Matthew Farrell

2. conceptos basicos

software de sistema
- sistema operativo drivers y controladores
- programas que permiten interactuar con el sistema operativo

software de aplicacion
- Suites aof, navegadores, juegos, edcion de imagen
- programas que permiten hacer trabajos

software de desarrollo
- editores, compiladores e interpretes
- programas que permite crear otro software y programas

Firmware: software de bajo nivel que esta cercano y comunica con el hardware

2.2. relacion entre hardware software

- DISCO DURO almacena de forma permanente los archivos ejecutables y archivos de datos

- memoria RAM almacena de forma temporal el codigo binario de los archivos ejecutables y los archivos de datos necesarios

- CPU lee y ejecuta instrucciones almacenadas en RAM, asi como los datos necesarios.
- La CPU tiene cache L1,L2 y L3

- E/S recoge nuevos datos desde la entrada, se muestran resultados, se leen/guardan a disco

2.3. Codigo fuente, objeto y ejecutable

- código fuente: archivo de texto legible escrito en un lenguaje de programacion.
- Permite modificar el programa de una forma sencilla

- codigo objeto: archivo binario no ejecutable
- codgio que se genera a partir del codigo fuente que no se entiende

- codigo ejecutable: archivo binario ejecutable
- solo valido para lenguajes compilados como JAVA C C++
- En lenguajes interpretados no existe codigo binario, solo codifo fuente como PHP y javascript
- el codigo objetivo en java se denomina ByteCode

3: ciclo de vida de software

- 3.1 ingenieria de software
- Estudio de los principios y metodologias para desarrollo y mantenimiento de software

Fases principales

- analisis saber que necesitamos

- diseño como implementar lo que sabemos para solucionar y crear el software

- codificacion escribir el codigo

- pruebas testear y probar el codigo para saber si funciona bien el codigo

- mantenimiento y documentacion

ANALISIS
- definir los requisitos del cliente para cumplir el software a desarrollar
  - ser completo y sin omisiones
  - ser conciso y sin trivialidades
  - evitar ambiguedades y utilizar lenguaje formal
  - evitar detalles de diseño
  - sea entendible por el cliente
  - separar requisitos funcionales y no funcionales
  - dividir y ordenar por jerarquia el modelo
  - fijar criterios de validacion

DISEÑO
- se descompone y organiza el sistema en elementos que se pueden ser desarrollados por separado
- se especifica la interrelacion y funcionalidad de los elementos componentes
- Las actividades habituales son las siguientes:
  - diseño arquitectonico,
  - diseño detallado
  - diseño de datos
  - diseño de interfaz de usuario

CODIFICACION

- se escribe el codigo fuente
- se utilizan lenguajes
  - de programacion como C, C++, Java, Javascript
  - Otro tipo como HTML, XML, JSON

PRUEBAS

- conseguir y chequear que el programa funciona correctamente sin fallos
  - someter el programa al maximo numero de situaciones diferentes

MANTENIMIENTO

- en la vida del producto hay que realizar cambios ocasionales y revisiones
- hay que rehacer parte del trabajo realizado en las fases previas
  - correctivo: se corrigen defectos

  - Perfectivo: se mejora la funcionalidad

  - evolutivo: se añade funcionalidades nuevas

  - adaptativo: se adapta a nuevos entornos

3.2.6 resultados de cada fase

- analisis: especificacion de requisitosde l software
- DISEÑO aequitectonico: documento de arquitetura del softwre

- DISEÑO detallado: especificacion de modulos y funciones
- CODIFICACION: codigo fuente
- PRUEBAS de unidades
- pruebas de integracion: sistema utilizable
- Pruebas del sistema: sistema aceptado
- Documentacion: Documentacion tecnica y usuario
- Mantenimiento informes de errores y control de cambios

4. Modelos de desarrollo

4.2 Modelo en cascada:

Modelo tiene problemas, las fases van por orden y el resultado de una fase es el empiezo de otra, si un a fase va mal la otra fase puede tener problemas al empezar

4.4 Modelo en V:
Vision jerarquizada con distintos nibeles

4.6: Prototipos
A menudo los requisitos no estan especificados claramente

- por no existir experiencia previa
- por omision o falta de concrecion del usuario/cliente

4.7 Prototipos 2
Se crea un prototipo durante la fase de analisis y es probado por el usuario/cliente para refinar los requis itos del software a desarrollar
- Se repite el paso anterior las veces necesarias

4.8 Prototipos 3
- Prototipos rapidos
  - el prototipo puede estar desarrollado usando otro lenguaje o herramientas
- Prototipos evolutivos
  - El prototipo se dieseña en el mismo lenguaje y herramientas del proyecto
  - El prototipo se usa como base de desarrollo del proyecto

4.11 Metodologias agiles 1
- Son metodos de ingenieria del software basaados en el desarrollo iterativoe incremental
- Los requisitos y soluciones evolucionan con el tiempo segun la necesidad del proyecto
- El trabajo es realizado mediante la colaboracion de equipos auto-organizafos y multidisciplinarios, inme rsos en un proceso compartido de toma de decisiones a corto plazo
- Las metodologias mas conocidas son
  -KanBan
  -Scrum
  -XP extreme programming

