Javier Sáez López 1DAM

**Práctica 1 – Metodologías ágiles**

En esta práctica aprenderemos los conceptos básicos de las principales metodologías ágiles en el desarrollo de software, de gran utilización en la programación moderna.

**Práctica**

1. Realiza un estudio y documenta en un trabajo las principales características, fases, ventajas, modos de implantación, etc. de la **metodología ágil SCRUM**. Es importante que te bases para ello en multitud de fuentes de información, sin utilizar el copia /pega de grandes párrafos como forma única de realizar el trabajo. Deberás incluir al final del trabajo enlaces a todas las fuentes de información que hayas tenido en cuenta o en las que te hayas basado para realizar el trabajo.

**Las principales características de la metodología Scrum son**:

Permite una correcta gestión de las expectativas del cliente

Fija un límite de tiempo para cumplir objetivos

Da mayor prioridad o relevancia a aquello que le interesa al cliente

Y funciona en equipos autodirigidos y equipos de alto rendimiento

**Y sus principales ventajas:**

Claridad en los objetivos y alcances.

Flexibilidad.

Mayor control de imprevistos.

Equipos altamente productivos.

Predictibilidad.

Fácil de implementar.

Las principales fases de Scrum son 5 pero se reparten en 16 procesos o tareas las cuales vamos a ver ahora divididas en sus principales fases.

**La primera seria la fase de inicio** en esta fase se encarga de estudiar y analizar el proyecto identificando las necesidades básicas del sprint

Las preguntas que se hacen en esta fase de inicio son:

Que quiero

Como lo quiero

Cuando lo quiero

La metodología Scrum da preferencia a la formación de equipos pequeños de mínimo 3 y máximo 5 personas, pues se facilita la fluidez de las ideas y se aporta creatividad al grupo.

Entre sus primeros pasos tenemos 6 procesos:

1. crear la visión del proyecto

2. identificar al líder del proyecto y a las personas a las que va dirigido

3. formar equipos Scrum

4.desarrolar épicas que sirven para estructurar los temas e iniciativas

5. crear **backlog**, que podría definirse como un listado ordenado y **priorizado** de los requisitos necesarios para la implementación de un proyecto.

6. planificar el lanzamiento.

**La segunda fase seria planificación y estimación**

Incluiría los siguientes pasos:

1.crear, estimar y comprometer historias de usuario

2.identificar y estimar las tareas

3.crear sprint backlog o iteración de tareas

Será la fase más importante en ella se delegarán las tareas a los grupos y se estimara el tiempo de entrega y prioridades.

**La tercera fase será la implementación del proyecto**

Donde se discute el sprint y se explora como optimizar el trabajo de cada grupo para darle forma definitiva al proyecto estos serían sus procesos:

1.crear entregables.

2.realizar reuniones diarias con los miembros del equipo para sincronizar las tareas.

3.se vuelve a estudiar las prioridades del producto.

En esta fase ya no deberían hacerse cambios innecesarios a no ser para que el sprint tenga éxito y previa consulta con el grupo de stakeholders (interesados).

**La cuarta fase seria la revisión y retrospectiva**

Una vez que el proyecto está terminado se hace una revisión del proceso y una autocrítica respecto al propio trabajo para aportar opiniones constructivas y soluciones viables con estos pasos:

1.demostrar y validar el sprint (para saber si los resultados entregados son aceptados)

2.retrospectiva del sprint (para buscar oportunidades de mejora)

**La quinta y última fase sería la de lanzamiento**

En esta fase es donde se entrega el producto que debería cumplir con las dos últimas tareas:

1.enviar entregables (aquellos que alcanzan los Criterios de Aceptación y la Definición de Terminado)

2.enviar retrospectiva del proyecto (que sirve para identificar, documentar e internalizar las lecciones aprendidas y las mejoras para proyectos futuros)

Bibliografía:

<https://coworkingfy.com/metodologia-scrum>

<https://blog.comparasoftware.com/fases-metodologia-scrum>

1. Realiza lo indicado en el apartado 1 para la **metodología ágil Programación Extrema (XP)**.

La metodología ágil programación extrema xp es un conjunto de técnicas que dan agilidad y flexibilidad en la gestión de proyectos y se centra en seguir los requisitos de los clientes para la creación del producto.

