

Metodologías Agiles - Repaso

GESTIÓN DE CALIDAD EN EL SOFTWARE

Virginia Cuomo
Mariela Castarés

Primer cuatrimestre 2016

Agenda

- Origen
- Valores y Principios
- Ejemplos
 - Scrum
 - XP
 - Scrum y XP
- Agilidad o Disciplina
- Bibliografía

Metodologías Agiles - Origen



El **mundo de los negocios** ha cambiado (la democratización de la tecnología, las finanzas e información, la globalización, etc.), los ciclos de los productos son más cortos, los clientes son más fluctuantes; por ende, no es sorprendente que metodologías que funcionaron correctamente en los años 70's , hayan comenzado a requerir cambios, a partir de la década de los 90's.

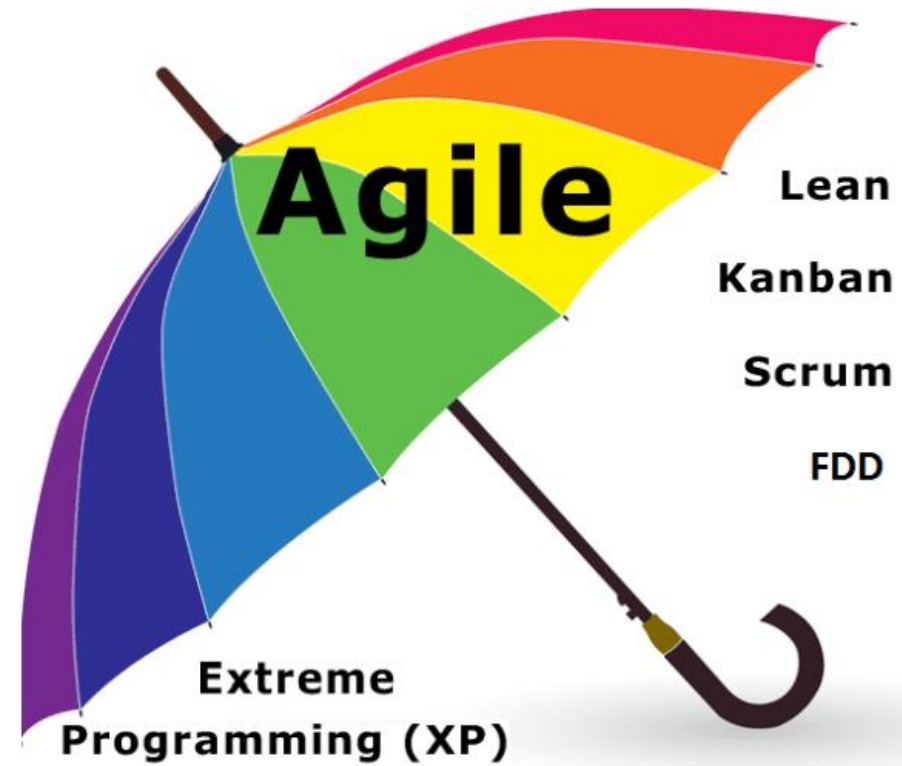
Metodologías Ágiles - Origen

Por ello, durante los 90, un grupo de diferentes personas que se dieron cuenta de que se habían producido estos cambios comenzaron a interesarse en metodologías para desarrollo de software que se adaptaran más fácilmente al *nuevo mundo de los negocios*.

El cambio es inevitable

Metodologías Ágiles - Origen

A pesar de que los detalles de estas metodologías difieren, **todas comparten principios básicos**, por este motivo todas ellas, se agruparon bajo el nombre de “**Metodologías Ágiles**”.



Metodologías Agiles – Valores y principios

Todas las metodologías ágiles, difieren en algunos detalles, pero presentan determinados ***valores y principios*** los cuales se acuerdan en una



Alianza Ágil

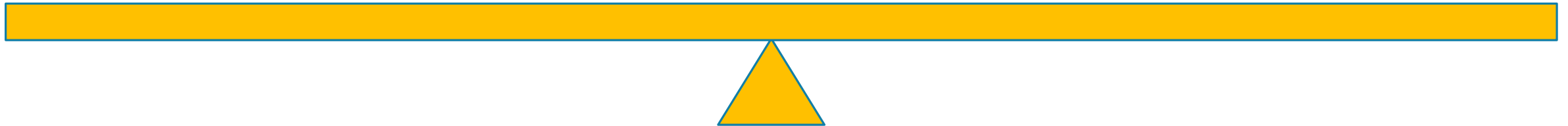


Valores y principios - Manifiesto Agil

“Estamos poniendo al descubierto mejores métodos para desarrollar software, haciéndolo y ayudando a otros a que lo hagan. Con este trabajo hemos llegado a valorar: “

Individuos e interacciones
Software funcionando
Colaboración con el cliente
Respuesta ante el cambio

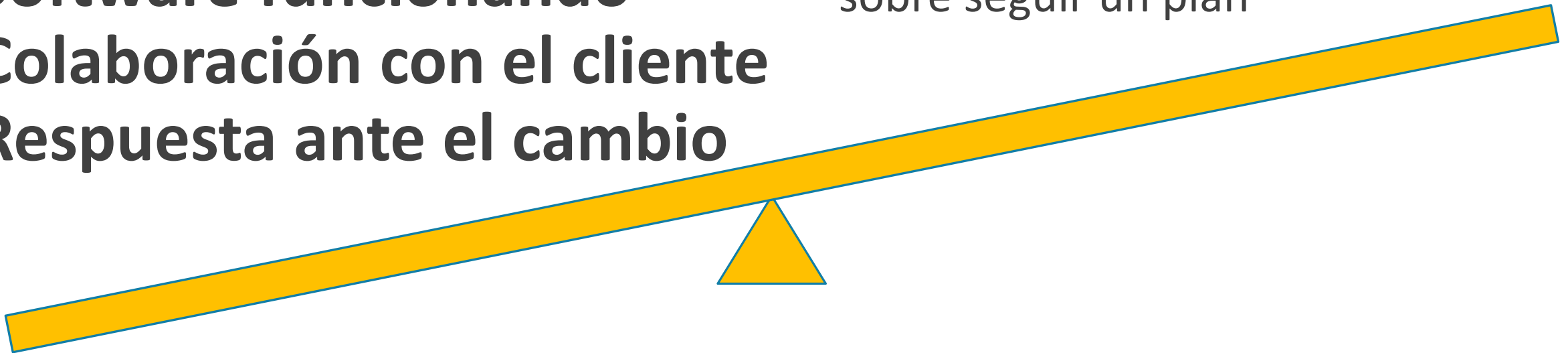
sobre procesos y herramientas
sobre documentación extensiva
sobre negociación contractual
sobre seguir un plan



Valores y principios - Manifiesto Agil

Individuos e interacciones
Software funcionando
Colaboración con el cliente
Respuesta ante el cambio

sobre procesos y herramientas
sobre documentación extensiva
sobre negociación contractual
sobre seguir un plan



Esto es, aunque valoramos los elementos de la derecha, valoramos más los de la izquierda.

Valores y principios I

- 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.

Valores y principios II

- 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**.
- 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**.

Valores y principios III

- 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.

Ejemplos de Metodologías Ágiles

- Metodologías ágiles para la gestión de Proyectos
 - ✓ SCRUM

- Metodologías ágiles para el desarrollo de software
 - ✓ XP



Metodologías Agiles

Todas buscan:

- La **satisfacción del Cliente** dando respuestas con calidad y rapidez a sus necesidades.
- Velocidad y flexibilidad
- Potenciar al máximo el **trabajo del equipo**
- Se basan en modelos **Adaptativos**

“No es más fuerte la especie que sobrevive sino aquella que es capaz de adaptarse”

(Charles Darwin)



Scrum

- En **1993 Jeff Sutherland** concibió, ejecutó y documentó el primer Scrum para desarrollo ágil de software con el convencimiento de que los equipos multidisciplinarios lograrían mejor performance.
- En **1995, Ken Schwaber** formalizó el proceso para la industria del software.

