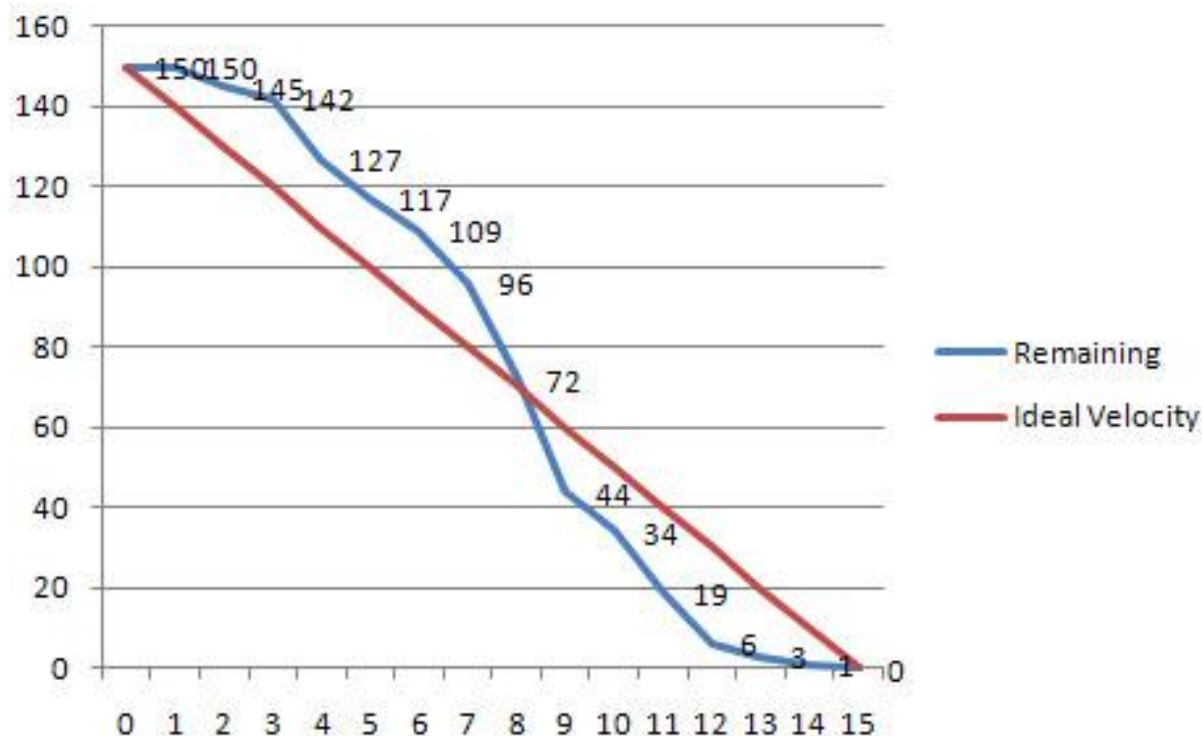
Actualización Sprint diario

- Equipo de desarrollo actualiza estimaciones y trabajo restante

				Nuevo esfuerzo estimado Lo que queda al final del día.					
Elemento de la Pila de Producto	Tarea del Sprint	Voluntario	Esfuerzo estimado inicial	1	2	3	4	5	6
Como comprador quiero poner un libro en el carrito de la compra	modificar base de datos	Sanjay	5	4	3	0	0	0	
	crear página web (interfaz de usuario)	Jing	3	3	3	2	0	0	
	crear página web (lógica Javascript)	Tracy & Sam	2	2	2	2	1	0	
	escribir pruebas de aceptación automáticas	Sarah	5	5	5	5	5	0	
	actualizar la página de ayuda del comprador	Sanjay & Jing	3	3	3	3	3	0	
	...								
Mejorar el rendimiento de procesamiento de transacciones	juntar el código DCP y completar los test del nivel de capa		5	5	5	5	5	5	
	completar la orden de máquina para pRank		3	3	8	8	8	8	
	Cambiar el DCP y el lector para usar el API http de pRank		5	5	5	5	5	5	

