

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 cuatro elementos a relacionar 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 almacenados en memoria RAM, así como las datos necesarios.
- La E/S recoge nuevos datos desde la entrada, se muestran los resultados, se leen/guardan 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. Es decir, se compone de un texto plano, y 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 que están diseñados para ser procesados y ejecutados por una computadora. Estas instrucciones están escritas en un lenguaje de programación específico. A su vez, 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 pasando a su vez por toda una serie de fases como son, el análisis, el diseño, la codificación, las pruebas y el mantenimiento. Además, este ciclo que se sigue se puede repetir para mejorar dicho software en futuras versiones.

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

Análisis: se determina, define y recopila los objetivos y necesidades del software que solicita el cliente.

Diseño: crear una estructura/arquitectura para el software, además de la planificación de la interfaz del usuario.

Codificación: como su propio nombre indica, se trata de codificar de acuerdo a los diseños y requisitos previamente establecidos.

Pruebas: fase en la que se realiza una evaluación y corrección de los errores para así garantizar que el software funciona debidamente.

Mantenimiento: en este caso se trata de actualizaciones, correcciones de errores y/o mejoras para garantizar su funcionalidad a lo largo del tiempo.

3. Avantatges i inconvenients del model en cascada.

Las ventajas: es de fácil uso ya que sigue un camino claro y conciso por tanto es fácil de seguir, por tanto lo hace adecuado para realizar trabajos/proyectos que sean simples y estén bien definidos desde un inicio.

Las desventajas: posee poca flexibilidad, además a la hora de realizar proyectos de gran tamaño no sería el modelo adecuado (se realizan muchos cambios en el proceso cosa que en el modelo en cascada no se hace), a su vez el cliente solo recibe algo en su fase final por tanto no se participa en el proceso o no poder ver como avanza puede acarrear problemas la espera.

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

Se trata de un enfoque en el que se crean versiones iniciales y simplificadas del software, llamadas prototipos (antes de desarrollar la versión definitiva). Estos suelen contener las características esenciales del software y se construyen de forma rápida. Seguido, se presentan a los usuarios o las partes interesadas con el fin de obtener sus opiniones. El resultado que se obtiene de la información por parte de estos usuarios en esta etapa de inicio es crucial, ya que permite ajustar y perfeccionar los requisitos del proyecto antes de iniciar el desarrollo completo. Además, con las repeticiones sucesivas, se mejoran los prototipos y se incorporan las sugerencias/recomendaciones por parte de los usuarios.

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

<https://youtu.be/tGEYCIsYyuw?feature=shared>

En total se encuentran un total de doce principios: 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.

6. Què és una història d'usuari? Consulta el següent enllaç i posa un exemple propi.

Les podem formular de la següent manera: “Com a [perfil], vull [objectiu del software], per a poder [resultat]”.

https://es.wikipedia.org/wiki/Historias_de_usuario

Se trata de una descripción de una funcionalidad o característica que un usuario o cliente desea de un producto o software. Un ejemplo de ello sería: “Como [gerente de producto], quiero [manera en que los miembros del equipo entiendan como los trabajos individuales contribuyen en 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. Pots consultar els següents enllaços:

<https://leankit.com/learn/kanban/kanban-board/>

<https://trello.com/es>

<https://taiga.io/>

<https://kanbantool.com/es/>

Ventajas: acceso remoto (poder acceder desde cualquier punto); actualización en tiempo real (cualquier integrante del trabajo en la pizarra web tienen información actualizada al momento); facilita la colaboración entre miembros; automatización; personalización.

Desventajas: coste (hay funciones en la pizarra que suponen pagar un extra si quieres utilizarlas); aprendizaje (el tiempo que se pierde aprendiendo a utilizar una herramienta nunca antes utilizada); dependes de su estado (si por algún casual el servicio en línea deja de funcionar interrumpe el trabajo de manera indefinida); privacidad (los datos que se manejen hay que tratarlos con cuidado, aún más si son confidenciales); cambio de metodología (puede haber gente que no quiera adaptarse a una manera nueva herramienta y utilizar la que siempre ha utilizado).

8. KANBAN. Fes un resum de la metodologia Kanban i indica les seves diferències davant de SCRUM. Pots consultar el següent enllaç:

<https://ca.atlassian.com/agile/kanban>

Esta metodología es un sistema visual de administración que se emplea para mejorar la eficacia y la secuencia de actividades en diversas situaciones, aunque se encuentra particularmente extendido en el ámbito de la dirección de proyectos y la creación de software. El funcionamiento de este método se realiza mediante el uso de tablero (es el núcleo, en él se representa mediante columnas diferentes etapas/columnas del trabajo a realizar) y tarjetas (las tareas del trabajo a realizar, además poseen la información necesaria para poder realizarla, poseen diferentes estados, “por hacer”, “en progreso” y “hecho”) Kanban. Además los límites de trabajo en proceso (WIP), esto se refiere a que en cada etapa/columna se establece un límite de tarjeta al mismo tiempo, su objetivo es evitar la sobrecarga y que se puede mantener un trabajo constante.

A su vez, comparte algunos conceptos con la metodología Scrum, aunque se encuentra diferencias en cinco campos: la cadencia; la metodología de publicación; las funciones; las métricas clave; la filosofía de cambios.

