



UD 1. DESARROLLO DE SOFTWARE

METODOLOGÍAS ÁGILES

ceedcv
CENTRE ESPECÍFIC
D'EDUCACIÓ A DISTÀNCIA DE
LA COMUNITAT VALENCIANA

Entornos de desarrollo (ED)

Sergio Badal
Raúl Palao

UD 1. DESARROLLO DE SOFTWARE

METODOLOGÍAS ÁGILES

1 METODOLOGÍAS ÁGILES

1.1 INGENIERÍA DE SOFTWARE

1.2 MANIFIESTO ÁGIL

1.3 EXTREME PROGRAMMING

1.4 SCRUM

1.5 KANBAN

1.6 TDD



UD 1. DESARROLLO DE SOFTWARE

METODOLOGÍAS ÁGILES

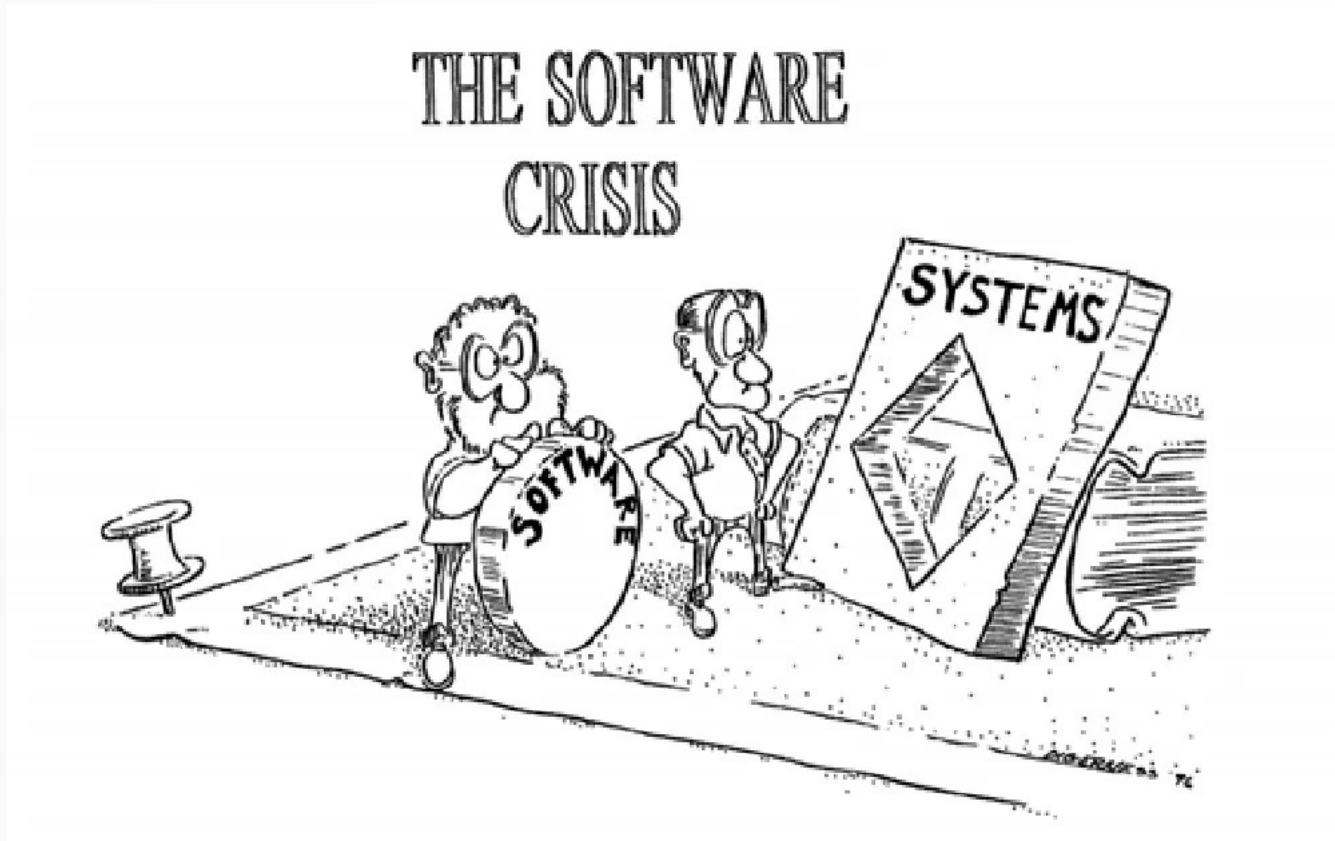


Años 50:

- **La importancia del hardware sobre el software era más que evidente.**
- **Consecuencia:**
 - **Entregas fuera de plazo, sobre-costes excesivos.**
 - **CRISIS DEL SOFTWARE**

UD 1. DESARROLLO DE SOFTWARE

METODOLOGÍAS ÁGILES



Solución:

- **Un conjunto de expertos ingenieros y matemáticos presentan numerosas iniciativas, técnicas y metodologías que confluyen en lo que se acabará denominando:**
 - **INGENIERÍA DE/DEL SOFTWARE**

UD 1. DESARROLLO DE SOFTWARE

METODOLOGÍAS ÁGILES

1 METODOLOGÍAS ÁGILES

1.1 INGENIERÍA DE SOFTWARE

1.2 MANIFIESTO ÁGIL

1.3 EXTREME PROGRAMMING

1.4 SCRUM

1.5 KANBAN

1.6 TDD



UD 1. DESARROLLO DE SOFTWARE

METODOLOGÍAS ÁGILES

¿Cuándo surge la agilidad?

- Se considera el inicio “oficial” de las metodologías ágiles la publicación del **Manifiesto Ágil**
- Es un documento redactado en 2001 por 17 expertos en programación que supuso un cambio radical en la forma de desarrollar ‘software’.
- Estos gurús propusieron 12 puntos que inspiran las diferentes metodologías ágiles que han surgido desde entonces.
- Aunque nació en el mundo del ‘software’, la filosofía que promueve este manifiesto es extensible al desarrollo de cualquier otro producto.

UD 1. DESARROLLO DE SOFTWARE

METODOLOGÍAS ÁGILES

Manifiesto Ágil

- 1) Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de 'software' con valor.
- 2) Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.
- 3) Entregamos 'software' funcional frecuentemente, entre dos semanas y dos meses, preferentemente en el periodo de tiempo más corto posible.
- 4) Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.
- 5) 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.

UD 1. DESARROLLO DE SOFTWARE

METODOLOGÍAS ÁGILES

Manifiesto Ágil

- 6) El método más eficiente y efectivo de comunicar información al equipo de desarrollo, y entre los miembros del equipo, es la conversación cara a cara.
- 7) El 'software' funcionando es la medida principal de progreso.
- 8) Los procesos ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.
- 9) La atención continua a la excelencia técnica y al buen diseño mejora la agilidad.
- 10) La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
- 11) Las mejores arquitecturas, requisitos y diseños emergen de equipos autoorganizados.
- 12) A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.

