

MÓDULO 5: Gestión de proyectos de investigación e innovación

b.1 Gestión de Proyectos Ágiles

ICSEM

Febrero de 2022

Introducción a la Gestión de Proyectos Ágil

Guía práctica de Ágil

Conceptos Ágiles

Capítulo 1-Introducción

Capítulo 2-Introducción a Agile: manifiesto, métodos, etc.

Capítulo 3-Selección del ciclo de vida de desarrollo del proyecto

Implementación de Agile

Capítulo 4-Crear un entorno ágil: roles del PM y del equipo

Capítulo 5-Entregar en un entorno ágil: técnicas y herramientas para proyectos ágiles

Capítulo 6-Consideraciones organizacionales: cambio, cultura, contratos, negocio, PMO

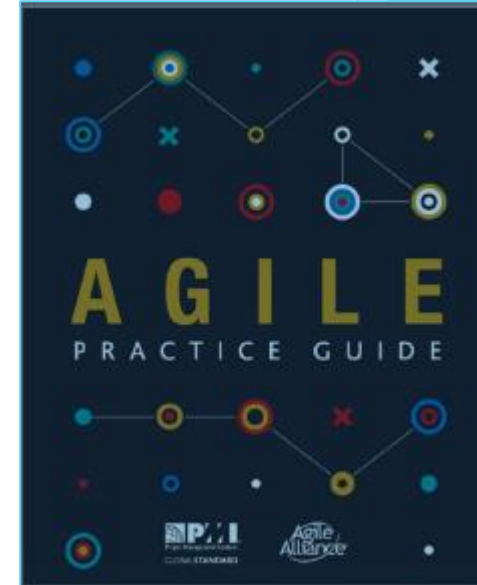
Anexos:

Anexo 1 -Relación con las 10 áreas de conocimiento de la Guía del PMBOK

Anexo 2 -Relación con los 4 valores y 12 principios del Manifiesto Ágil

Anexo 3 -Resumen de los marcos ágiles más conocidos

Apéndices -Consideraciones sobre Adaptación, Herramientas para elegir el modelo de ciclo de vida de desarrollo del proyecto, Referencias, Glosario, etc.



Aplicar Principios Ágiles en los Proyectos Ágiles



El Director de Proyectos Ágil debe explorar, fomentar y aplicar principios ágiles en el contexto del equipo del proyecto y la organización.

¿Por qué Agile?

“Diferentes tipos de proyectos, requieren diferentes enfoques de gestión”

Las personas nos adaptamos continuamente nuestra forma de hacer las cosas a la situación. No hacemos las cosas como un robot, siempre igual, siguiendo los mismos pasos.

Esto es particularmente necesario en Proyectos en el trabajo del conocimiento donde el cambio y la complejidad del entorno es una llamada directa a los enfoques ágiles.

Trabajo industrial vs. Trabajo del conocimiento

Trabajo del Conocimiento (Knowledge Work) =

Expertos en la materia (SME) colaborando para crear o mejorar un producto o servicio.

No sólo nos referimos a la industria de las TI. Ingenieros, abogados, científicos...etc son trabajadores del conocimiento.



1. Reducción de la tasa de error
2. Dirección autoritaria (presionar, más horas)
3. Recursos intercambiables
4. Ritmo de producción constante
5. Proceso estándar

1. Exigir una cuota de error
2. A los programadores les gusta su trabajo
3. Los programadores no son intercambiables
4. Un proyecto es algo dinámico
5. Replantearse el trabajo (no sólo hacerlo)

Trabajo del conocimiento

Cuando se empiezan a desarrollar proyectos en el ámbito del trabajo del conocimiento, se empiezan a constatar que:

- Hace falta mucha comunicación y colaboración
- El trabajo es mas incierto y menos definible
- Hace falta mas involucración por parte de las personas
- Surgen muchas indeterminaciones y no se conoce exactamente lo que se quiere conseguir, o como conseguirlo.
- Aparecen muchos cambios al no definirse de forma clara y especifica el trabajo a ser realizado y/o los resultados a obtener

Los Enfoques Agiles fueron desarrollados para responder a esta situación. Aunque comenzaron como hemos visto en el ámbito del software, hoy en día son aplicados en todo tipo de proyectos relacionados con el trabajo del conocimiento

Trabajo del conocimiento

Proceso definido: los pasos pueden ser definidos, es la forma mas eficiente si trabajamos en un entorno de **baja incertidumbre**...por ejemplo, proyectos industriales, de construcción, pueden ser gestionados por procesos definidos.

Proceso empírico: Al trabajar con **incertidumbre**, se requiere un proceso de **prueba y experimentación**, para determinar que trabajo desarrollar y construir mediante **pequeñas iteraciones**. El resultado es un avance iterativo e incremental con revisiones frecuentes y adaptación.

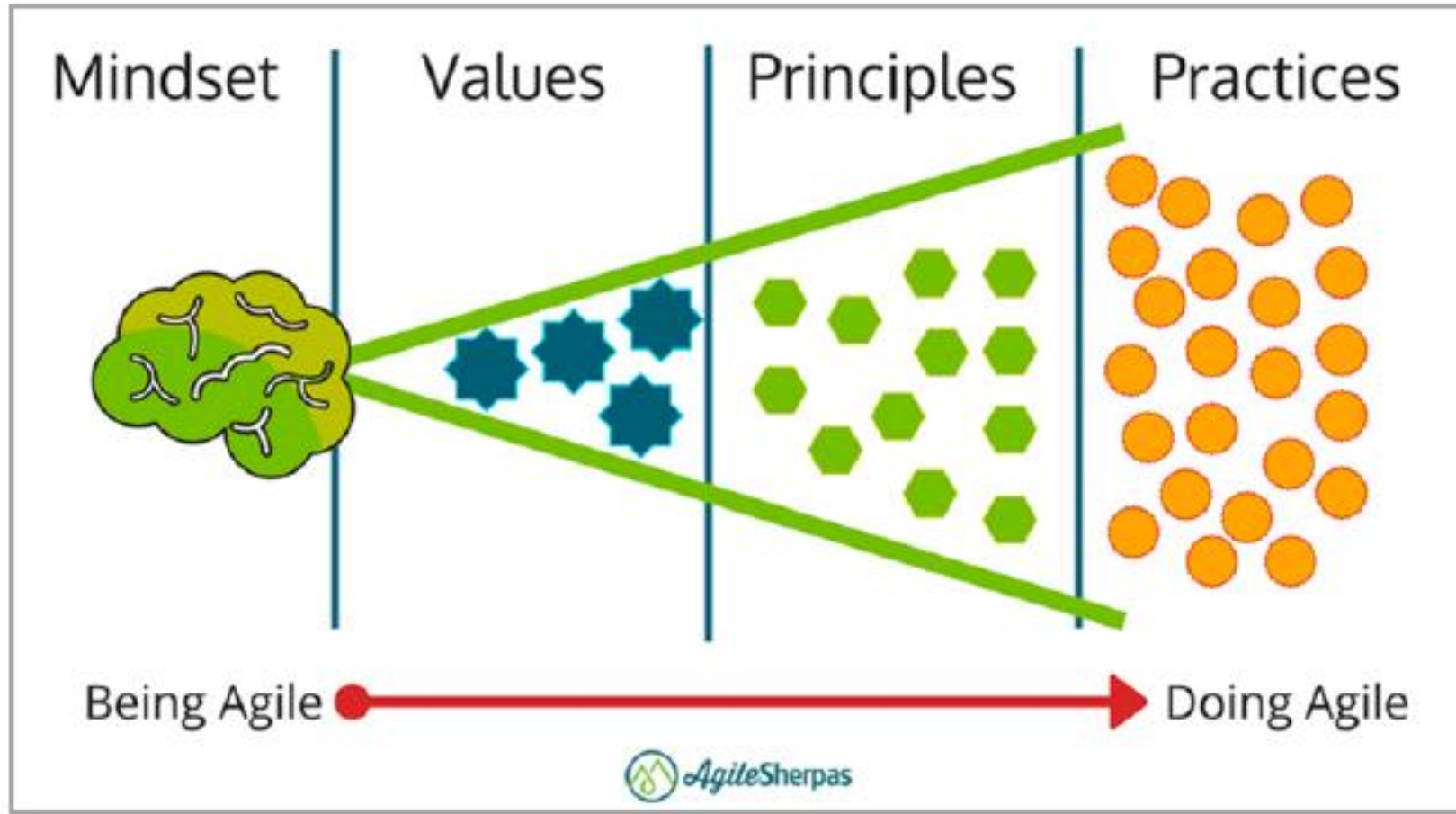
Agile

Podemos resumir los principales principios:

- Bienvenidos los cambios
- Trabajar en pequeños incrementos de valor añadido
- Utilizar ciclos de construcción y feedback
- Descubrir mediante el aprendizaje
- Desarrollar con el foco en el valor
- Fallar rápido con aprendizaje
- Entregar de forma continua
- Mejorar de forma continua

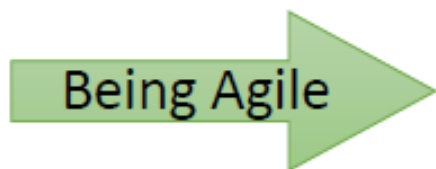
*Agile es una mentalidad, una forma de entender y abordar los proyectos, fuertemente arraigadas en principios, valores, y, por ultimo, herramientas.
Diferencia entre “Ser agile” “Hacer agile”*

Mentalidad ágil

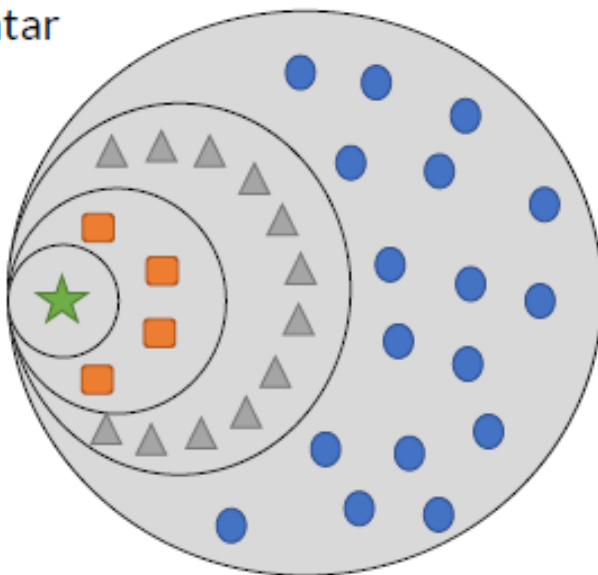


Agile

“Being agile” es la forma correcta de implementar agile



“Ser ágil” comienza interiorizando la mentalidad, y entonces usar ese entendimiento para seleccionar e implementar las prácticas correctas, adaptándolas



Agile es una mentalidad (★)
definida por **valores** (■)
guiada por **principios** (▲)
manifestada a través de muchas **prácticas** diferentes (●)

“Doing agile” es un problema común en la adopción de agile.

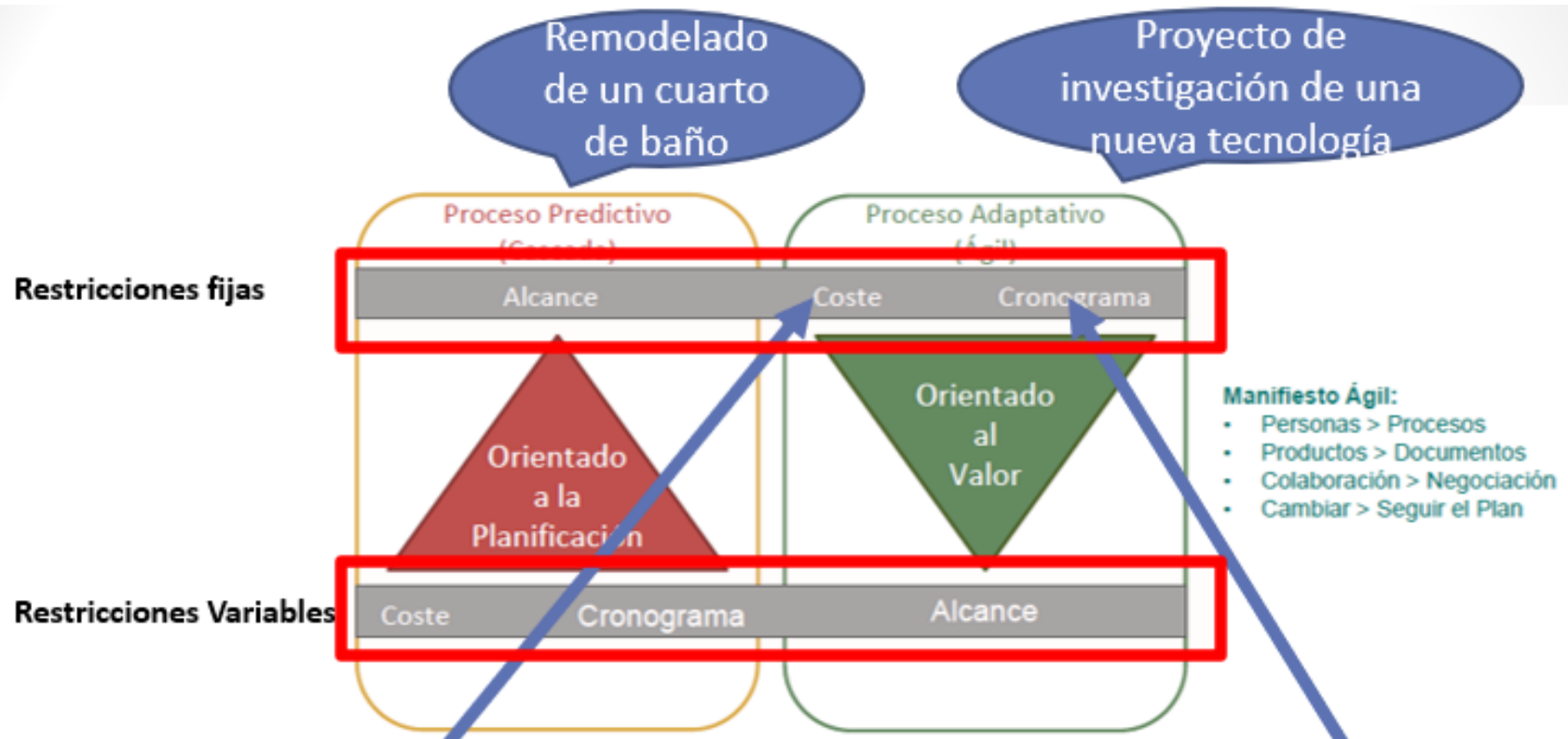


Usar prácticas ágiles sin adoptar la mentalidad ágil que nos permita encontrar el balance adecuado de prácticas y adaptarlas

Mentalidad ágil



Cambio de Paradigma en Proyectos Ágiles



En el enfoque Ágil, se permiten cambios en el alcance dentro de los parámetros fijos de tiempo y coste. Es decir, perseguimos entregar el máximo valor posible en la fecha fijada y con el presupuesto aprobado.

La declaración de interdependencia

Este manifiesto se firmó en el 2005 por un grupo de jefes de proyecto, gerentes y expertos incluyendo dos de los autores originales del “Manifiesto por el desarrollo ágil del software“. Este manifiesto, al contrario que el anterior, está más enfocado a la gestión ya que la idea original era extender el manifiesto por el desarrollo ágil de software a productos que no fueran solamente software, a gestión de proyectos en particular y a gestión en general.