La cadencia en Kanban es de flujo continuo, en Scrum es mediante sprints de longitud fija periódicos. La metodología de publicación en Kanban es mediante una entrega continua o a discreción del equipo, en Scrum sería al final de cada sprint (si lo aprueba el propietario del producto). Las funciones en Kanban no hay, ya que los equipos cuentan con la ayuda de un orientador, en Scrum es el propietario del producto, un experto en scrum o un equipo de desarrollo. Las métricas clave en Kanban es el tiempo de ciclo, en Scrum la velocidad. Y la filosofía de cambios en Kanban pueden suceder en cualquier momento, en Scrum los equipos evitan los cambios en la previsión durante el sprint, ya que se sacrificaría el aprendizaje sobre la estimación.

9. SCRUM. Explica com funciona Scrum. Consulta els enllaços següents:

<https://proyectosagiles.org/que-es-scrum/>

<https://proyectosagiles.org/como-funciona-scrum/>

Es un marco de trabajo ágil que se enfoca en la entrega repetitiva de productos. Su método se basa en ciclos de desarrollo llamados “sprints” que generalmente tienen una duración de dos a cuatro semanas. Durante estos “sprints”, un equipo autoorganizado planifica, desarrolla y entrega un conjunto de funcionalidades prioritarias. Para asegurar la transparencia y fomentar la mejora continua, se llevan a cabo reuniones diarias de seguimiento conocidas como “Daily Standup”, así como reuniones de planificación de “sprint”, revisión y retrospectiva. Scrum se destaca por su enfoque en la colaboración, la adaptabilidad y la entrega constante de valor al cliente.

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 deber ser desarrollados en el proyecto. Esta lista es responsabilidad del “producto owner” y se utiliza como hoja de ruta para dirigir el proceso de desarrollo repetitivo del producto. El objetivo primordial es garantizar que las necesidades más críticas del cliente se atiendan primero.
- Sprint backlog: es una lista de actividades concretas y específicas extraídas del “producto backlog” que se eligen para ser llevadas a cabo durante un “sprint” en particular.

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. Pots consultar el següent enllaç:

<http://www.davidvalverde.com/blog/introduccion-a-la-programacion-extrema-xp/>

-Metàfora (metaphor): una historia de cómo funciona el sistema. Se utilizan para ayudar a que cualquier persona pueda entender el objeto del programa.

-Refactorización (refactoring): proceso mediante el cual a la hora de implementar nuevas características en los programas se busca la manera de hacerlo lo más simple posible, y su busca a su vez hacer el programa más simple sin perder funcionalidad.

-Programación por parejas (pair programming): a la hora de trabajar sobre el código de producción trabaja dos personas juntas, pero con un solo ordenador.

-Propiedad colectiva (collective ownership): se busca que cualquier persona pueda aportar al código en sí, es decir, nadie es el propietario.

-Estándares de codificación (coding standards): buscar un estándar en conjunto para todo el equipo.

-Cliente en casa (on-site customer): conlleva que un cliente este junto a los programadores durante el trabajo, para así poder estar disponible para responder a sus preguntas, resolver discusiones y fijar las prioridades.

-Integración continua (continuous integration): como su nombre indica se debe integrar, concretamente el código y realizar todas las pruebas necesarias para revisar que funcionan correctamente, se realizará con una parejas de programadores.

Llenguatges de programació

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

Los declarativos indican el resultado a obtener sin especificar los pasos, en cambio, los imperativos es al revés, indican los pasos que se deben seguir para obtener un resultado.

Declarativos: SQL y Prolog.

Imperativos: Java y C++.

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

Compilar implica transformar el código fuente creado 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 proceso de análisis por parte de un compilador, el cual verifica la estructura del código, resuelve las 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 creación de un archivo binario separado.

3. Avantatges dels llenguatges compilats.

Ventajas: rendimiento superior en comparación a un lenguaje interpretado; oculta el código fuente (capa extra de seguridad); cierta portabilidad; enfocados para proyectos grandes y complejos;

4. Avantatges dels llenguatges interpretats

Ventajas: portabilidad; flexibilidad en el desarrollo; depuración interactiva (ejecutar línea por línea permite revisar en tiempo real y poder corregir); facilidad a la hora de aprender (más sencillo para un principiante); menos tiempo de compilación.

5. Anomena 2 llenguatges compilats i altres 2 interpretats.

Lenguajes compilados: C y C++.

Lenguajes interpretados: PHP y JavaScript.

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

Si, pero no como el código objeto nativo, ya que es un lenguaje intermedio destinado a ser utilizado en la máquina virtual de Java.

7. Posa un exemple de llenguatge dels tipus següents:

- Baix nivel: lenguaje ensamblador (assembly language).
- Nivell mitjà: C++.
- Alt nivel: Python.

8. Quin paradigma de programació segueixen els llenguatges següents?

- C → Imperativa.
- C++ → admite varios: 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 posee a su vez: imperativa, estructurada, concurrente, genérica y funcional.
- Javascript → admite varios: orientada a objetos, funcional, basada en eventos, asíncrona, orientada a aspectos.
- Lisp → funcional.
- Prolog → lógica.

Pots consultar el següent enllaç:

https://ca.wikipedia.org/wiki/Paradigma_de_programaci%C3%B3n

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

Se tienen que tener en cuenta toda una serie: campo de aplicación; experiencia previa; herramientas de desarrollo; documentación disponible; base de usuarios; reusabilidad; transportabilidad; imposición del cliente.

Activitat final (no realitzar encara)

Elabora les respostes de les preguntes d'aquesta Unitat i guarda l'arxiu a GitHub en un repositori anomenat 'ActivitatsUD1_ED'.