Desde entonces, se volvió una de las metodologías ágiles de referencia en SW

La **Scrum Alliance** es la organización sin fines de lucro que se encarga de difundir Scrum en el ámbito del desarrollo de software.

www.scrumalliance.org



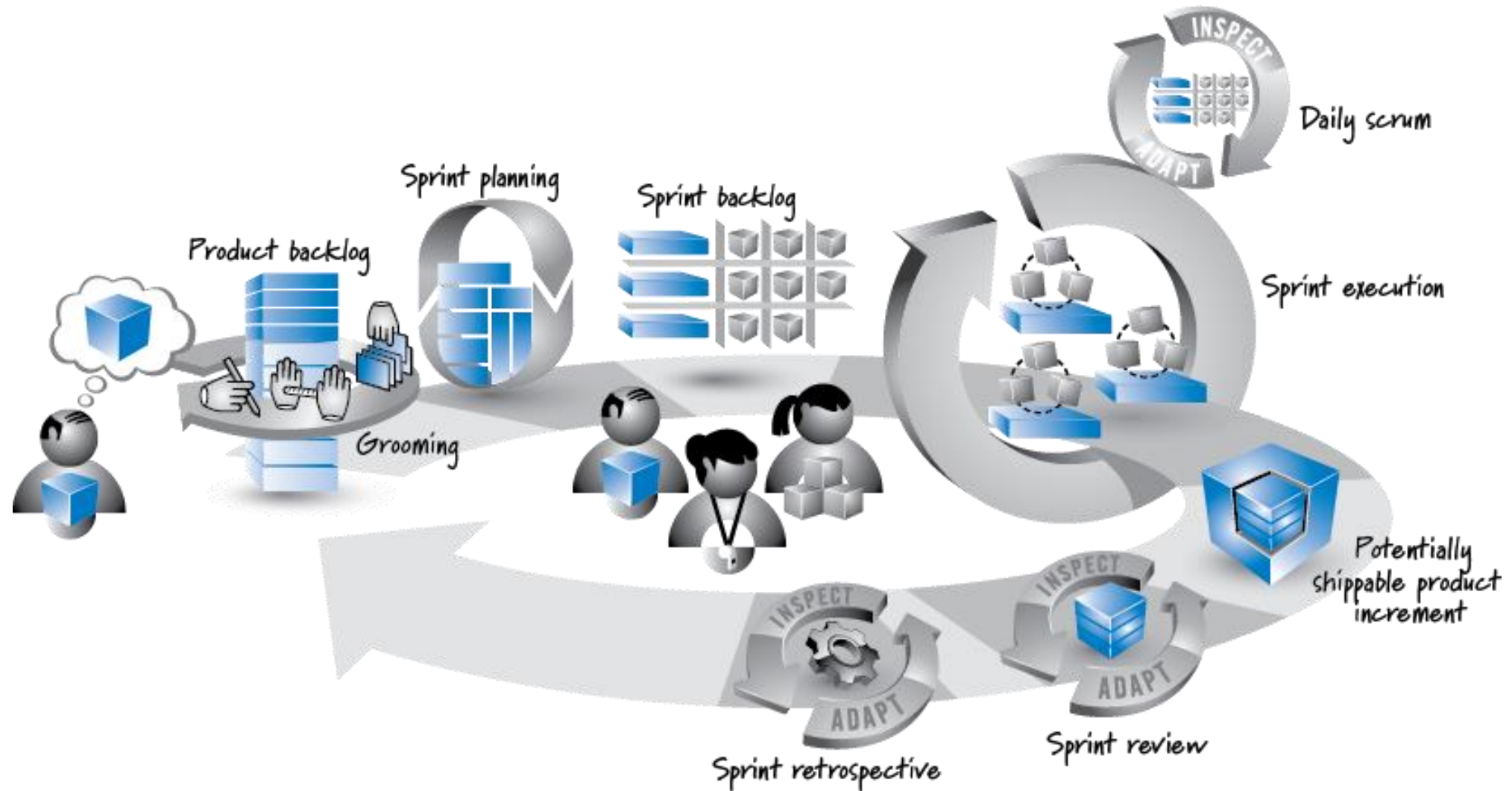
¿En qué consiste SCRUM? I

- Scrum es un framework para **la gestión ágil de proyectos** que puede ser aplicado tanto en software, como en cualquier otra industria.
- No contiene prácticas de ingeniería o ninguna otra práctica específica de trabajo.
- Busca siempre una forma de **maximizar la ganancia sobre la inversión de trabajo**.
- Ciclos de desarrollo iterativos incrementales.
- El objetivo es tener **software entregable** al final de cada iteración incluso si constituye sólo una pequeña porción del producto deseado.
- **Entregas parciales** priorizadas por el valor directo percibido por el receptor del entregable.

¿En qué consiste SCRUM? II

- **Equipos pequeños** de trabajo que maximizan la comunicación, minimizan el overhead, maximizan la cooperación y el conocimiento compartido.
- **Flexible y adaptable** a los cambios tanto técnicos como de negocio con el fin de asegurar el mejor producto construido para el Cliente.
- Promueve la **detección y eliminación de obstáculos** que se interpongan en el proceso de desarrollo y entrega de los productos.
- Forma de trabajo **escalable** de un simple proyecto a grandes organizaciones o empresas.

¿En qué consiste SCRUM? III



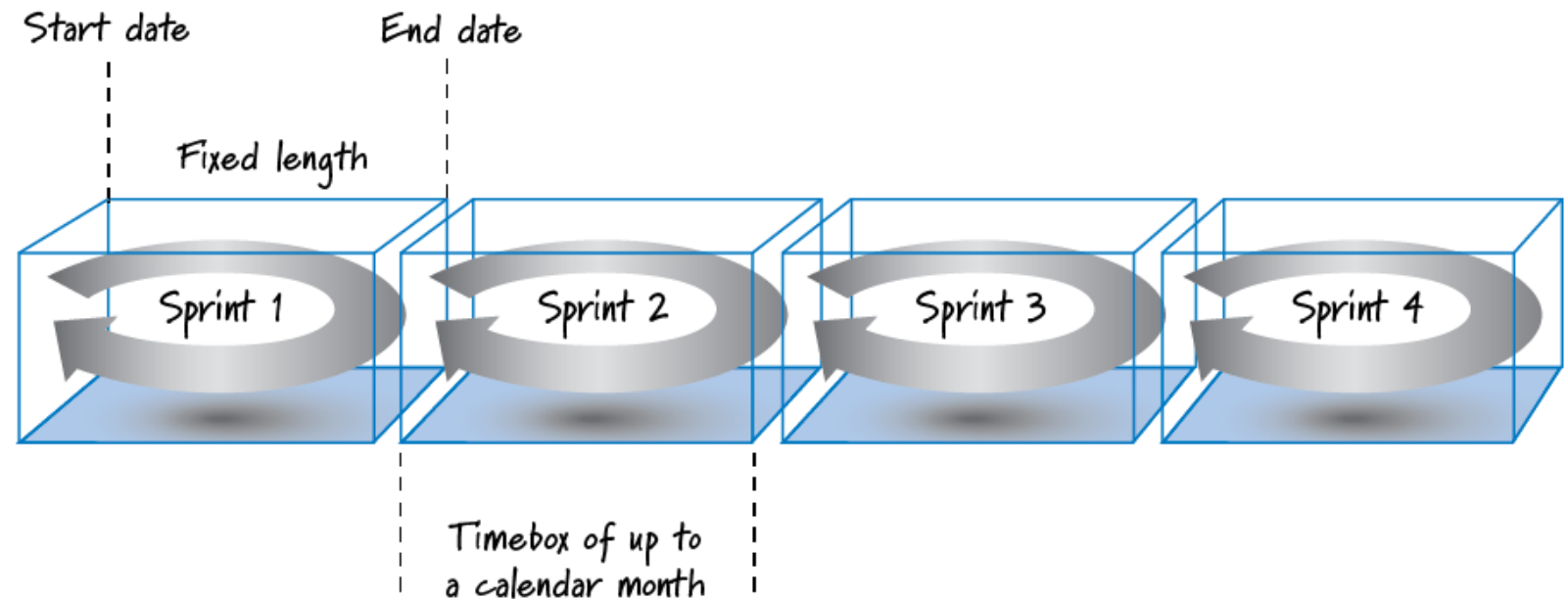
Elementos de SCRUM



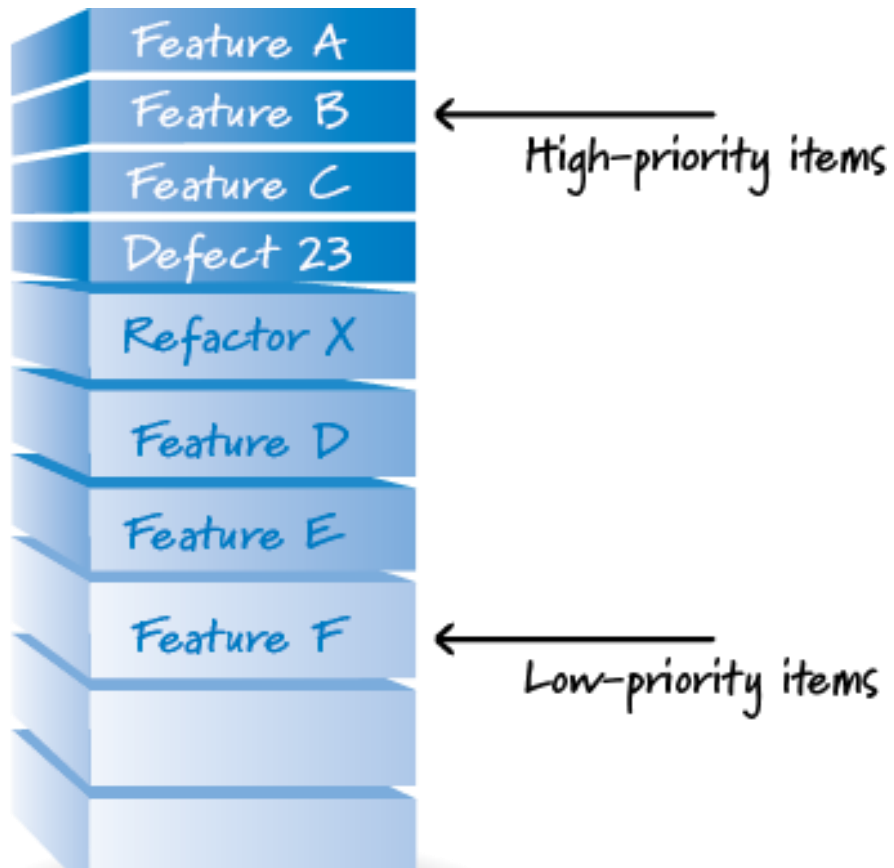
- Sprint
- Product Backlog
- Sprint Backlog
- Uncommitted Backlog
- Product Backlog Item

SPRINT

- Período en el tiempo en el cual se desarrollan una serie de tareas definidas y priorizadas en el backlog.
- Todo el equipo trabaja para conseguir el objetivo común.
- Durante el Sprint no se pueden agregar/eliminar tareas, en cambio se pueden dejar tareas sin realizar dentro de un sprint, que quedaran pendientes en el backlog.