Somos una comunidad de líderes de proyectos. Conseguimos que los proyectos produzcan resultados de éxito. Para ello:

- Incrementamos el ROI enfocándonos en el flujo continuo del valor.
- Entregamos resultados fiables involucrando a los clientes, en interacciones frecuentes y compartiendo la propiedad de los resultados.
- Esperamos incertidumbre y la gestionamos a través de iteraciones, anticipación y adaptación.
- Liberamos la creatividad y la innovación reconociendo que los individuos son la verdadera fuente de valor y creando un entorno donde puedan ser diferenciales.
- Optimizamos el desempeño a través de la responsabilidad en grupo sobre los resultados y responsabilidad compartida de la efectividad del equipo.
- Mejoramos la efectividad y fiabilidad a través de estrategias específicas contextuales, procesos y prácticas.

El manifiesto ágil

Investigando mejores métodos para desarrollar software, por experiencia propia y ayudando a otros, concluimos que hay que valorar lo siguiente:

Individuos e interacciones	<i>sobre</i>	Procesos y herramientas
Software que funcione	<i>sobre</i>	Documentación extensiva
Colaboración con el cliente	<i>sobre</i>	Negociación contractual
Responder a los cambios	<i>sobre</i>	Seguir el plan

Aunque valoramos los puntos de la derecha, valoramos más los de la izquierda.

1. *Personas > Procesos*
2. *Productos > Documentos*
3. *Colaboración > Negociación*
4. *Cambiar > Seguir el Plan*

www.agilemanifesto.org

Principios ágiles

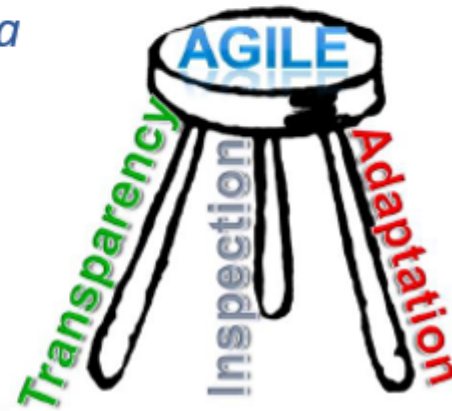
1. Nuestra máxima prioridad satisfacer al **cliente** través de la entrega, desde el **principio** y de forma continua, de **software con valor**.
2. Se reciben de buen grado los **cambios en los requisitos**, incluso en una fase tardía del desarrollo. Los procesos Ágiles estimulan los cambios para favorecer la ventaja competitiva del cliente.
3. Entregar **frecuentemente software que funcione** (con valor para el negocio), desde una vez cada 2 semanas hasta una vez cada 2 meses, cuanto más breve el ciclo de entrega, mejor.
4. Los representantes del negocio y el equipo de desarrollo deben **trabajar juntos** día a día durante el proyecto.
5. Desarrollar equipos con **personas motivadas**. Proporcionándoles el entorno y soporte que necesiten, y confiando en ellos para que hagan el trabajo.
6. El método más eficiente de transmitir información al equipo y dentro del equipo mediante conversaciones **cara a cara**.

Principios ágiles

7. **El software que funcione** (entregas con valor) es la principal medida de progreso.
8. Los procesos ágiles facilitan el **desarrollo sostenible**. Patrocinadores, desarrolladores y usuarios deberían ser capaces de mantener un ritmo constante de forma indefinida.
9. Prestar atención continua a la **excelencia técnica** y al **buen diseño**, como factores para mejorar la agilidad.
10. La simplicidad (el arte de) es esencial.
11. Las mejores arquitecturas, requisitos y diseños, surgen de equipos **auto-organizados**.
12. A intervalos regulares, el **equipo reflexiona** sobre cómo ser más efectivo, **ajustando** su comportamiento en consecuencia.

Objetivos comunes de los Métodos Ágiles

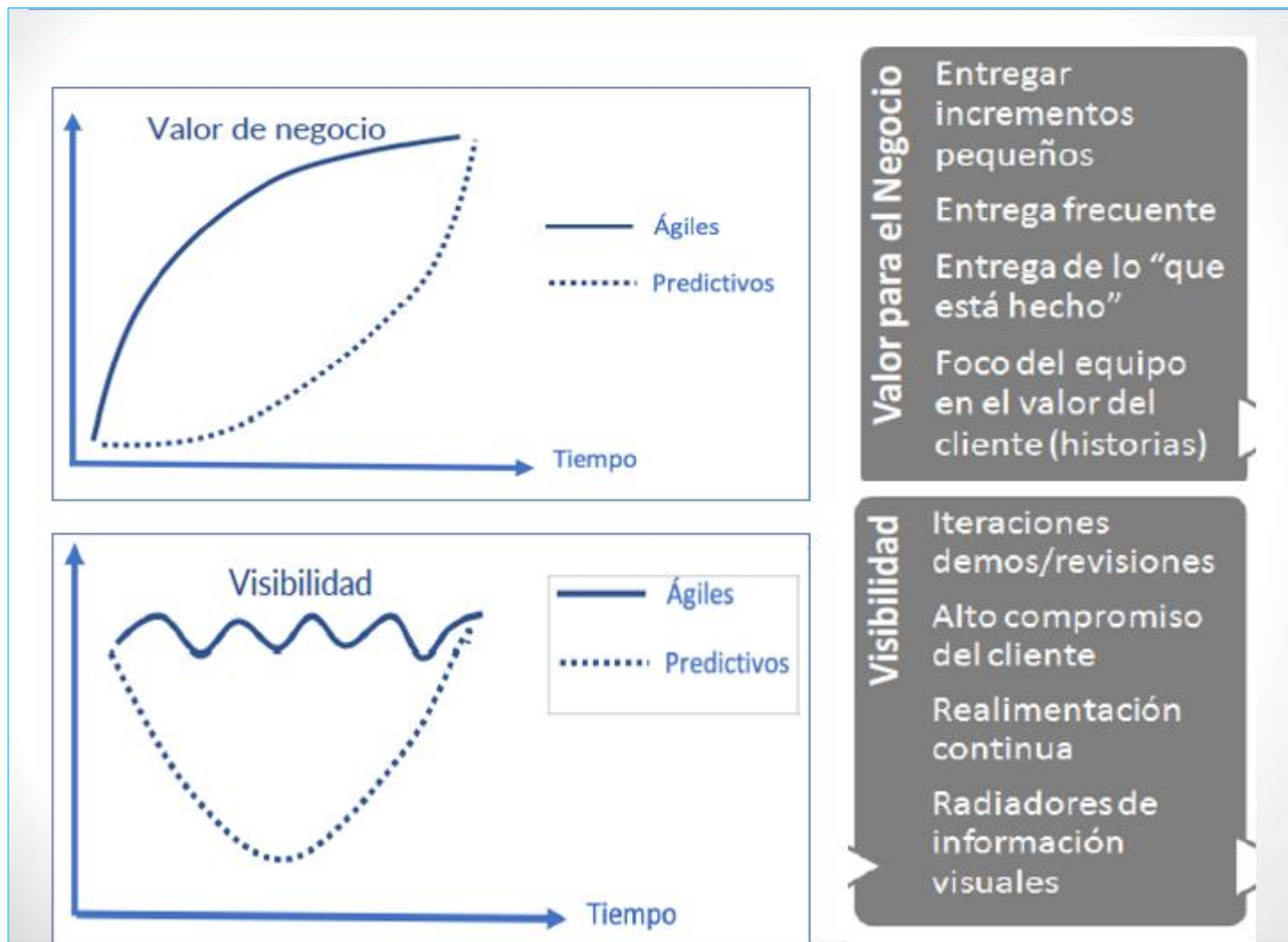
1. Entregas incrementales de **valor** a los interesados
2. **Retroalimentación** desde el principio y continua de los interesados
3. Trabajar en **equipos crossfuncionales** y altamente colaborativos
4. Enfocarse en **completar** el trabajo de mayor valor
5. Alta **calidad** y foco en la excelencia técnica
6. Alta visibilidad, medición y control **visual**
7. Inspección y adaptación **continuas**



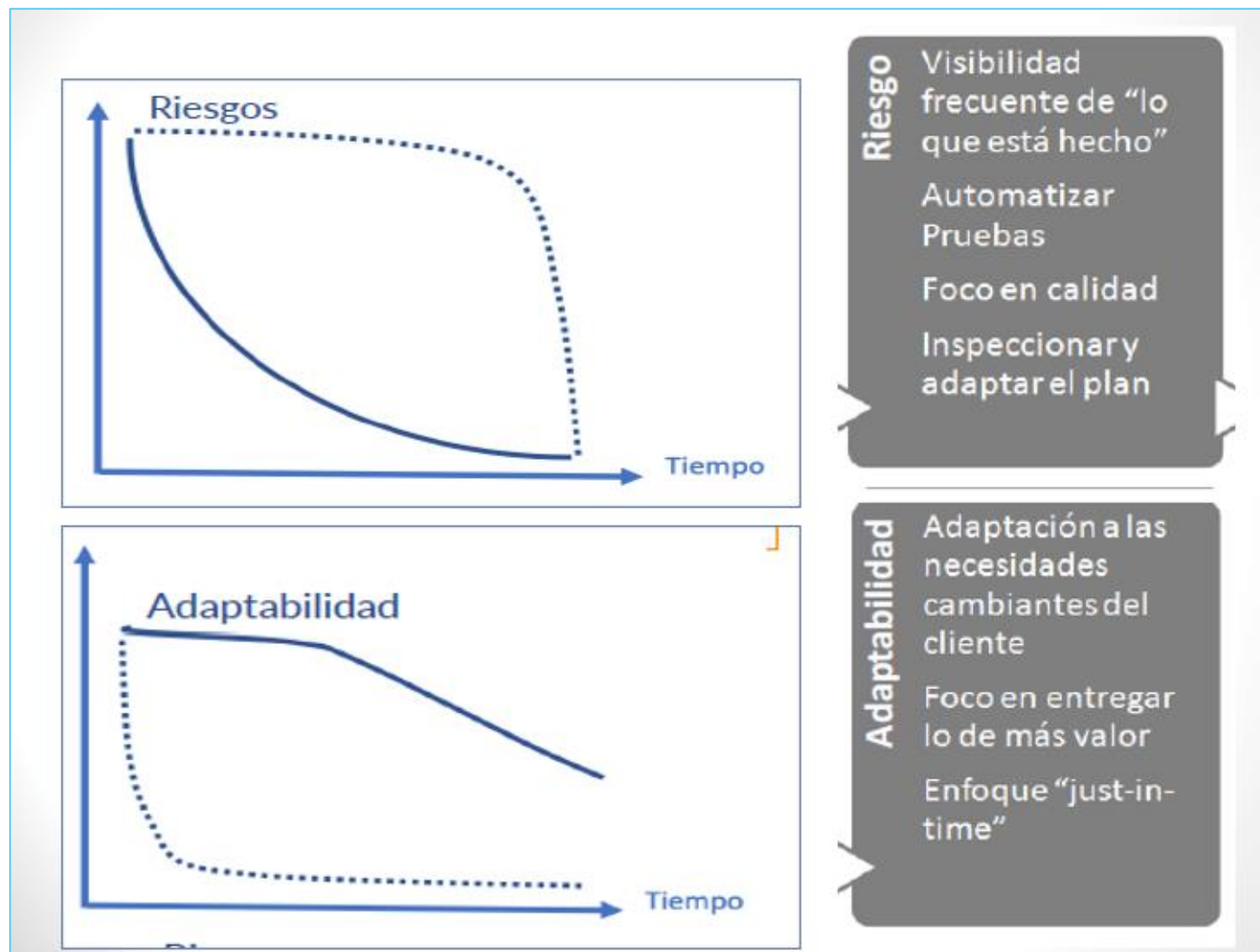
Objetivos comunes de los Métodos Ágiles



La entrega basada en valor



La entrega basada en valor



Entrega de valor temprano

Uno de los caminos claves es maximizar valor mediante la entrega de valor temprano. ¿Por qué entregar temprano?

- 1. Las cosas pasan, los riesgos aparacen, cuanto mas grande es el horizonte, mas incertidumbre de riesgos que pueden aparecer y trastornar. Para maximizar el éxito debemos entregar tantos componentes de alto valor como sea posible, antes de que las cosas puedan cambiar.*
- 2. Entregando temprano, la satisfacción de los stakeholders, se incrementa, se involucran más, el equipo demuestra el entendimiento de las necesidades de los stakeholders, aumentando la confianza de estos*

Es recomendable tomar decisiones que prioricen las actividades de alto valor añadido y esfuerzos para reducir riesgos y después ejecutar con estas prioridades.

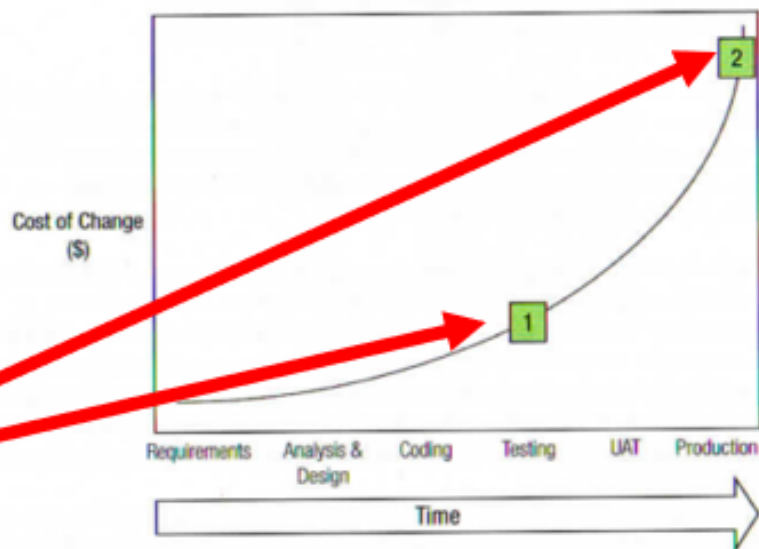
Entregando incrementalmente

Las entregas incrementales son otra forma de optimizar la entrega de valor por parte de las metodologías ágiles.

El equipo despliega regularmente incrementos del producto a lo largo del proyecto.

En el caso de proyectos de desarrollo de software, estos despliegues son al área de testeo, aunque si tiene sentido para el negocio, pueden ser despliegues a producción.

Es mucho mas barato solucionar un problema localizado en la fase de test que en la fase de producción.

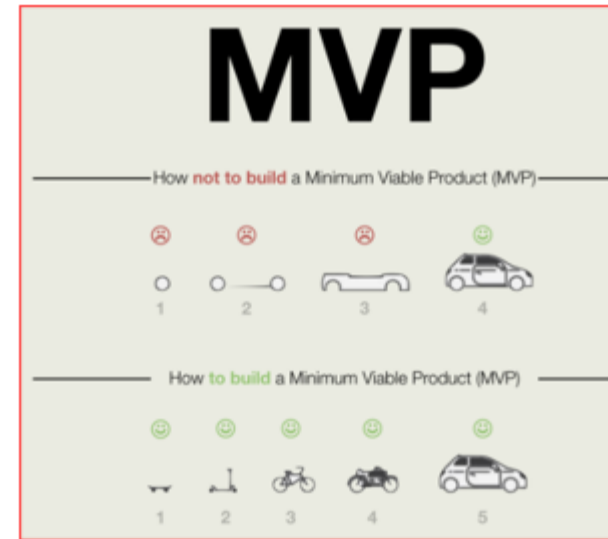


Producto mínimo viable

Es la versión creada con mínimo esfuerzo que puede ser usada para recopilar la **máxima cantidad de aprendizaje** sobre la opción del cliente

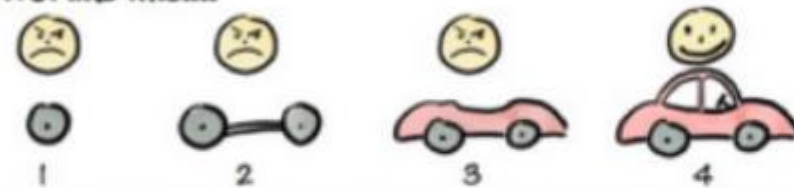
Evita construir productos que los clientes no quieren, y busca maximizar aprendizaje sobre el cliente por unidad de coste invertida.