4.13 Metodologias agiles 2
- Individuos e interacciones sobre procesos y herramientas
- Software funcionande sobre documentacion extensiva
- Colaboracion con el cliente sobre negociacion contractual
- respuesta ante al cambio sobre seguir un plan

4.12.1 Kanban 1

- Sistema de tarjetas
- Controla por demanada la fabricacion de los productos necesarios en la cantidad y tiempos necesarios
- Enfocado a entregar el máximo valor para los clientes utilizando los recursos justos

4.12.3 SCRUM conceptos
- Modelo de desarrollo incremental
- Iteraciones (sprint) cada 2 a 4 semanas
- Al principio de cada iteracion se establecen sus objetivos priorizados (sprint backlog)
- Al finalizar cada iteracion se obtiene una entrega parcial utilizable por el cliente
- Existen reuniones diarias para tratar la marcha del sprint

4.12.5 SCRUM roles principales
- Product owner es la "voz del cliente" Define criterios de aceptacion y asegura que se cumplan
- Scrum Master Asegura que se sigue la metodologia Scrum. Motiva y facilita el trabajo del equipo
- Team es el equipo de desarrollo auto-organizado y multifuncional. Entre 6 y 10 miembros.

4.12.6 SCRUM artefactos
- Product Backlog es la lista ordenada con los requistios del producto
- Sprint backlog es la lista de requisitos sacados del backlog para su desarrollo durante el sprint
- Incremento es el estado del producto despues de cada sprint

4.12.7 SCRUM eventos
- Sprint evento principal, que contiene el resto de eventos como mucho dura 1 mes
- Sprint planning reunion inicial donde se planifica el trabajo del sprint dura como mucho 8 horas
- daily scrum es la reunion diaria de puesta en comun sobe la marcha del Sprint. Duracion maxima es de 15 minutos
- Scrum review es la reunion final para evaluar el incremento obtenido. duracion maxima es de 4 horas

5. lenguajes de programacion

5.1 Obtencion de codigo ejecutable

Para obtener codigo binario ejecutable tenemos
/ compilar
/ interpretar

5.2 Porceso de compilacion/interpretacion

La compilacion/interpretacion del codigo fuente se lleva a cabo en dos fases:
1: Analisis léxico
2: Analisis sintáctico
- si no hay errores se genera el codigo objeto correspondiente
- Un codigo fuente correctamente escrito no significa que funciona segun lo deseado
- NO se realiza un análisis semántico

5.3 Lenguajes compilados
- C, C++
- Principal ventaja: Ejecucion muy eficiente
- PRincipal desventaja: Es necesario compilar cada vez que el codigo fuente es modificado

5.4 Lenguajes interpretados
- PHP, javascript
- Principal ventaja: el codigo fuente se interpreta directamente
- principal desventaja: ejecucion menos eficiente

5.5 JAVA 1

- lenguajes compilados e interpretados
- El codigo fuente Java se compila y se obtiene un código binario intermedio llamado bytecode
- Puede considerarse código objeto peero destinado a la máquina virtual de Java en lugar de código obje to nativo
- Después eeste bytecode se interpreta para ejecutarlo

5.6 JAVA 2

- Ventajas
  estructurado y orientado a objetos
  relativamente facil de aprender
  buena documentacion y base de usuarios
- Desventajas
  Menos eficiente que los lenguajes compilados

5.7 Tipos 1

- Segun en la forma que operan
  Declarativo: indicamos el resultado a obtener sin especificar los pasos
  Imperativo: indicamos los pasos a seguir para obtener un resultado

5.8 Tipos 2

- Tipos lenguajes declarativos:
  - Logicos: utilizan reglas. Ej: Prolog
  - Funcionales: utilizan funciones: Ej Lisp, HAskell
  - Algebraicos: utlizan sentencias: Ej SQL
- normalmente son lenguajes interpretados

5.9 Tipos 3

- Tipos de lenguajes imperativos:
  - Estructurados: C
  - Orientados a objeto: Java
  - Multiparadigma: C++, Javascript
- Los lenguajes orientados a objetos son también lenguajes estructurados
- Muchos de estos lenguajes son compilados

5.10 Tipos (IV)

- Tipos de lenguajes segun nivel de abstraccion
  - Bajo nivel: ensamblador
  - Medio nivel: C
  - Alto nivel: C++, Java

5.11 evolucion

- Codigo binario
- Ensamblador
- Lenguajes estructurados
- Lenguajes orientados a objetos

5.12 Criterios para la seleccion de un lenguaje

- Campo de aplicacion
- experiencia previa
- herramientas de desarrollo
- Documentacion disponibl
- base de usuarios
- reusabilidad
- transportabilidad
- imposicion del cliente

nota unidad 1 tener repositorio bien
examen tipo test en el ordenador 20 preguntas