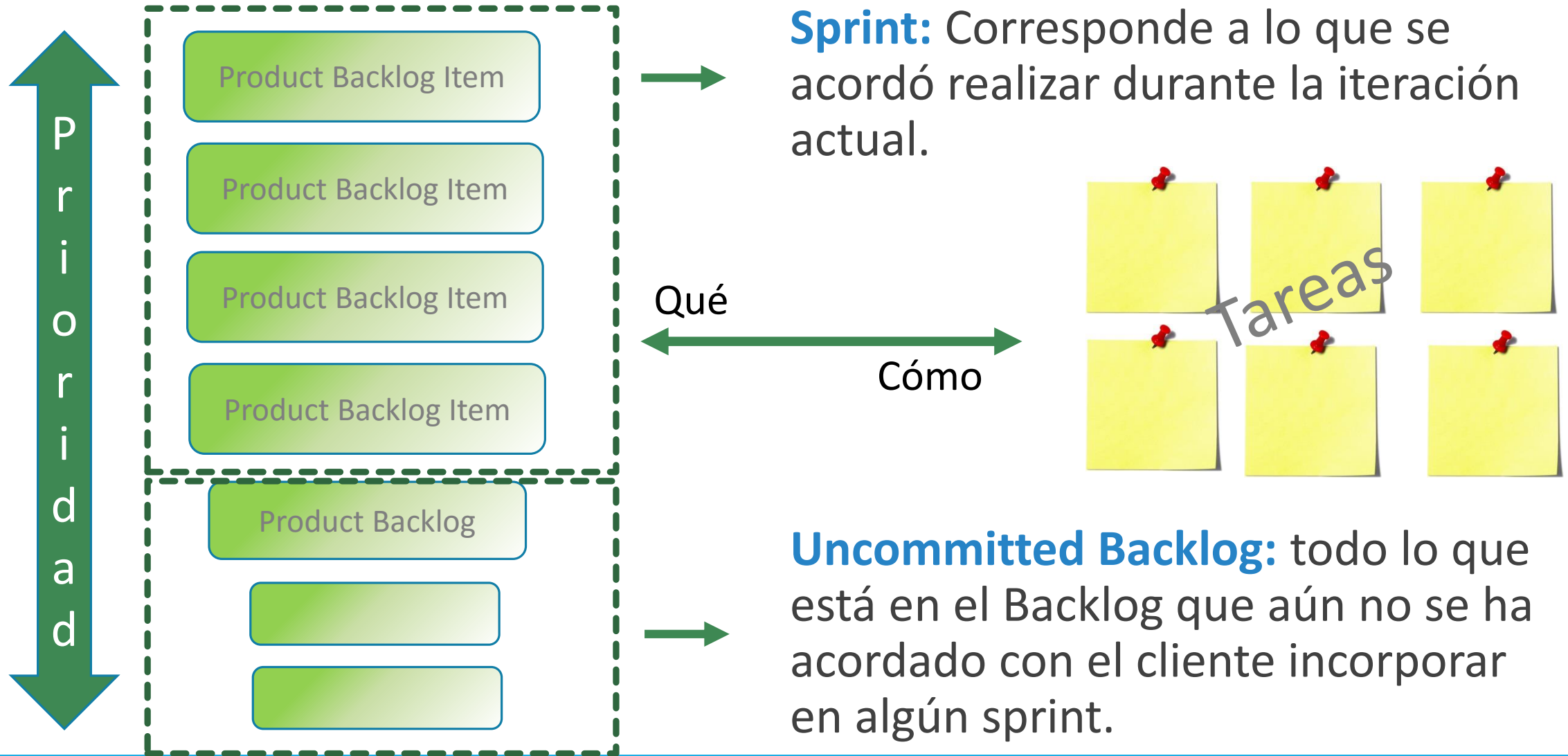


Product Backlog



- El product backlog es una lista de todo lo que debe hacerse en el proyecto → Product Backlog Item.
- La principal característica del product backlog es que está priorizado.
- La prioridad no es para siempre, se revisa y ajusta constantemente con el cliente.
- Hay un único ítem en el tope de la pila
- Los ítems de arriba tienen mayor granularidad.

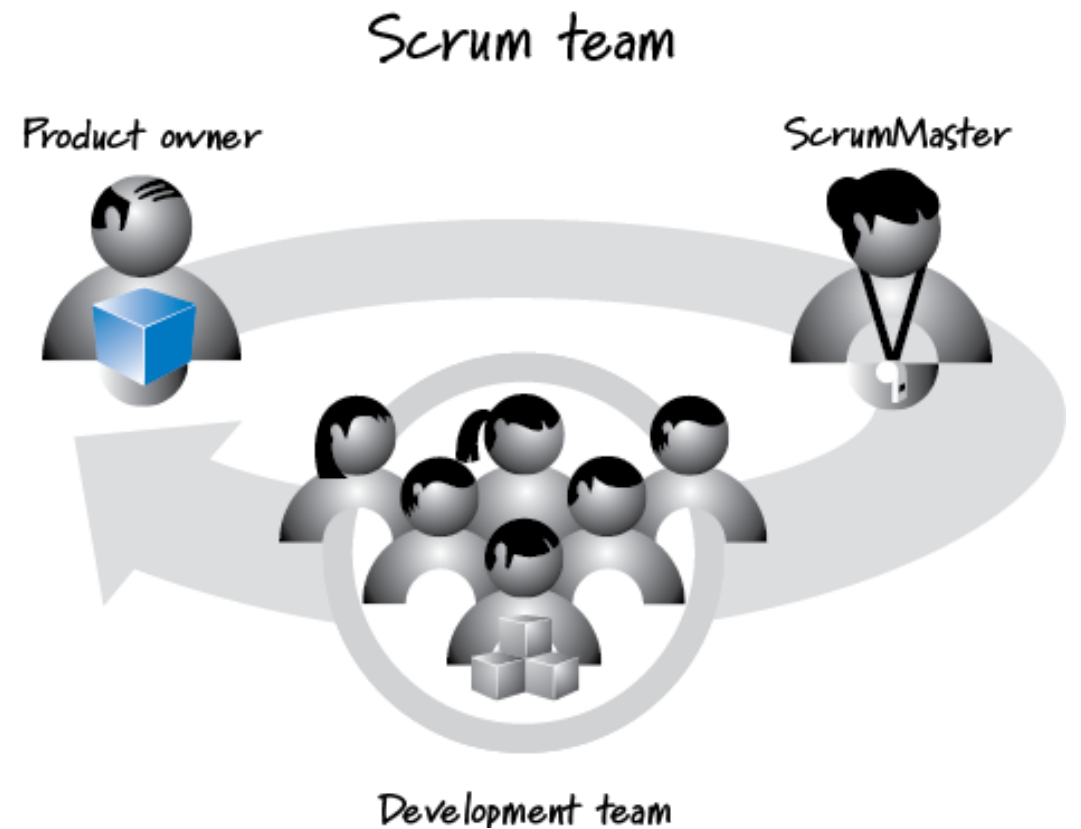
Sprint Backlog – Uncommitted Backlog



Roles en SCRUM



- Product Owner
- Scrum Master
- Equipo de Desarrollo



Product Owner

Responsable de:

- Definir las características del producto y decidir fechas de release y contenidos.
- Consolidar input de los usuarios, stakeholders y otras partes interesadas para obtener una única visión de los requerimientos priorizados del sistema.
- La rentabilidad del producto (ROI)
- La priorización de los requerimientos acordes al valor al negocio.
- Administrar los cambios y prioridades, sprint a sprint.
- Aceptar o rechazar los entregables generados.