UD 1. DESARROLLO DE SOFTWARE

METODOLOGÍAS ÁGILES

Manifiesto Ágil (resumen)

- 1) El factor humano es el principal factor de éxito de un proyecto (**peopleware**)
- 2) Documentar solo si aporta valor o sirve para tomar decisiones
- 3) Interacción constante con el cliente
- 4) Requisitos cambiantes son aceptables
- 5) Entregas continuas, tempranas y funcionales
- 6) Cara a cara es la mejor manera de comunicarse
- 7) Simplicidad en todas las decisiones

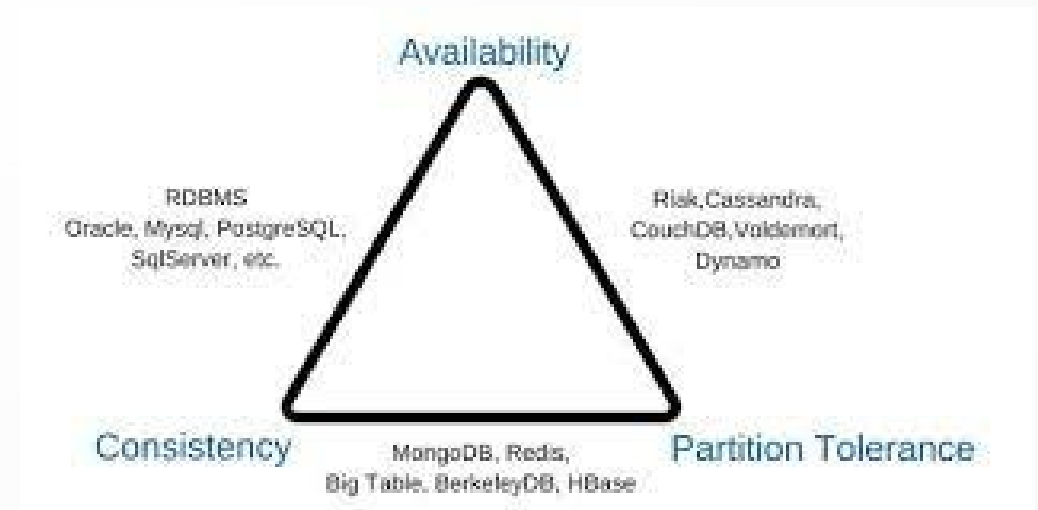
UD 1. DESARROLLO DE SOFTWARE

METODOLOGÍAS ÁGILES

Triángulo de hierro

- Los triángulos son muy comunes en ISW y te encontrarás con muchos si sigues tus estudios en informática.
- Los dos más conocidos son:
 - El **triángulo CAP de bases de datos** o Conjetura de Brewer.
 - El **triángulo de hierro del coste, tiempo y alcance**.

<https://proyectosagiles.org/triangulo-hierro/>



UD 1. DESARROLLO DE SOFTWARE

METODOLOGÍAS ÁGILES

Triángulo de hierro

- Imagina que, al empezar este ciclo, te propones tres metas:
 - 1) Superar todos los módulos de primer curso de DAW
 - 2) Trabajar a jornada completa
 - 3) Viajar todos los fines de semana
- ¿Qué te parece ese reto? ¿Lo ves factible?
- ¿Y si ponemos las tres cosas en un triángulo y descartamos una? **¿Sería factible?**

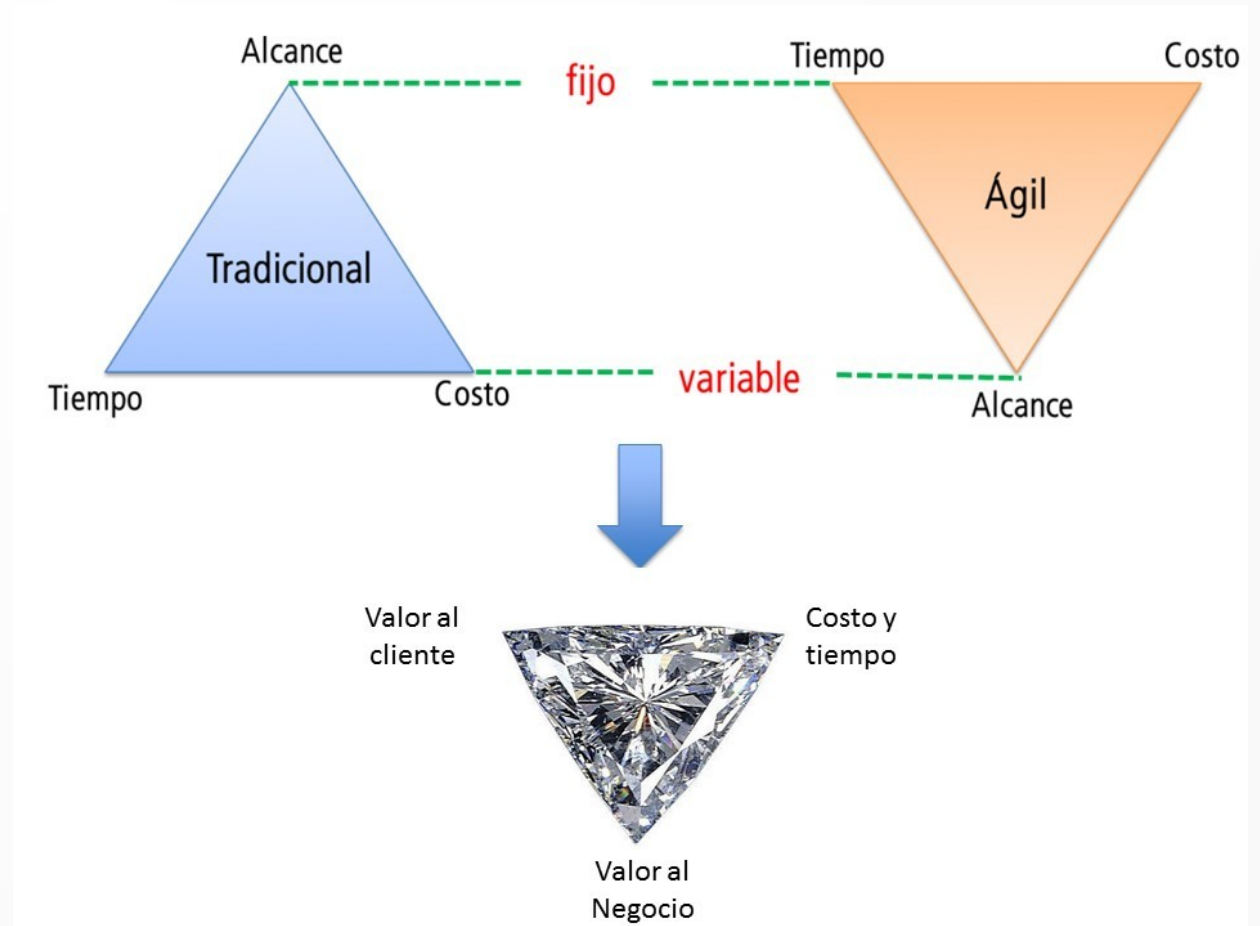


UD 1. DESARROLLO DE SOFTWARE

METODOLOGÍAS ÁGILES

Triángulo de hierro

- Básicamente, **las metodologías tradicionales se empeñan en conseguir los tres elementos del triángulo** y las ágiles son realistas y dicen que **solo pueden asegurar dos de ellas, concretamente el coste y el tiempo, dejando abierto el alcance.**
- Si entiendes este concepto, has comprendido la filosofía de la AGILIDAD.

