

EDI 1 UX / UI

Experiencia de Usuario / Interfaz de Usuario

Año 2024 - Profesor: Federico Carrizo

ISFT N°93

Clase 4: Metodologías de Desarrollo de Software (Parte 2)

En esta presentación pasada, vimos metodologías de desarrollo de software que están ambientadas para un desarrollo con fines UX. En esta presentación vamos a clasificar las metodologías para conocer las innovaciones en el desarrollo del sw.

Metodologías de Desarrollo de SW

¿Por qué existen? La respuesta es fácil:

EL SOFTWARE TIENE UN CICLO

Todo software es creado a partir de una necesidad y actualizado a raíz de nuevos requerimientos, en el medio de estos pasos, tenemos análisis, diseño, codificación, pruebas y post-mantenimiento.

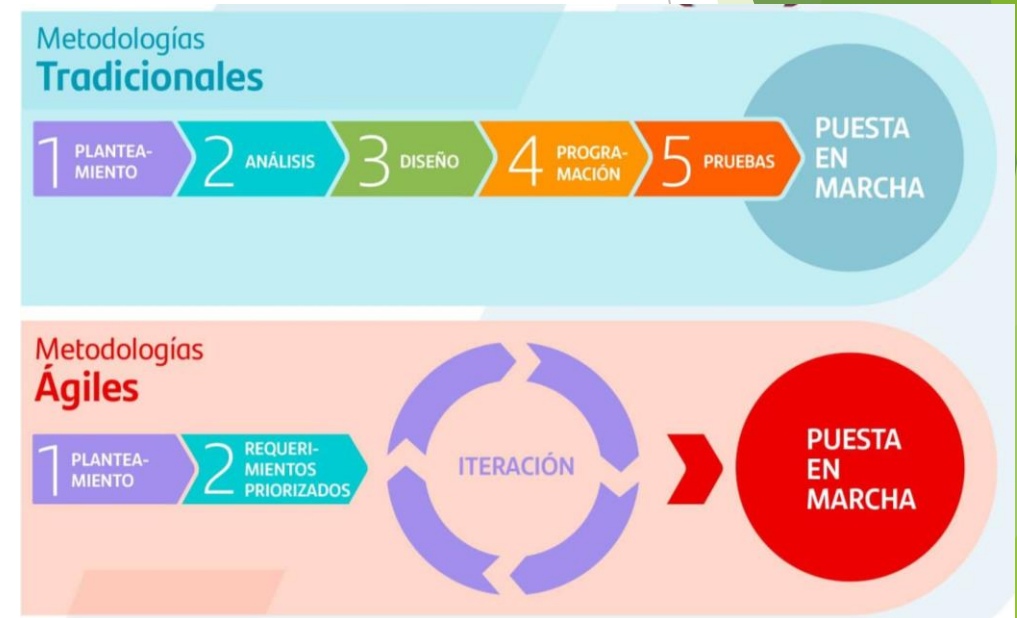


Metodologías de Desarrollo de SW

Las MDS funcionan como guías de trabajo, que te indica como, cuando y donde debes de llevar a cabo una determinada acción dentro de tu proyecto, con el objetivo de optimizar el proceso de desarrollo.

Existen varios tipos, cada MDS está diseñada para un tipo de proyecto en específico. Su capacidad para afrontar diversos retos, no es garantía de que será adecuada para utilizarse en cualquier proyecto

Definir una metodología de desarrollo de software para tu proyecto, te garantiza un control absoluto del proceso. Así como, un producto de calidad, bien organizado y apegado a sus requerimientos.

