

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.

- Disc dur: emmagatzema de forma permanent els fitxers executables i els fitxers de dades
- Memòria RAM: emmagatzema de manera temporal el codi binari dels fitxers executables i els fitxers de dades necessaris.
- CPU: llegeix i executa instruccions emmagatzemades en memòria RAM, així com les dades necessàries.
- E/S: recull noves dades des de l'entrada, es mostren els resultats, es llegeixen/guarden a disc, ...

2.Defineix els conceptes següents:

- **Codi font:** fitxer de text llegible escrit en un llenguatge de programació
- Codi objecte:** fitxer binari no executable
- Codi executable:** fitxer binari executable

Cicle de vida del programari

1.Defineix "Cicle de vida del programari".

Es el conjunt d'estats del progrés dels projecte de creació d'aplicacions informàtiques, per a poder identificar quant s'ha avançat i quant queda fins al final

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

- **Anàlisi:** es determina i defineix clarament les necessitats del client i s'especifiquen els requisits que ha de complir el programari a desenvolupar.
- **Disseny:** S'especifica la interrelació i la funcionalitat dels elements components
- **Codificació:** S'escriu el codi font de cada component.
- **Proves:** El principal objectiu de les proves ha de ser aconseguir que el programa funcioni incorrectament i que es descobreixin defectes.
- **Manteniment:** Durant l'explotació del sistema programari cal fer canvis ocasionals.

3. Avantatges i inconvenients del model en cascada.

Avantatges:

- Identifica les fases principals del desenvolupament de programari
- Hi ha diferents variants amb més o menys quantitat d'activitats.

Inconvenients:

- Model mes antic
- Es un model força rígid que s'adapta malament al canvi continu d'especificacions.
- Les fases s'han de fer en l'ordre indicat

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

Es crea un prototip durant la fase d'anàlisi i és provat per l'usuari/client per refinar els requisits del programari a desenvolupar .

Existeixen 2 tipus de prototips:

- **Prototips ràpids:** el prototip pot estar desenvolupat usant un altre llenguatge i/o eines.
- **Prototips evolutius:** el prototip està dissenyat en el mateix llenguatge i eines del projecte, este s'usa com a base per a desenvolupar el projecte.

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

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

- Entregues ràpides i contínues de programari utilitzable
- Recepció de requeriments canviants fins i tot en desenvolupament tardà
- Programari amb lliuraments constants
- Col·laboració diària i pròxima entre els negocis, els clients i els desenvolupadors
- Els crèdits es construeixen entre individus motius en el que s'ha de confiar
- Sessions cara a cara
- Programació en parells
- Desenvolupament sostenible
- Excel·lència a traves de la reflexió
- Simplicitat
- Equips auto organitzats
- Canvi constant a les circumstancies 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

Una història d'usuari és una descripció breu i informal del que vol fer l'usuari dins d'un producte de programari per a obtenir resultats òptims del projecte final.

Ex: Com a Client, vull consultar els productes venuts per a poder tindre un seguiment.

7. KANBAN. Estudia els avantatges i els inconvenients de tenir una pissarra web digital per a la metodologia Kanban. Pots consultar els següents enllaços:**Avantatges:**

- Fàcil de manejar.
- Benefici de l'enfocament visual, utilitzant columnes, carrils i les targetes de colors.
- Es pot portal un control del material per a facilitar el detectar els materials defectuosos

Inconvenients:

- El sistema no té cap anticipació en cas d'imprevisions en la demanda.
- No s'implementa bé en cicles productius llargs.

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

La metodologia KanBan es tracta d'un mètode visual de gestió de projectes que permet als equips visualitzar els seus fluxos de treball i la càrrega de treball. En un tauler Kanban, el treball es mostra en un projecte en forma de tauler organitzat per columnes.

Mentre que Scrum s'utilitza per al desenvolupament de productes més complexos. És un mètode de treball per a aconseguir el millor resultat d'un projecte.

