

Introducción a las metodologías ágiles:

1

Las metodologías ágiles son un conjunto de principios y prácticas para la gestión de proyectos que se centran en la entrega **iterativa y frecuente de valor al cliente**. En lugar de planificar cada detalle desde el principio (como se hace en las metodologías tradicionales).

2

las metodologías ágiles se adaptan a los cambios y se enfocan en mejorar constantemente el producto a través de ciclos de **trabajo cortos** llamados "iteraciones" o "sprints".

3

Los equipos ágiles trabajan en **colaboración** estrecha, incluyendo a los clientes y usuarios finales, para comprender mejor sus necesidades y ajustar el producto en consecuencia.

4

Metodologías ágiles más populares: Scrum, Kanban, Lean y XP (Extreme Programming).

Las metodologías ágiles son ideales para proyectos donde:

1. **Los requisitos no están completamente definidos o pueden cambiar:** Dado que las metodologías ágiles se adaptan fácilmente a los cambios, son ideales para proyectos donde los requisitos pueden evolucionar a medida que el proyecto avanza.
2. **La entrega rápida de valor es importante:** Si necesitas lanzar un producto rápidamente y luego iterar basándote en los comentarios de los usuarios, las metodologías ágiles son una excelente opción.
3. **La colaboración y comunicación son claves:** Las metodologías ágiles fomentan la comunicación frecuente entre los miembros del equipo y con los clientes, por lo que son ideales para entornos donde la colaboración es importante.
4. **La calidad es fundamental:** Los enfoques ágiles permiten la entrega constante de versiones de alta calidad del producto, ya que el proceso incluye revisiones y mejoras frecuentes.

Las metodologías ágiles proporcionan flexibilidad, capacidad de respuesta al cambio y entrega continua de valor, lo que las hace ideales para muchos proyectos de desarrollo de software y otros tipos de proyectos complejos.

Roles y responsabilidades:

1

Product Owner: Representa a los interesados del producto y es responsable de maximizar el valor del producto resultante. Normalmente es alguien de la organización (como un gerente de producto) que entiende la visión del producto, el mercado, los clientes y el negocio en general.