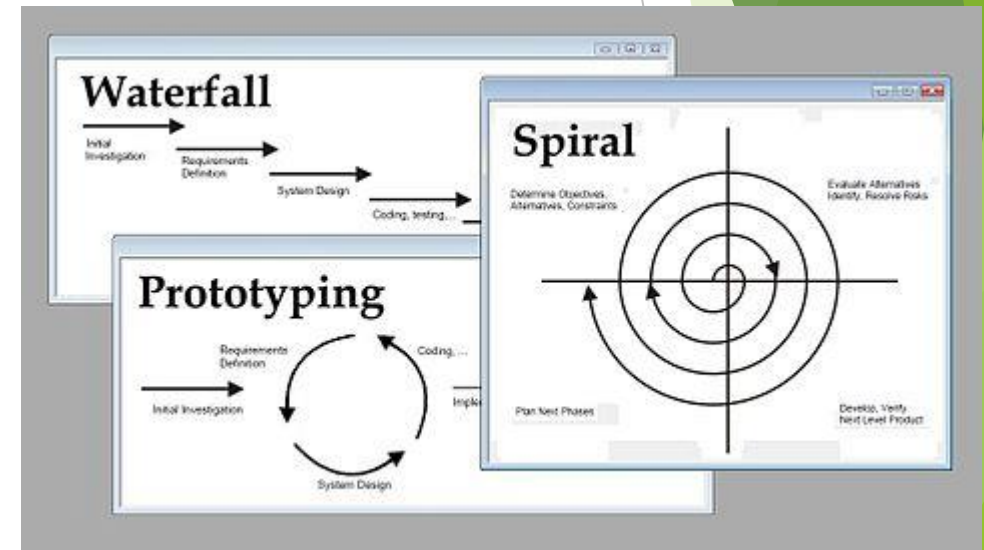


Tipos de MDS: Tradicionales

Las MDS tradicionales se caracterizan por definir total y rígidamente los requisitos al inicio de los proyectos de ingeniería de software. Los ciclos de desarrollo son poco flexibles y no permiten realizar cambios, al contrario que las metodologías ágiles; lo que ha propiciado el incremento en el uso de las segundas.

La organización del trabajo de las metodologías tradicionales es lineal, es decir, las etapas se suceden una tras otra y no se puede empezar la siguiente sin terminar la anterior. Tampoco se puede volver hacia atrás una vez se ha cambiado de etapa. Estas metodologías, no se adaptan nada bien a los cambios, y el mundo actual cambia constantemente...

Las principales metodologías tradicionales o clásicas son:

