

Activitats UD1-ED

Introducció

1. Descriu breument la relació que hi ha entre els components de maquinari principals d'un ordinador i l'emmagatzematge i l'execució del programari.

En concreto, se pueden relacionar cuatro elementos entre el hardware y el software:

- El disco duro almacena de forma permanente los ficheros ejecutables y los ficheros de datos.
- La memoria RAM almacena de manera temporal el código binario de los ficheros ejecutables y los ficheros de datos necesarios.
- La CPU lee y ejecuta instrucciones almacenadas en la memoria RAM, así como los datos necesarios.
- La E/S recoge nuevos datos desde la entrada, muestra los resultados, y lee/guarda en un disco.

2. Defineix els conceptes següents:

- Codi font: Es un conjunto de instrucciones escritas en un lenguaje de programación específico que los programadores utilizan para crear software. Su función principal es servir como un plano o diseño detallado que los desarrolladores utilizan para crear, mantener y mejorar el software.
- Codi objecte: Es un tipo de archivo binario que contiene representaciones intermedias de un programa de software. Además, se trata de una etapa intermedia en el proceso de creación de un programa ejecutable.
- Codi executable: Se trata de un conjunto de instrucciones y datos diseñados para ser procesados y ejecutados por una computadora. Estas instrucciones están escritas en un lenguaje de programación específico. Han sido previamente compiladas o interpretadas para que la computadora pueda comprender y ejecutar las tareas específicas.

Cicle de vida del programari

1. Defineix "Cicle de vida del programari".

Es un proceso que abarca todas las etapas de desarrollo de software, desde la concepción de una idea hasta su implementación, incluyendo el análisis, diseño, codificación, pruebas y mantenimiento. Este ciclo puede repetirse para mejorar el software en futuras versiones.

2. Anomena les fases principals del desenvolupament de programari i explica breument que es fa a cadascuna.

- Anàlisis: Determina, define y recopila los objetivos y necesidades del software que solicita el cliente.
- Disseny: Crea una estructura/arquitectura para el software, además de planificar la interfaz del usuario.
- Codificació: Como su nombre indica, se trata de traducir los diseños y requisitos previamente establecidos en código.

- Proves: Fase en la que se evalúan y corrigen errores para garantizar el correcto funcionamiento del software.
- Manteniment: Incluye actualizaciones, correcciones de errores y mejoras para garantizar la funcionalidad a lo largo del tiempo.

3. Avantatges i inconvenients del model en cascada.

Ventajas: Es fácil de seguir y adecuado para proyectos simples y bien definidos desde el inicio.

Desventajas: Posee poca flexibilidad y no es adecuado para proyectos de gran envergadura con muchos cambios en el proceso. El cliente solo recibe algo al final, lo que puede generar problemas de expectativas y comunicación.

4. Explica com funciona el model de desenvolupament mitjançant la creació de prototips.

Se trata de crear en la fase de análisis versiones iniciales y simplificadas del software, llamadas prototipos, que contienen las características esenciales. Estos se presentan a los usuarios de forma rápida para obtener sus opiniones y ajustar los requisitos antes del desarrollo completo.

5. Quins principis regeixen el desenvolupament àgil expressats al Manifest Àgil.

En total hay doce principios, incluyendo satisfacción del cliente a través de entregas en un corto espacio de tiempo y continuas de software utilizable; la recepción de requerimientos cambiantes incluso en desarrollo tardío; software con entregas constante; colaboración diaria y cercana entre los negocios, los clientes y los desarrolladores; los proyectos se construyen entre individuos motivados en los que se debe confiar; conversaciones cara a cara; programación en pareja (pueden trabajar en la misma estación de trabajo); desarrollo sostenible capaz de mantener un ritmo constante; excelencia lograda a través de la reflexión; simplicidad; equipos autoorganizados; cambio constante a las circunstancias variables.

Puedes encontrar más detalles en [este enlace](<https://youtu.be/tGEYClSYyuw?feature=shared>).

6. Què és una història d'usuari? Posa un exemple propi.

Una historia de usuario es una descripción de un requisito, una funcionalidad o característica que un usuario o cliente desea en un producto o software, expresada en lenguaje comun del usuario..