Actualización Sprint diario

- Representa en forma gráfica lo que queda por hacer (Gráfico de Tiempo Restante: Burndown Chart).
Velocidad



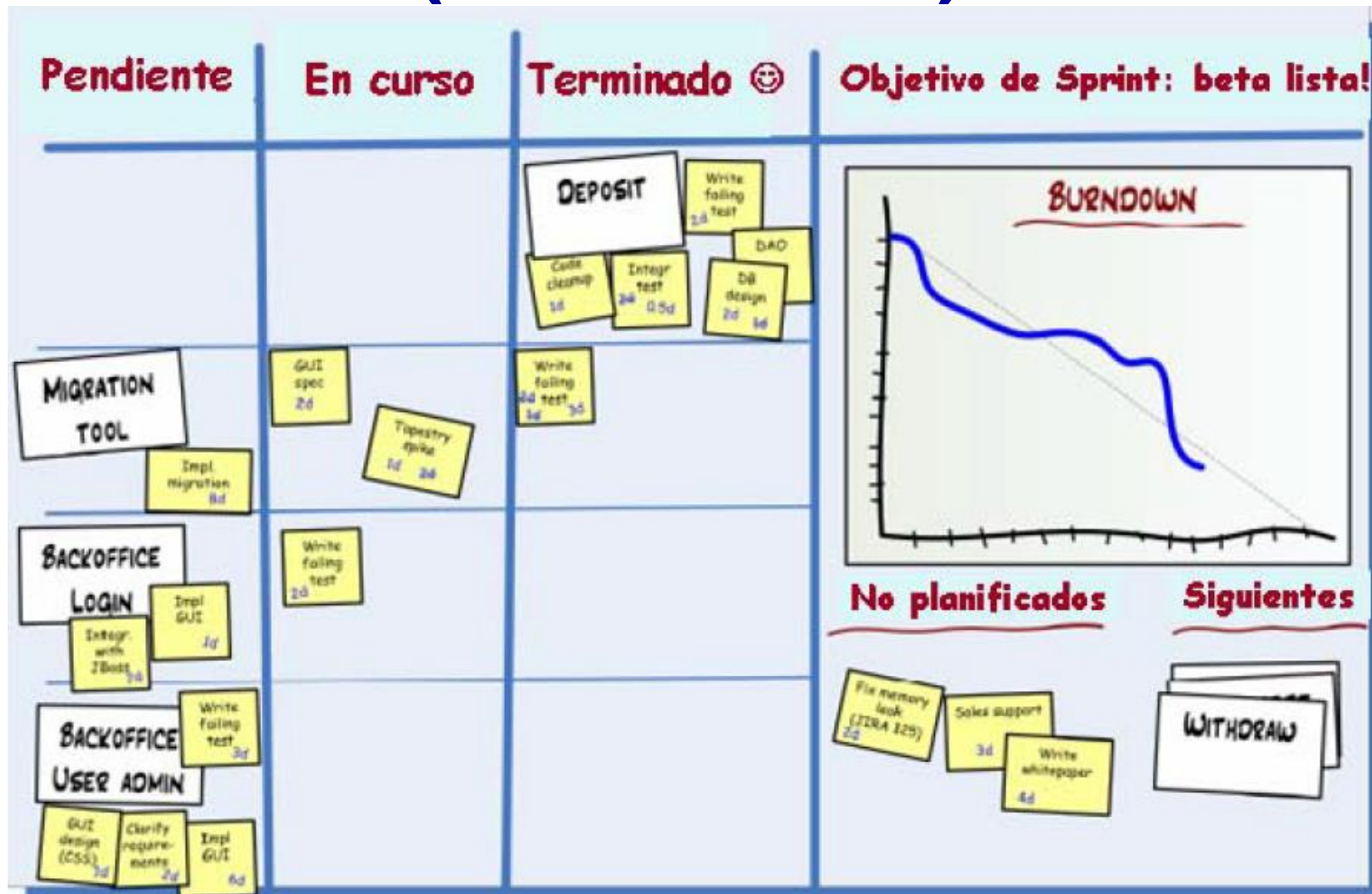
Refinamiento de pila de producto (backlog grooming)