Scrum Master

Responsable de:

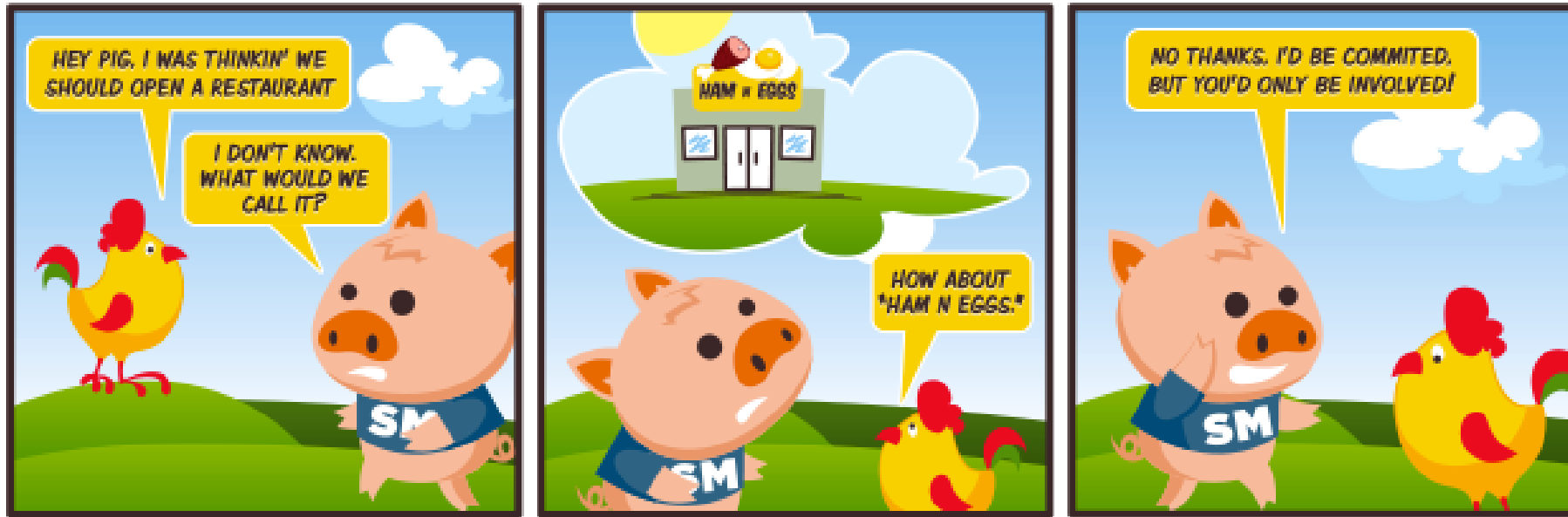
- Mejorar la vida y productividad del equipo facilitando la creatividad y empowerment (equipo autoorganizados)
- Fomentar estrecha cooperación entre todos los roles eliminando cualquier tipo de barreras.
- Proteger al equipo de interferencias externas y eliminar “impedimentos”
- Asegurar la adherencia al proceso.
- Colaborar en la resolución de impedimentos
- Mejorar las herramientas y prácticas de ingeniería de manera de favorecer el desarrollo de incrementos de funcionalidad potencialmente “entregable”.

Development Team

Responsable de:

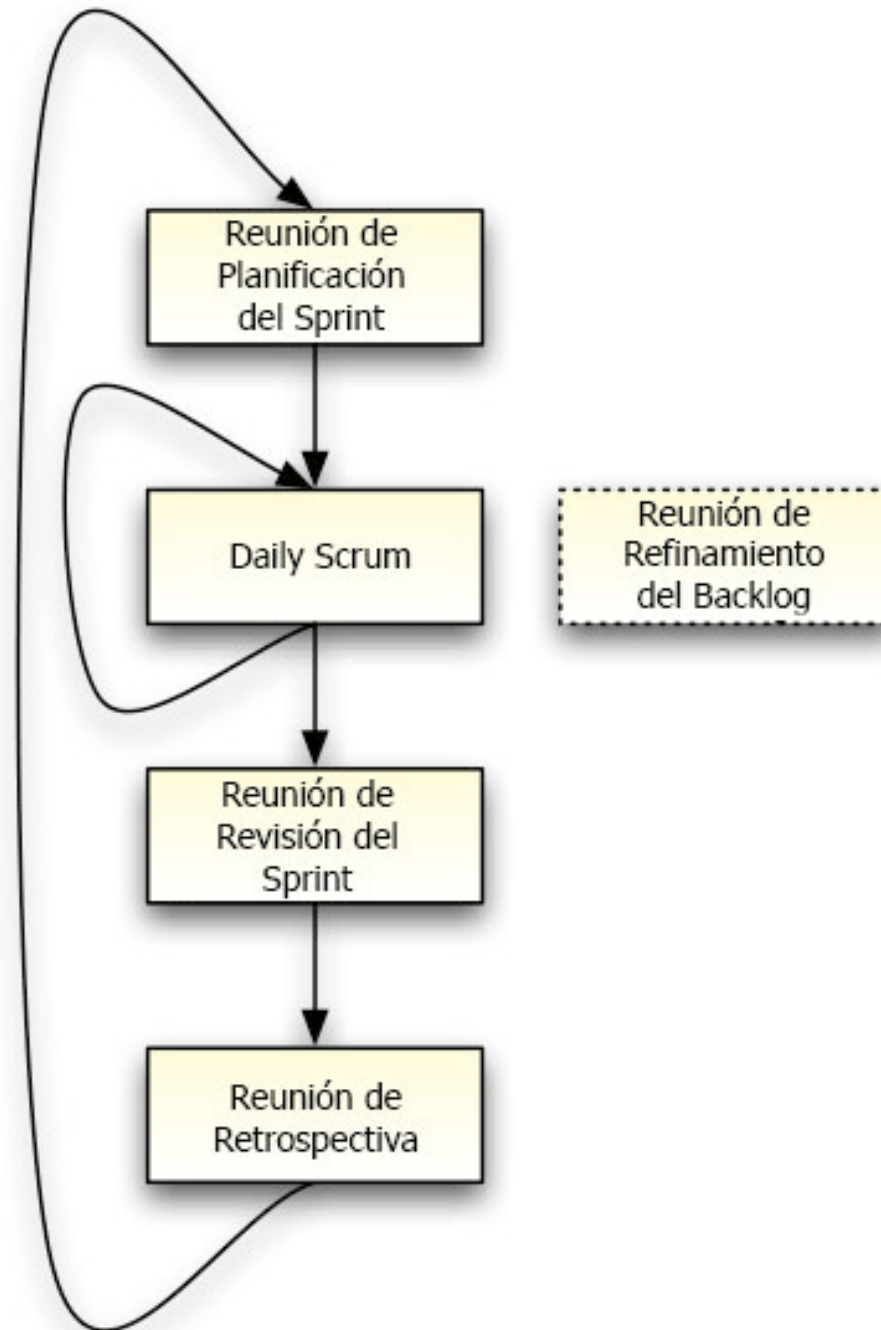
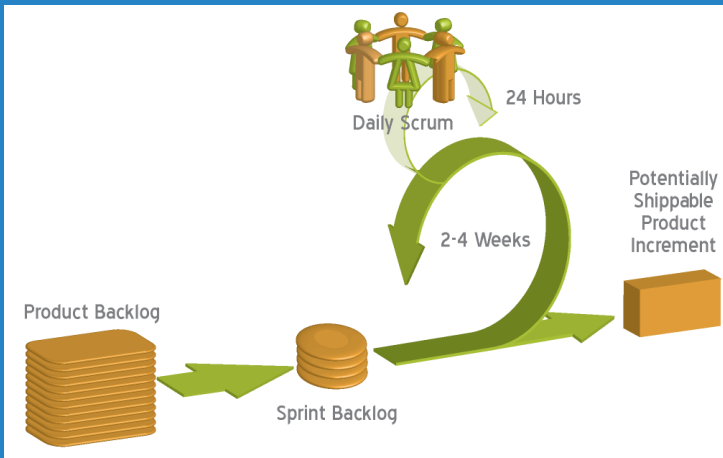
- Convertir los requerimientos en “software funcionando”.
- Acordar con el Product Owner el alcance de cada iteración. Primero acordar luego hacer...
- Identificar el objetivo de cada iteración y especificar los resultados esperados que validará el Product Owner.
- Organizar su propio trabajo.
- Mostrar los resultados de lo realizado durante la iteración al Product Owner y Stakeholders
- Entregar software de alta calidad.

Roles en Scrum



- Product Owner, Scrum Master y Team Members son los **“cerdos”**
 - **Están comprometidos** a construir software de manera regular y frecuente
 - **Ponen su propio “pellejo”** para sacar el proyecto adelante.
- Otros stakeholders son **“gallinas”**, deben participar y aportar al proyecto pero no afectar el trabajo de los “cerdos”

El proceso en SCRUM - Reuniones



Reunión de Planificación

- El Product Owner y el equipo negocian qué ítems del Backlog del Producto intentarán convertir en producto funcionando durante el Sprint.
- El Product Owner es el responsable de declarar cuáles son los ítems más importantes para el negocio.
- El equipo es responsable de seleccionar la cantidad de trabajo que cree que podrán realizar sin acumular deuda técnica. El equipo “toma” el trabajo desde el Product Backlog hacia el Sprint Backlog.