AYUDA A VALIDAR HIPOTESIS DE NEGOCIO




Producto mínimo viable

Not like this....



1 2 3 4

Like this!



1 2 3 4 5

No es iterativo ni incremental...el cliente no percibe valor hasta el final

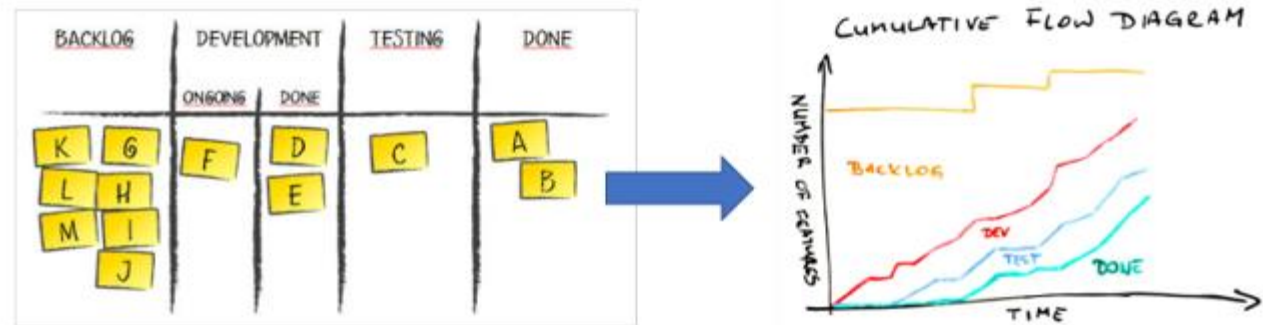
Iterativo e incremental, proporcionamos valor en cada entrega y recibimos feedback, el cliente percibe valor de forma continua

Diagrama de flujo acumulado

Son utiliza para monitorizar y pronosticar la **entrega de valor**.

Se representa en vertical el numero de elementos acumulado en cada columna del tablero Kanban, y los valores se van acumulando

En horizontal el eje del tiempo del proyecto



Involucrando a los interesados

Manteniendo a los interesados involucrados:

ITERACIONES CORTAS: Ayudan mantener el interés. Buen aliado par mantenerles enganchados.



GESTIÓN CONTINUA DE CAMBIOS: Asegurar que las peticiones y cambios emerjan cuanto antes, para identificar riesgos, defectos e incidencias.

HABILIDADES BALNDAS: Para que las relaciones sean mas fluidas.

ESCALADO DE INCIDENCIAS O DECISIONES: Establecer un procedimiento ágil de escalado de aquellas decisiones que están fuera del equipo, para minimizar tiempo.

Involucrando a los interesados

Definición de hecho (Definition of Done)

*¿Que es? Una **lista sencilla y auditable** que pertenece al equipo*

Depende de la organización, las especificaciones del producto o de la entrega.

*Es **única** para distintos equipo y organizaciones*

*Se utiliza como **filtro** para decidir lo que está listo para ser lanzado*



Trabajando colaborativamente

Involucrando un gran grupo de interesados, llevará a una mejor toma de dediciones.

Los beneficios reales vienen en lo que se produce dentro del grupo:

- *Resolver problemas frente a seguir instrucciones*
- *Iniciativa frente a pasividad*
- *Entendimiento colectivo frente a silos funcionales.*
- *Propiedad colectiva de las ideas*



Trabajando colaborativamente

Algunas prácticas que fomentan el enfoque colaborativo:

Daily stand ups

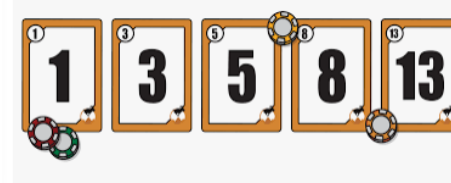
Toma de decisiones participativa

Estimación mediante planning poker

Resolución de problemas basado en equipo

OTROS:

Talleres (Workshops), Lluvia de ideas (Brainstorming), Juegos colaborativos (Recuerda el futuro, Forma el árbol de producto, el velero)



Desempeño del equipo

Tratar sobre el desarrollo y soporte de los equipos auto-organizados, (y auto-empoderados) y ayudarles a desarrollarse y tener éxito.

- Numero pequeño (12 o menos), si son mas desglosarlos en sub-equipos
- **Habilidades complementarias**, una persona no tiene todas las habilidades, tiene varias. Entre todos, tienen todas. (Multi-Funcionalidad)
- **Compromiso a propósito común**, alineados entorno a los objetivos del proyecto
- **Responsabilidad compartida**: Comparten la responsabilidad sobre el resultado obtenido

Desempeño del equipo



“Un equipo es un pequeño número de personas con habilidades complementarias que están comprometidos en un propósito común, unos objetivos de desempeño y un enfoque que les mantiene con la responsabilidad compartida”

(The Wisdom of Teams D. Smith & J. Katzenbach)

Beneficios de especialistas generalizados

Son personas que tienen conocimientos especializados en un área concreta, pero conocimientos generales en un marco más amplio.

- Ayuda a minimizar transferencias (de conocimiento) y evitar cuellos de botella.
- Se reducen las transferencias de información.
- Se reducen los cuellos de botella e ineficiencias, ya que se agiliza el flujo, y se optimiza el trabajo, mejorando la comunicación y minimizando el desperdicio.



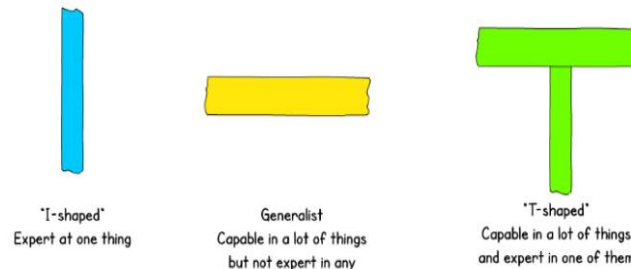
Beneficios de especialistas generalizados

Especialistas generalizados: son personas que tienen conocimientos **especializados** en un área concreta, pero conocimientos **generales** en un marco más amplio

Esto es lo que se llama **HABILIDADES EN T (T-SHAPED)**, pero tienen amplitud de conocimiento en otras...frente a la **especialización I (I-SHAPED)**, en la que sólo están especializados en un trabajo.

Beneficios de los equipos tipo T

- **Transferencias:** Si sólo tenemos especialistas puros, para cada proyecto hay mucho traspaso de conocimiento de uno a otro y se pierde información por el camino.
- **Cuellos de botella,** se reducen, ya que se agiliza el flujo, y se optimiza el trabajo, mejorando la comunicación y minimizando desperdicio.



Características de los equipos de alto rendimiento

- Crean una **visión compartida** para el equipo: Permite ir más rápido y genera confianza.
- Ponen **objetivos realistas**: Que sean alcanzables para motivar pero no frustrar.
- **Limitan el número máximo**: Para mejorar la comunicación y la eficiencia. Si son más grandes sub-dividir.
- Construyen **sentimiento de pertenencia**: Incrementa la lealtad y la ayuda al resto de miembros.
- Desarrollan un **fuerte liderazgo**: Los líderes deben señalar el camino, y dejar que el equipo lo recorra.

TeamWork (C.Larson & F. LaFasto)



PERÚ
Ministerio
de Educación

PMESUT
PROGRAMA PARA EL MEJORAMIENTO Y FORTALECIMIENTO DE LOS CENTROS
DE EDUCACIÓN SUPERIOR UNIVERSITARIA Y TECNOLÓGICA A NIVEL NACIONAL



BID
Mejorando vidas

Universidad de Vigo



Características de los equipos de alto rendimiento

- Son **auto-organizados**, más que basados en roles o títulos.
- Están empoderados para **tomar decisiones**.
- **Confían** en que como equipo pueden resolver **cualquier problema**.
- Están **comprometidos en el éxito** del equipo por encima del éxito con cualquier coste.
- El equipo es **responsable** de sus decisiones y compromisos.
- Los miembros son **motivados por confianza**, en vez de miedo o enfado.
- Están **enfocados al consenso** con divergencias que luego convergen.
- Están en **constante desacuerdo constructivo**.

Coaching agile Teams (Lyssa Adkins)

Equipos auto-dirigidos

En contra de los modelos de **Command-and-Control**, podemos utilizar el conocimiento para que el **equipo** determine la **mejor forma de llevar a cabo el trabajo**.

Hay que darles **autonomía**, para que puedan **gestionar mejor la complejidad del trabajo** (área del conocimiento).

Se establecen **objetivos de alto nivel**, y, es el **equipo el que tiene que decidir cómo y cual es la mejor forma de hacerlo** cumpliendo las normas aceptables de una organización.

Las **personas** están mucho **mas orgullosas** de sus trabajos cuando se les **reconoce como expertos**.

Así, además, se les permite **elegir el trabajo y evitar bloqueos** que surgen cuando **alguien externo decide como y que hacer**.



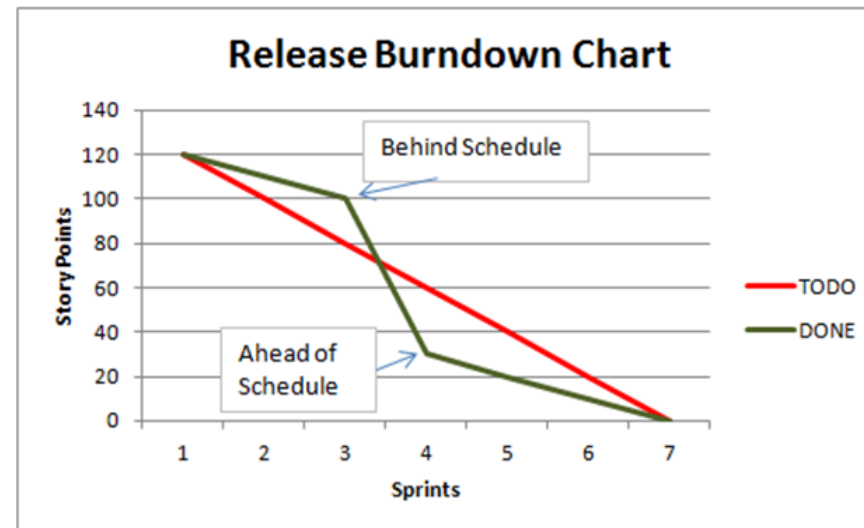
Monitorizando el desempeño

Diagramas de quemado

Herramientas muy utilizadas para dar visibilidad sobre el avance del proyecto, mostrándolo en los radiadores de información

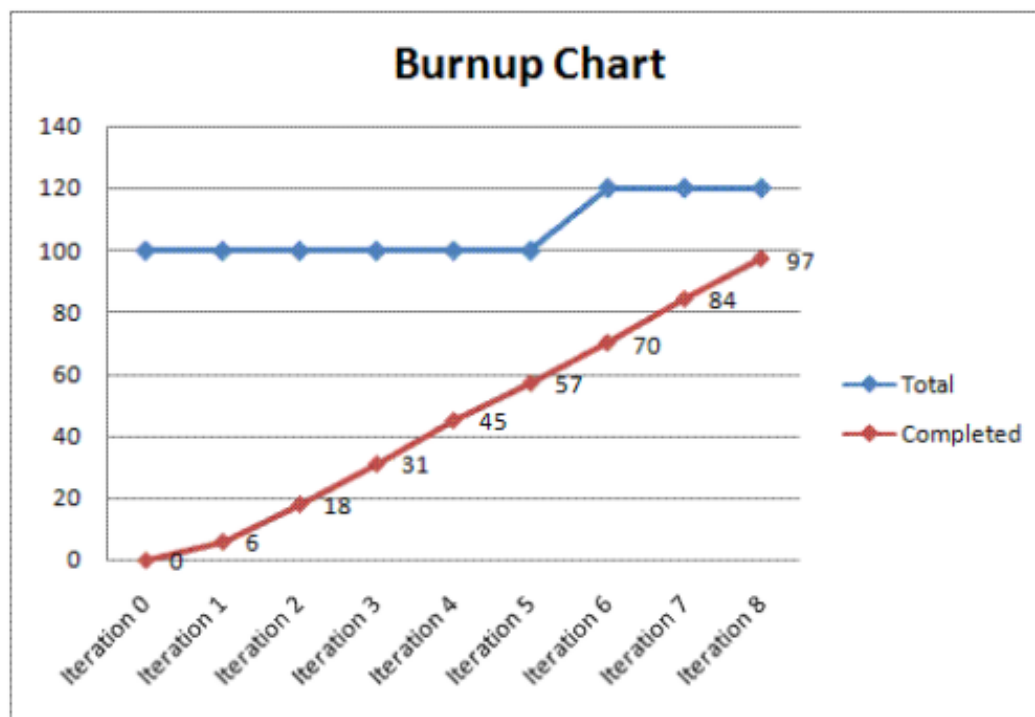
DIAGRAMAS DE QUEMADO

(Burndown chart) Muestran el esfuerzo estimado pendiente, a medida que el proyecto se completa la línea de progreso se mueve hacia abajo



Monitorizando el desempeño

Burn-up chart: Muestra la cantidad de trabajo (funcionalidades que YA han sido entregadas)

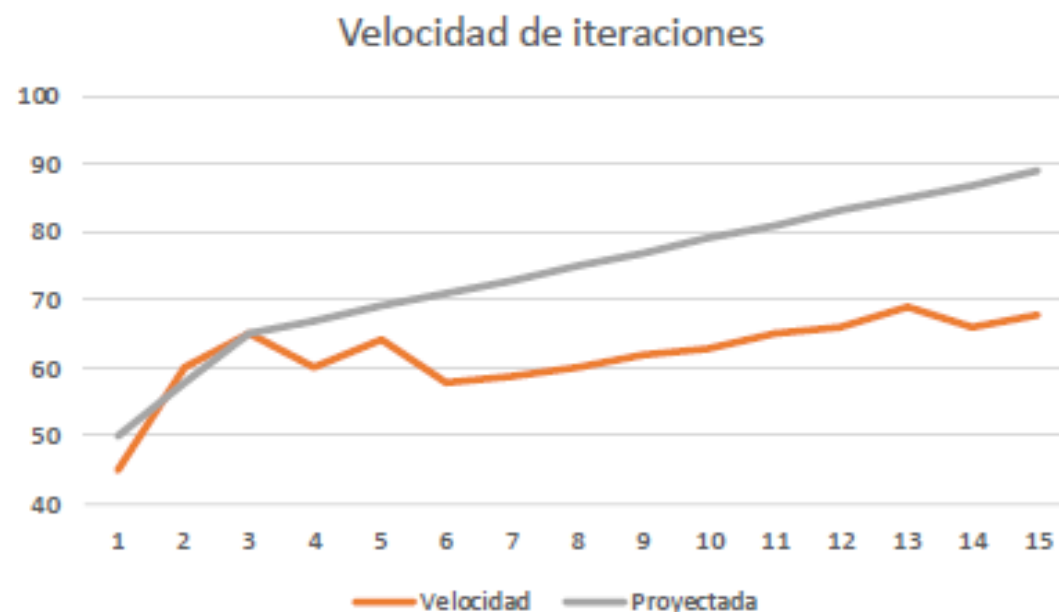


Monitorizando el desempeño

VELOCIDAD

Capacidad de un equipo para trabajar en una iteración.

Permite estimar cuanto trabajo serán capaces de hacer en estimaciones futuras, basado en lo que han hecho en las iteraciones pasadas.



Planificación adaptativa

¿Qué significa esta expresión? Se denomina “adaptativa” como enfoque contrario a la “estática” planificación tradicional que vemos en un diagrama de Gantt.

Al inicio del proyecto NO se realiza el ejercicio de detallar todas las tareas que se deben realizar para construir o desarrollar los requisitos del proyecto.