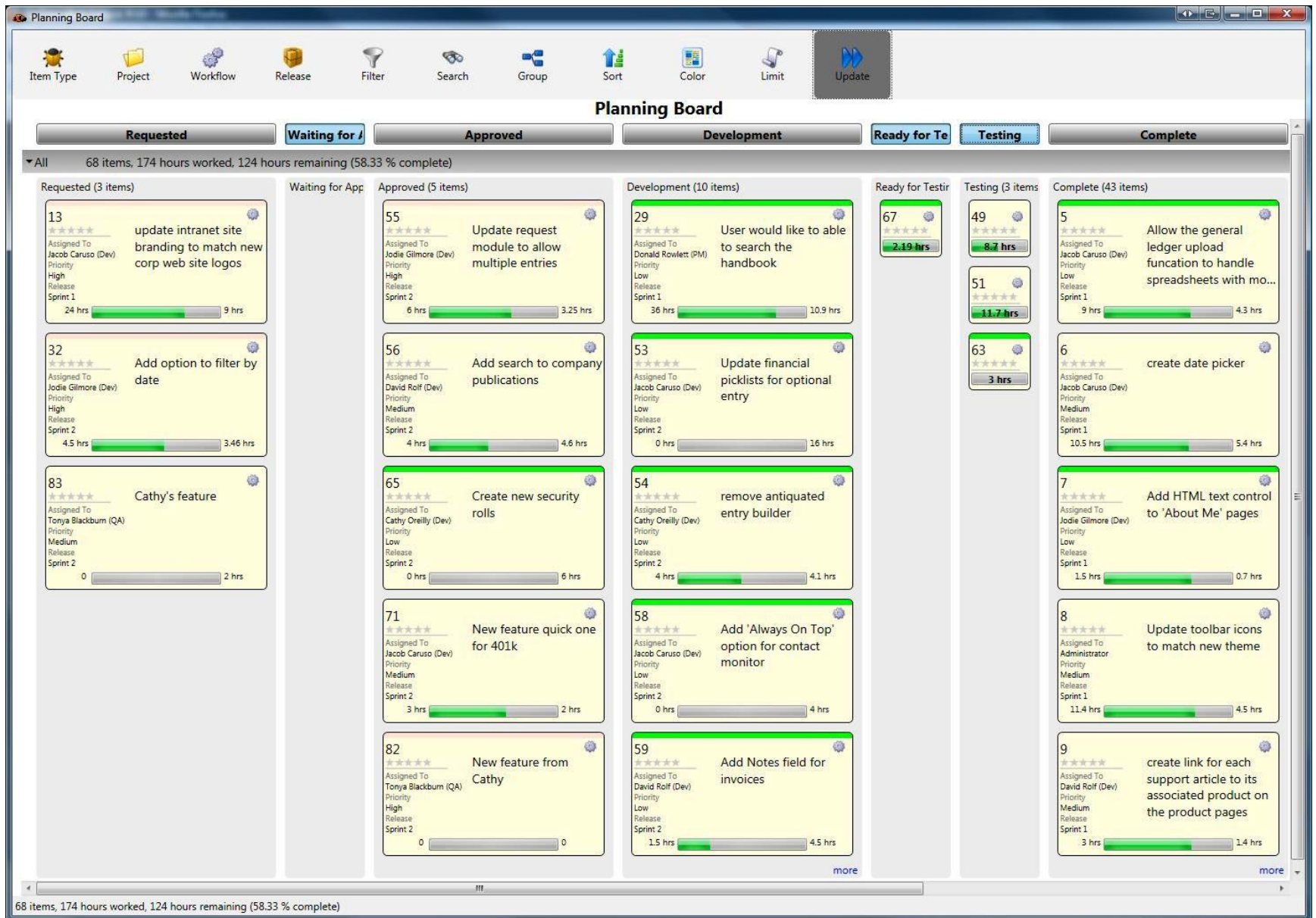
- Participa todo el equipo de SCRUM
- Facilita la planificación de sprints posteriores, manteniendo características del producto backlog (DEEP)
- Se hace durante un sprint, en referencia a los elementos futuros (que no están incluidos en él)
 - Analizar los requisitos
 - Generar nuevos elementos; identificar elementos obsoletos del backlog del producto
 - Separar elementos grandes en pequeños
 - Estimar nuevos elementos
 - Reestimar existentes
 - Cambiar prioridad de tareas existentes
- Típicamente 5-10% del tiempo de un sprint, cerca del final, en forma de taller (otros recomiendan semanalmente, no al final)