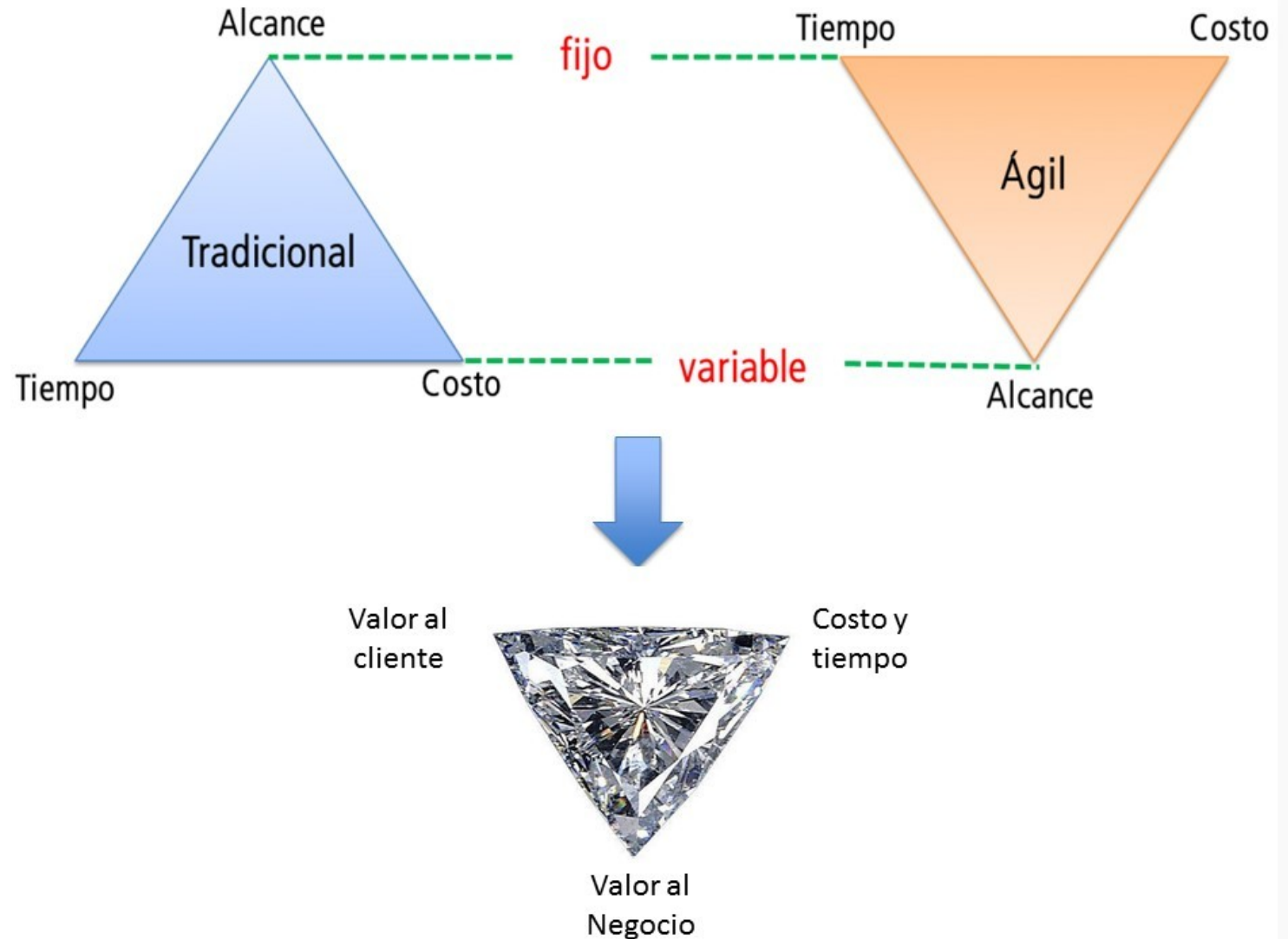


UD 1. DESARROLLO DE SOFTWARE

METODOLOGÍAS ÁGILES

Triángulo de hierro

¡Si entiendes este concepto, has comprendido la filosofía de la AGILIDAD!



UD 1. DESARROLLO DE SOFTWARE

METODOLOGÍAS ÁGILES

1 METODOLOGÍAS ÁGILES

1.1 INGENIERÍA DE SOFTWARE

1.2 MANIFIESTO ÁGIL

1.3 EXTREME PROGRAMMING

1.4 SCRUM

1.5 KANBAN

1.6 TDD



UD 1. DESARROLLO DE SOFTWARE

METODOLOGÍAS ÁGILES

Extreme Programming (XP)

- Feedback
 - Es importante que el cliente tenga numerosas oportunidades para expresar sus críticas.
 - El cliente puede estar in-house (con el equipo de programadores).
- Respeto
 - Rendimiento excelente necesita respeto e implica, también, que un desarrollador no realice modificaciones que tengan un impacto negativo en el trabajo de un compañero.



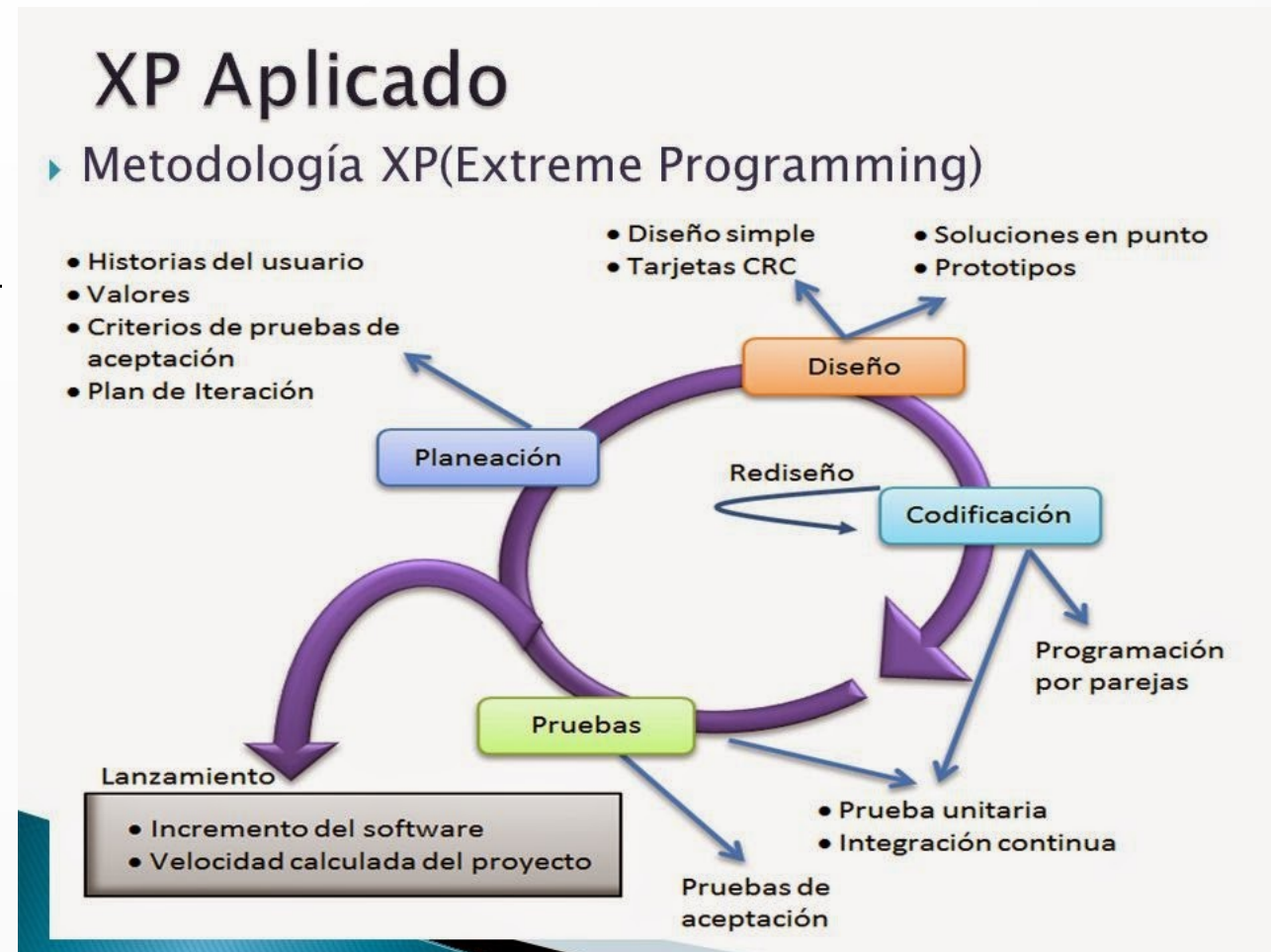
Imagen elaborada por oficinaproyectosinformatica.blogspot.com