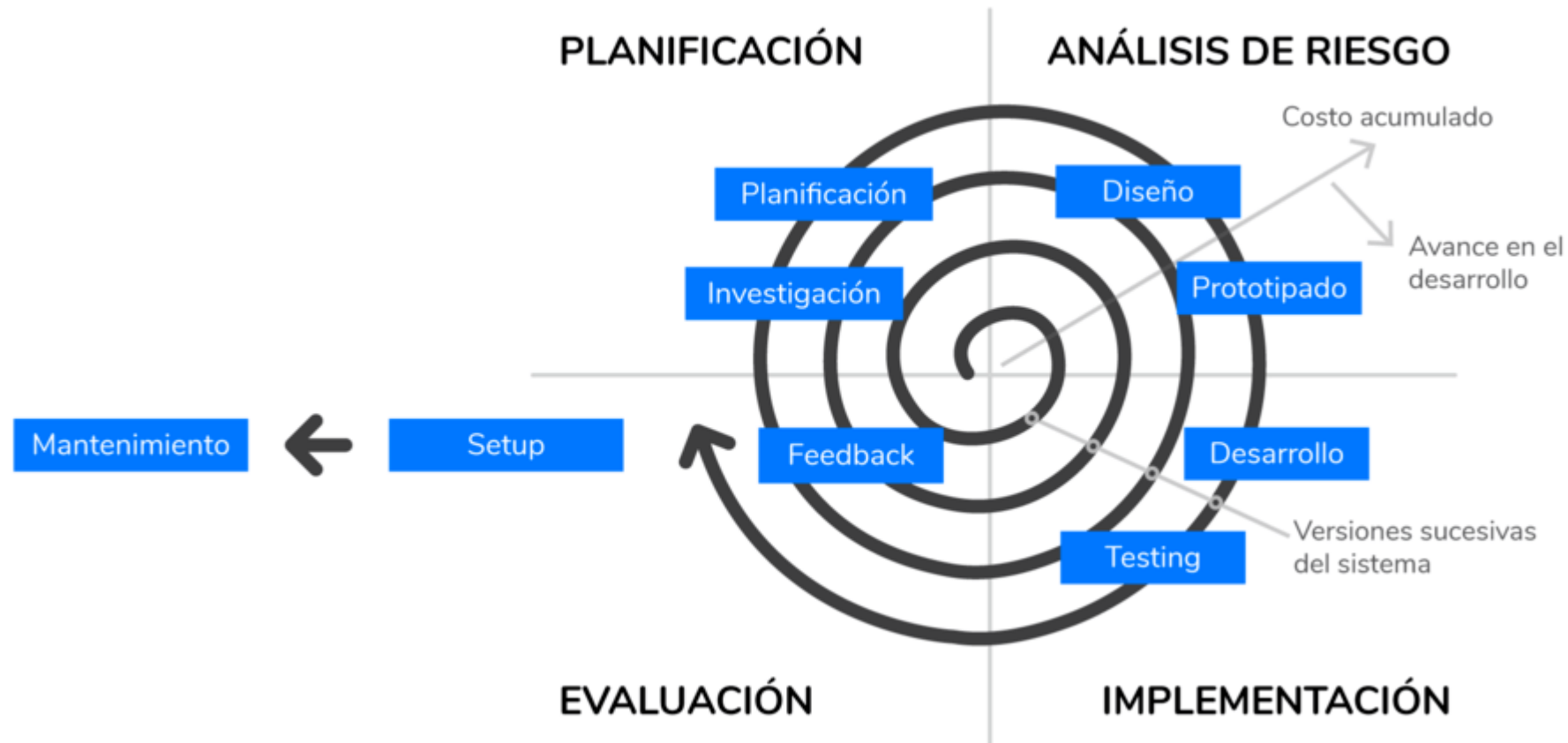


Tipos de MDS: Tradicionales



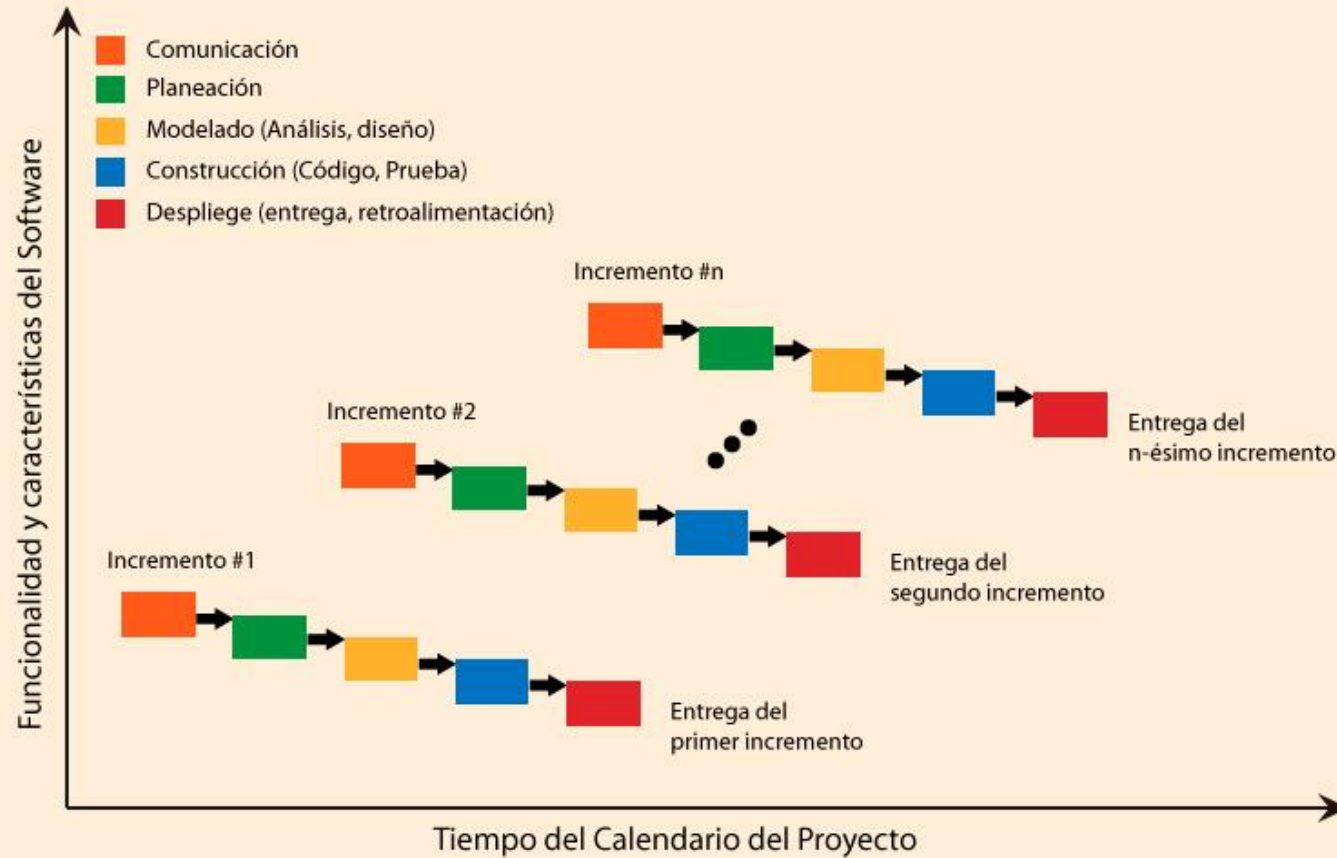
MODELO CASCADA (Royce 1970)