Cierre del Sprint

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

Pizarras (Task Board)





Otras prácticas (variaciones)

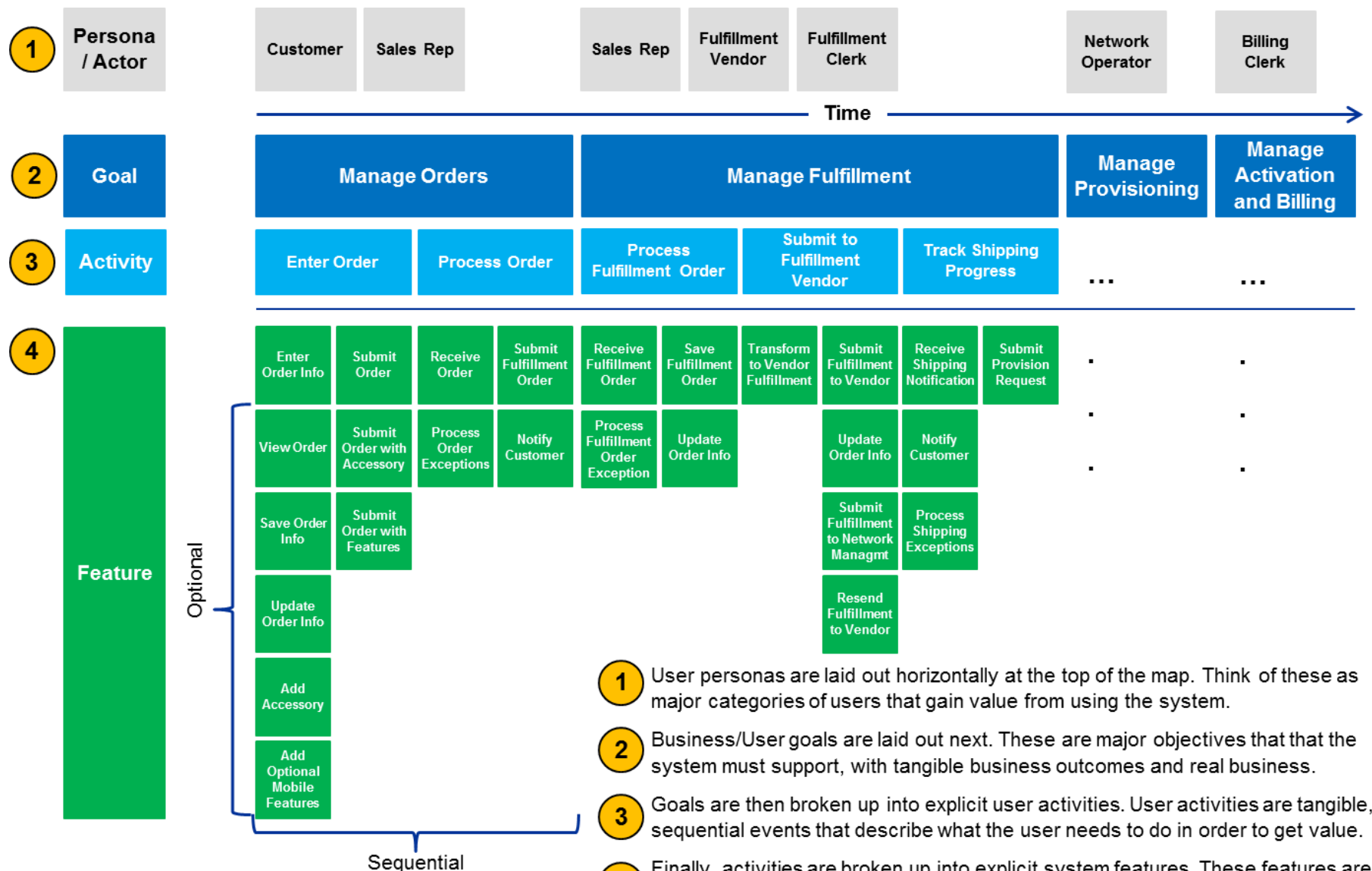
- Adaptación a proyecto y organización. Algunos ejemplos
- Productos nuevos o mantenimientos que adoptan Scrum por primera vez:
 - Refinar pila de producto antes del primer sprint
 - Especificaciones/Diseño arquitectura previo
- Grado de confianza en el equipo de desarrollo (alcance cerrado primera fase, alcance abierto siguientes fases)
- Proyectos muy cerrados con poca disponibilidad de cliente final (papel del product owner y los requisitos)
- Equipos con especialistas en UI/UX, front-end, back-end, base de datos, pruebas
- Discusión: prácticas de laboratorio