UD 1. DESARROLLO DE SOFTWARE

METODOLOGÍAS ÁGILES

Extreme Programming (XP)

- 1) Esta herramienta es muy útil sobre todo para startups o empresas que están en proceso de consolidación.
- 2) Su principal objetivo es ayudar en las relaciones entre los empleados y clientes.
- 3) La clave del éxito del Extreme Programming XP es potenciar las relaciones personales, a través, del trabajo en equipo, fomentando la comunicación y eliminando los tiempos muertos.



UD 1. DESARROLLO DE SOFTWARE

METODOLOGÍAS ÁGILES

Extreme Programming (XP)

- Comunicación
 - Solo si todos los implicados están en contacto de forma permanente se pueden evitar y detectar rápidamente los malos entendidos
- Valentía
 - Si hay errores en el producto, hay que señalarlos. Incluso los propios.
 - En un equipo que trabaja con XP tampoco hace falta disculparse.

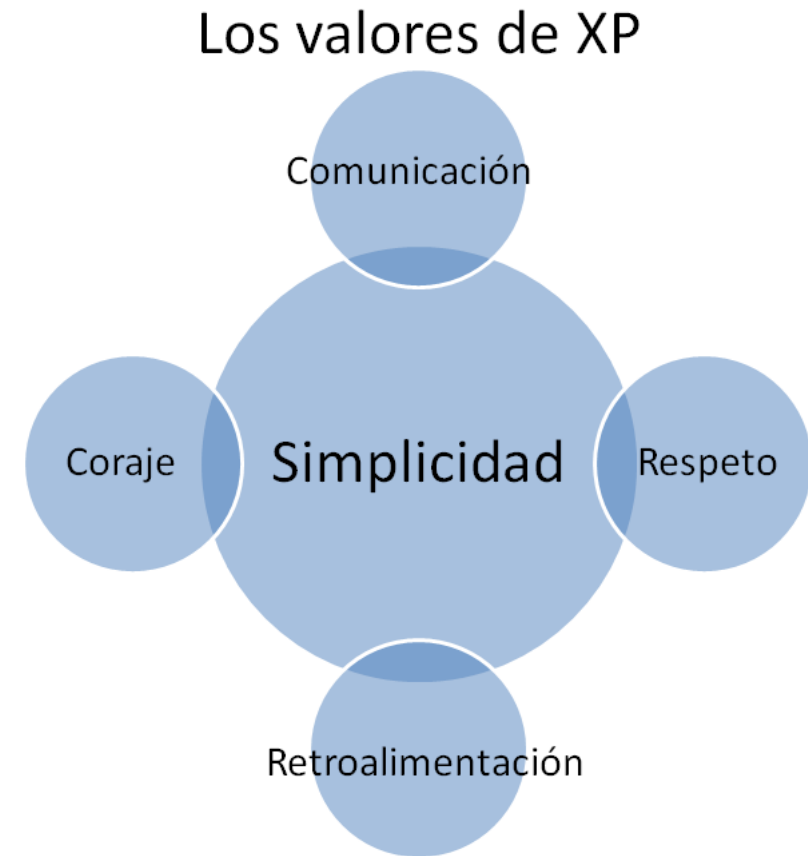


Imagen elaborada por oficinaproyectosinformatica.blogspot.com

UD 1. DESARROLLO DE SOFTWARE

METODOLOGÍAS ÁGILES

1 METODOLOGÍAS ÁGILES

1.1 INGENIERÍA DE SOFTWARE

1.2 MANIFIESTO ÁGIL

1.3 EXTREME PROGRAMMING

1.4 SCRUM

1.5 KANBAN

1.6 TDD

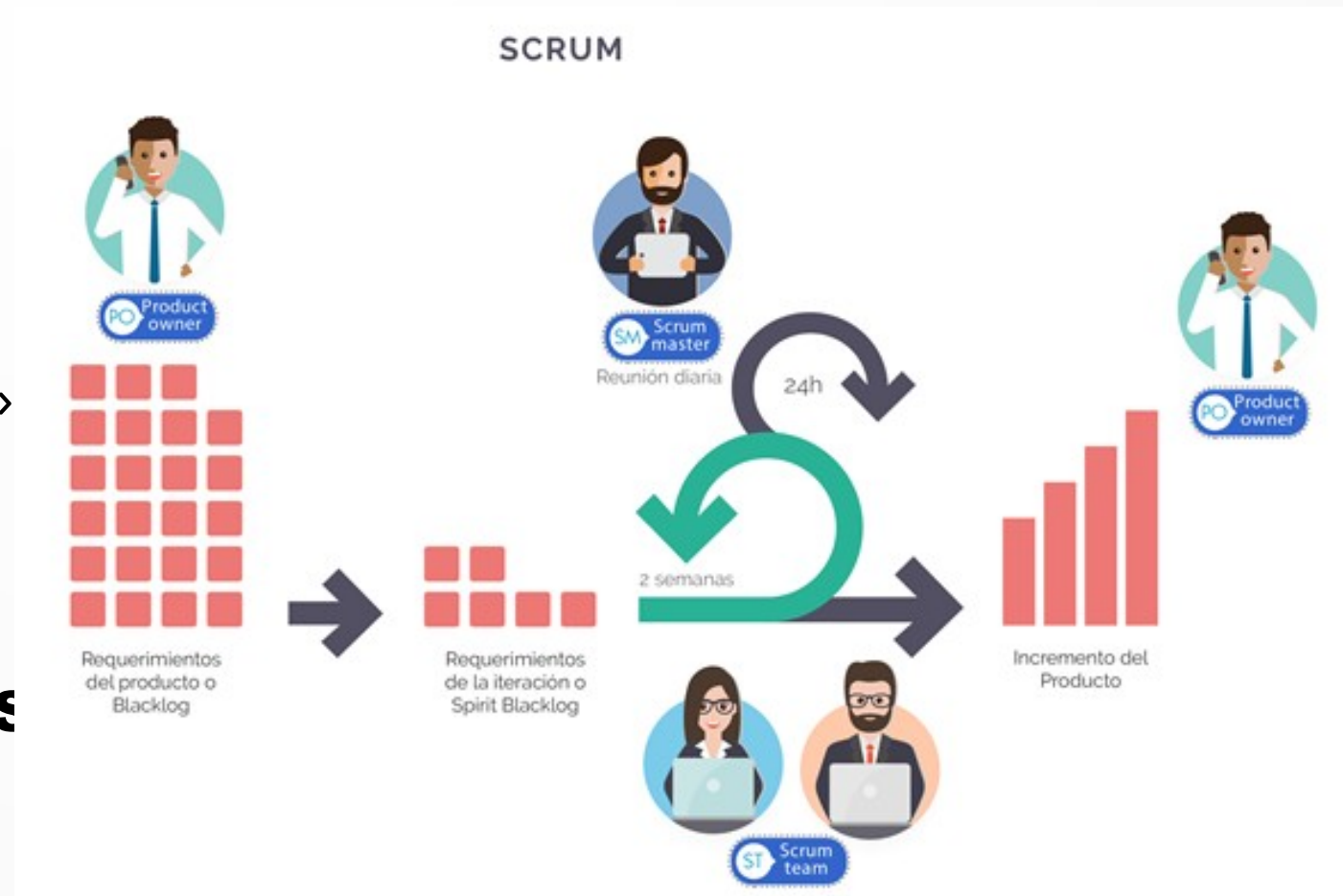


UD 1. DESARROLLO DE SOFTWARE

METODOLOGÍAS ÁGILES

SCRUM

- Se basa en una estructura de desarrollo incremental (iterativo con entregas funcionales).
- Cualquier ciclo de desarrollo se desgrana en «pequeños proyectos» divididos en distintas etapas:
 - Análisis, desarrollo y testing.
- En desarrollo encontramos iteraciones del proceso o **SPRINTS** es decir, **entregas regulares y parciales del producto final.**