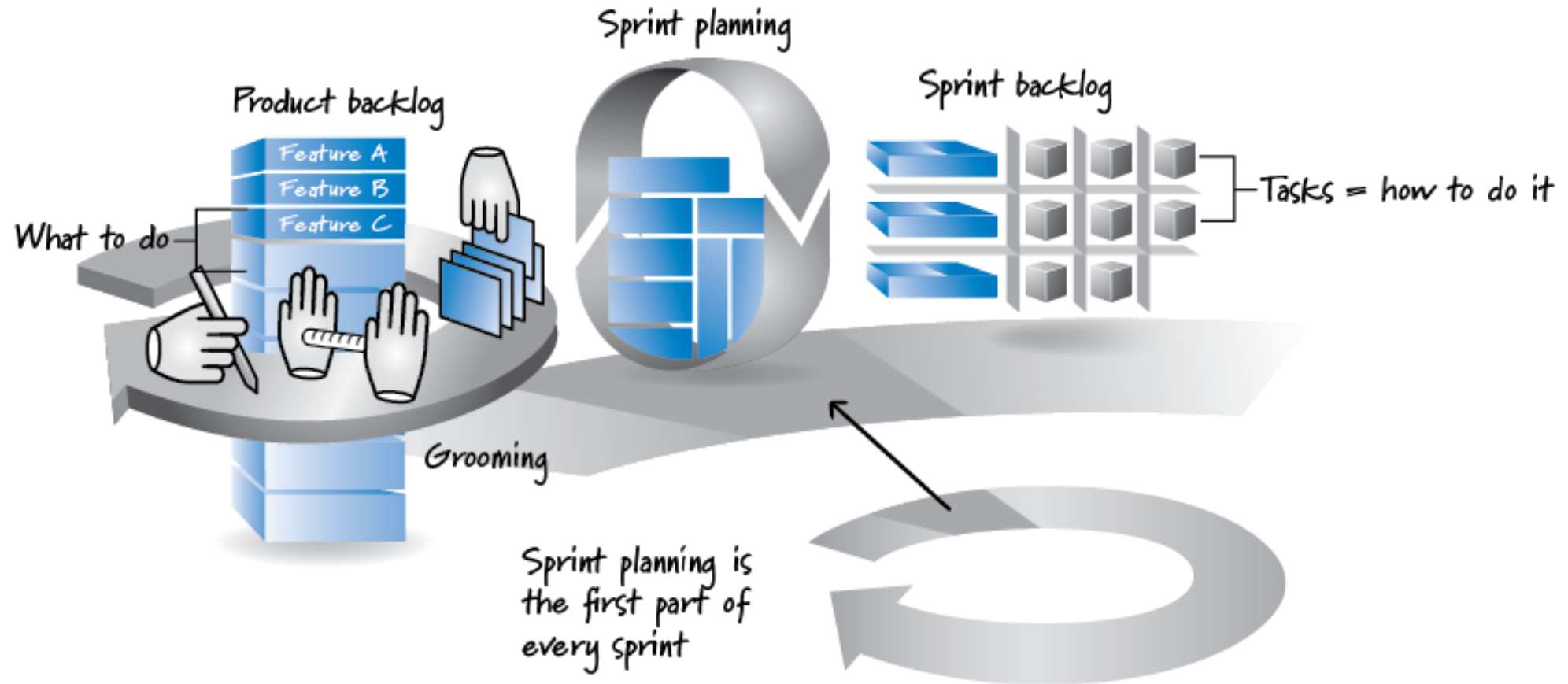
Aspectos claves:

Repriorizar el Backlog

Determinar el trabajo que puede ser completado en el siguiente sprint

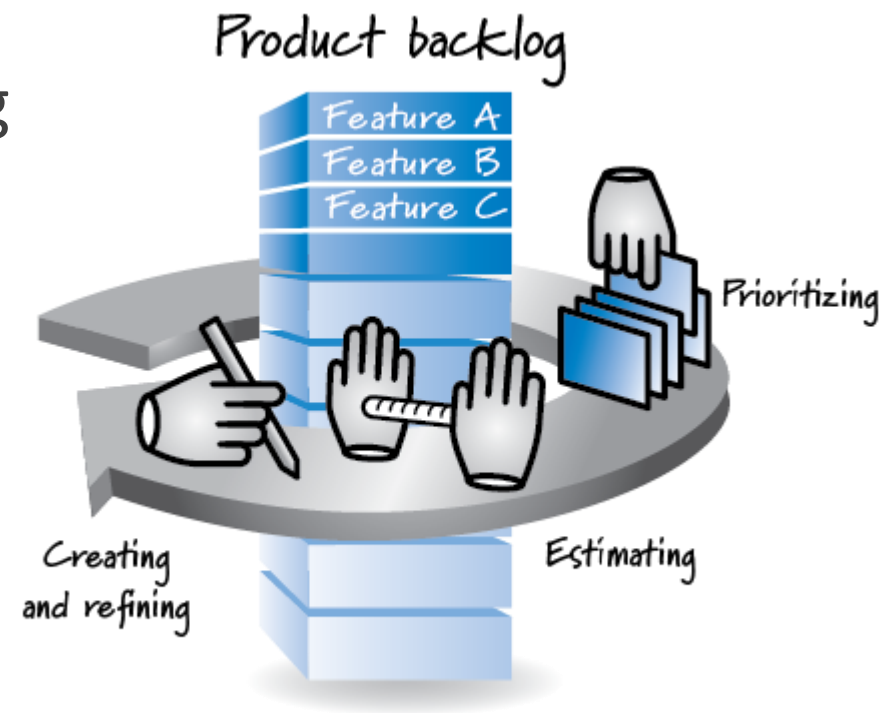
Desglosar los ítems del Backlog en tareas a ser hechas en el sprint

Reunión de Planificación



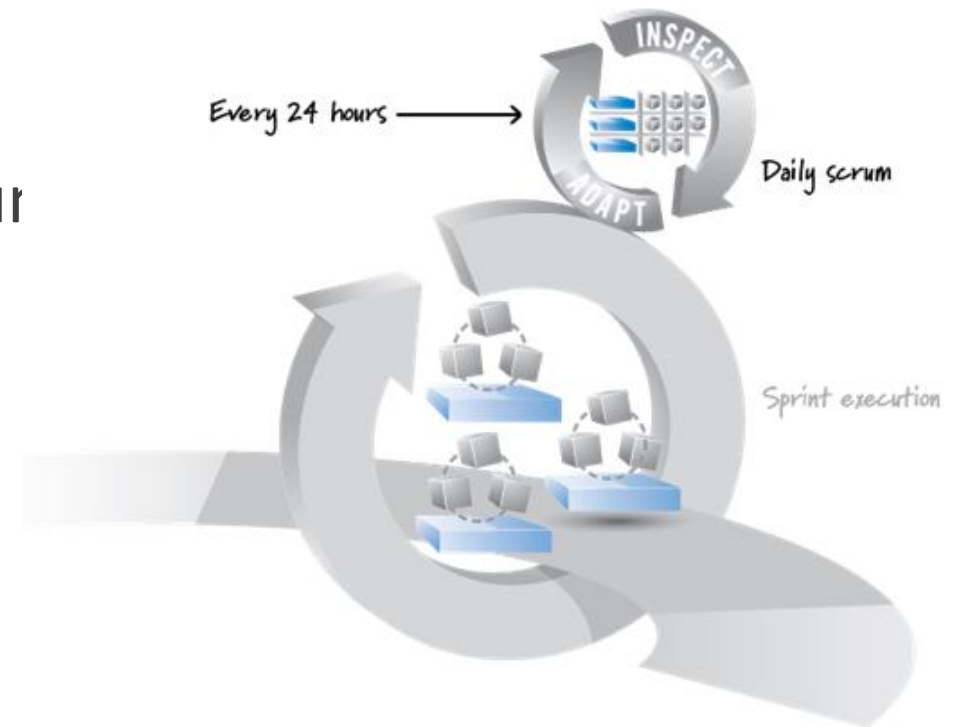
Reunión de Refinamiento

- La mayoría de los PBIs inicialmente deben refinarse, ya que son grandes y poco comprendidos.
- Se prepara el Backlog del Producto para la próxima Reunión de Planificación.
- El equipo estima la cantidad de esfuerzo que se debe invertir para completar los Ítems del Backlog del Producto y proporciona información técnica para ayudar al Product Owner a priorizarlos .
- Los grandes ítems se dividen y se clarifican, teniendo en cuenta temas tanto de negocio como técnicos
- Se repriorizan los ítems del backlog.
- También conocida como Grooming.