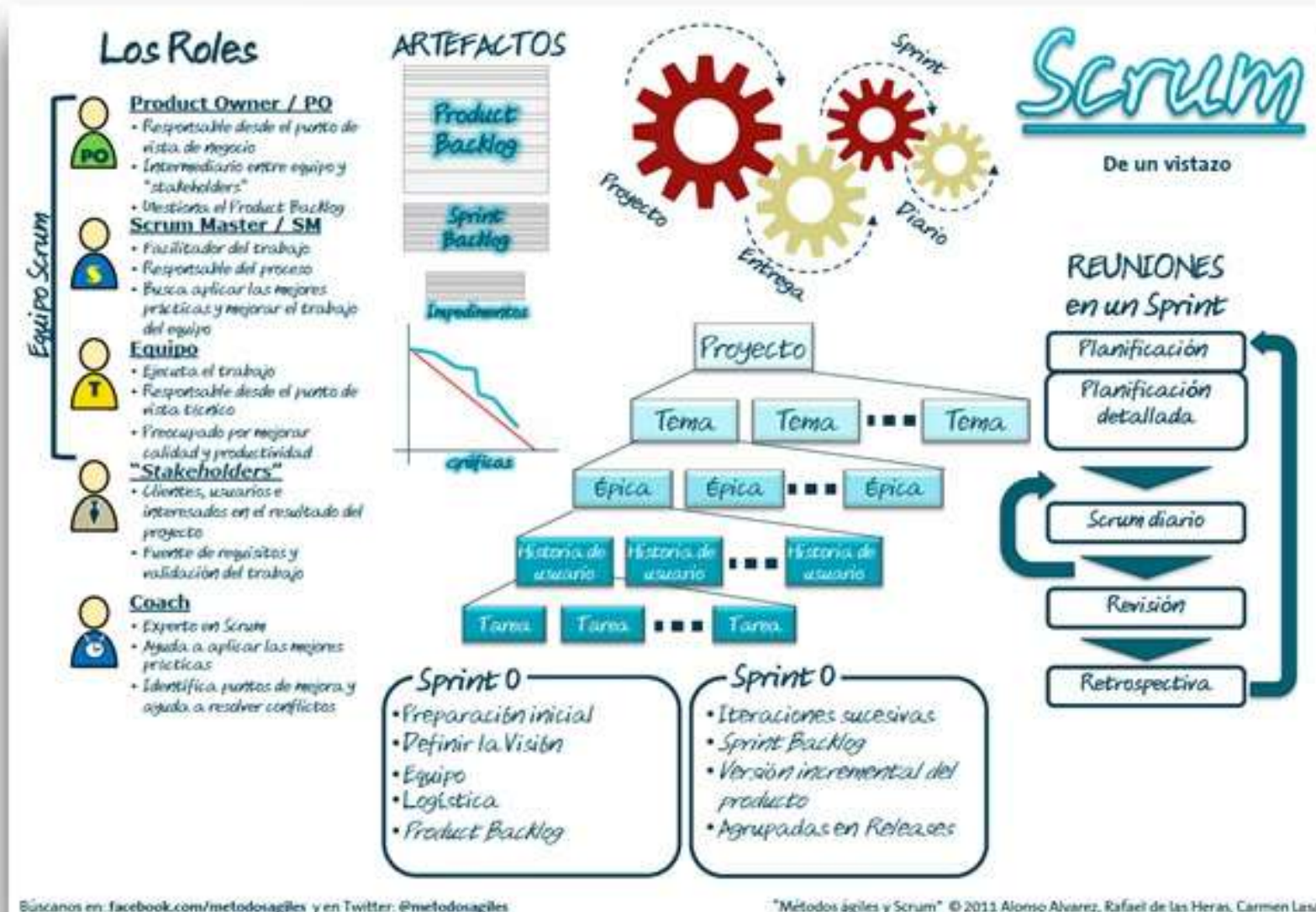
2

Scrum Master: Es el facilitador y coach del equipo. No es un jefe o gerente en el sentido tradicional, sino alguien que ayuda a eliminar los obstáculos que el equipo pueda enfrentar y garantiza que se sigan las reglas y principios de Scrum.