Por ejemplo: "Como gerente de producto, quiero que los miembros del equipo entiendan cómo sus trabajos individuales contribuyen a los objetivos comerciales para aumentar la eficiencia."

7. KANBAN. Estudia els avantatges i els inconvenients de tenir una pissarra web digital per a la metodologia Kanban.

Ventajas:

- Acceso remoto, actualización en tiempo real.
- Ofrece una visión más amplia del trabajo total a realizar.
- Es útil para equipos de trabajo que trabajan de manera remota.
- Facilita la colaboración, automatización y personalización.
- Fomenta la flexibilidad del equipo de trabajo.
- Ayuda a cumplir con los tiempos de entrega.

Desventajas:

- Si hay demasiadas tareas a realizar al mismo tiempo, puede resultar abrumador visualmente.

- Coste adicional por funciones, tiempo de aprendizaje, dependencia del estado en línea, preocupaciones de privacidad y resistencia al cambio.

8. KANBAN. Fes un resum de la metodologia Kanban i indica les seves diferències davant de SCRUM.

La metodología Kanban es un sistema visual de administración que se enfoca en la eficacia y la secuencia de actividades. Se basa en dividir el proyecto en elementos de trabajo (tareas más pequeñas) y representarlos visualmente mediante un tablero, en este tablero se dividen las tareas según el estado en el que se encuentran, normalmente: por hacer, en proceso, para revisar y hecho. Esta metodología permite que equipos de trabajo distintos o separados puedan trabajar en un mismo proyecto a la vez de manera eficiente.

Se diferencia de Scrum en aspectos como la cadencia, metodología de publicación, funciones, métricas clave y filosofía de cambios:

- Scrum se utiliza para tareas intensivas (Sprints) y de corta duración, mientras que Kanban es para equipos de trabajo continuo.
- La publicación del trabajo en Scrum se realiza al final de cada Sprint, mientras que en Kanban la entrega es normalmente continua.
- Scrum tiene roles definidos (Propietario del producto, experto en Scrum, Equipo de desarrollo), mientras que Kanban no los tiene.
- La métrica principal en Scrum es la Velocidad, mientras que en Kanban es el Tiempo de ciclo.
- En Scrum, los equipos deben evitar cambios en las previsiones durante el Sprint, en Kanban los cambios pueden ocurrir en cualquier momento.

9. SCRUM. Explica com funciona Scrum.

Scrum es un marco de trabajo ágil que se centra en la entrega repetitiva de productos, mediante ciclos. Se basa en ciclos de duración corta y fija de desarrollo llamados "sprints", donde un equipo autoorganizado planifica, desarrolla y entrega funcionalidades prioritarias. Se llevan a cabo reuniones diarias de seguimiento. Cada sprint debe proporcionar un resultado completo que se pueda entregar al cliente. Se divide en las fases:

- Planificación de la iteración: Se determinan los requisitos y se planifican las tareas necesarias para lograrlos.
- Ejecución de la iteración: Las tareas suelen representarse en un tablero visual para que el equipo pueda verlas mejor.
- Inspección y adaptación: El último día de la iteración se realiza una reunión para revisarla.

Componentes:

El Product Owner es la persona responsable de representar los intereses y necesidades del cliente o usuarios finales ante el equipo de desarrollo.

Scrum Master es la persona que trabaja para eliminar impedimentos y crear un ambiente propicio para la colaboración y el desarrollo efectivo, es decir, en la facilitación del trabajo del equipo.

El Team es uno de los tres roles fundamentales en el marco de trabajo Scrum. Este equipo es el grupo de profesionales que son responsables de llevar a cabo el trabajo necesario para entregar un incremento potencialmente entregable del producto al final de cada Sprint.

10. SCRUM. Defineix els termes següents:

- Product backlog: Es una lista en constante evolución que enumera de manera ordenada todas las funcionalidades, características y requisitos que deben ser desarrollados en el proyecto. Es responsabilidad del "Product Owner" y sirve como hoja de ruta.

- Sprint backlog: Es una lista de actividades concretas extraídas del "Product backlog" para ser realizadas durante un sprint específico.

11. SCRUM. A la terminologia Scrum quins termes s'utilitzen com a sinònim de:

- Cap de projecte: "Scrum Master".
- Client: "Product Owner".
- Equip de desenvolupament: "Team".