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

La metodologia SCRUM consisteix a abordar qualsevol projecte dividint-lo en sprints o parts més xicotetes. Dins d'aquest entorn de treball cal seguir una sèrie de fases per a abordar cada tasca, i participen uns rols específics que garanteixen el compliment d'aquesta filosofia de treball.

10.SCRUM. Defineix els termes següents:

- **Product backlog:** és una eina essencial per a la gestió de projectes que consisteix en l'elaboració d'un llistat de totes aquelles tasques que volem realitzar durant el desenvolupament d'un projecte amb l'objectiu que aquestes siguin visibles per a tot l'equip.
- **Sprint backlog:** és el subconjunt d'elements del Product Backlog triats per a abordar-se durant l'sprint. Aquests elements es componen de tasques tècniques més xicotetes que aconsegueixen un increment del programari.

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

- Cap de projecte:** Product Owner
- Client:** Scrum Master
- 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/>

- **Sienzillesa:** Es busca un disseny simple i elegant, evitant la sobreingenieria.
- **Comunicació:** La XP ajuda mitjançant les seues pràctiques a la comunicació entre els integrants del grup de treball: caps de projecte, clients i desenvolupadors.
- **Refactorització contínua:** Millora del codi sense canviar la seua funcionalitat.
- **Estàndards de codificació:** Ús d'estàndards i pràctiques de desenvolupament coherents.
- **Propietat col·lectiva del codi:** L'equip comparteix la responsabilitat del codi.
- **Client en el lloc:** Un representant del client treballa de prop amb l'equip de desenvolupament.

Llenguatges de programació

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

Un llenguatge declaratiu sol defineix l'estat final, sempre s'acaba en el mateix lloc, sense importar on es comence. D'altra banda, un llenguatge imperatiu concep una tasca com una sèrie de passos predefinits que poden portar-li a un punt final diferent depenent del punt de partida.

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

Un compilador és un programa que tradueix codi escrit en un llenguatge de programació a un altre llenguatge. En aquesta mena de traductor el llenguatge font és generalment un llenguatge d'alt nivell i l'objecte un llenguatge de baix nivell, com assembly o codi màquina

La interpretació és un procés en el qual un programa de programari, executa directament el codi font d'un programa sense necessitat de compilar-lo prèviament a codi de màquina.

3. Avantatges dels llenguatges compilats.

- Major rendiment
- Execució més ràpida
- Major seguretat
- Optimització de recursos
- Programes independents

4. Avantatges dels llenguatges interpretats

- Portabilitat
- Multiplataforma
- Desenvolupament ràpid
- Facilitat de depuració
- Flexibilitat i dinamisme

5. Anomena 2 llenguatges compilats i altres 2 interpretats.

Compilats: Erlang, C++

Interpretats: Perl, JavaScript

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

El bytecode a Java és una forma de codi objecte. Encara que no és codi màquina directament executable, és una representació intermèdia en forma de codi binari dissenyada per a la Màquina Virtual Java.

7. Posa un exemple de llenguatge dels tipus següents:

- Baix nivell: Assembler
- Nivell mitjà: C
- Alt nivell: Java

8. Quin paradigma de programació segueixen els llenguatges següents?

- C: Imperatiu
- C++: Multiparadigma
- SQL: Declaratiu
- Java: Orientat a objectes
- Javascript: Multiparadigma
- Lisp: Multiparadigma
- Prolog: Lògic

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.

Requisits del projecte: Selecciona un llenguatge que s'adapte a les necessitats específiques del teu projecte.

Experiència de l'equip: Opta per un llenguatge que el teu equip conega i manege bé per a una major eficiència.

Rendiment: Avalua el rendiment del llenguatge, especialment si necessites una alta velocitat o eficiència.

Facilitat de manteniment: Considera la facilitat de manteniment i l'escalabilitat a llarg termini del llenguatge triat.

Seguretat: Considera la seguretat del llenguatge i la seua capacitat per a previndre vulnerabilitats.

Plataforma i sistema operatiu: Verifica si el llenguatge és compatible amb la plataforma i el sistema operatiu.

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'.