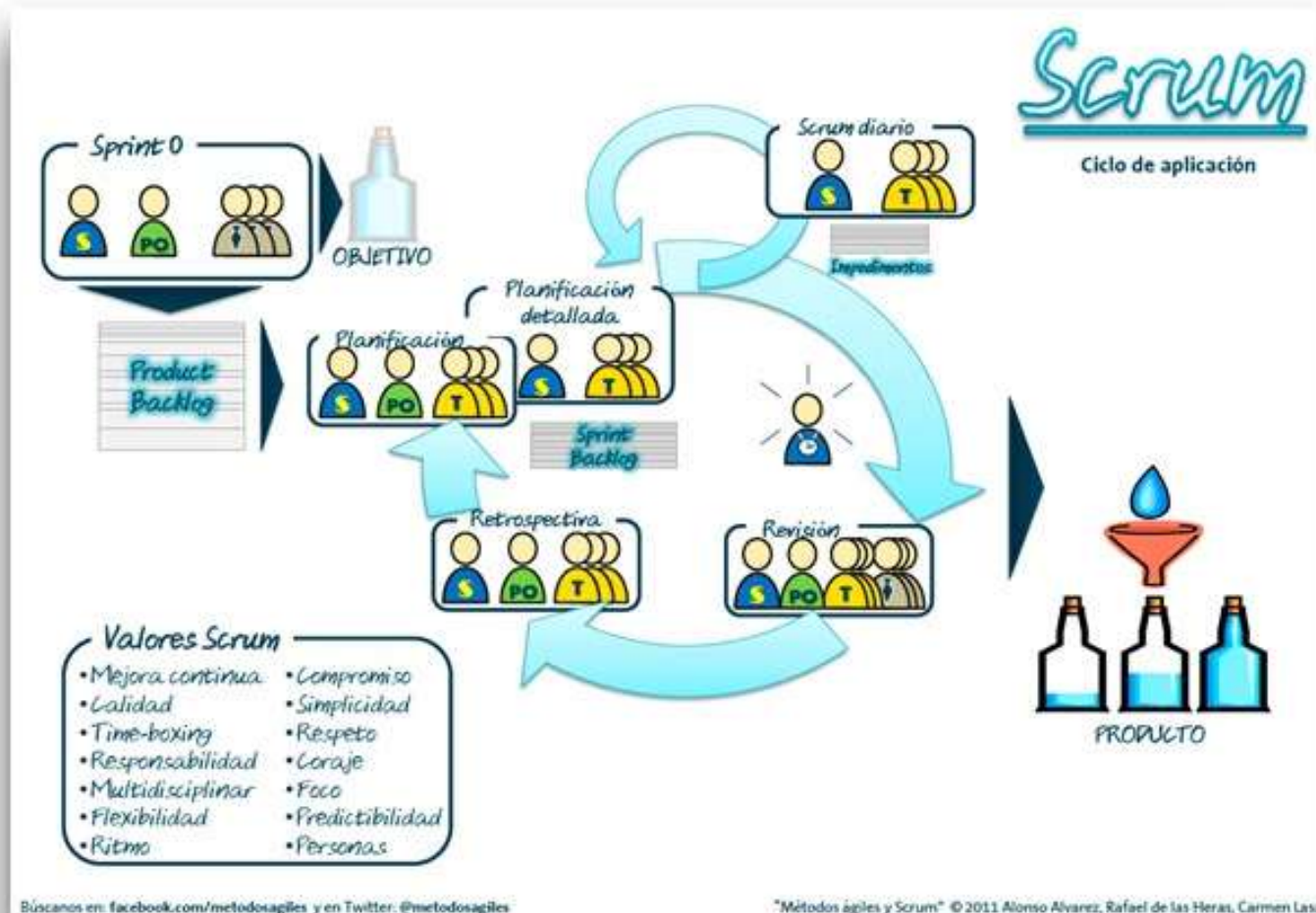
3

Equipo de Desarrollo: Es un grupo multidisciplinario de profesionales responsables de entregar incrementos potencialmente liberables del "Producto Terminado" al final de cada sprint. Los equipos de desarrollo en Scrum son autoorganizados, lo que significa que deciden internamente quién hace qué, cuándo y cómo.

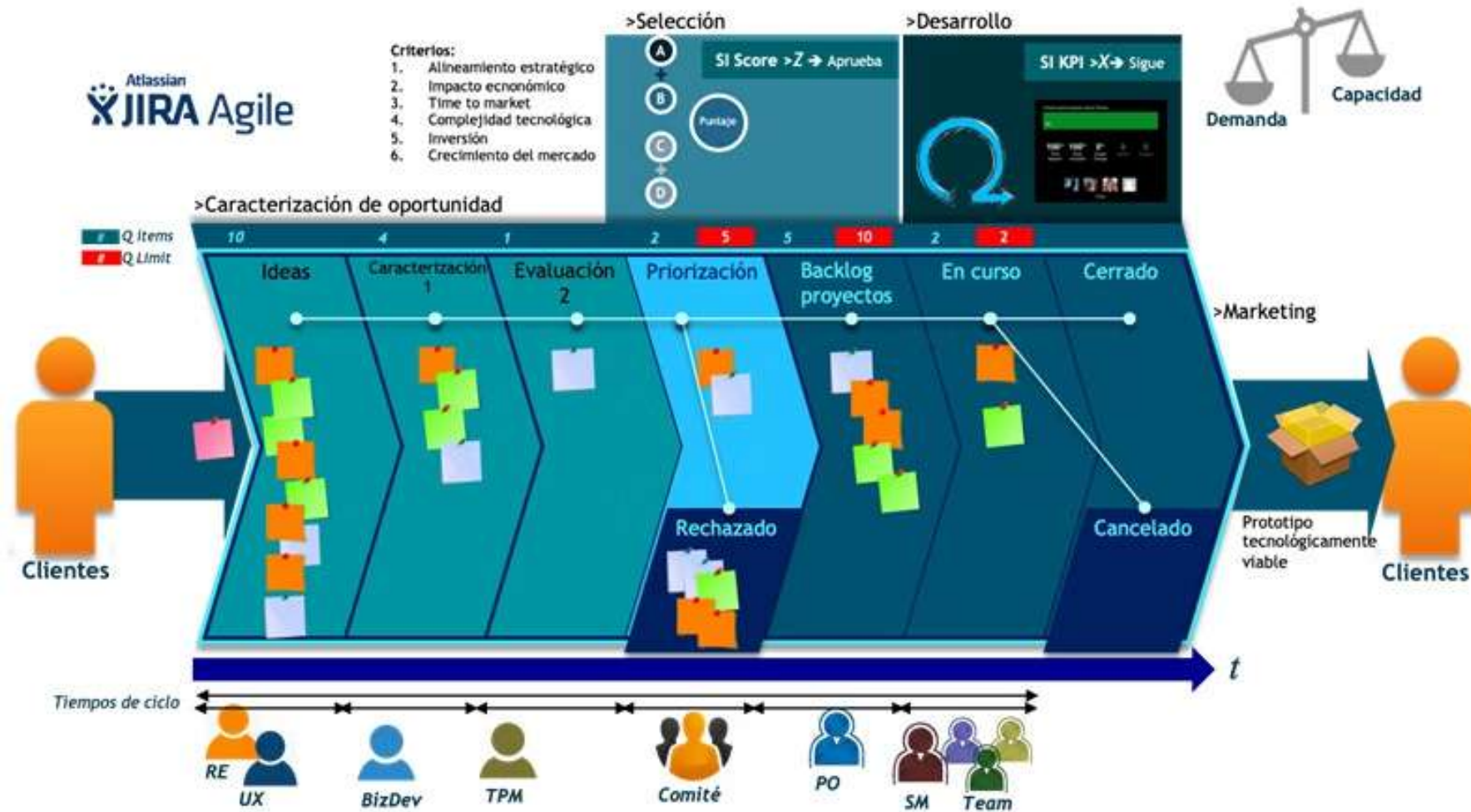
Roles y responsabilidades:



Artefactos y eventos ágiles



Kanban de proyectos - Visión ejecutiva



Kanban de proyectos - construccion

¿Qué métricas vamos a usar para mejorar nuestro proceso?