“Planificar que tendremos que re-planificar”

- *Vamos planificando solo lo que sabemos, y a medida que surgen cambios, etc...vamos re-planificando*
- *Minimizar el desperdicio (Waste) y entregar la máxima cantidad de valor*

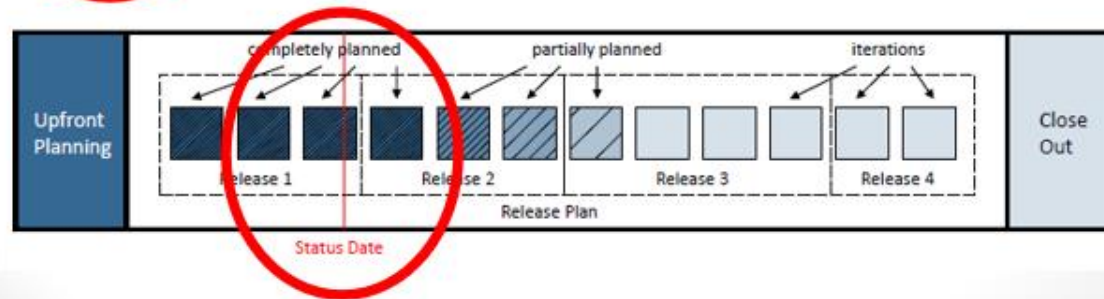
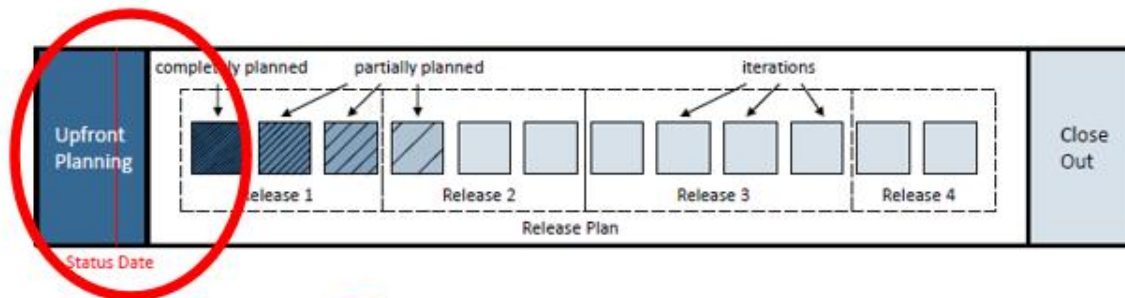


Elaboración progresiva

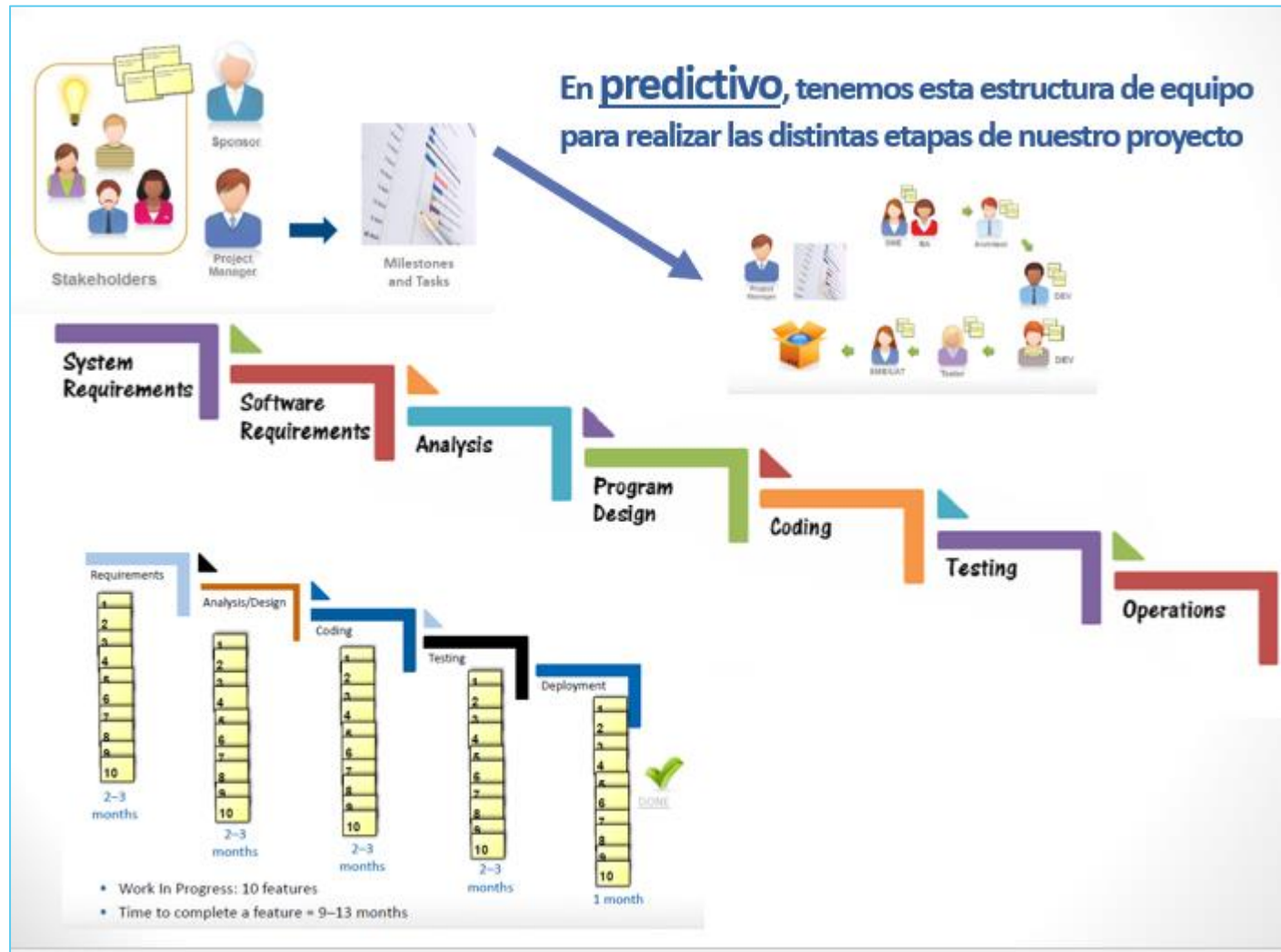
Añadir mas detalle a medida que la nueva información va surgiendo

Aplicado a : - Planes - Estimaciones - Evaluación de riesgos

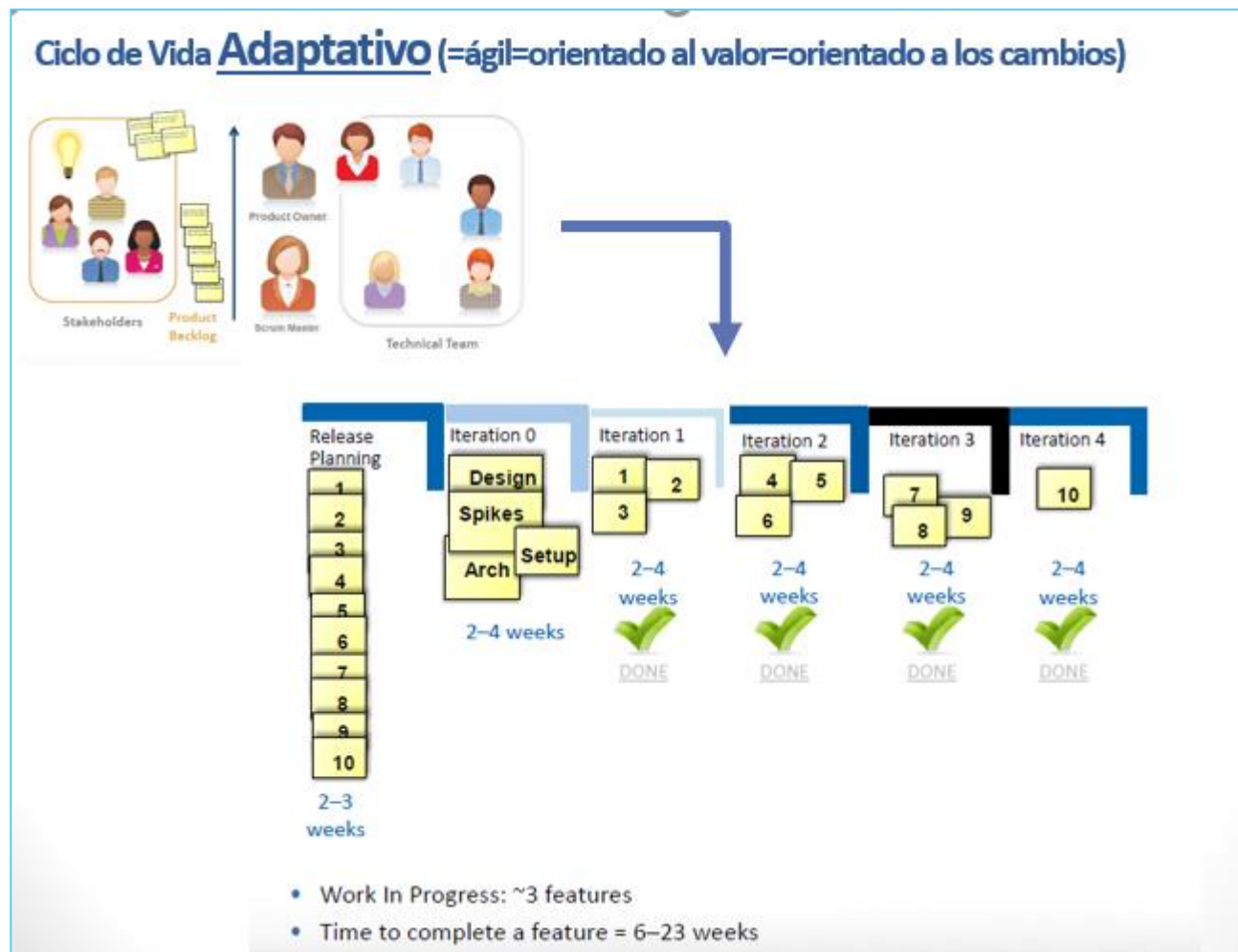
- Diseños - Criterios de aceptación - Escenarios



Ciclo de vida Predictivo contra adaptativo



Ciclo de vida Predictivo contra adaptativo



Tamaño, estimación, planificación

HISTORIAS DE USUARIO: Las historias de usuario son un pequeño trozo de funcionalidad de negocio dentro de una funcionalidad que puede llevar unos días (aprox...)

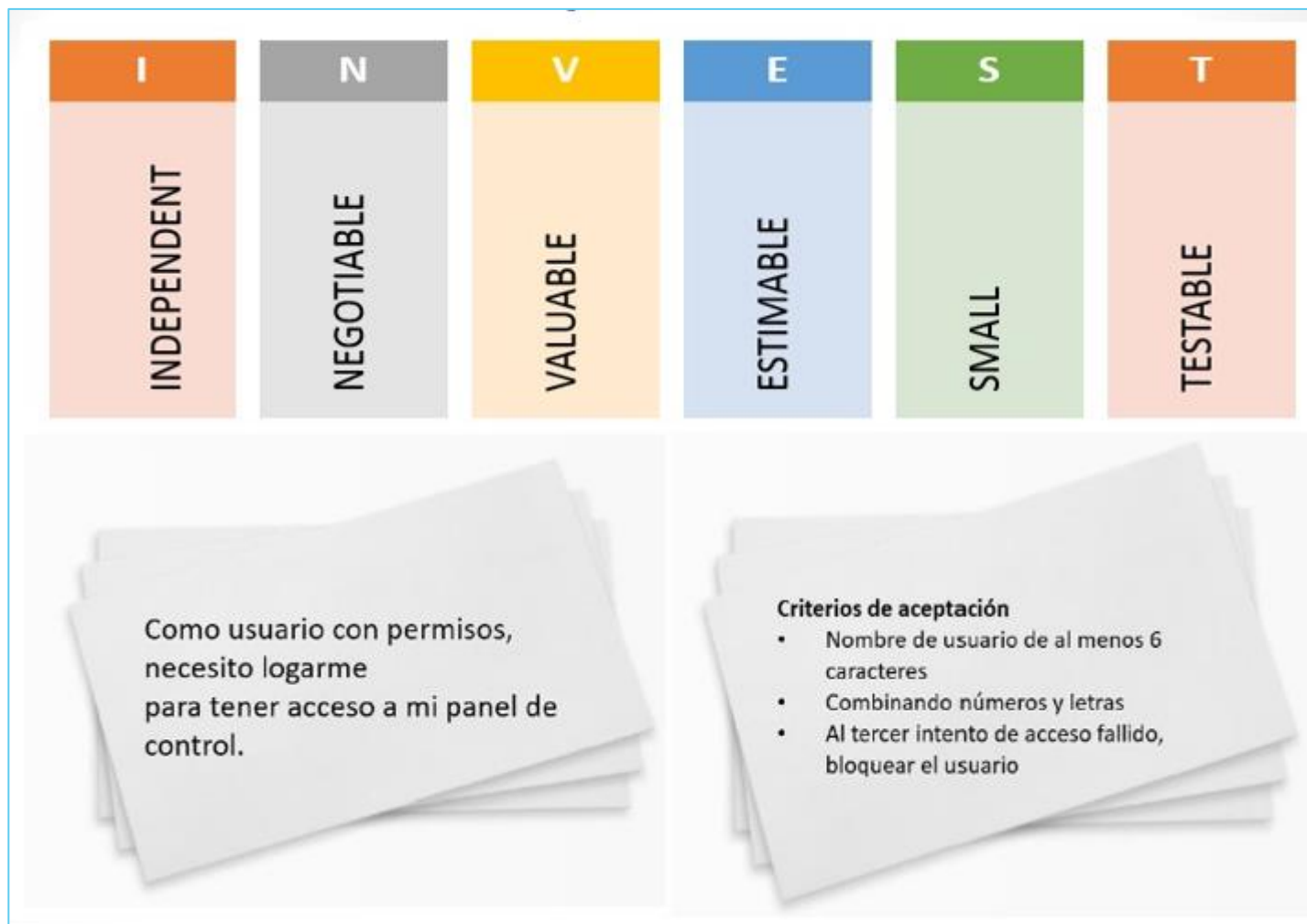


Tamaño, estimación, planificación

Cada historia de usuario debe incluir tres “Cs”:

- **Card (tarjeta):** Una historia de usuario tiene que caber en una tarjeta pequeña de cartón (tamaño cuartilla). El hecho de restringir el espacio físico para escribirla fuerza a redactarla de forma simple, atómica, resumida, y fácil de entender por cualquiera que la lea.
- **Conversación:** Una historia de usuario no es fruto del desarrollo de documentación, sino de conversación (cara a cara, si es posible) entre Negocio y los miembros del equipo de trabajo. La tarjeta de cartón o papel es el soporte físico sobre el que gira la conversación para definir el requisito.
- **Confirmación:** Una historia de usuario tiene que estar escrita de forma que pueda ser confirmada. Es decir, cuando una historia de usuario está finalizada y se presenta para su validación, tiene que estar claro cuáles son los criterios para confirmar si el requisito está satisfecho o no lo está.

Tamaño, estimación, planificación



Herramientas para estimaciones

PUNTOS DE HISTORIA: Los puntos de historia o puntos función son unidades de medida relativas, no absolutas, del tamaño de un requisito.

El equipo de trabajo asigna puntos de historia a cada requisito.

Factores a tener en cuenta:

Complejidad: una historia que es mas compleja que otra tendrá mas puntos historia.