<https://www.iebschool.com/blog/que-son-metodologias-agiles-agile-scrum/>

UD 1. DESARROLLO DE SOFTWARE

METODOLOGÍAS ÁGILES

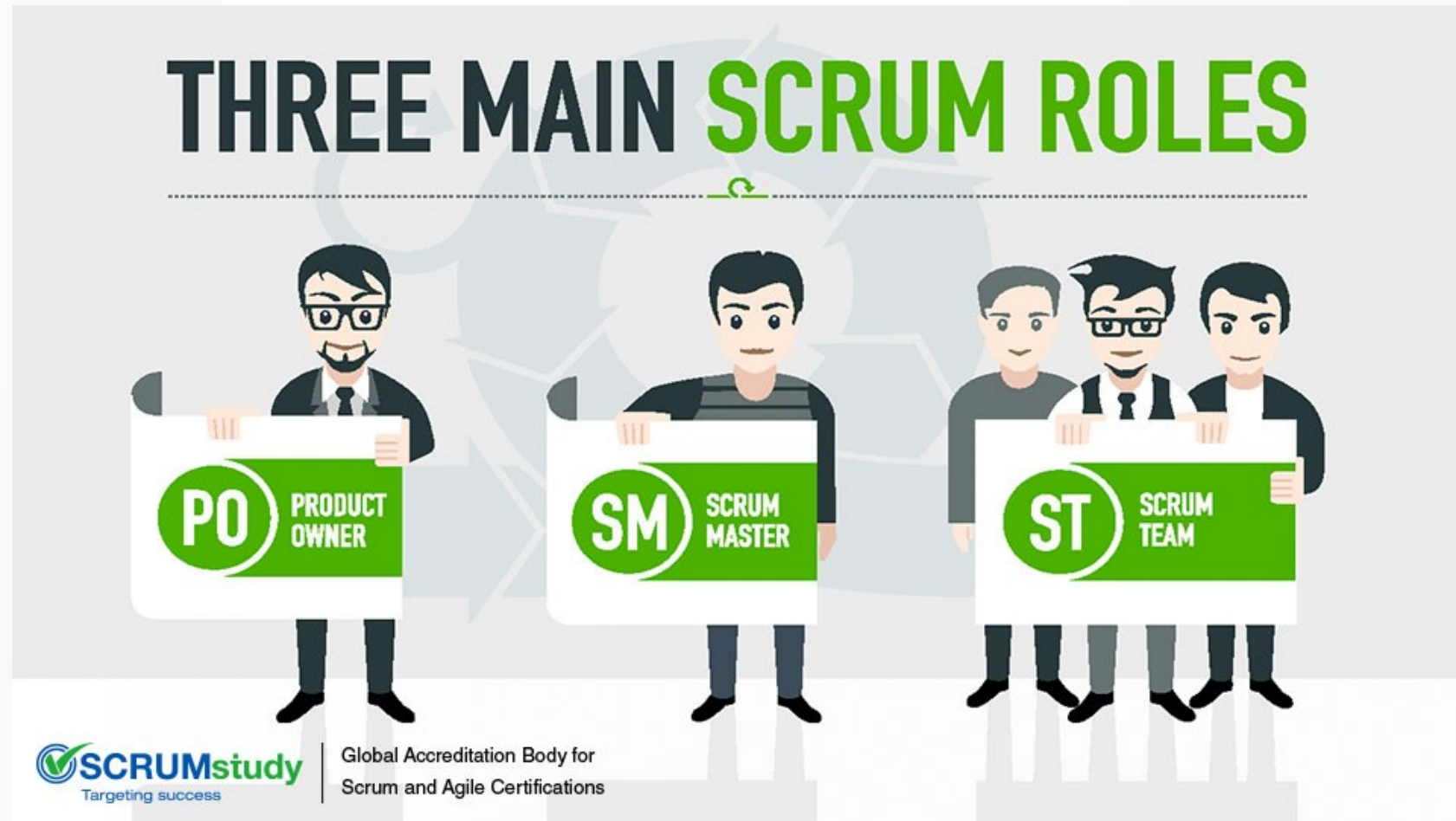
SCRUM CONCEPTOS

- **Sprint**
 - Iteración
- Sprint **Planning**
 - ¿Qué vamos a hacer?
- **Daily** Scrum
 - Reunión diaria muy breve
 - ¿Cómo vamos?
- Product **Backlog**
 - Tareas que tenemos pendientes en todo el proyecto
- Sprint **Backlog**
 - Tareas que hemos seleccionado para este sprint
- Sprint **Review**
 - ¿Qué hemos hecho?
 - ¿Qué vamos a entregar?
- Sprint **Retrospective**
 - ¿Cómo lo hemos hecho?
 - ¿Qué podemos mejorar?

UD 1. DESARROLLO DE SOFTWARE

METODOLOGÍAS ÁGILES

SCRUM: MIEMBROS



UD 1. DESARROLLO DE SOFTWARE

METODOLOGÍAS ÁGILES

SCRUM: MIEMBROS

- **PO (Product Owner)**

- **Debéis memorizar esas siglas: PO**

- Es es el representante del CLIENTE en el proyecto y puede ser el propio cliente, aunque **no es lo habitual**.

- Es el responsable de maximizar el valor del producto y el trabajo del equipo.

- **El Product Owner es el ÚNICO que gestiona el Product Backlog.**

- **Scrum Master**

- **NO es la persona más importante ni la que tiene mayor responsabilidad.**

- Este rol es responsable que **Scrum sea entendido y promulgado**, por lo que se aseguran que el Scrum Team, sigue las prácticas y reglas marcadas.

- El Scrum Master es un sirviente-leader para el Scrum Team y para el PO.

- Podríamos decir que es un “coach” que se encarga de que todo funcione bien.

UD 1. DESARROLLO DE SOFTWARE

METODOLOGÍAS ÁGILES

SCRUM: MIEMBROS

- **Development Team**

- Está compuesto por aquellos profesionales que realizan el trabajo de desarrollar una potencial e incremental versión del producto o servicio al final de cada Sprint.
- Solo los miembros del Development Team pueden crear ese incremento.
- El Development Team está estructurado y tiene el suficiente poder para organizar y **gestionar su propio trabajo.**
- Esto da como resultado que se optimiza la eficiencia y efectividad general del trabajo, al no depender de factores externos.