Dinamica a desarrollar sprint 1

Definición del sprint:

El proyecto se enfoca en realizar un análisis exploratorio detallado de un dataset relacionado con el mantenimiento predictivo, con el objetivo de descubrir patrones, tendencias, y posibles áreas de interés que puedan ser útiles para futuros análisis predictivos o prescriptivos. Este análisis inicial es crucial para entender la naturaleza de los datos con los que contamos, y para planear acciones futuras que permitan mejorar la efectividad y eficiencia de los programas de mantenimiento.

Objetivos:

1. **Descubrimiento de Datos:** Explorar el dataset para entender su estructura, tipo de datos y calidad de los mismos.
2. **Limpieza de Datos:** Identificar y corregir o eliminar datos incorrectos, incompletos o irrelevantes.
3. **Análisis Estadístico:** Realizar análisis estadísticos para descubrir patrones y tendencias en los datos.
4. **Visualización de Datos:** Crear visualizaciones que ayuden a interpretar los datos y compartir hallazgos con las partes interesadas.
5. **Identificación de Áreas de Interés:** Identificar áreas que podrían beneficiarse de un análisis más profundo o de la aplicación de modelos de mantenimiento predictivo.

Historias de usuario

Una Historia de Usuario debe estar estructurada de manera que describa claramente quién es el usuario, qué necesita y por qué lo necesita. Una estructura común es

Como [rol del usuario], quiero [acción o característica], para que [beneficio o resultado].

Además, cada historia de usuario debe incluir Criterios de Aceptación que detallen cómo se verificará que la historia está completada correctamente.

Ejemplos de Historias de Usuario para el proyecto:

1. Exploración Inicial de Datos:

- Como Analista de Datos, quiero inspeccionar la estructura del dataset, para entender los tipos de datos y la calidad de los mismos.
 - *Criterios de Aceptación:*
 - Identificación y documentación de los tipos de datos.
 - Identificación y documentación de datos faltantes o incorrectos.

2. Limpieza de Datos:

- Como Analista de Datos, quiero limpiar y preprocesar los datos, para asegurar la precisión en el análisis posterior.
 - *Criterios de Aceptación:*
 - Eliminación o corrección de datos incorrectos o faltantes.
 - Documentación del proceso de limpieza.

Ejemplos

Historia de Usuario: Generación de Boxplots

- **Título:** Generación de Boxplots para Analizar Casos de Falla y No Falla
- **Como** Analista de Datos,
- **Quiero** generar boxplots para diferentes variables relevantes,
- **Para que** pueda entender la distribución y variabilidad de estas variables en casos de falla y no falla, y así identificar cualquier diferencia notable que pueda ayudar en la predicción y prevención de futuras fallas.

Criterios de Aceptación:

1. Selección de Variables:

- Identificación y selección de variables relevantes para el análisis basado en el conocimiento previo o la correlación con los casos de falla.

2. Generación de Boxplots:

- Creación de boxplots que muestren la distribución de las variables seleccionadas para los casos de falla y no falla.

3. Análisis Comparativo:

- Comparación visual y estadística de las distribuciones entre los casos de falla y no falla utilizando los boxplots generados.

4. Identificación de Outliers:

- Identificación y documentación de outliers que podrían indicar condiciones anómalas leading a fallas.

5. Documentación de Hallazgos:

- Documentación clara y comprensible de los hallazgos, incluyendo cualquier diferencia notable en las distribuciones entre los casos de falla y no falla.

Criterios generales de Éxito:

1. **Calidad de Análisis:** El análisis debe ser exhaustivo y los hallazgos deben estar bien fundamentados y claramente comunicados.
2. **Cumplimiento de Plazos:** Todas las fases del proyecto deben ser completadas dentro de los plazos acordados.
3. **Satisfacción de las Partes Interesadas:** Las partes interesadas deben estar satisfechas con los hallazgos y las recomendaciones proporcionadas.
4. **Preparación para Análisis Futuros:** El análisis exploratorio debe proporcionar una base sólida para futuros análisis predictivos o prescriptivos.

Stakeholders:

- **Equipo de Mantenimiento:** Beneficiarios de los hallazgos y recomendaciones.
- **Gestión de Proyectos:** Supervisión y aseguramiento del cumplimiento de objetivos y plazos.
- **Dirección Técnica:** Proveedores de dirección estratégica y toma de decisiones basada en los hallazgos del proyecto.