Tipos de MDS: Tradicionales



Tipos de MDS: Tradicionales

Modelo Incremental

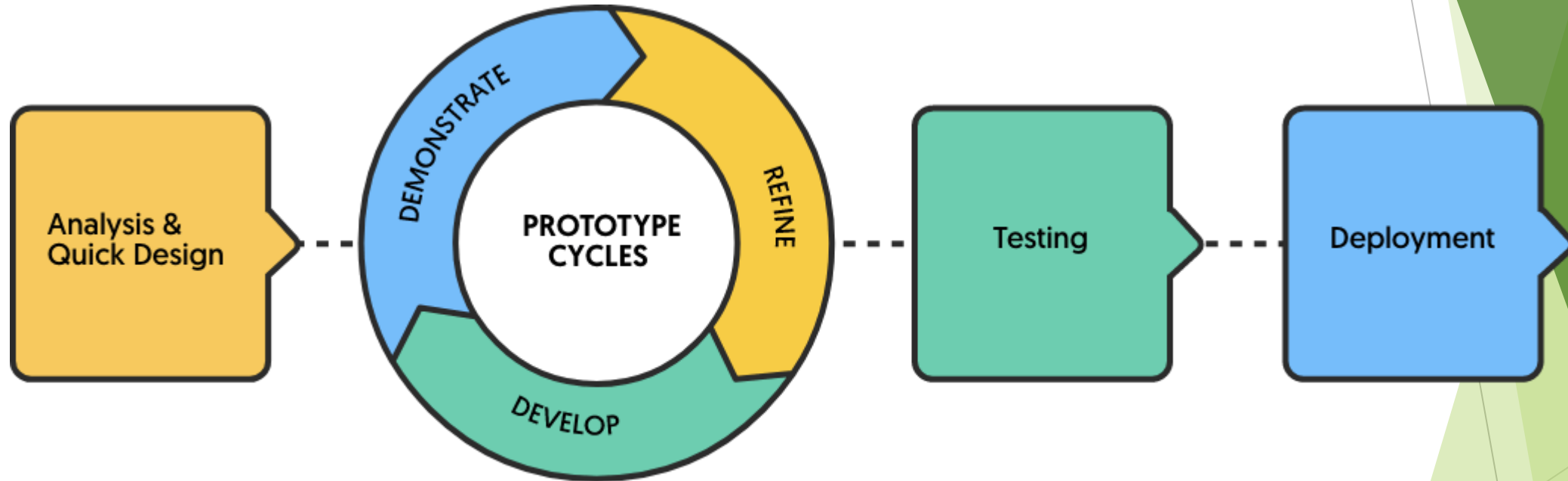


Tipos de MDS: Tradicionales

MODELO PROTOTIPOS



Tipos de MDS: Tradicionales



MODELO DESARROLLO RÁPIDO DE APLICACIONES
(DRA)

Tipos de MDS: ÁGILES

A diferencia de las metodologías anteriores, las metodologías ágiles se basan en un enfoque incremental e iterativo, donde las fases de la ejecución del proyecto son flexibles, evolucionan y a menudo se superponen entre sí.

En lugar de una planificación en profundidad al comienzo del proyecto, las metodologías ágiles están abiertas a requisitos cambiantes a lo largo del tiempo y fomentan la retroalimentación constante de los usuarios finales.

El objetivo de cada interacción es generar un producto que funcione, lo cual lo hace una elección más adecuada para proyectos en los que el cliente no está seguro del resultado deseado, busca un tiempo de entrega rápido y quiere participar de cerca en el proceso de diseño.