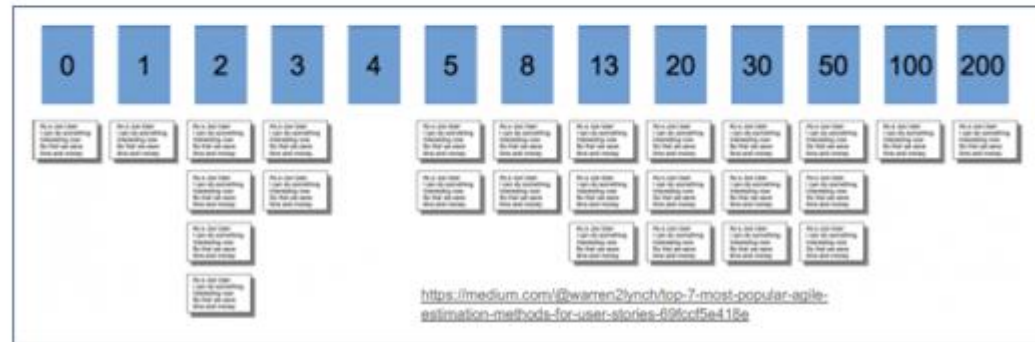
Esfuerzo: igual no tiene complejidad pero si requiere esfuerzo, también tendrá mas puntos de historia que otra mas corta.

Incertidumbre o riesgo: Es la probabilidad de que algo no esperado aparezca en la historia, siempre por falta de información.

Herramientas para estimaciones

Estimaciones de puntos de historia:

Estimaciones por afinidad agrupas elementos en categorías similares



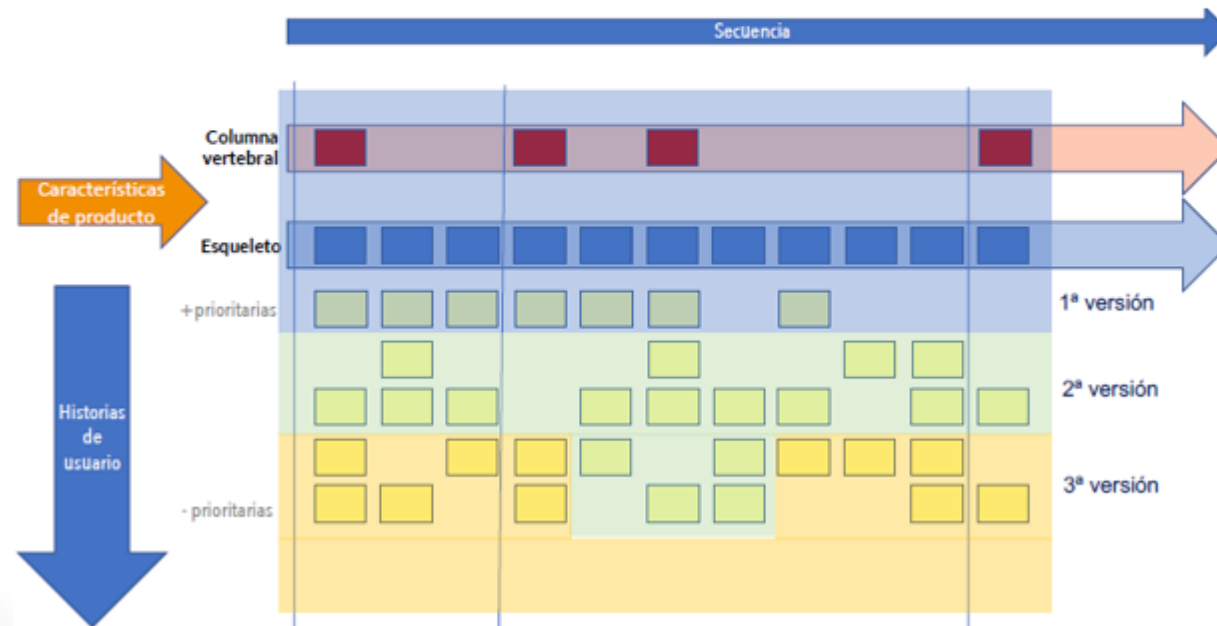
Tallas de camiseta estimación alto nivel.



Herramientas para estimaciones

Estimaciones de puntos de historia:

Mapas de historia mapear las prioridades del proyecto...basado en la información del proyecto. Descripción visual de las versiones del producto y los principales componentes que serán incluidas en cada una das versiones .



Metodologías ágiles

Aunque existen otras, hay 4 metodologías ágiles que se utilizan muy habitualmente: Scrum, XP, Kanban y Lean.



Método simple

- Iterativo e incremental
- 5 valores
- 3 roles
- 3 artefactos
- 5 tipos de reunión



- 13 prácticas de ingeniería
- 5 valores
- 4 actividades
- 4 roles



- Comenzar ahora con lo que hay que hacer
- Poner como objetivo pequeños cambios incrementales
- Cada miembro del equipo es un líder
- Mejorar de forma continua y colaborativa
- Todos los implicados ven el proceso claramente
- Limitar el trabajo en curso
- Hacer visible el trabajo
- Visualizar el flujo de trabajo
- Gestionar el flujo de trabajo



- Un enfoque, más que una metodología
- Eliminar el desperdicio
- Llevar la simplicidad al máximo

Scrum

Scrum es un método para solventar problemas complejos, entregando productos que aporten el mayor valor posible. Es una metodología:

- **Ligera:** Scrum tiene poca teoría, únicamente define algunas reuniones o ceremonias, los roles, y unos pocos principios básicos. El contenido teórico se lee en menos de 5 minutos.
- **Fácil de entender:** Es una metodología abierta, que no propone reglas complicadas ni demasiado específicas en función del proyecto.
- **Difícil de dominar:** La clave es adaptarla correctamente al entorno y al proyecto concreto. Por eso está definido el rol de Scrum Master, que es la figura que domina el método y ayuda a su aplicación y ajuste.

Está basada en **procesos empíricos de control**, es decir, el conocimiento viene de la experiencia, y se toman decisiones en función de la información que se tiene.

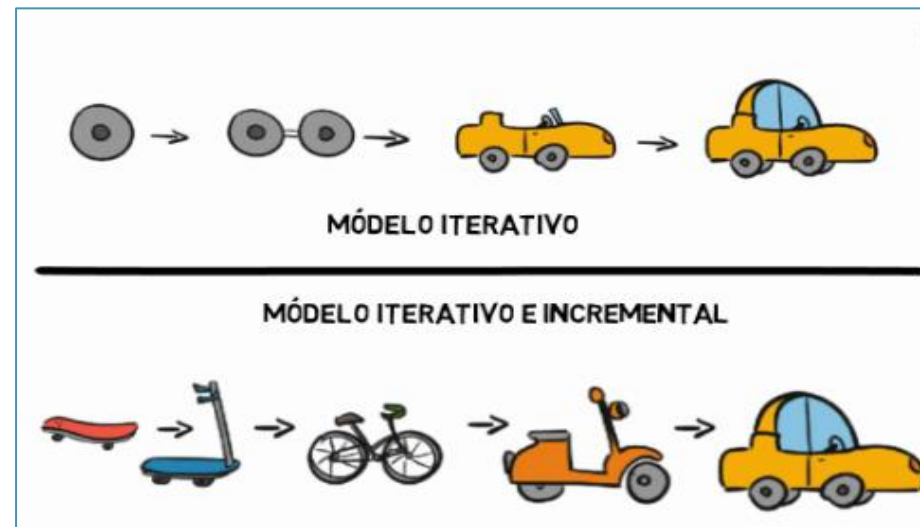


Scrum

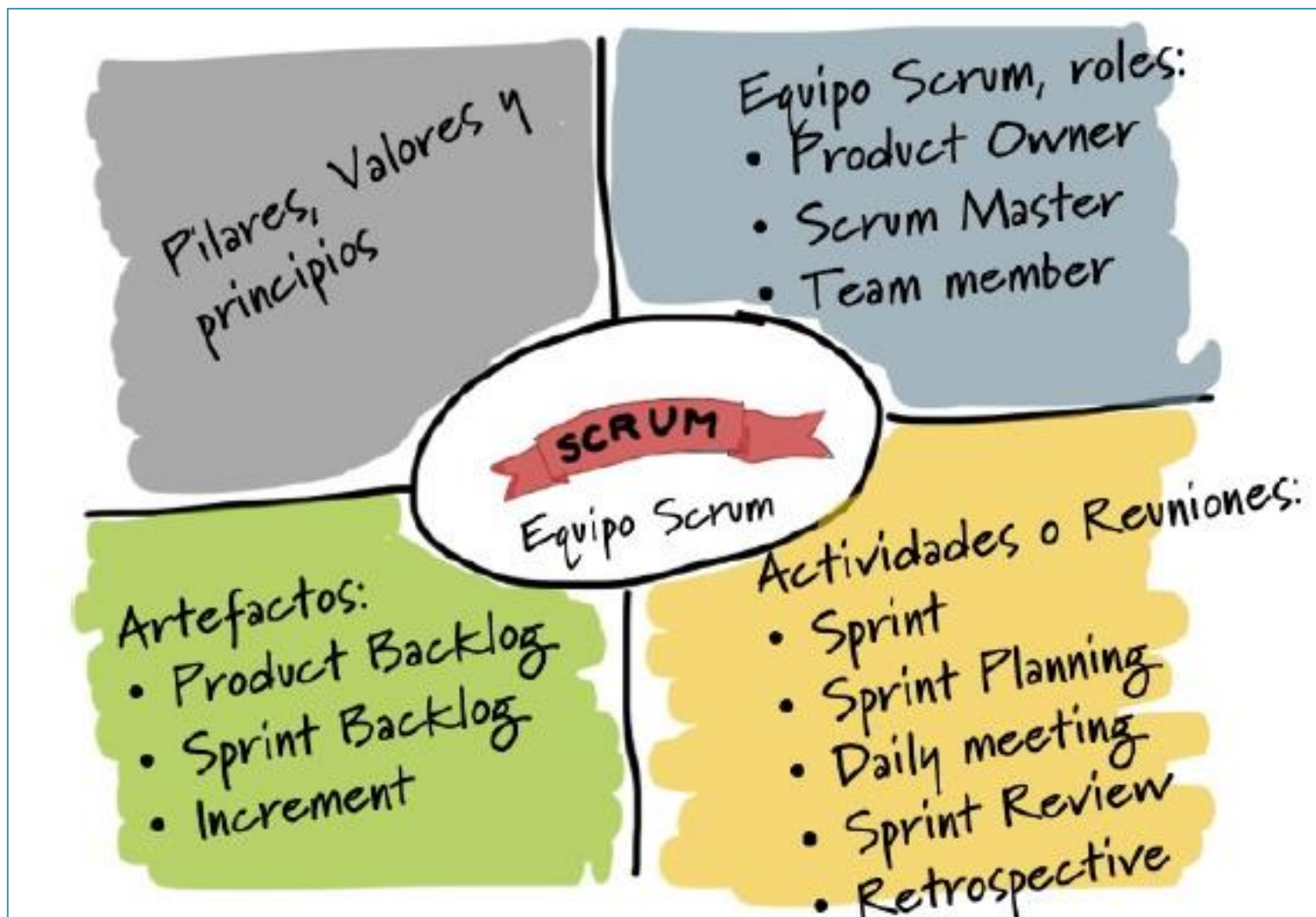
Su enfoque es iterativo e incremental.

Iterativo: En cada sprint, se genera una nueva versión del producto, que mejora la versión del sprint anterior. Se trata de ir refinando y mejorando las propiedades del producto conforme avanza el proyecto.

Incremental: En cada sprint, se añade alguna nueva característica al producto. Se trata de ir añadiendo nuevas capacidades o características al producto conforme avanza el proyecto.



¿Qué es Scrum?



Pilares de Scrum

Scrum se basa en tres pilares:



Pilares de Scrum

Transparencia

- Todos los aspectos relativos al proceso deben ser **visibles** a los responsables del resultado.
- Para ello se requiere un **estándar** y nomenclatura común a todo el equipo Scrum y los interesados del proyecto
- *Manteniendo visibilidad sobre el estado del proyecto*
- *Invitando a los interesados a las Sprints reviews*
- *Compartiendo una definición de lo que se considera hecho.*

Pilares de Scrum

Inspección

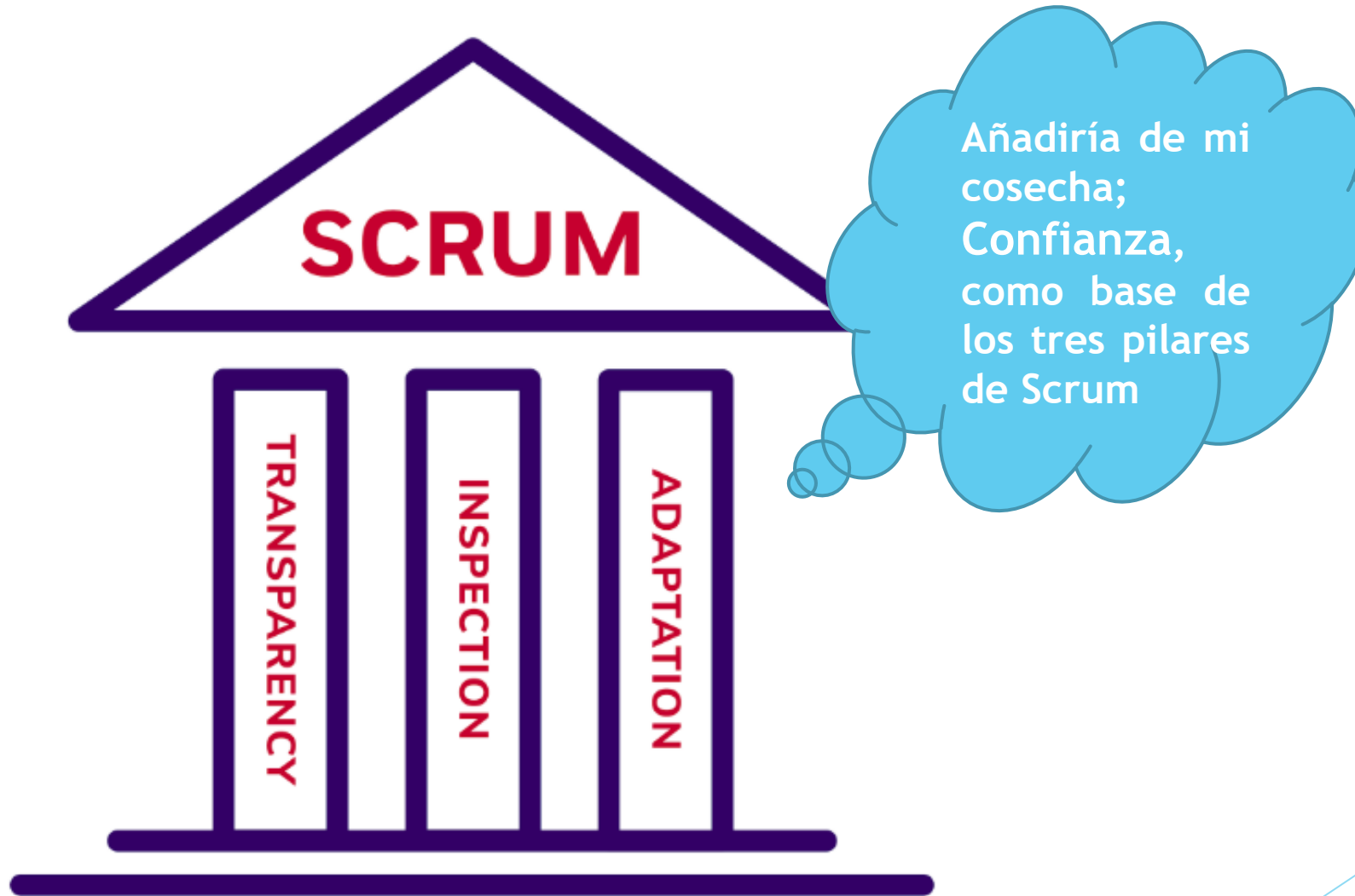
- Scrum promueve la inspección frecuente de artefactos y el proceso para **identificar y corregir variaciones no deseadas.**
- La inspección tiene lugar durante los **eventos o ceremonias del Sprint.**
- *Manteniendo gráficos de evolución del trabajo actualizados*
- *Realizando las reuniones diarias de sincronización, limitadas a 15 minutos y a las tres preguntas clave*

Pilares de Scrum

Adaptación

- Tras la inspección deben realizarse los **ajustes** al proceso, artefactos, comportamiento de las personas y a la interacción con el entorno, para **minimizar desviaciones adicionales**.
- *Redefiniendo objetivos del Sprint*
- *Re-priorizando trabajo*
- *Adaptando el trabajo*
- *Mejorando de forma continua el funcionamiento del equipo*

Pilares de Scrum



Valores de Scrum



Principios de Scrum

Divide tu organización en equipos pequeños, interdisciplinarios y auto-organizados.