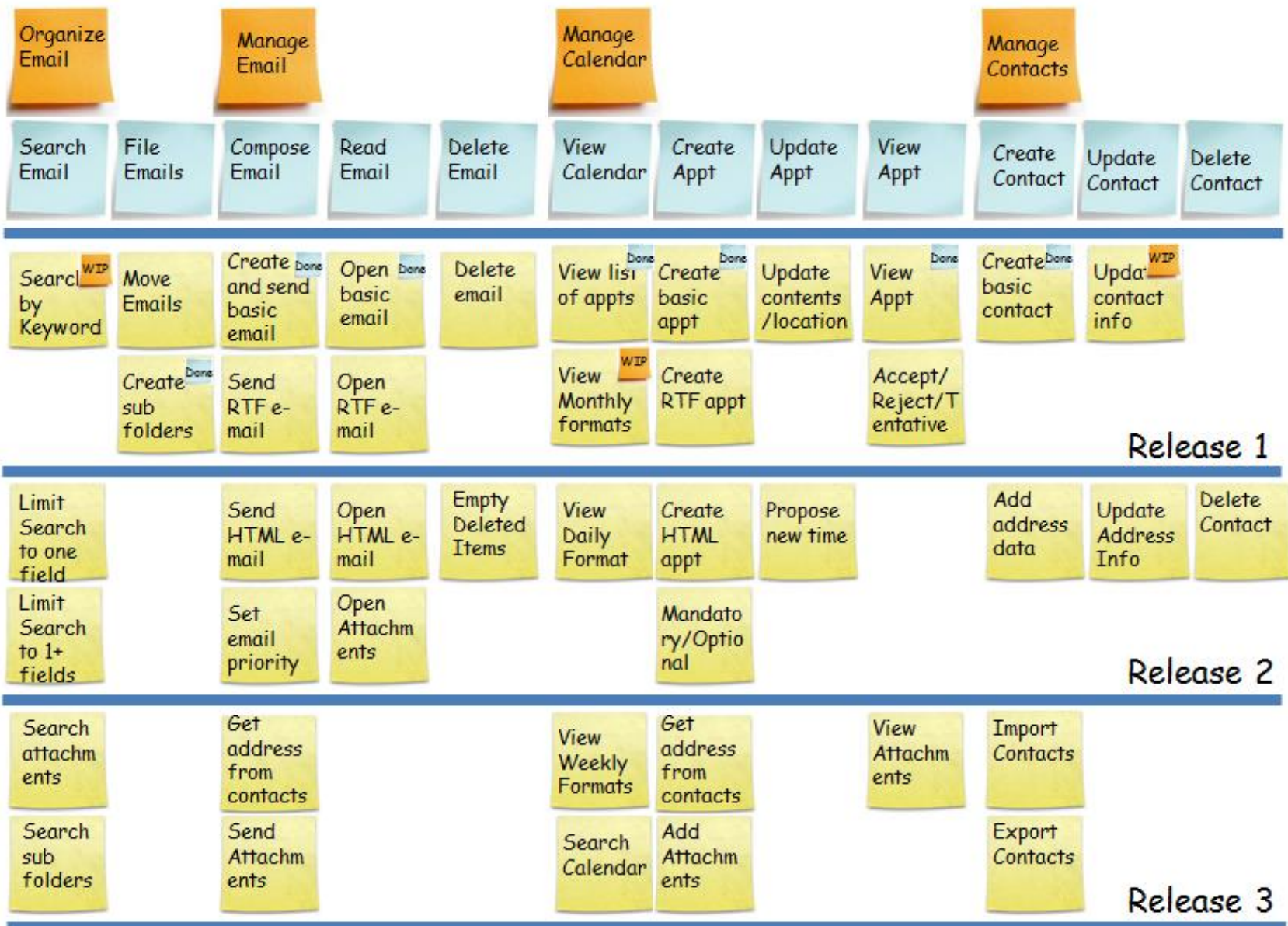
Otras prácticas

- Planificación de entregas (Release Planning).
Elegir qué historias se incluirán en la siguiente versión y cuándo (incluirlá varios sprints)
 - ☐ Alcance fijo (fecha variable)
 - ☐ Fecha fija (alcance variable)
 - ☐ Alcance y fecha fijo (problemas)
 - ☐ Compromiso alcance, velocidad, coste (triángulo de oro)

Otras prácticas

- Mapeo de Historias (Story Mapping)
 - El backlog del producto es unidimensional
 - Con esto le damos estructura bidimensional
 - Columnas: Agrupa los procesos/usuarios/funcionalidades. Recomendable ordenar siguiendo el proceso
 - Filas: Las historias con sus prioridades
 - Mejorar la visión de producto y construir una hoja de ruta clara y compartida por el equipo
 - Dinámico, evoluciona con el proyecto





Pruebas

- Parte integral y esencial de todos los métodos ágiles
- TDD: Test-Driven Development
 - Definir primero las pruebas (automatizadas)
 - Implementar y ejecutar las pruebas (lo necesario para que no fallen)
 - Refactorizar
- BDD: Behaviour-Driven Development
 - Evolución de TDD
 - Usar historias de usuario con estructura, formato semi-formal que se convierten en pruebas
- Otros tipos de pruebas, niveles, aceptación (Asignatura CVVS)

Otros métodos Ágiles

