**Exercici 1: Implantació vs Implementació**

**En les fases del cicle de vida del software existeixen dos fases amb un nom similar: implantació i implementació. Identifica de quines fases es tracta, i indica on es troben en el cicle de vida del programari.**

**Pensa un exemple de desenvolupament de programari i identifica en què consistirien aquestes fases.**

Implementación o codificación es la etapa del proceso de desarrollo donde los desarrolladores traducen a código todo el diseño del programa.

Implantación o despliegue es la etapa del proceso donde los desarrolladores publican el proyecto final en el formato acordado.

Como ejemplo de una tienda online la fase de implementación seria todo el desarrollo del código de la web y la fase de implantación la publicación final para que pueda ser utilizado por los usuarios.

**Exercici 2: Scrum**

**Busca i explica en què consiteix la metodologia SCRUM.**

Scrum es un marco de trabajo para el desarrollo ágil de software que se centra en la colaboración, la comunicación y la transparencia. Es un proceso iterativo e incremental, en el que el trabajo se divide en pequeñas tareas que se completan en ciclos de tiempo cortos, conocidos como sprints.

Los ***principios*** de Scrum se basan en los siguientes valores:

* Scrum se centra en las personas y sus interacciones, en lugar de los procesos y las herramientas.
* El software funciona es más importante que la documentación exhaustiva.
* El cliente debe estar involucrado en el proceso de desarrollo desde el principio.
* Es flexible y se adapta al cambio.
* El software se entrega al cliente de forma temprana y continua.

Scrum define tres ***roles*** principales:

* Product Owner: Es el responsable de definir el producto y sus requisitos.
* Scrum Master: Es el responsable de facilitar el proceso de Scrum y eliminar los obstáculos que puedan impedir el progreso.
* Equipo Scrum: Es el equipo responsable de desarrollar el producto.

El proceso de Scrum (***metodología***) se divide en los siguientes pasos o eventos:

* Sprint Planning: El equipo Scrum se reúne para planificar el trabajo que se realizará en el sprint. El Product Owner presenta los requisitos del sprint y el equipo Scrum planifica cómo los va a completar.
* Daily Scrum: El equipo Scrum se reúne diariamente para realizar un seguimiento del progreso y planificar el trabajo del día.
* Sprint Review: El equipo Scrum presenta el trabajo completado al Product Owner y a otros interesados. El Product Owner acepta o rechaza el trabajo.
* Sprint Retrospective: El equipo Scrum se reúne para reflexionar sobre el sprint anterior y planificar mejoras para el siguiente.

**Intenta buscar opinions a favor i en contra d'utilitzar aquesta metodologia.**

Las ***ventajas*** de Scrum incluyen:

* Fomenta la colaboración entre los miembros del equipo y entre el equipo y el cliente.
* Proporciona una visión clara del progreso del proyecto.
* Es flexible y adaptable al cambio.
* Divide el trabajo en pequeñas tareas, lo que reduce el riesgo de fracaso.

Las ***desventajas*** de Scrum incluyen:

* Es un proceso iterativo, por lo que requiere un compromiso a largo plazo por parte del equipo y del cliente.
* Puede ser complejo de implementar: Supone un cambio de mentalidad por parte de los equipos de desarrollo.
* Es un buen enfoque para proyectos de desarrollo de software complejos, pero puede no ser adecuado para proyectos pequeños o simples.

**Exercici 3: TDD**

**Busca i explica en què consiteix el TDD o Test-Driven Development.**

Test-Driven Development (TDD) es una metodología de desarrollo de software que se centra en la escritura de pruebas automatizadas antes de escribir el código.

El proceso se divide en los siguientes pasos:

* Se escribe una prueba que describe una funcionalidad o comportamiento esperado.
* Se ejecuta la prueba y se asegura de que falle.
* Se escribe el código mínimo necesario para que la prueba pase.
* Se refactoriza el código para mejorar su calidad sin alterar su comportamiento.

Las ***ventajas*** de TDD incluyen:

* Obliga a los desarrolladores a pensar en el comportamiento del código antes de escribirlo, lo que mejora la calidad del código.
* Ayuda a identificar y corregir errores temprano en el proceso de desarrollo.
* Facilita la modificación del código existente y mejora la mantenibilidad
* Proporciona una mayor confianza en que el código funciona correctamente.

Las ***desventajas*** de TDD incluyen:

* Requiere que los desarrolladores escriban pruebas antes de escribir el código y puede ocasionar que sea lento.
* Puede ser difícil de aprender y requiere un cambio de mentalidad por parte de los desarrolladores.
* Puede ser demasiado complejo para proyectos pequeños o simples.

**Exercici 4: CI/CD**

**Busca i explica en què consiteix CI/CD o Continuous Integration/Continuous Delivery.**

Es una práctica de desarrollo de software que se centra en la automatización de los procesos de desarrollo, prueba y despliegue.

La integración continua (CI) es el proceso de combinar el código de los desarrolladores en un repositorio central de forma regular. Esto ayuda a identificar y corregir problemas de integración temprano en el proceso de desarrollo.

La entrega continua (CD) es el proceso de desplegar automáticamente las nuevas versiones del software a un entorno de producción. Esto ayuda a los equipos a entregar software de forma más rápida y fiable.

Las ***ventajas*** de CI/CD incluyen:

* Ayuda a identificar y corregir errores temprano en el proceso de desarrollo.
* Reduce el riesgo de que las nuevas versiones del software introduzcan errores o problemas.
* Permite entregar software de forma más rápida y fiable.
* Facilita la colaboración entre los desarrolladores y los equipos de operaciones.

Las ***desventajas*** de CI/CD incluyen:

* Requiere una inversión en herramientas y procesos.
* Puede ser complejo de implementar, especialmente en proyectos grandes o complejos.
* No es adecuado para proyectos pequeños o simples.