Divide el trabajo en una lista de entregables pequeños y concretos. Ordena la lista por orden de prioridad y estima el esfuerzo relativo de cada elemento

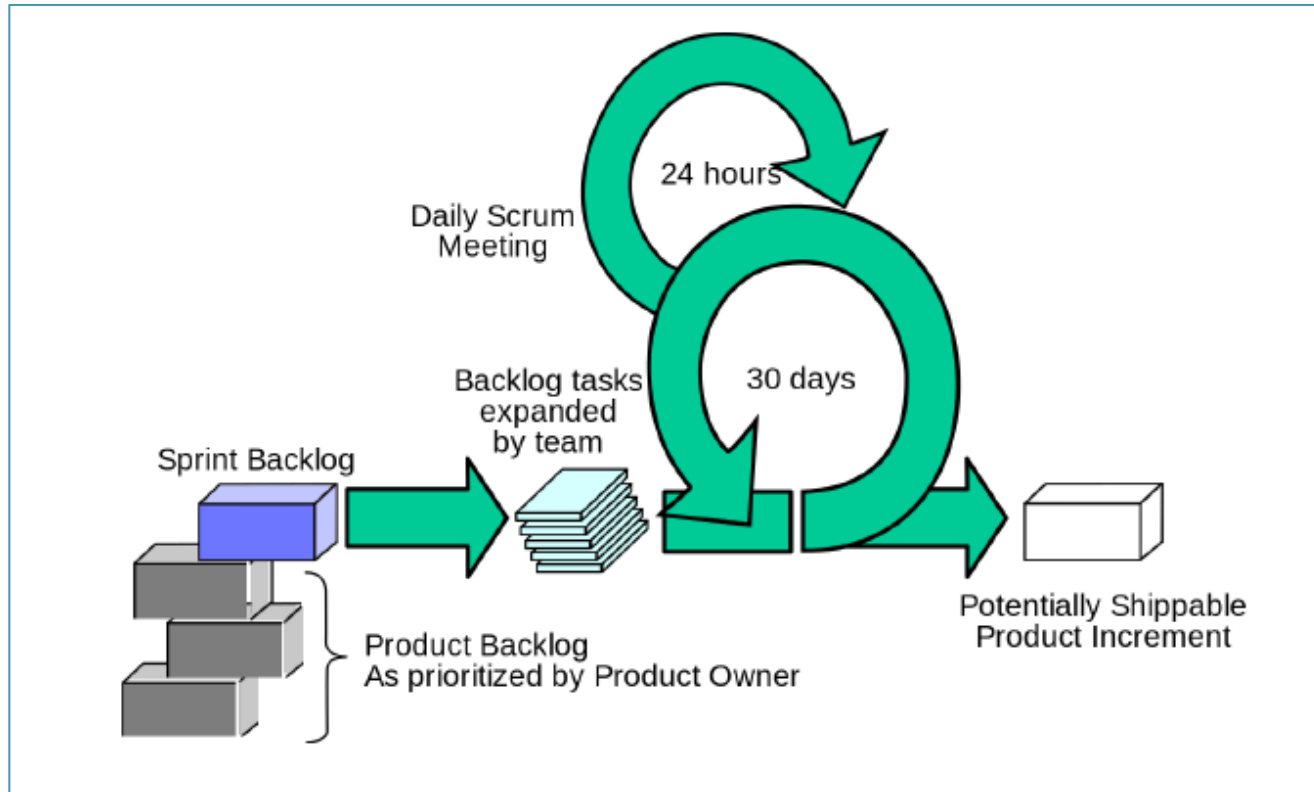
Divide en tiempo en iteraciones cortas de longitud fija, con sub-producto potencialmente entregable y demostrado después de cada iteración.

Optimiza el plan de entregas y actualiza las prioridades de la mano del cliente, basada en los conocimientos adquiridos mediante la inspección del entregable después de cada iteración.

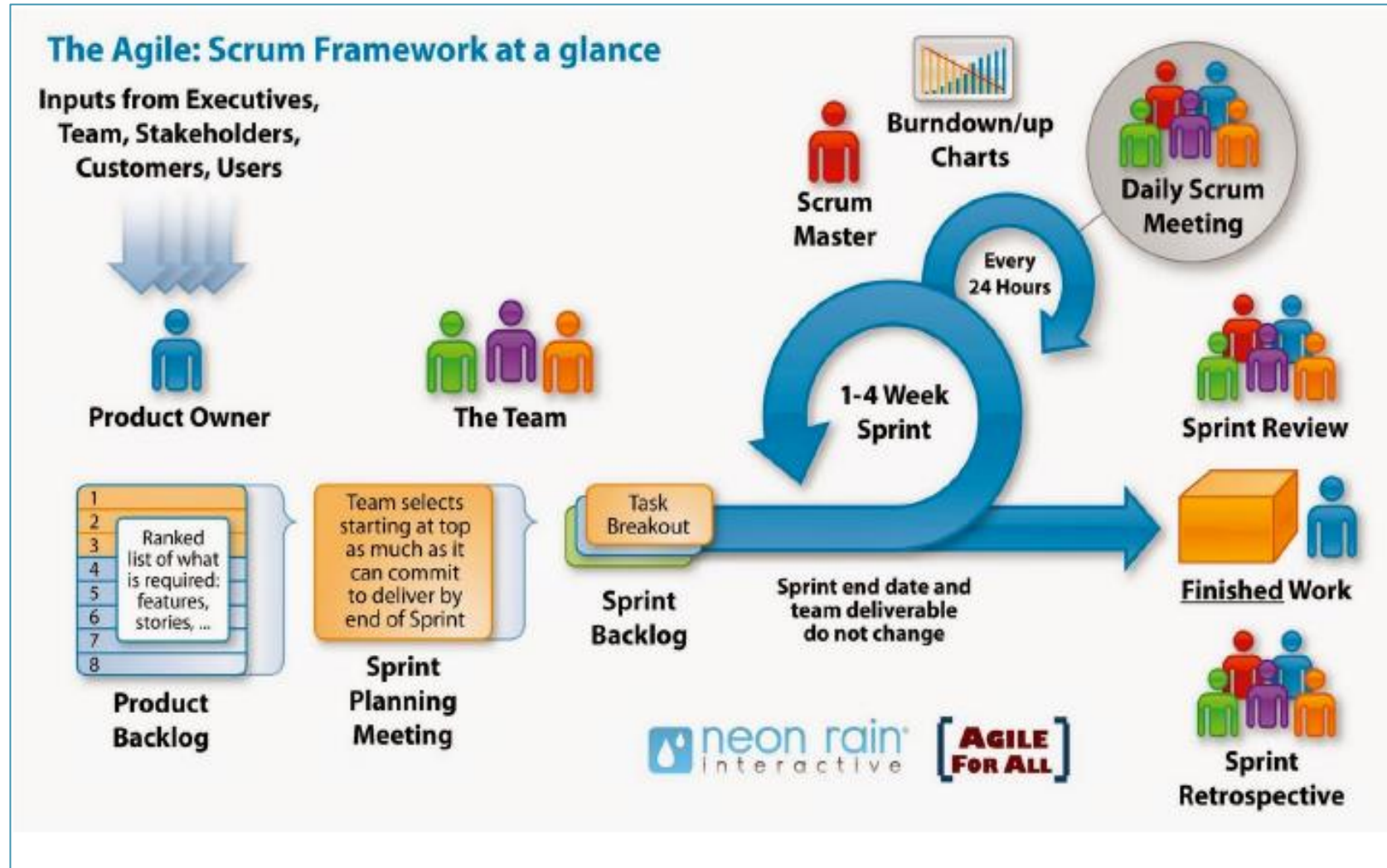
Optimiza el proceso teniendo en retrospectiva después de cada iteración.

Scrum - Ciclo de Scrum

El proyecto se ejecuta en base a **sprints**, de duración fija, que se planifican al arrancar cada sprint, con las Daily cada 24 horas. En cada sprint se resuelve o construye el Sprint backlog, que se integra al final del sprint con el resultado de sprints anteriores, conformando un producto entregable.



Scrum - Ciclo de Scrum



Scrum - Sprints

El producto se construye de forma incremental en base a períodos de tiempo cortos, denominados **Sprints**.

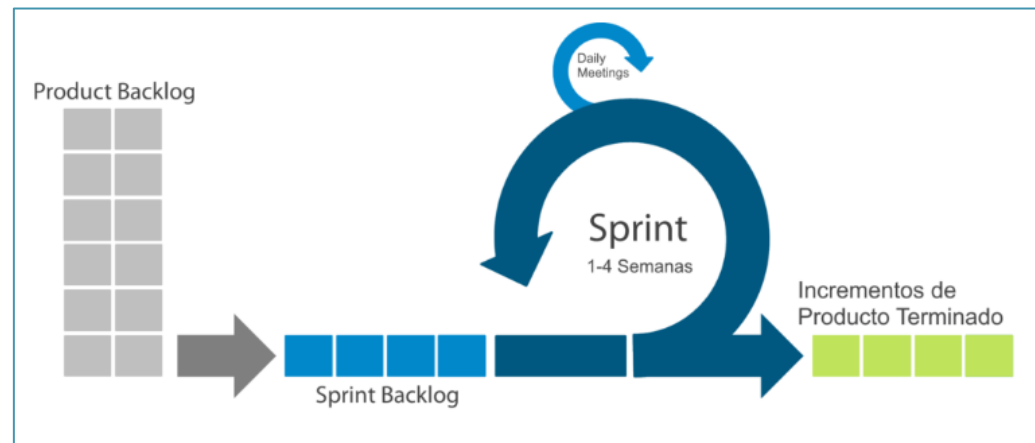
Los Sprints tienen una duración fija y determinada, entre 1 y 4 semanas; mejor cuanto más cortas, es decir mejor 1 semana que 4.

Todos los Sprints tienen la misma duración a lo largo del proyecto, porque se rigen según el principio de “**timeboxing**”: cada elemento tiene un tiempo asignado que termina cuando acaba este tiempo.

El equipo **construye un incremento** (parte de producto potencialmente entregable).

Un Sprint puede ser cancelado si el **objetivo se vuelve obsoleto**, por cambios en la **dirección de negocio** o por **condiciones de tecnología**.

Sólo el **Product Owner puede cancelar el Sprint**. El trabajo pendiente volverá al Product Backlog.



Scrum - Roles

En un proyecto ágil, típicamente bajo metodología Scrum, distinguimos 3 roles:

Product Owner (PO)

- Decide qué se incluye (y qué no) en el backlog del proyecto
- Ordena los ítems en el backlog en función de su prioridad de negocio
- Explica y hace entender al equipo de trabajo en qué consisten esos ítems (historias de usuario)
- Asegurarse de que equipo y Negocio tengas la misma visión de proyecto
- Centraliza la comunicación entre los stakeholders y el Scrum Team
- Establece en qué debe trabajar el equipo
- El Equipo no hace caso más en cuanto a qué hacer



Scrum - Roles

Scrum Master (SM)

- Es el responsable de asegurar que la metodología es entendida y aplicada con éxito
- Es un Servant Leader para el equipo de desarrollo:
 - Eliminando impedimentos
 - Facilitando eventos
 - Entrenando al equipo (auto-organización y multifuncionalidad)
- Colabora con el Producto Owner:
 - Ayudando a gestionar el producto backlog
 - Comunicando la visión, objetivos y detalle del Backlog
- Colabora con la organización:
 - Facilitando la adopción de Scrum
 - Ayudando a entender Scrum
 - Provocando cambios que aumenten la productividad
 - Colaborando con otros Scrum Master para implantar Scrum en la organización



Scrum - Roles

Equipo de desarrollo (DT)

- Es el grupo de profesionales que construye el incremento de producto en cada sprint
- Constituyen un equipo auto-organizado, están empoderados para gestionar su propio trabajo.
- Son responsables de decidir cómo se realiza el trabajo
- Constituyen un equipo multifuncional (cada miembro puede desempeñar más de un rol de los necesarios)
- No hay un gestor que asigna, delimita o coordina las tareas
- No hay sub-equipos.



Scrum - Otros roles

Patrocinador y resto de interesados

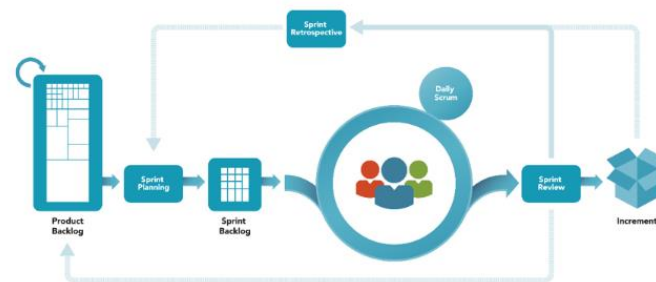
- Ayudar al PO a alinear los entregables con las necesidades globales de la organización
- Se involucran estrechamente con el Product Owner para definir las necesidades, impactos y resultados deseados.
- Asisten a las demostraciones del equipo y dan retroalimentación.
- Están accesibles para responder preguntas.
- Respetan las reglas Ágiles.
- Eliminan impedimentos.
- Reconocen los logros del equipo.



Scrum - Actividades (Eventos, reuniones o ceremonias)

El framework Scrum define 5 reuniones.

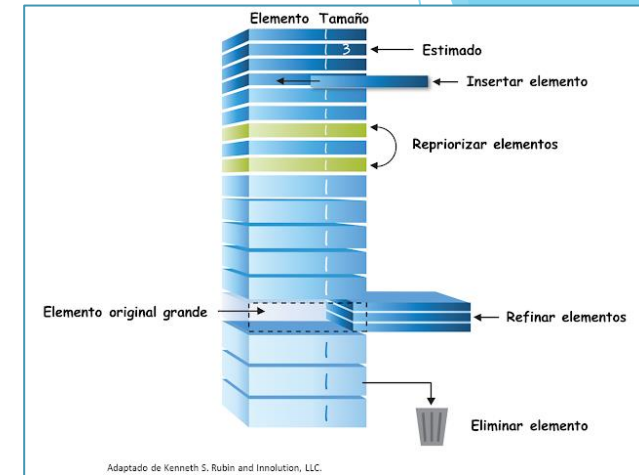
- **Refinado del Backlog:** Discutir y actualizar los elementos del Product
- **Planificación (Sprint Planning):** Planificación del trabajo a realizar en el sprint
- **Reunión diaria (Daily Scrum):** Reunión de seguimiento diario
- **Revisión (Sprint Review):** Revisión al final del Sprint con cliente, interesados y equipo para revisar el incremento del trabajo
- **Retrospectiva (Sprint Retrospective):** Actividad final de inspección y adaptación para recopilar aprendizaje e identificar oportunidades de mejora.



Scrum - Actividades (Eventos, reuniones o ceremonias)

Refinado del Backlog (Backlog Refinement)

- Reunión donde se refina el Backlog
- Participa todo el equipo Scrum
- En ella se discuten y actualizan todos los elementos del Backlog:
 - Proporcionando más detalle
 - Clarificando todo lo necesario (Criterios de aceptación, diagramas, ejemplos, información)
- Se realiza sobre la marcha, durante el Sprint. No tiene un momento específico.
- Algunos equipos la planifican a mitad de Sprint, lo que permite trabajar con elementos que serán candidatos para el siguiente Sprint.