UD 1. DESARROLLO DE SOFTWARE

METODOLOGÍAS ÁGILES

SCRUM: MIEMBROS

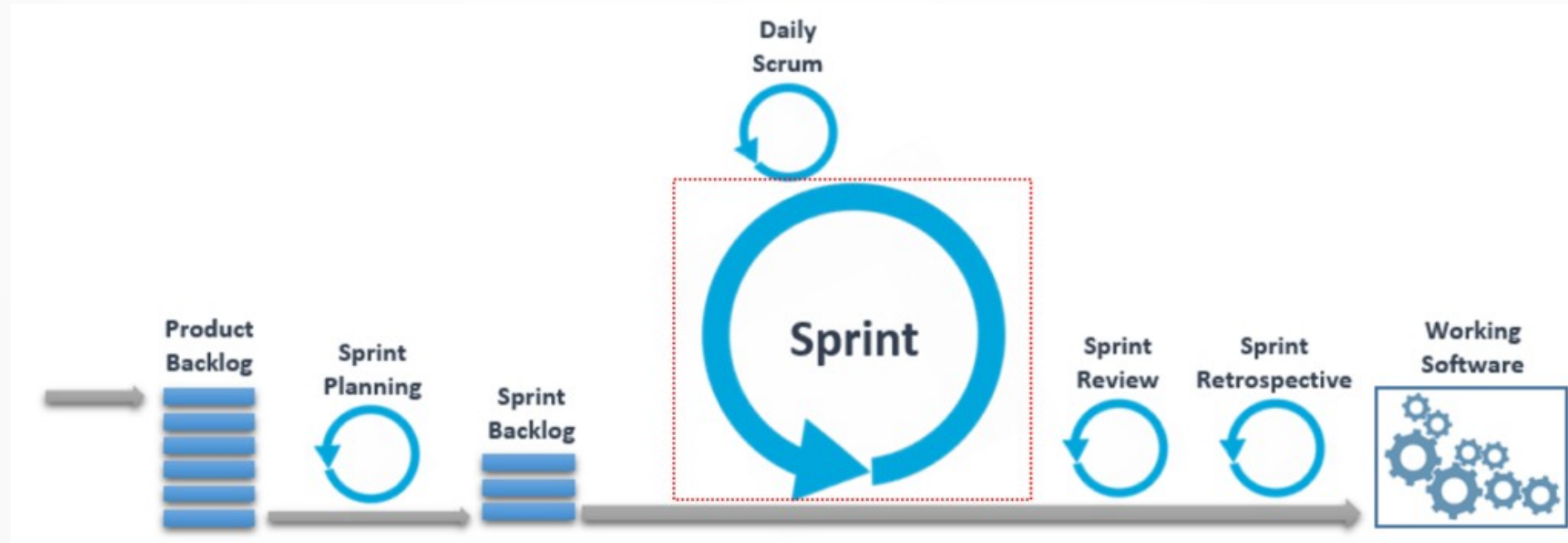
- **Cualidades del PO, del Scrum Master (y de CUALQUIER GESTOR DE PERSONAS)**
 - Como en cualquier proceso y aspecto de la vida, pueden delegar sus tareas pero NUNCA sus responsabilidades.
 - Deben ser personas **empáticas**, colaboradoras, pacientes, y con el coraje de defender los roles y funciones de Scrum (o de la metodología usada).
 - Aquí tenéis una aclaración sobre ambos roles:

<https://www.saraclip.com/funciones-del-scrum-master/>

UD 1. DESARROLLO DE SOFTWARE

METODOLOGÍAS ÁGILES

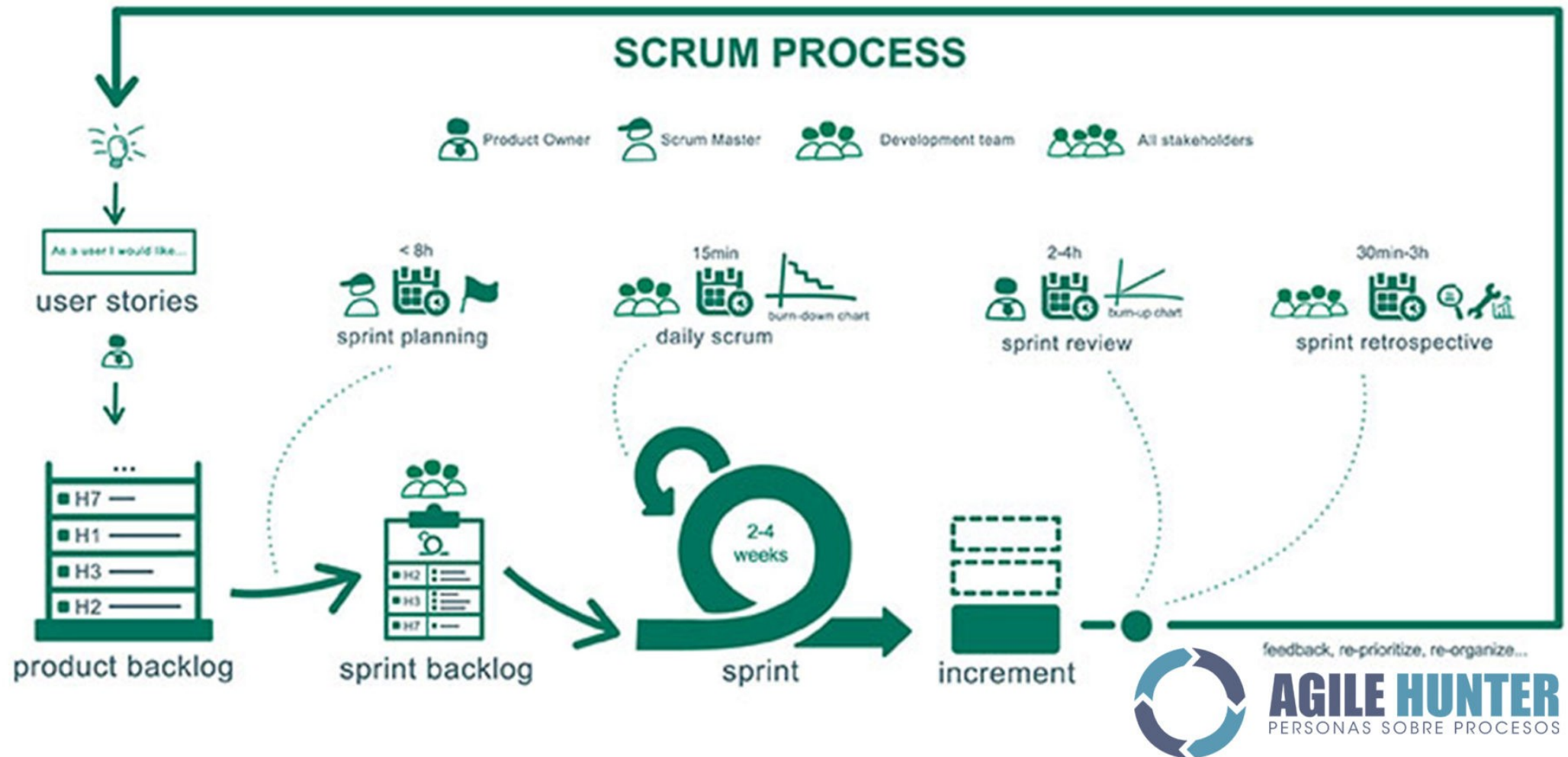
SCRUM



UD 1. DESARROLLO DE SOFTWARE

METODOLOGÍAS ÁGILES

SCRUM



UD 1. DESARROLLO DE SOFTWARE

METODOLOGÍAS ÁGILES

1 METODOLOGÍAS ÁGILES

1.1 INGENIERÍA DE SOFTWARE

1.2 MANIFIESTO ÁGIL

1.3 EXTREME PROGRAMMING

1.4 SCRUM

1.5 KANBAN

1.6 TDD



UD 1. DESARROLLO DE SOFTWARE

METODOLOGÍAS ÁGILES

KANBAN

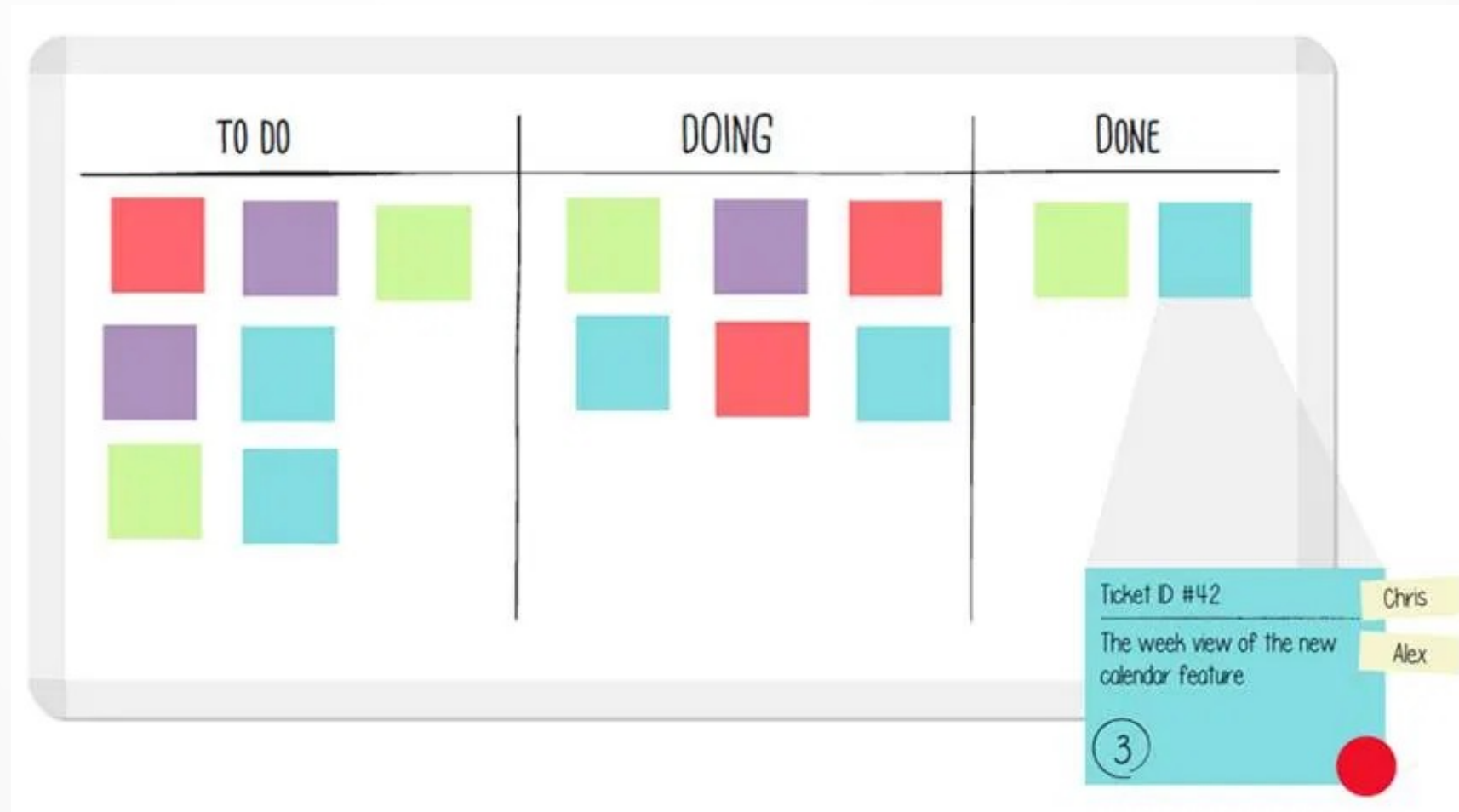
- La estrategia Kanban conocida como 'Tarjeta Visual» muy útil para los responsables de proyectos.
- Esta consiste en la elaboración de un cuadro o diagrama en el que se reflejan tres columnas de tareas; pendientes, en proceso o terminadas.
- Este cuadro debe estar al alcance de todos los miembros del equipo, evitando así la repetición de tareas o la posibilidad de que se olvide alguna de ellas.
- Por tanto, ayuda a mejorar la productividad y eficiencia del equipo de trabajo.