Daily Scrum

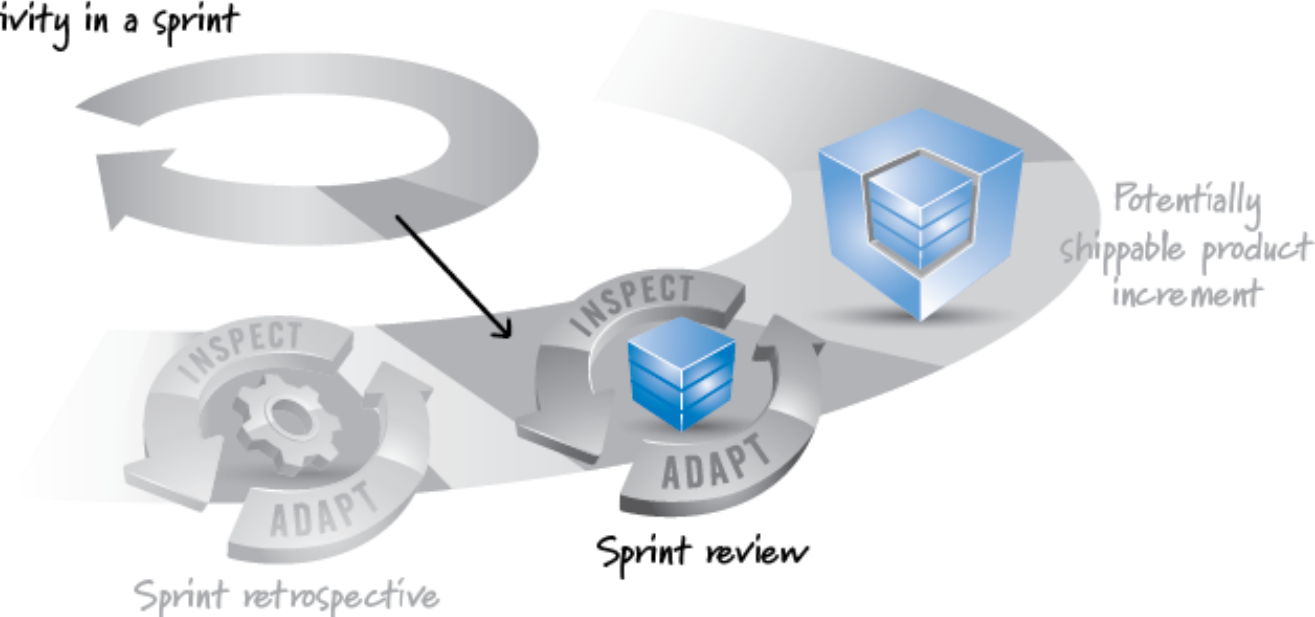
- Todo el equipo participa y debe ser puntual (comienza en hora)
- Tres preguntas: **Qué se hizo, qué harás, qué problemas?**
- No es una reunión de reporte al Scrum Master, el equipo conduce el proyecto.
- Solo una persona habla a la vez.
- Si se detectan necesidades de soporte de un Team Member a otro, se acuerda trabajo conjunto para después de la reunión.
- Tiene un timebox de quince minutos
- El PO indica los ítems que considera finalizados o no. Concepto de DONE.



Revisión del Sprint

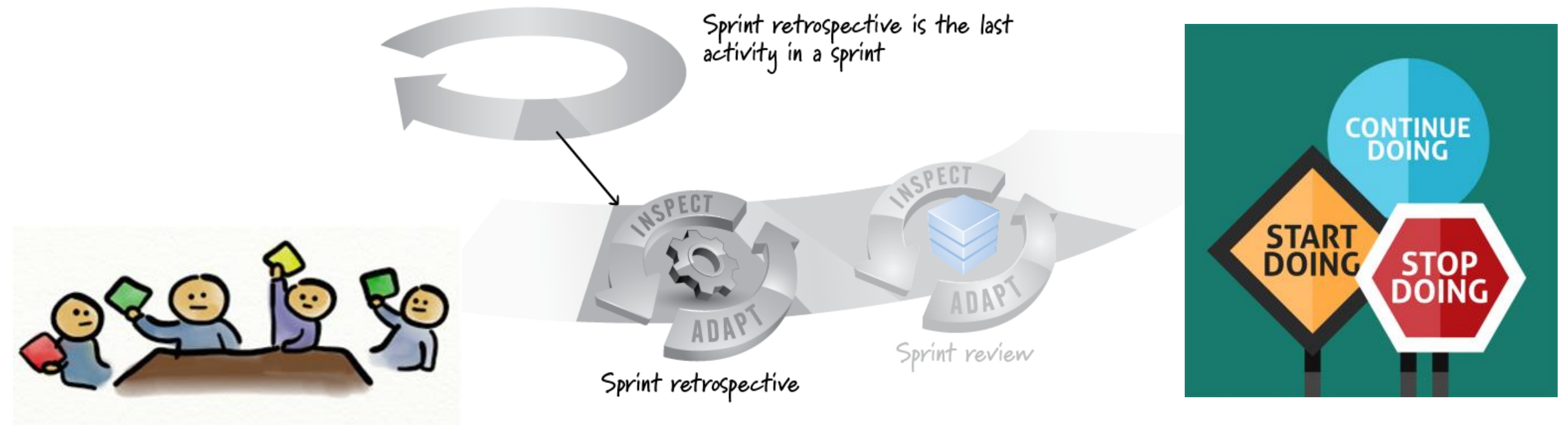
- Product Owner, usuarios, stakeholders, etc. validan los entregables generados.
- El equipo presenta los objetivos del sprint, el alcance acordado y muestra los resultados del trabajo realizado.
- Se muestra software funcionando.
- Los cambios que surjan se incorporan al backlog.
- Se recolecta feedback de cada asistente finalizada la demo.
- Se discute la re priorización de los ítems del backlog según el feedback obtenido.

Sprint review is the next-to-last activity in a sprint

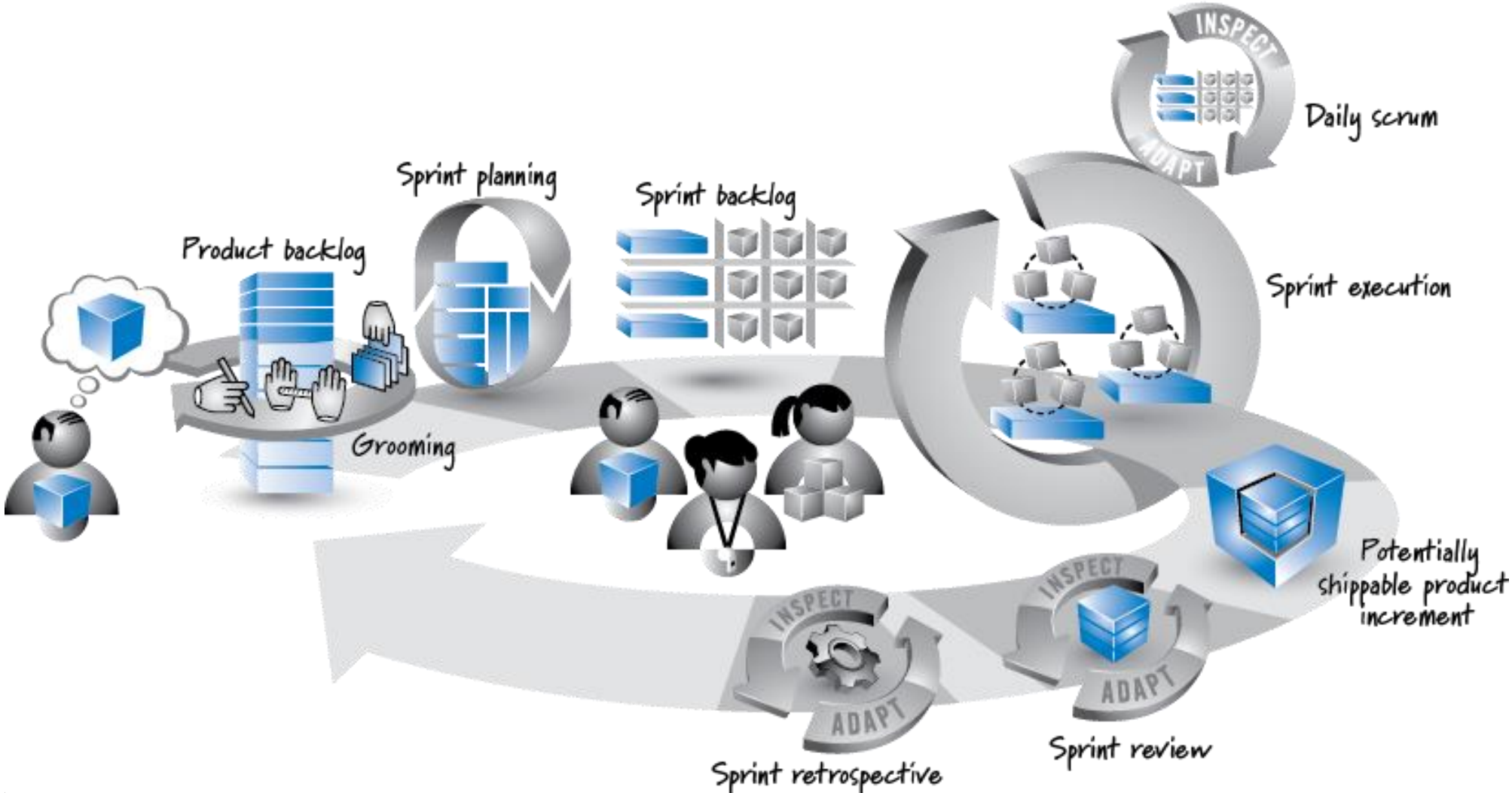


Reunión de Retrospectiva

- Qué se hizo bien, qué se podría haber hecho mejor, cómo se puede mejorar en el siguiente sprint.
- Deben contemplarse procesos, prácticas, mecanismos de comunicación, entorno, artefactos y herramientas, entre otros



Dinámica del proyecto ágil



XP

Extreme Programming

La programación extrema **es una metodología ágil de ingeniería para el desarrollo de Software**. Fue creada por Kent Beck en respuesta a los problemas de dominio cuyos requerimientos son muy cambiantes.