Scrum - Actividades (Eventos, reuniones o ceremonias)

Planificación del Sprint (Sprint Planning meeting)

- Es la reunión donde se decide qué trabajo se realizará y cómo será realizado
- Participa todo el Equipo Scrum (PO, SM, DT)
- Pasos:
 - El PO presenta el backlog actualizado
 - Todo el grupo lo discute para comprobar que hay entendimiento común
 - El equipo de desarrollo estima cuanto trabajo podrá realizar (basado en capacidad, velocidad, sprints previos...)
 - Se define un objetivo del Sprint
 - El equipo define cómo realizarán el trabajo y se organizan para alcanzar el objetivo



Scrum - Actividades (Eventos, reuniones o ceremonias)

Scrum diario (Daily Scrum)

- Es una reunión diaria de 15 minutos (máximo) que tiene lugar en el mismo sitio a la misma hora
- Participa el equipo de desarrollo (apoyado por el SM)
- Objetivo: Sincronizar trabajo e impedimentos
- Cada miembro del equipo responde a tres preguntas:
 - ¿Qué hice ayer?
 - ¿Qué voy a hacer hoy?
 - ¿Qué impedimentos tengo o visualizo?
- Cualquier discusión técnica, o de resolución de problemas, debe ser aplazada para después, para mantener la daily en el timebox



Scrum - Actividades (Eventos, reuniones o ceremonias)

Revisión del Sprint (Sprint Review)

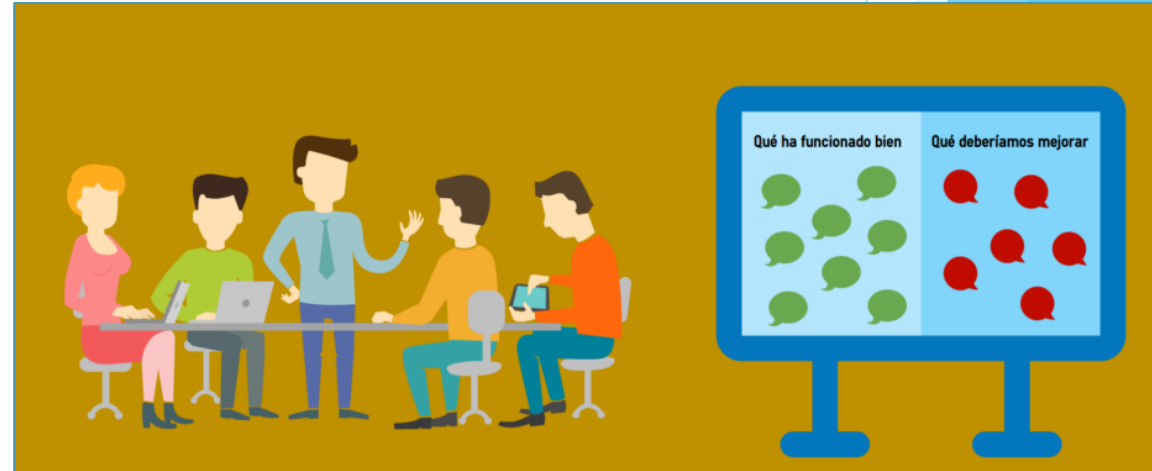
- Es una reunión al final del Sprint en la que se muestra el incremento de producto desarrollado
- Participa el equipo Scrum (PO SM, DT) y otros interesados (Clientes, interesados internos, ...etc.)
- Objetivo: Mostrar el trabajo realizado y decidir si está hecho, actualizando el backlog
- Útil para los interesados, comprueban el desarrollo de los requisitos, verifican expectativas, entienden mejor sus expectativas...
- Y útil para el equipo, comprueba si entendió las necesidades, que puntos mejorar, y puede mostrar sus resultados.



Características de los equipos de alto rendimiento

Retrospectiva del Sprint (Sprint Retrospective)

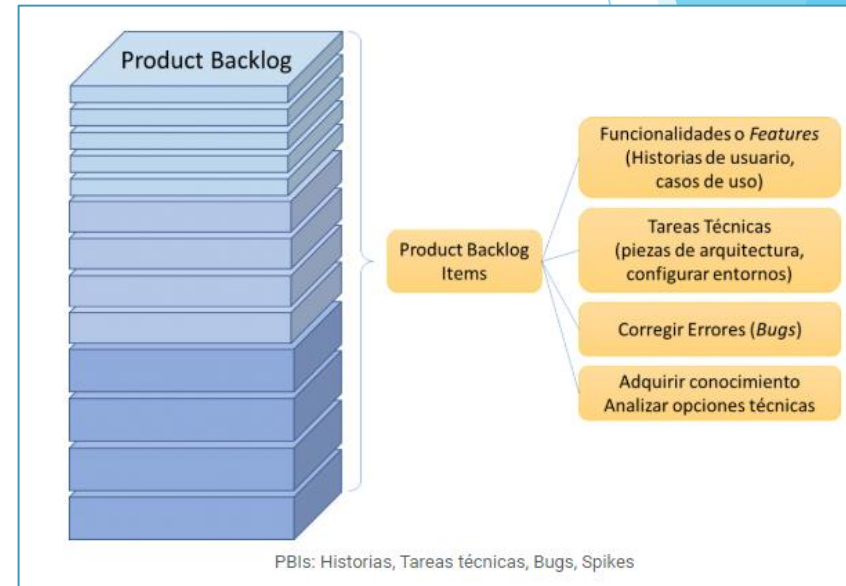
- Es una reunión después de la Sprint Review, pero antes de la siguiente Planning.
- Participa el equipo Scrum (PO, SM, DT)
- Objetivo: recopilar aprendizaje. Identificar oportunidades y acciones concretas de mejora
- Algunos aspectos clave:
 - Cubre cualquier situación ocurrida durante el sprint
 - Hay que tener en cuenta el feedback recibido durante la Sprint Review
 - El equipo decide qué acciones de mejora va a implantar a partir del siguiente Sprint



Scrum - Artefactos

Pila del producto (Product Backlog)

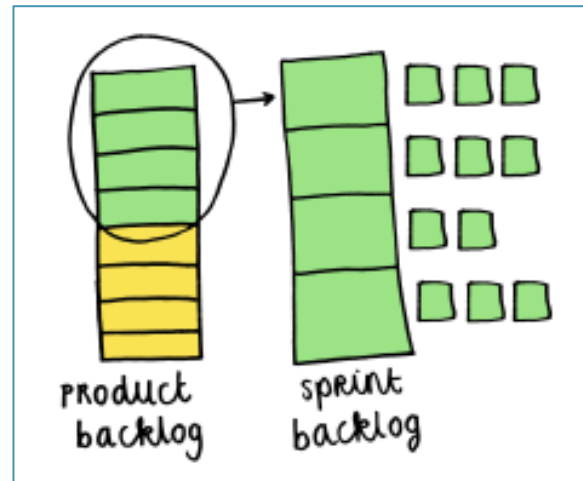
- Es una Lista priorizada de todo el trabajo que necesita ser hecho para construir el producto
- Puede incluir: Características, funciones, requisitos, atributos de calidad, mejoras y reparaciones
- Es una lista dinámica que va variando
- Lo mas importante está en la parte de arriba, que se realizará lo primero
- A media que se avanza, se va detallando el backlog
- Los elementos con prioridad mas baja, podrían no ser desarrollados, ya que pueden ser pospuestos por otros de mayos prioridad
- El proceso de dar mas detalle se realiza en el refinado del Backlog



Scrum - Artefactos

Pila del Sprint (Sprint Backlog)

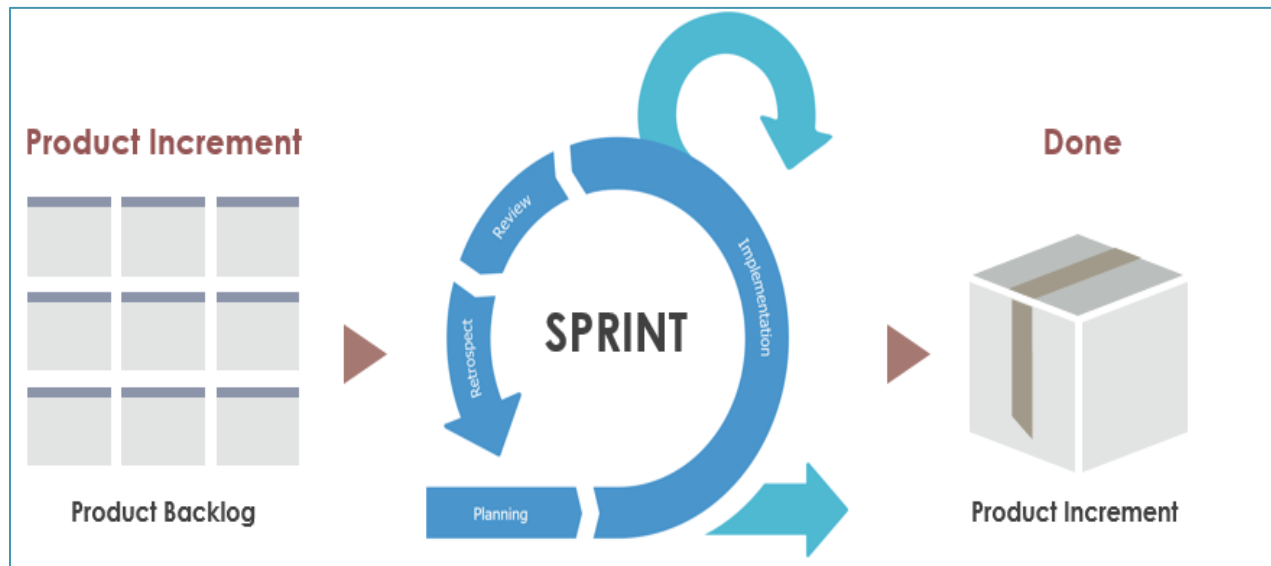
- Es un conjunto de elementos del backlog, que han sido seleccionados para realizar el Sprint vigente
- A lo largo del Sprint, el equipo desarrolla el plan de cómo alcanzará el objetivo del sprint
- Sirve como una lista de alta visibilidad del trabajo que está siendo realizado
- Sólo puede ser actualizado por el equipo de desarrollo



Scrum - Artefactos

Incremento del producto (Product Increment)

- Es la parte del producto construido por el equipo durante el sprint
- Se va acumulando a lo construido anteriormente
- El equipo obtiene feedback sobre el incremento en la Sprint Review y tratan de conseguir que el elemento esté hecho (Done).



Scrum - Definición de hecho (Done)

El equipo de trabajo tiene que encontrar una definición para el concepto de "hecho" -Done-. Cada incremento del producto debe cumplir dicha definición de "hecho" para darlo por finalizado, y poder ser entregado.

NOTA: La definición de "hecho" puede aplicar a requisitos, sprints, releases, entornos... es decir, en cualquier elemento sobre el que se pueda plantear la cuestión de "¿está finalizado y puede continuar al paso siguiente del proyecto?" Son las condiciones para considerar el elemento terminado con éxito.



Veamos un ejemplo de "Hecho" para un requisito (habitualmente denominadas "historias de usuario"):

- Todo el código escrito sin errores
- Todas las pruebas unitarias pasadas correctamente
- Pruebas funcionales pasadas con éxito
- Documentación del requisito generada
- Pruebas de aceptación pasadas con éxito

Ejemplo de "Hecho" para un Sprint:

- Pruebas de rendimiento pasadas con éxito
- Elementos resultantes del sprint integrados
- Todos los bugs resueltos
- Pruebas de integración pasadas con éxito

Ejemplo de "Hecho" para el despliegue en Producción:

- Pruebas de stress pasadas con éxito
- Pruebas de seguridad validadas
- Plan de marcha atrás definido
- Entorno y servicio estable

Scrum - A recordar

1. Scrum es muy sencillo de aprender pero muy difícil de practicar porque exige un cambio de mentalidad y de comportamientos.
2. Scrum tiene tres roles: Scrum Master, Product Owner y el equipo.
3. Scrum tiene 4 reuniones: planificación del Sprint, scrum diario, revisión del sprint y retrospectiva.
4. Scrum tiene 3 artefactos: pila de producto, pila del sprint e incrementos de producto.
5. Scrum persigue alta visibilidad, colaboración y mejora continua, trabajando en periodos fijos de tiempo de llamados sprints (típicamente de 2 semanas de duración).
6. Durante el scrum diario, solo puede hablar el equipo, no pueden hablar los interesados. Todos deben estar de pie y responder las tres preguntas.
7. El tamaño típico de un equipo Scrum es de 7 ± 2 .
8. El equipo Scrum tiene la última palabra sobre lo que puede hacerse durante un sprint.
9. El Product Owner es responsable de la priorización del producto backlog. Cualquiera puede solicitar que se añada un nuevo elemento, pero siempre a través del Product Owner.
10. Cada sprint debe producir funcionalidades "potencialmente usables".

Scrum - A recordar

Scrum

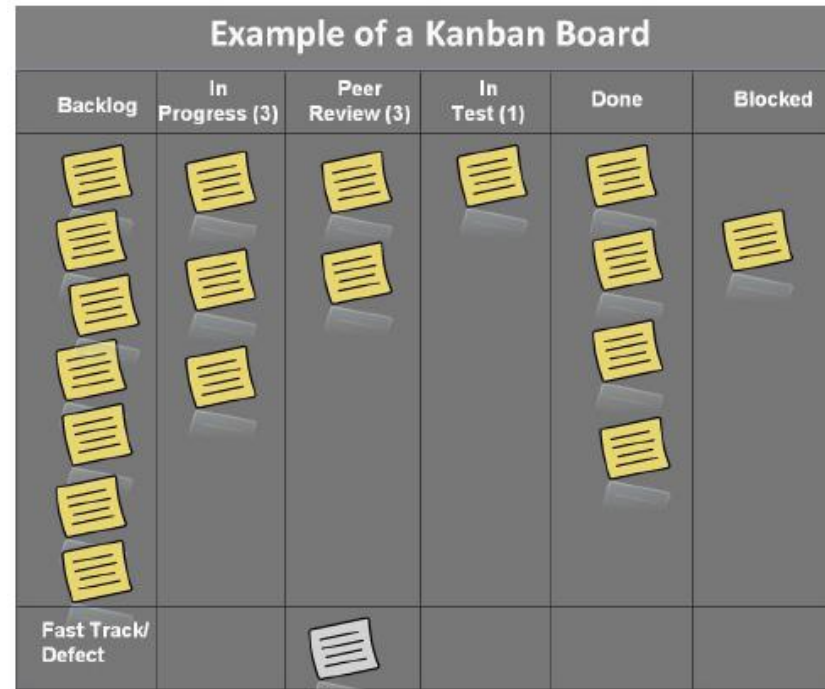
- **3 Roles**
 - Product Owner
 - Scrum Master
 - Agile Team
- **3 Artefactos**
 - Product Backlog
 - Sprint Backlog
 - Increment
- **4 Ceremonias**
 - Sprint Planning
 - Daily Standup
 - Sprint Demo
 - Retrospective
- **Management 3.0**
 - Equipos auto-organizados
 - 5-9 personas
 - Holocracia
- **Separar**
 - Cliente: aceptar
 - Analista (Product Owner): ¿qué?
 - Técnicos (Agile Team): ¿cómo?
- **Manifiesto Ágil**
 - Personas > Procesos
 - Productos > Documentos
 - Colaboración > Negociación
 - Cambiar > Seguir el Plan

Kanban

Se utiliza un "Kanban" para buscar cambios de estado en el trabajo, que reflejen el progreso y el trabajo en curso.

Es un método derivado del sistema de producción Lean desarrollado por TOYOTA.