Las principales características que tiene son:

. Comunicación constante entre el cliente y el equipo de desarrollo.

. Respuesta rápida a los cambios constantes.

. La planificación es abierta con un cronograma de actividades flexible.

. El software que funciona está por encima de cualquier otra documentación.

. Los requisitos del cliente y el trabajo del equipo del proyecto son los principales factores de éxito del mismo.

La metodología de programación extrema xp:

Lista de equipo xp:

**Los clientes** que son los responsables de definir los objetivos del proyecto, así como su gestión y marcar las prioridades del proyecto.

**Los programadores** ayudaran a cumplir los objetivos, así como a delimitar la duración y estimar el tiempo, planificaran los requisitos acordados con los clientes sobre el proyecto.

**Los testers** serán los encargados de pruebas y estarán en comunicación con el cliente para alinear los resultados con los requisitos estimados.

**El tracker o encargado de seguimiento** se encargará del control y estará en comunicación constante con el cliente para definir los hitos y puntos de control en la planificación en función a los objetivos del cliente y las estimaciones en el tiempo de ejecución de las tareas de los programadores.

**Los coach** son los guías del proyecto, para que todos sepan bien qué, cómo y cuándo hacerlo.

**El mánager** es el responsable de coordinar las comunicaciones entre las distintas partes y ofrecer y gestionar los recursos necesarios, debe tener una idea general del funcionamiento del proyecto y su estado en todo momento

**METODOLOGIA XP: FASES**

**Planificación:** va de acuerdo con las historias de usuario, estas se priorizan y descomponen en mini versiones, luego la planificación se va revisando cada dos semanas aproximadamente, después de las iteraciones, para obtener un software útil, funcional, listo para las pruebas y su lanzamiento.

**Diseño:** se trabaja con un código sencillo, realizando lo mínimo necesario para que funcione, se obtiene el prototipo. Luego para el diseño del software si va orientado a objetos se generan tarjetas CRC (Clase-Responsabilidad-Colaboración).

**Codificación:** se hace en parejas frente al mismo ordenador, en algunos casos se intercambian las parejas, para asegurar que el código sea más universal, para que cualquier otro trabajador pueda trabajar en él y entenderlo. Debe parecer hecho por una sola persona, para obtener una programación organizada y planificada.

**Pruebas:** deben ser automáticas y continuas, esto es clave para proyectos a corto plazo. Incluso el mismo cliente puede hacer pruebas, proponer pruebas nuevas y validar las mini versiones.

**Lanzamiento:** si se llega a este punto es porque se han probado todas las historias de usuario o mini versiones y han tenido éxito, ajustándose a los requerimientos del cliente. Generando un software útil y que puede incorporarse en el producto.

<https://ginzo.tech/blog/como-funciona-metodologia-xp-desarrollo-software>

<https://www.sinnaps.com/blog-gestion-proyectos/metodologia-xp>

1. Realiza lo indicado en el apartado 1 para la **metodología ágil Kanban**.

Para la metodología ágil Kanban se usa un tablero con diferentes columnas y es primordial saber las tareas que se necesitan gestionar y organizar en el proyecto.

Se usan tarjetas que representan los diferentes elementos de trabajo, y las columnas representan el flujo. Las tarjetas van pasando por las diferentes columnas a medida que están por realizar, en proceso, y terminadas, de ese modo, es muy sencillo identificar los cuellos de botella, que serán las tareas que dificultan más y las que más tardan en hacerse.

Una vez identificados los cuellos de botella, se debe minimizar las interrupciones, estableciendo un número máximo de elementos por etapa, lo cual permite aumentar exponencialmente el rendimiento de un proyecto, ajustarlo y conseguir una eficiencia más alta en cada etapa.

**Ventajas del modelo Kanban**

Fácil de utilizar.

Permite actualizarse rápidamente.

Entendible por todos los miembros del equipo.

Técnica de gestión visual.

Ágil de consultar.

Pautar el desarrollo del trabajo muy efectivo.

Estímulo del rendimiento.

Permite una colaboración más cercana.

<https://www.epitech-it.es/kanban>