- Extreme Programming (pionero)
- Concepto de Lean:
 - Eliminar desperdicio, todo lo innecesario que no aporta valor
- Kanban (Manufactura Japón, 50's). Just In Time
 - Limitar el trabajo en proceso.
 - No preasignación de las tareas, empezar una al acabar otra
- Scrumban (híbrido Scrum-Kanban)
 - En vez de estimar cada ítem del backlog, tener estimación del promedio de cada uno (número de slots fijo)
 - Reducción en planificación de las iteraciones

Resumen/Discusión

- De modelo centrado en proyecto a centrado en desarrollo continuo de producto
- Define fundamentalmente cómo organizarse
- No prescribe todas las prácticas que se pueden utilizar, no prohíbe
- Multifuncionalidad y responsabilidad del equipo de desarrollo
- Involucración del cliente
- No soluciona los problemas de desarrollo, pero los hace visibles pronto

Bibliografía

- <http://www.scrum.org/> (accedido Octubre 2021)
- Deemer P, Benefield G, Larman C, Vodde B (2012). The Scrum Primer (Version 2.0). The Scrum Primer. <http://scrumprimer.org/> (accedido Octubre 2021)
- Sims C, Johnson HL (2011). The Elements of SCRUM
- Cohn M (2010). Succeeding with Agile Software Development Using Scrum. Addison-Wesley
- Cohn M (2006) Agile Estimating and Planning (2006). Pearson