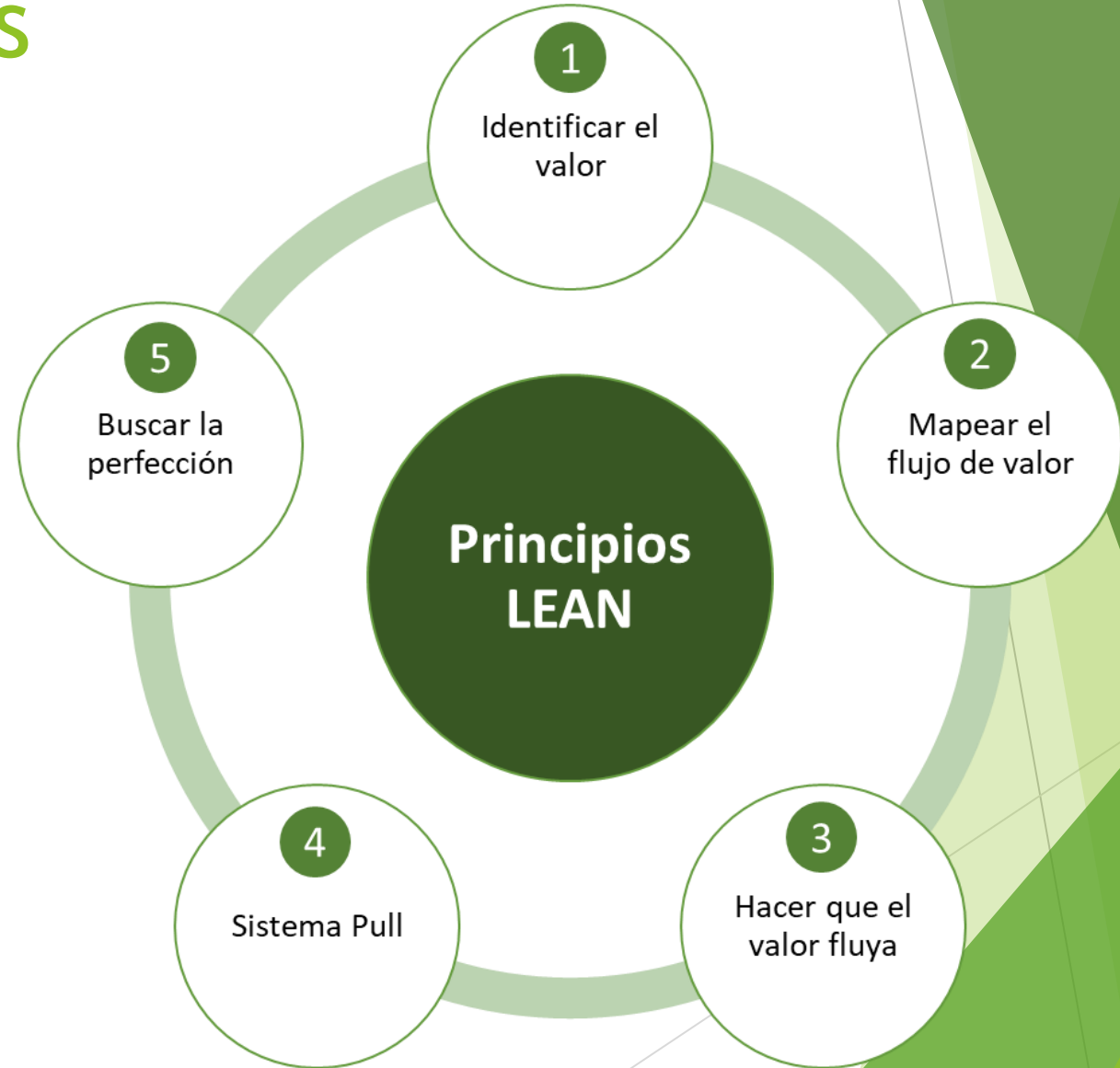
Tipos de MDS: Tradicionales vs Ágiles

Metodologías ágiles	Metodologías tradicionales
Están preparadas para cambios durante el proyecto	Son poco flexibles a los cambios
Proceso menos controlado, con pocos principios	Proceso mucho más controlado, con numerosas normas
No existe contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado
El cliente es parte activa en el proceso de desarrollo	El cliente interactúa con el equipo solo mediante reuniones de entregas
Grupos pequeños, 10 integrantes o menos y trabajando en el mismo sitio en el cual todos tienen conocimiento sobre todo el proceso de desarrollo	Grupos grandes y posiblemente distribuidos donde a cada integrante se le asignan tareas específicas
Menos énfasis en la arquitectura del software	La arquitectura del software es esencial y se expresa mediante modelos

Tipos de MDS: Ágiles

Lean Development

Fue creada por Bob Charette en 2003, a raíz de los conocimientos adquiridos al trabajar en el sector automovilístico japonés durante la década de los 80, pero se dio a conocer en 2003 mediante el libro “Lean Software development: An Agile Toolkit” de los aurores Mary y Tom Poppendieck, donde se establecieron las bases de la aplicación de esta metodología.



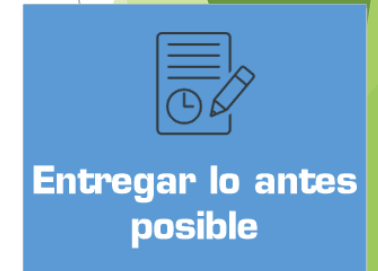
Tipos de MDS: Ágiles

La metodología Lean en el desarrollo de software es una traducción de los principios y las prácticas de la forma de producir lean, hacia el área del desarrollo de software.

Inicialmente, originado en el Sistema de Producción de Toyota y ahora, apoyado por una corriente que está surgiendo desde la comunidad Ágil.

El desarrollo de software Lean es un conjunto de principios que se pueden aplicar al desarrollo de software los podemos resumir en 7:

- Eliminar desperdicios/restos
- Amplificar el aprendizaje
- Tomar decisiones tardías
- Entregar lo antes posible
- Potenciar el equipo
- Crear la integridad
- Visualizar todo el conjunto



Tipos de MDS: Ágiles

Kanban

Actualmente, el término Kanban ha pasado a formar parte de las llamadas metodologías ágiles, cuyo objetivo es gestionar de manera general cómo se van completando las tareas. Kanban es una palabra japonesa que significa “tarjetas visuales”, donde Kan es “visual”, y Ban corresponde a “tarjeta”.

Inicialmente, surgió como un sistema de programación para la fabricación JIT, antes mencionada, originado por el Sistema de Producción Toyota (TPS). A finales de los 40, Toyota introdujo este sistema en su producción.