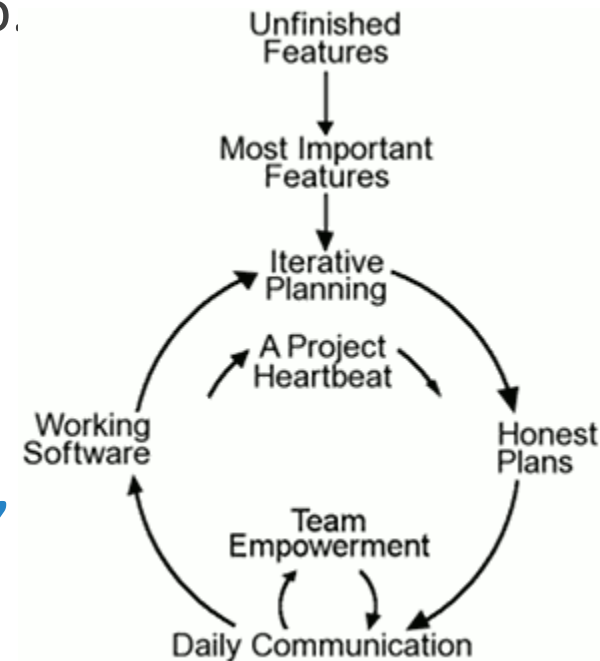
«Todo en el software cambia. Los requisitos cambian. El diseño cambia. El negocio cambia. La tecnología cambia. El equipo cambia. Los miembros del equipo cambian. El problema no es el cambio en sí mismo, puesto que sabemos que el cambio va a suceder; el problema es la incapacidad de adaptarnos a dicho cambio cuando éste tiene lugar.»

Kent Beck.



XP

- Hace hincapié en el trabajo en equipo. Gerente, clientes y desarrolladores son todos socios iguales en un equipo colaborativo.
- Implementa un entorno sencillo, pero eficaz, permitiendo a los equipos ser altamente productivos.
- El equipo se auto-organiza alrededor del problema para resolverlo lo más eficientemente posible.
- Se entrega software de alta calidad en forma rápida y continua.
- **Promueve alto compromiso del cliente, ciclos cortos de feedback, testing continuo, planificación continua y equipo único para entregar sw funcionando en intervalos frecuentes (1-3 semanas)**

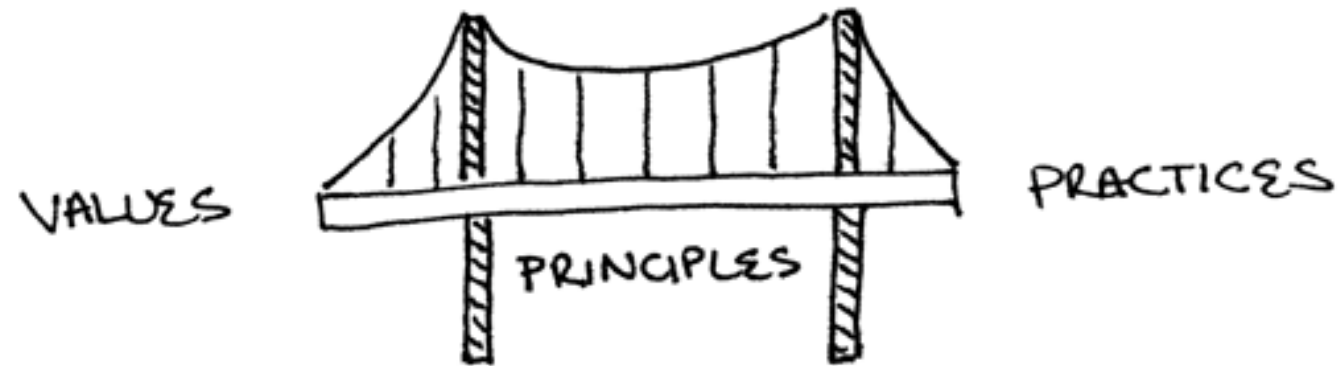


XP es una metodología de desarrollo de software ágil, adaptativa, con un conjunto bien definido de valores y prácticas

XP - Elementos

El Framework XP consiste de tres elementos principales:

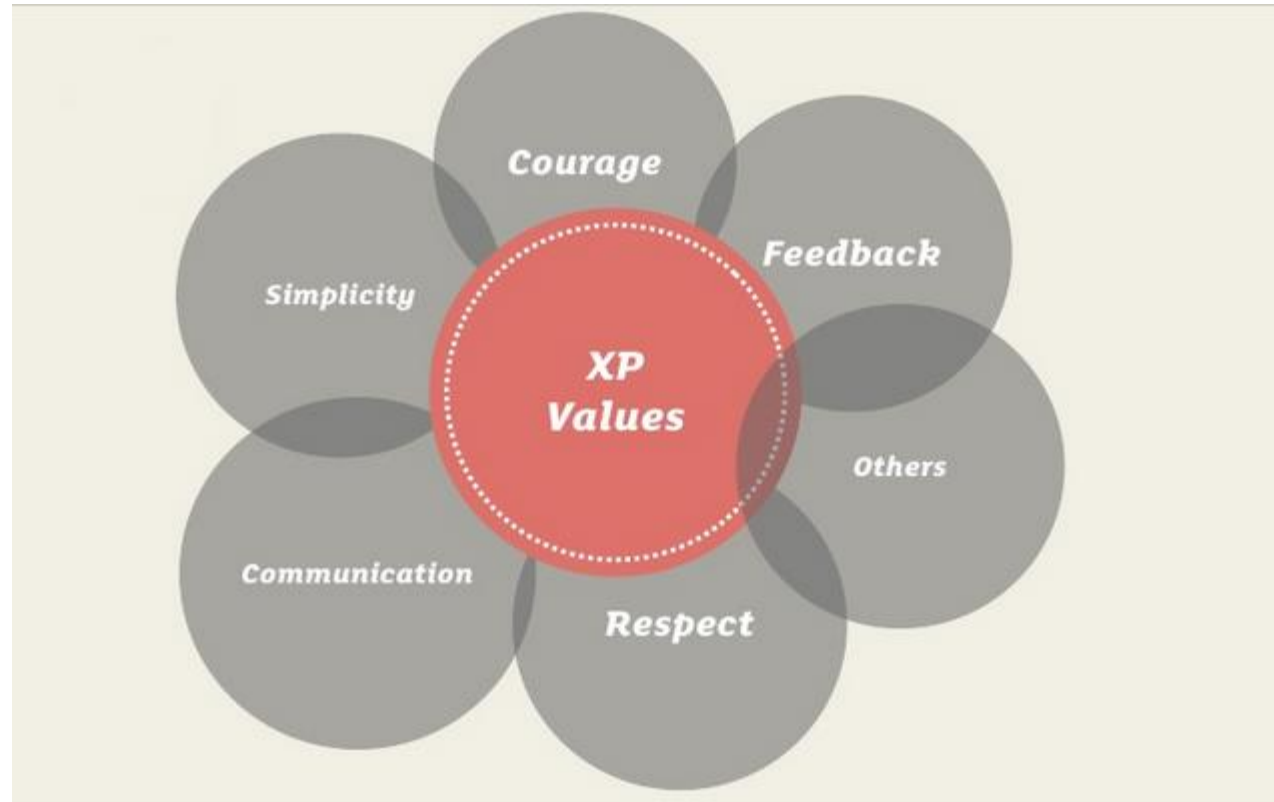
- **Valores**
- **Principios**
- **Prácticas**



XP - Valores

Extreme Programming mejora un proyecto de software de 5 formas:

- **Comunicación**
- **Simplicidad**
- **Feedback**
- **Corage**
- **Respeto**



XP - Principios

Los **principios** son el puente entre los **valores** que son sintéticos y abstractos, y las **prácticas** que indican cómo desarrollar el software.