12. XP. Quines són les característiques distintives de XP davant d'altres metodologies àgils? Explica-les.

- Metàfora (metaphor): Una historia que explica cómo funciona el sistema, facilitando su comprensión.
- Refactorització (refactoring): Proceso para simplificar y mejorar el código sin perder funcionalidad al implementar nuevas características.
- Programació en parella (pair programming): Dos personas trabajan juntas en una estación de trabajo para mejorar la calidad del código.
- Propietat col·lectiva (collective ownership): Todos pueden contribuir al código, promoviendo la colaboración.
- Estàndards de codificació (coding standards): Se establecen normas para mantener la consistencia en el código.
- Client in situ (on-site customer): Un cliente trabaja junto a los programadores para facilitar la comunicación y priorización.
- Integració contínua (continuous integration): Se integra y prueba el código regularmente para asegurar su funcionamiento.

Llenguatges de programació

1. Quina diferència hi ha entre els llenguatges declaratius i els imperatius? Anomena almenys 2 de cada tipus.

La diferencia está en la forma en que funcionan.

Los lenguajes declarativos indican el resultado que se desea obtener sin especificar los pasos detallados para lograrlo. En cambio, los lenguajes imperativos proporcionan instrucciones detalladas sobre cómo lograr el resultado.

Ejemplos de lenguajes declarativos son SQL y Prolog, mientras que lenguajes imperativos incluyen Java y C++.

2. Explica què és compilar? Explica què és interpretar?

Compilar implica transformar el código fuente, escrito por un programador en un lenguaje de programación, en un conjunto de instrucciones comprensibles y ejecutables por una computadora. Durante este proceso, el código fuente se somete a un análisis por parte de un compilador, que verifica la estructura del código, resuelve referencias a variables y funciones, y genera un archivo ejecutable o una representación binaria del programa.

Interpretar implica la ejecución de un programa directamente a partir de su código fuente. Esto se logra traduciendo y ejecutando cada línea en tiempo real, sin la necesidad de crear un archivo binario separado.

3. Avantatges dels llenguatges compilats.

Ventajas de los lenguajes compilados incluyen un rendimiento superior en comparación a los lenguajes interpretados, la capacidad de ocultar el código fuente brindando una capa extra de seguridad, cierta portabilidad y un enfoque adecuado para proyectos grandes y complejos.

4. Avantatges dels llenguatges interpretats

Ventajas de los lenguajes interpretados incluyen portabilidad, flexibilidad en el desarrollo, depuración interactiva que permite ejecutar línea por línea para revisar y corregir en tiempo real, facilidad de aprendizaje, y tiempos de compilación más cortos.

5. Anomena 2 llenguatges compilats i altres 2 interpretats.

Lenguajes compilados: C y C++.

Lenguajes interpretados: Python y Ruby.

6. Es pot considerar codi objecte el bytecode generat a Java després de la compilació? Explica'n la resposta.

Sí, el bytecode generado en Java después de la compilación puede considerarse código objeto, pero no en el sentido de código objeto nativo. Es un lenguaje intermedio destinado a ser ejecutado en la máquina virtual de Java.

7. Posa un exemple de llenguatge dels tipus següents:

- Baix nivell: Lenguaje ensamblador (assembly language).
- Nivell mitjà: C++.
- Alt nivell: Python.

8. Quin paradigma de programació segueixen els llenguatges següents?

- C: Imperativa.
- C++: Admite varios paradigmas: orientada a objetos, genérica, imperativa, orientada a aspectos y de sistemas.
- SQL: Declarativa, que se divide en DML (Data Manipulation Language) y DDL (Data Definition Language).
- Java: Principalmente orientada a objetos, aunque también posee aspectos imperativos, estructurados, concurrentes, genéricos y funcionales.
- Javascript: Admite varios paradigmas: orientada a objetos, funcional, basada en eventos, asíncrona, orientada a aspectos.
- Lisp: Funcional.
- Prolog: Lógica.

9. Explica quins criteris es poden seguir a l'hora d'escollir un llenguatge de programació per al desenvolupament de programari.

Al elegir un lenguaje de programación para el desarrollo de software, se deben considerar diversos criterios, como el campo de aplicación, experiencia previa, herramientas de desarrollo disponibles, documentación, base de usuarios, reusabilidad, portabilidad y requisitos del cliente.