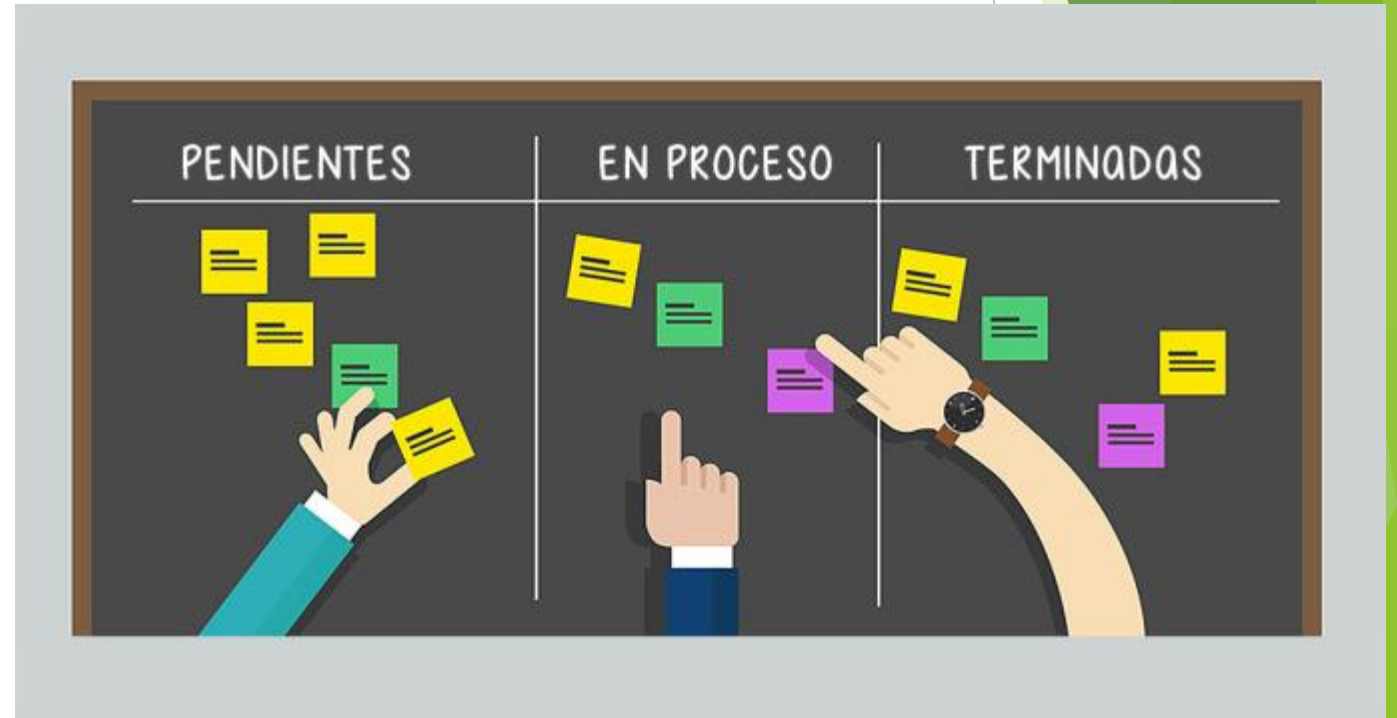
Tipos de MDS: Ágiles

Kanban

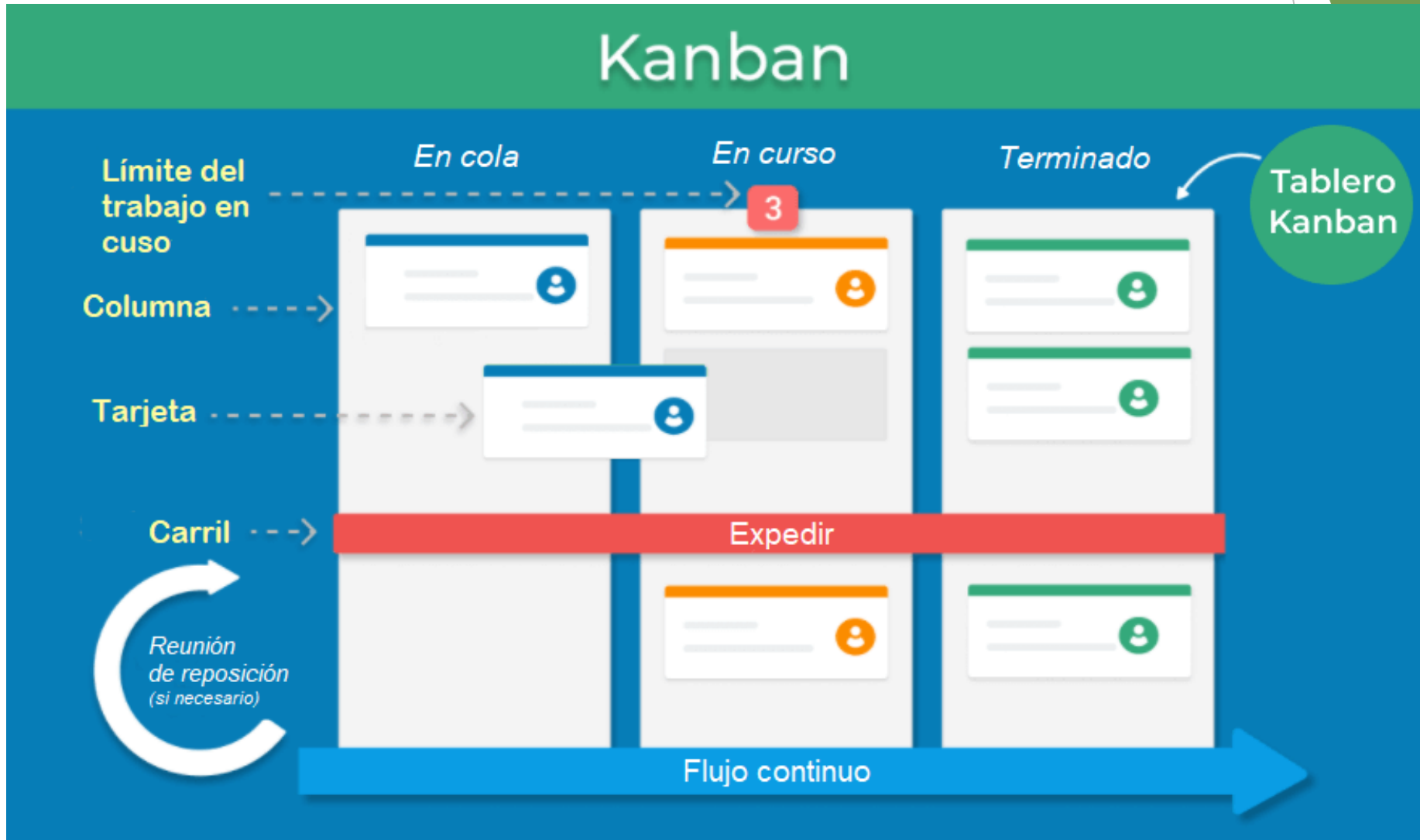
Las principales ventajas de esta metodología es que es muy fácil de utilizar, actualizar y asumir por parte del equipo. Además, destaca por ser una técnica de gestión de las tareas muy visual, que permite ver a golpe de vista el estado de los proyectos, así como también pautar el desarrollo del trabajo de manera efectiva.