Valores

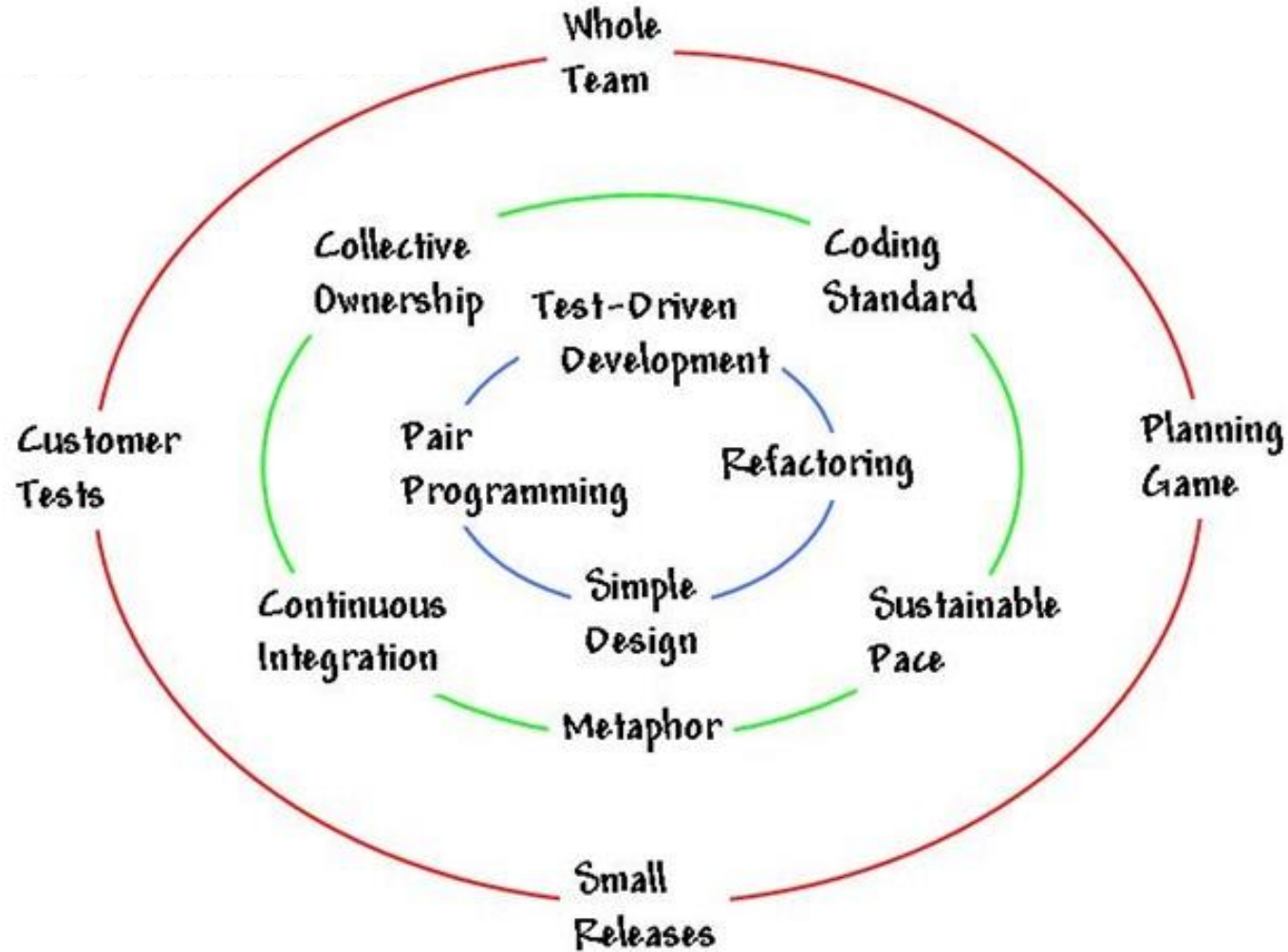
- Comunicación
- Simplicidad
- Feedback
- Coraje
- Respeto

Principios

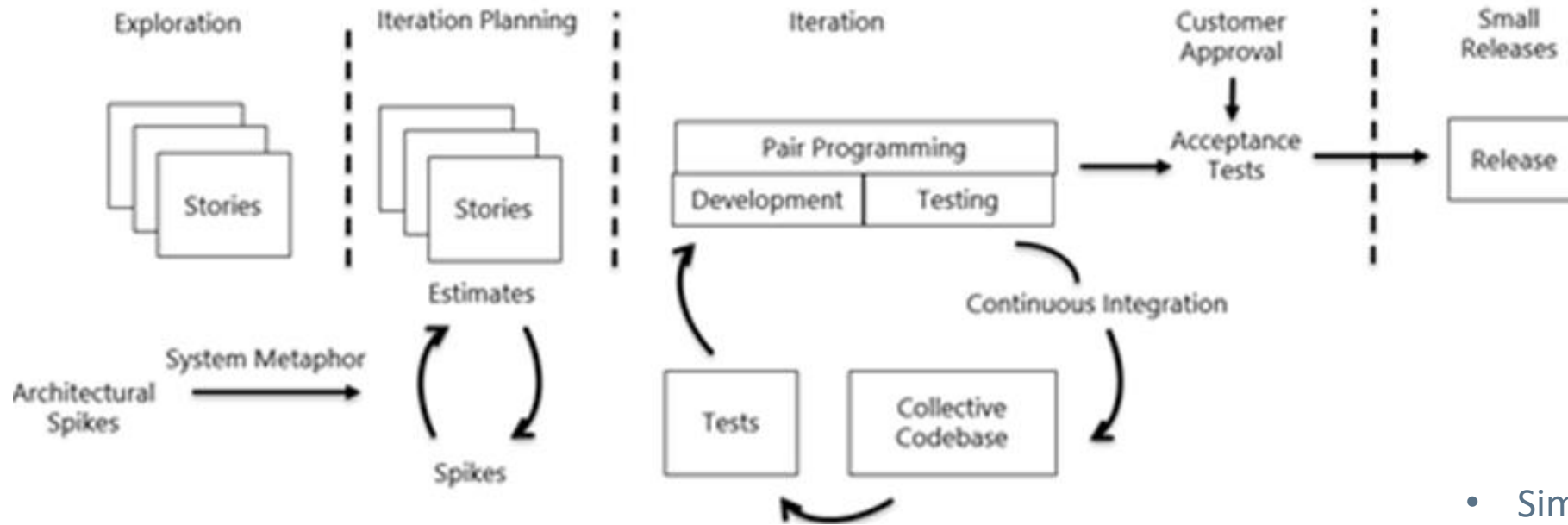
- Feedback Rápido
- Asumir simplicidad
- Cambios incrementales
- Adoptar el cambio
- Trabajar con calidad
- Enseñar a aprender
- Inversión inicial pequeña
- Jugar para ganar
- Experimentar en forma concreta
- Comunicación abierta y honesta
- Trabajar con los instintos de las personas – No en su contra
- Aceptar la responsabilidad
- Adaptación local
- Viajar liviano
- Medición honesta

Prácticas

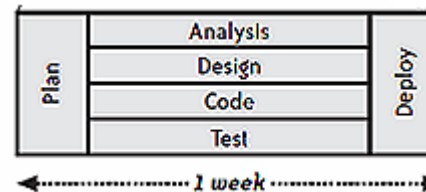
XP - Practicas



XP - Proceso



Iteration



- Simple Design
- Test-Driven Development
- Pair Programming
- Design Improvement
- Continuous Integration
- Re-factoring
- Collective Code Ownership

SCRUM Y XP

Las personas que utilizan **Scrum** en ambientes de desarrollo de software generalmente adoptan una o más prácticas **XP** para aplicar las prácticas de ingeniería en la organización de desarrollo.

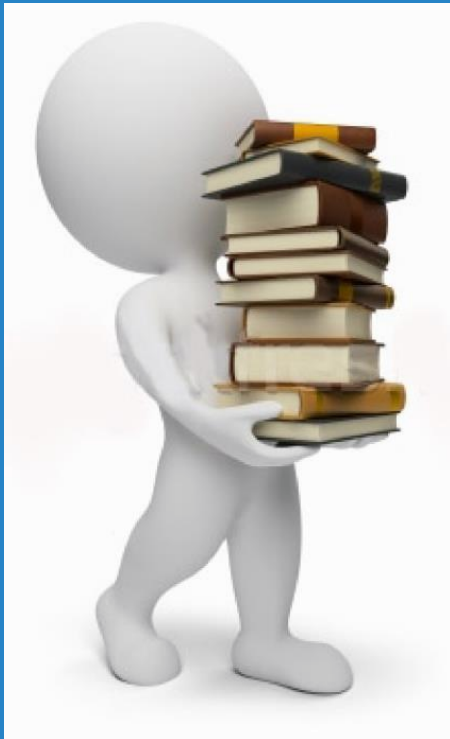


¿Agilidad o
disciplina?

Para lograr **AGILIDAD** necesitamos
DISCIPLINA



Bibliografía



- 📖 Mike Conn, “User Stories Applied for Agile Software Development”. Addison-Wesley (Signature Series)
- 📖 Ken Schwaber, Agile Project Management with Scrum. Microsoft Press.
- 📖 www.agilemanifesto.org
- 📖 www.agilealliance.org
- 📖 www.scrum.org
- 📖 www.extremeprogramming.org
- 📖 <http://agileatlas.org/articles/item/scrum-framework>
- 📖 <http://scrummethodology.com/>

Dudas y consultas

MARIELA CASTARÉS

VIRGINIA CUOMO