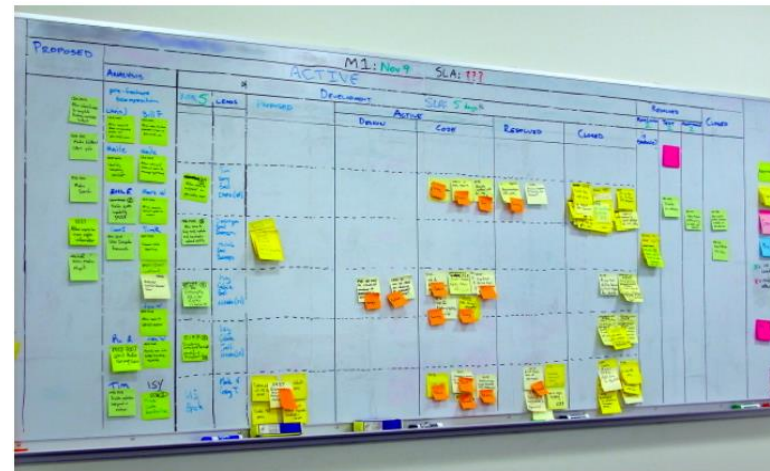
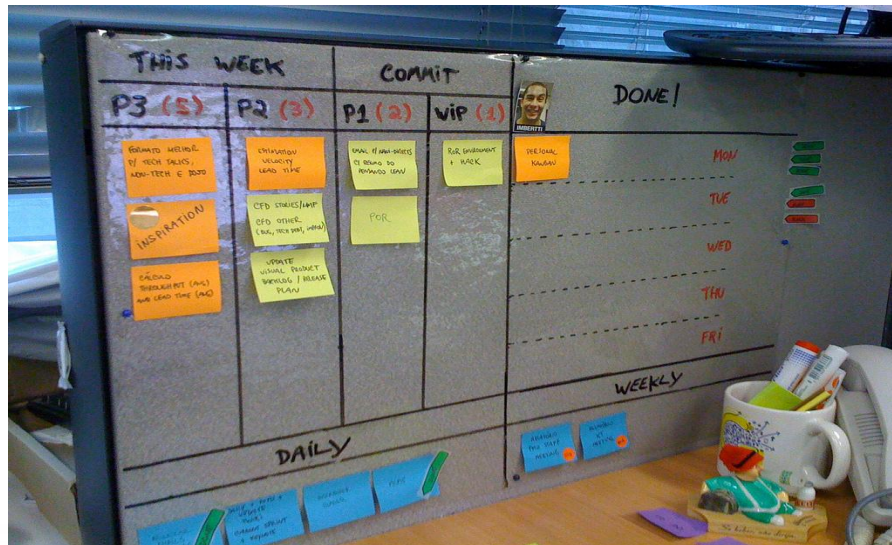
Esta metodología consiste en la organización del trabajo diario en base a un panel de tareas como el siguiente:



Kanban - ¿Qué es?

Pool of Ideas	Feature Preparation		Feature Selected	User Story Identified	User Story Preparation		User Story Development		Feature Acceptance		Deployment	Delivered
	In Progress	Ready			In Progress	Ready (Done)	In Progress	Ready (Done)	In Progress	Ready		
Epic 431			2-5	30							5	Epic 294
Epic 478	Epic 444	Epic 662					Story 602-02	Story 602-03	Story 602-04	Epic 401	Epic 609	Epic 694
Epic 562	Epic 589		Epic 602				Story 602-01	Story 602-02	Story 602-03	Epic 468	Epic 577	Epic 386
Epic 439	Epic 651		Epic 302		Story 302-01	Story 302-02	Story 302-03	Story 302-04		Epic 362		Epic 419
Epic 329			Epic 335		Story 335-01	Story 335-02	Story 335-03	Story 335-04				Epic 388
Epic 287			Epic 512		Story 512-01	Story 512-02	Story 512-03	Story 512-04				Epic 521
Epic 606	Discarded											Epic 287
	Epic 511	Epic 213										Epic 274
	Epic 221											

- Policy**
Business case showing value, cost of delay, size estimate and design outline.
- Policy**
Selection at Replenishment meeting chaired by Product Director.
- Policy**
Small, well-understood, testable, agreed with PD & Team
- Policy**
As per "Definition of Done" (see...)
- Policy**
Risk assessed per Continuous Deployment policy (see...)



PERÚ Ministerio de Educación



Universidad de Vigo



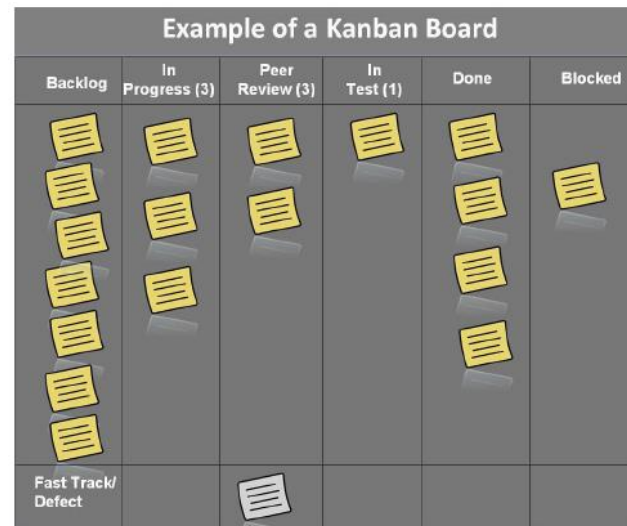
Kanban - 6 Prácticas

1. **Visualizar el flujo:** para poder organizarlo, optimizarlo y monitorizarlo
2. **Limitar el trabajo en curso** (W.I.P. Work in progress)
3. **Gestionar el flujo** (tratando de acelerarlo al máximo)
4. **Hacer explícitas las políticas** (acordadas por el equipo)
5. **Circuitos de retroalimentación**
6. **Mejorar colaborando y evolucionar experimentando**

Kanban - 6 Prácticas

1. Visualizar el flujo

En los proyectos de conocimiento, el trabajo es intangible e invisible. Visualizar el flujo es necesario para visibilizarlo, organizarlo, optimizarlo y seguirlo.



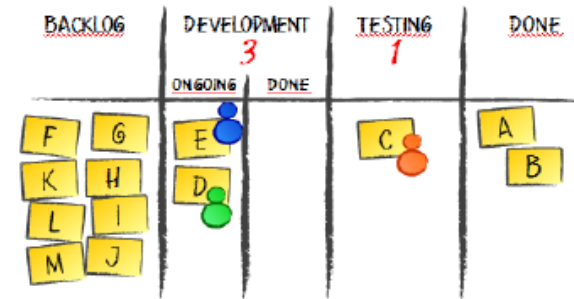
Kanban - 6 Prácticas

2. Limitar el trabajo en curso WIP

Restringir la cantidad de trabajo en curso:

- Mejor la productividad
- Reduce el tiempo de entrega
- Reduce los cuellos de botella
- Facilita la mejora continua

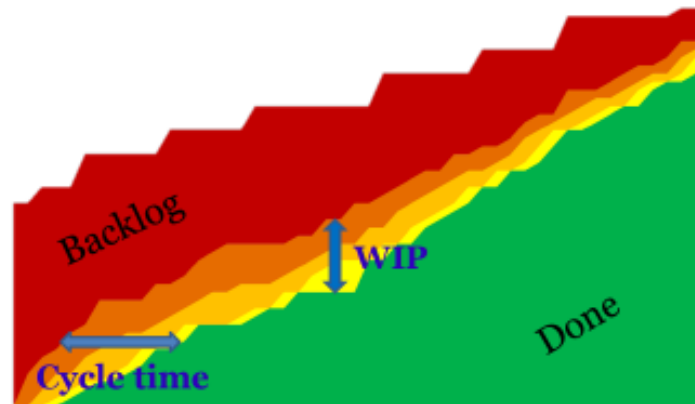
Pasamos de un sistema Push a otro Pull



Kanban - 6 Prácticas

3. Gestionar el flujo

Monitorizando el flujo de trabajo las incidencias son detectadas y los cambios son medidos para ver si efectividad



Kanban - 6 Prácticas

4. Hacer las políticas explícitas

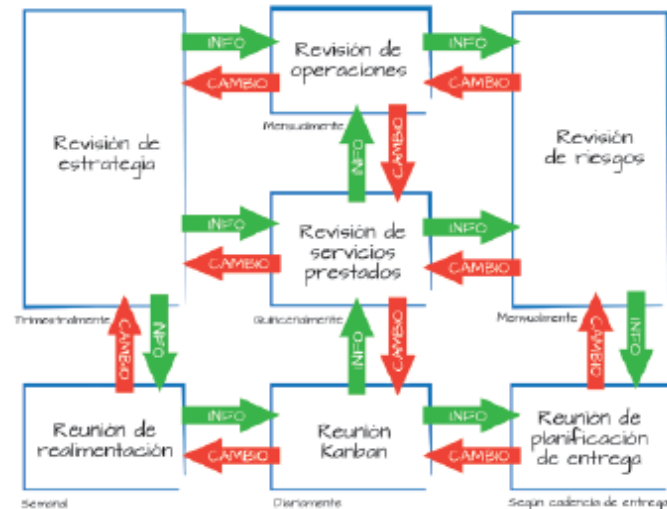
Para que quede claramente explicado como las cosas funcionan en el equipo y que se puedan mejorar. Las políticas las deciden aquellos cuyo trabajo le impacta en el tablero.



Kanban - 6 Prácticas

5. Circuitos de retroalimentación

Kanban define una serie de reuniones para gestionar el flujo de trabajo, ya sea a nivel organización, programa o equipo.



Kanban - sistema PULL

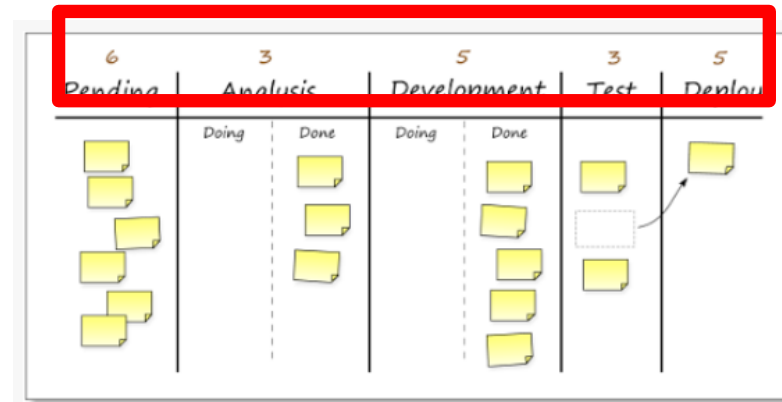
En un tablero Kanban de Scrum, el trabajo se mueve de izq. A der., pero no se limita el WIP. Es Push.

Kanban es un sistema PULL - cada vez que se completa una tarea esta dispara un aviso de “pull” y varias tareas pasan al siguiente estado , si están ya terminadas.

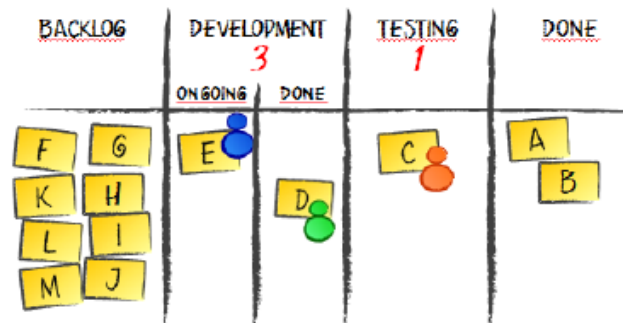
En cada columna hay un numero determinado de slots libres

Kanban - Limites W.I.P.

- Consiste en limitar cada estado del flujo (columnas) cuantos elementos puede haber



- Cuando el limite es alcanzado, no pueden moverse nuevos elementos a una columna hasta que no se libera el WIP



Ejemplo:

La tarea D no puede pasar a Testing hasta que C no esté terminada.

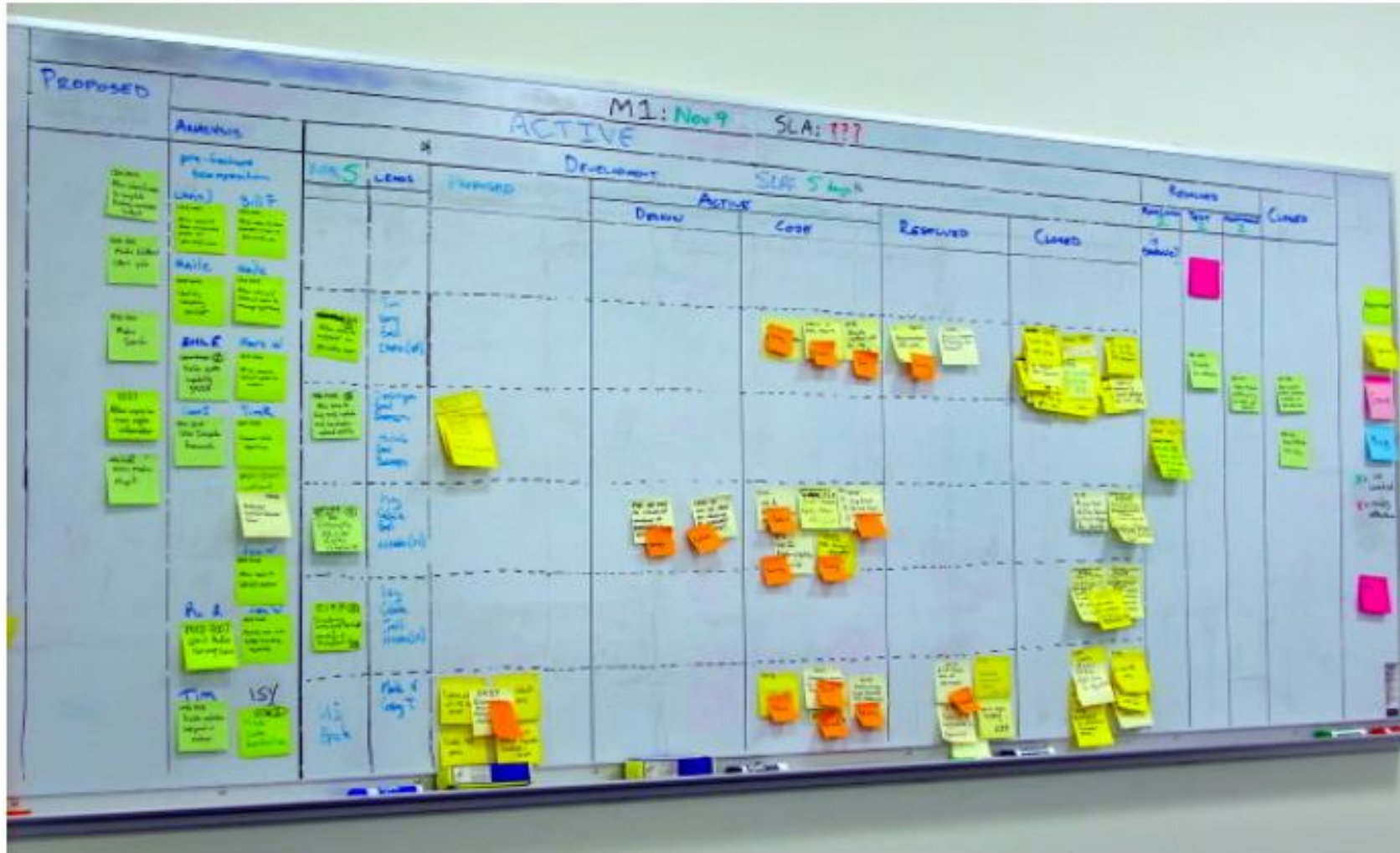
Kanban



Kanban



Kanban



Gracias