Sus características son:

- Visualizar el flujo de trabajo
- Limitar el trabajo en curso
- Gestionar el flujo
- Explicitar las políticas de procesos
- Aplicar bucles de retroalimentación
- Mejorar en colaboración



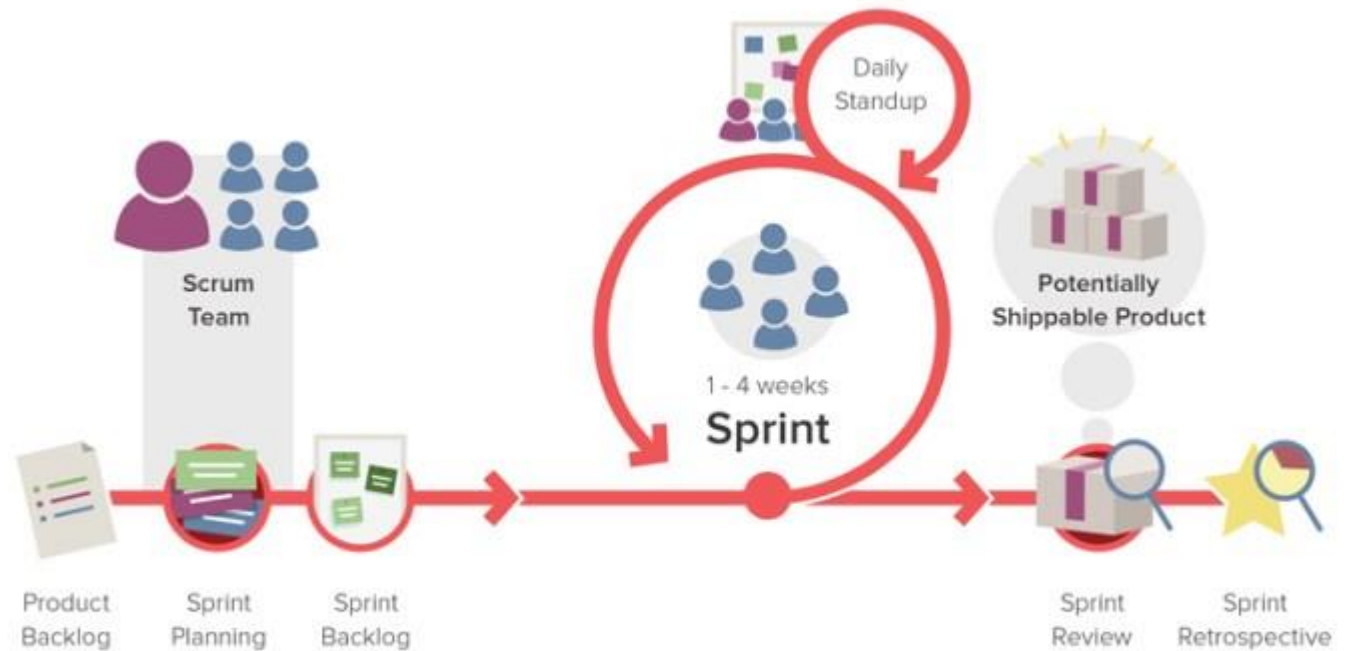
Tipos de MDS: Ágiles



Tipos de MDS: Ágiles

SCRUM

La metodología Scrum es un marco de trabajo ágil que permite abordar proyectos complejos en entornos dinámicos y cambiantes. Se basa en entregas parciales y regulares del producto final en base al valor que ofrecen a los clientes. Se trabaja con equipos pequeños multidisciplinarios que tienen un alto poder de decisión y se centran en la iteración y la mejora continua. El proceso se divide en ciclos cortos llamados sprints, que se planifican y ejecutan en reuniones periódicas