<https://www.iebschool.com/blog/que-son-metodologias-agiles-agile-scrum/>

UD 1. DESARROLLO DE SOFTWARE

METODOLOGÍAS ÁGILES

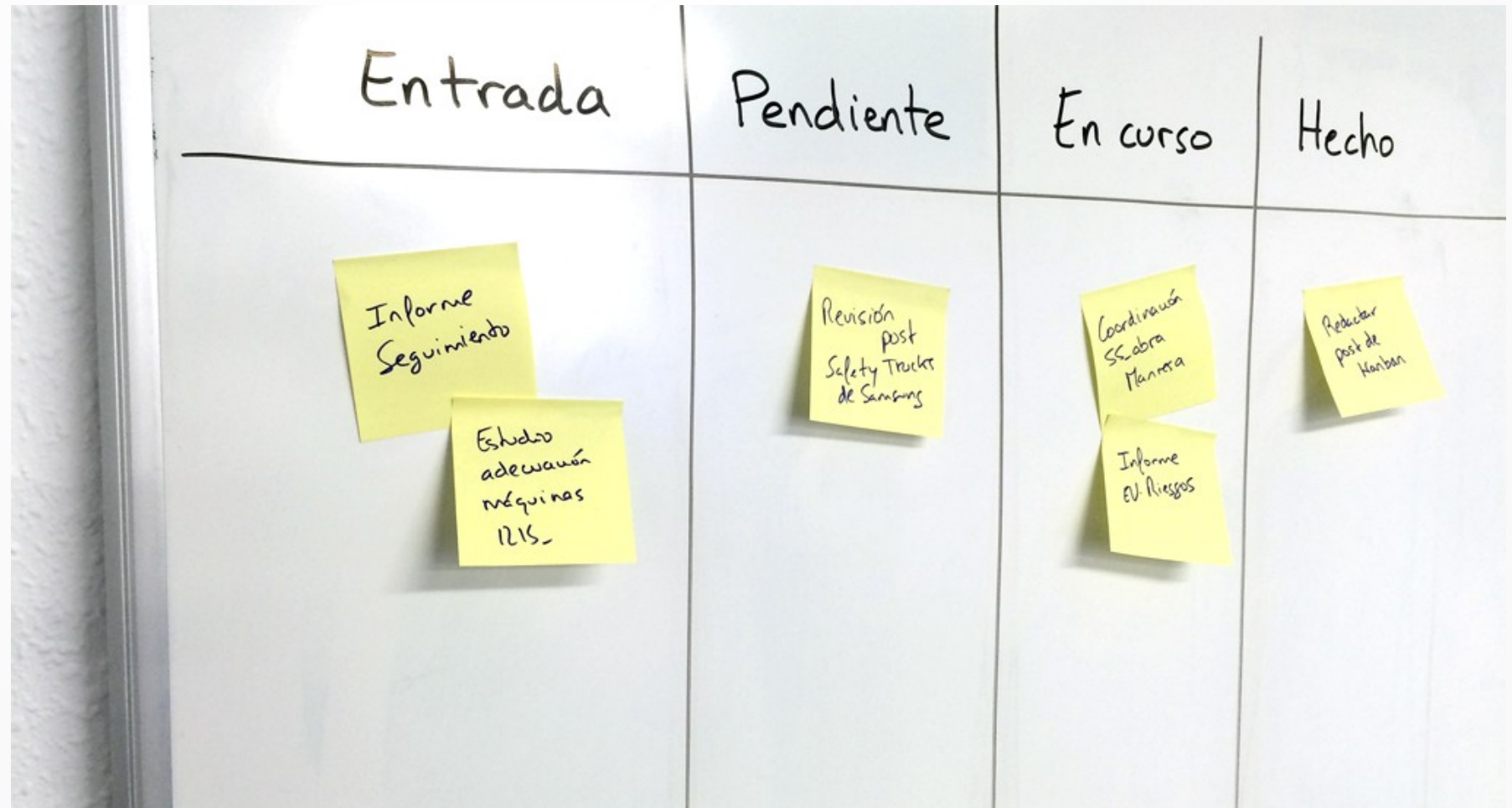
KANBAN



UD 1. DESARROLLO DE SOFTWARE

METODOLOGÍAS ÁGILES

KANBAN



UD 1. DESARROLLO DE SOFTWARE

METODOLOGÍAS ÁGILES

KANBAN



UD 1. DESARROLLO DE SOFTWARE

METODOLOGÍAS ÁGILES

1 METODOLOGÍAS ÁGILES

1.1 INGENIERÍA DE SOFTWARE

1.2 MANIFIESTO ÁGIL

1.3 EXTREME PROGRAMMING

1.4 SCRUM

1.5 KANBAN

1.6 TDD



UD 1. DESARROLLO DE SOFTWARE

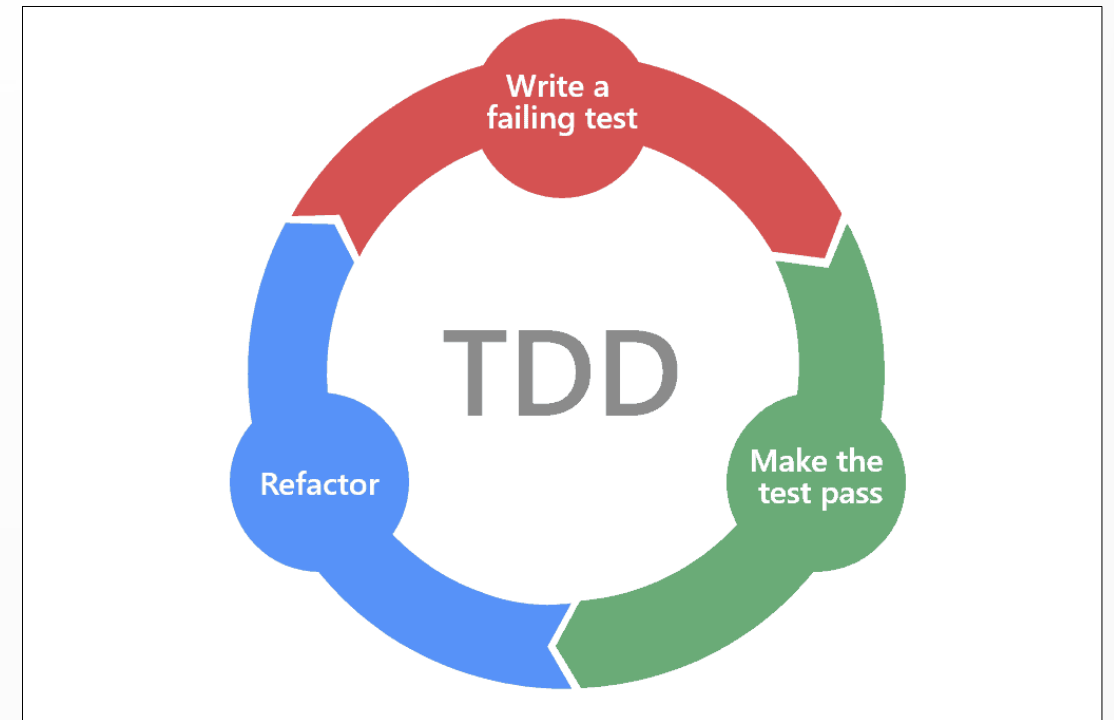
METODOLOGÍAS ÁGILES

TDD

Test-driven development (TDD).

Tiene tres fases principales:

- Test first.
- Code.
- Refactoring.

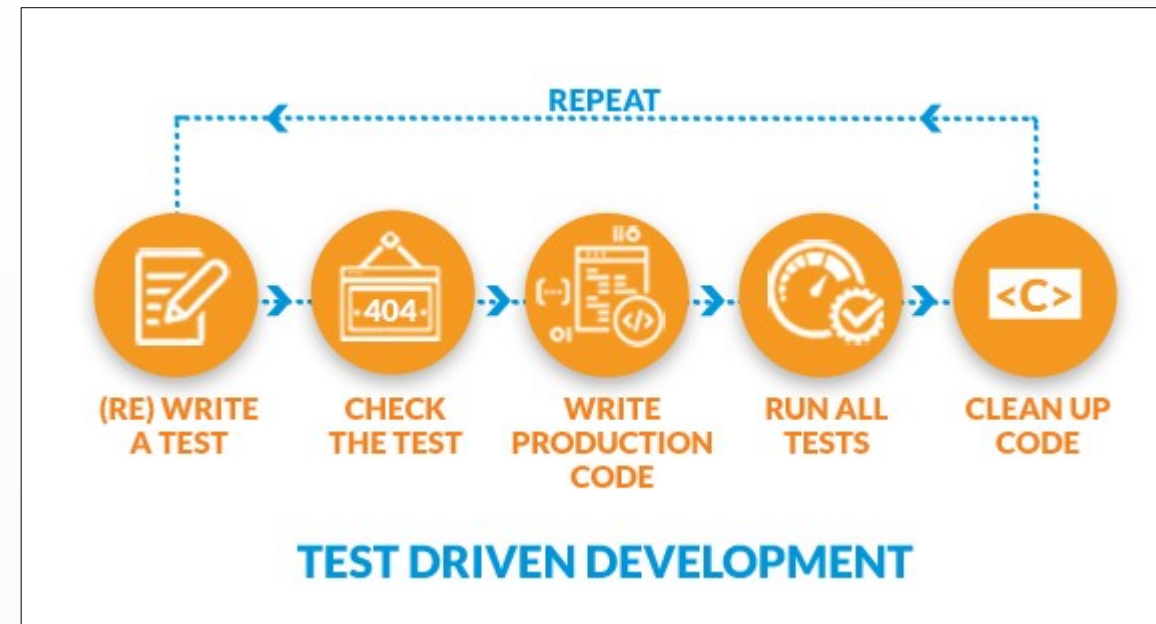


UD 1. DESARROLLO DE SOFTWARE

METODOLOGÍAS ÁGILES

TDD. Ciclo de desarrollo.

1. Elegir un requisito.
2. Escribir un test unitario que falle.
3. Verificar que el test falla.
4. Escribir el código más sencillo posible para que el test no falle.
5. Verificar si los test anteriores no dan fallo.
6. Refactorizar código.
7. Actualizar la lista de requisitos.



UD 1. DESARROLLO DE SOFTWARE

METODOLOGÍAS ÁGILES

TDD. Ejemplo.

<https://www.basicsbehind.com/2014/09/17/tdd-by-example-2/>

¡Solo si sabes de programación!