Tipos de MDS: Ágiles

SCRUM

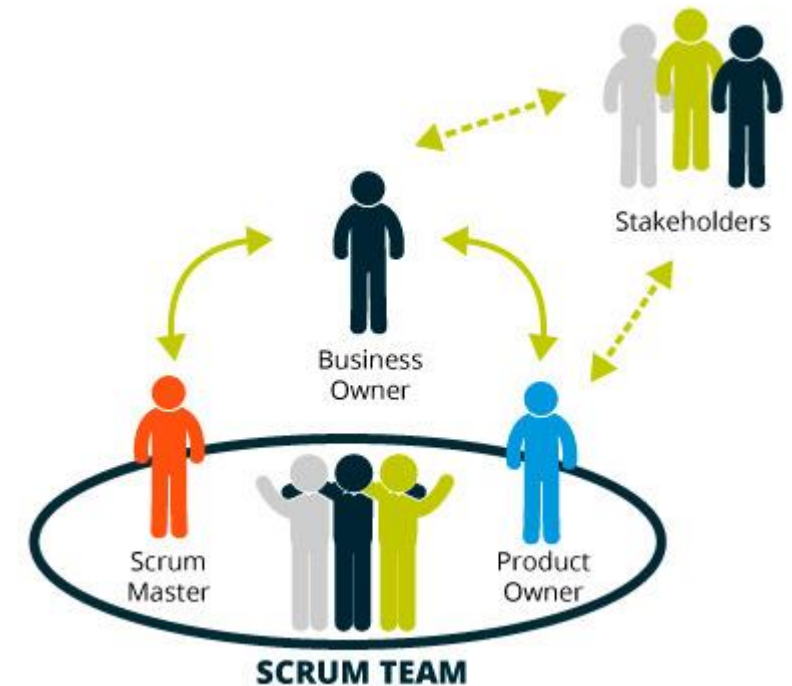
Este método no se puede llevar a cabo sin un equipo de trabajo:

Tenemos al PRODUCT OWNER que representa la voz del cliente y el resto de los interesados, pero no están implicados directamente en el proyecto. Este perfil se encarga de definir los objetivos del proyecto y de garantizar que el equipo trabaje de la mejor manera, para que los resultados finales sean los más óptimos.

SCRUM MASTER es otro perfil que contribuye a que el equipo se mantenga siempre muy activo y productivo.

SCRUM TEAM es el equipo encargado de desarrollar y entregar el producto. Su trabajo es muy importante, porque estamos hablando de una estructura horizontal auto-organizada capaz de autogestionarse.

STAKEHOLDERS este grupo comprende aquellos perfiles que están interesados sólo en el producto, hacen parte de este grupo: Los directores, los dueños, comerciantes. Aunque no formen parte del SCRUM TEAM, deben de todas formas ser tenidos en cuenta.



Tipos de MDS: Ágiles

SCRUM

Beneficios de Scrum

- **Flexibilidad a cambios.** Gran capacidad de reacción ante los cambiantes requerimientos generados por las necesidades del cliente o la evolución del mercado. El marco de trabajo está diseñado para adecuarse a las nuevas exigencias que implican proyectos complejos.
- **Reducción del Time to Market.** El cliente puede empezar a utilizar las características más importantes del proyecto antes de que esté completamente terminado.
- **Mayor calidad del software.** El trabajo metódico y la necesidad de obtener una versión de trabajo funcional después de cada iteración, ayuda a la obtención de un software de alta calidad.
- **Mayor productividad.** Se logra, entre otras razones, debido a la eliminación de la burocracia y la motivación del equipo proporcionado por el hecho de que pueden estructurarse de manera autónoma.
- **Maximiza el retorno de la inversión (ROI).** Creación de software solamente con las prestaciones que contribuyen a un mayor valor de negocio gracias a la priorización por retorno de inversión.
- **Predicciones de tiempos.** A través de este marco de trabajo se conoce la velocidad media del equipo por sprint, con lo que es posible estimar de manera fácil cuando se podrá hacer uso de una determinada funcionalidad que todavía está en el Backlog.
- **Reducción de riesgos.** El hecho de desarrollar, en primer lugar, las funcionalidades de mayor valor y de saber la velocidad a la que el equipo avanza en el proyecto, permite despejar riesgos efectivamente de manera anticipada.¹⁰

Fin de la Clase ¡Buen Viernes!

Año 2024 - Profesor: Federico Carrizo

ISFT N°93