CURS 1

Bla bla bla software engineering

Ingineria sistemelor soft presupune: modelare, rezolvare de probleme, acumularea de cunostinte, argumentare

Un model este o reprezentare abstracta (si simplificata) a unui sistem

Modelarea este procesul de repezentare a elementelor sau esentei unui sistem sau fenomen

Produs = artefact realizat in timpul procesului de dezvoltare

Activitate = o multimer de sarcini realizate cu un anumit scop

Sarcina = o unitate atomica, gestionabila de lucru

Resurse = bun utilizat pentru realizarea unor sarcini

Cerinta – functionala sau nefunctionala(ceva legat de operarea sistemului)

Metoda = succesiune de pasi apicati in scopul rezolvarii unei anumite probleme

Metodologie = colectie de metode folosite ptr rezolvarea unei anumite clase se probleme

ACTIVITATI TEHNICE ALE ING. SOFT ORIENTATA OB. – colectarea cerintelor, analiza cerintelor,

Proiectarea de sistem, proiectarea obiectuala, implementarea, testarea

MODEL FUNCTIONAL

Actor = rol jucat de on enitate externa sistemului, care interactioneaza cu sistemul

Caz de utilizare = secventa generala de evenimente descriind toate interactiunile posibile intre un actor si sistem

CU – ID and name, primary actor, secondary actor, description, trigger, preconditions, postconditions, normal flow, alternative flows, exceptions, conditions for quality

Model obiectual: Model conceptual -> diagrama de clase

Model dinamic -> diagrama de secventa si de comunicare/interactiune

CURS 2

O notatie (un limbaj) reprez o multimer de reguli folosite ptr specificarea modelelor

Relatiile posibile in diagramele de cazuri de utilizare: comunicare, incluziune, extindere, generalizare

UML – clase, data type, enumeration

Obiecte – instante ale claselor

Asociere, agregare, compunere, generalizare

Diagrama de tranzitie

Stare – o conditie satisfacuta de valorile atributelor unui obiect

-activ, inactive, inchis, arhivat

Actiune – o unit. Fundamentala de procesare, care poate primi input-uri, poate produce output-uri si poate schimba starea sistemului

Activitate – mutlime coordonata de actiuni

Diagrama de activitati – descrie modul de realizare a unui anumit comportament (intre diagrame de activitati ??)

-idk ceva noduri decizionale

CURS 3 – COLECTAREA CERINTELOR

Cerinta – un element de functionalitate pe care sistemul trebuie sa il ofere sau o constrangere pe care trebuie sa o indeplineasca

Ingineria cerintelor = colectare + analiza

Colectarea cerintelor – identificarea actorilor, scenariilor, cazurilor de utilizare, rafinarea cazurilor de utilizare, ident. relatiilor intre cazurile de utilizare, ident. cerintelor nefunctionale

Validarea cerintelor presupune verificarea completitudinii(cuprinde toate aspectele de interes ptr clienti), consistentei(sa nu existe cerinta contradictorii), claritatii, corectitudinii(sa reprez fidel insteresele clientului)

-Realism, verificabilitate si trasabilitate

Metoda JAD

Activitati JAD – project definition, research, preparation, session, final document preparation

CURS 4

Entity – responsabile de informatia persistenta din sistem

Boundary – reprez inerfata sistemului cu actorii

Control – responsabile de coordonare claselor entity si boundary

Diagrame de secventa - prezinta comportamentul din perspectiva unui singur caz de utilizare

Diagrame de tranzitie a starilor - prezinta comportamentul din perspectiva unui singur obiect

CURS 5

Proiectarea de sistem – obiective de proiectare, arhitectura sistemului

Descompunere in subsisteme

Serviciu = multime de operatii inrudite

Subsistemele sunt caracterizate de serviciile pe care le ofera altor subsisteme

Ceva interfata

API(application programming interface)

Conector ball-and-socket

Ball – interfata oferita, socket, interfata solicitata

Cuplare = masura a dependentei dintre doua subisteme

Coeziune = masura a dependentelor din interiorul unui subsistem\

Se vrea cuplare slaba, coeziune inalta/ridicata

STILURI ARHITECTURALE

1.Repository

2.Model-View-Controller

◦ Model - reprezinta informatii/cunostinte din domeniul problemei

◦ View - afiseaza aceste informatii utilizatorului

◦ Controller - translateaza interactiunile cu view-ul în actiuni asupra modelului

3.Client-Server

4.Peer-to-peer

5.Three-tier architecture = user interface + application logic + storage

6.Four-tier architecture = presentation client + presentation server + application logic + storage

7.Pipes and filters – fiecare subsistem filtru are un canal de intrare si unul de iesire

Proiectare de sistem – definirea politicilor privind controlul accesului, stabilirea fluxului global de control, descrierea cazurilor limita(pornire, oprire, gestionare chestii – admin stuff)

Sablon de proiectare

Façade – permite reducerea dependetelor dintre un subsistem si clientii sai, prin intermediul unei interfete unificate, de nivel inalt, (inlocuind) pentru un grup de interfete ale unui subsistem

Proxy – surogat in scopul controlarii accesului la acesta

CURS 7_8

SOLID

Sabloane de proiectare (bune pentru modificari in proiectare)

1.Adapter – o alta interfata intre o clasa “legacy” si interfata clientului

2.Bridge – permite substituirea dinamica a implementarilor posibilie ale unei aceeasi interfete (care pot fi mai abstracte sau mai concrete)

3.Strategy (ptr incapsulare algoritmilor) – ptr interschimbarea in mod dinamic a unui algoritm folosit

4.Composite – permite reprezentarea unor ierarhii de latime si adancime variabila, astefl incat agregatele si frunzele sa fie tratate uniform, prin intermediul unei interfete commune

5.Abstract factory – o interfata cu operatii pentru crearea de diferite obiecte de tipul unei clase abstracte

6.Command – incapsuleaza o cerere ca si un obiect, folositor pentru cereri de la client si formarea de cozi de cereri

7.State

8.Singleton

9.Proxy

10.Facade

CURS 9

DESIGN BY CONTRACT

La apelul unei subrutini sunt necesare niste specificatii contractuale intre client si cel care furnizeaza subrutina

- Pre-conditii – obligatii pentru client si beneficii pentru furnizator, post-conditii – obligatii pentru furnizator si beneficii pentru client

In unele cazuri, aceste conditii nu mai sunt verificate in momentul apelului la furnizor si pot declansa erori (daca clientul nu respecta pre-conditiile)

PROIECTAREA OBIECTUALA: SPECIFICAREA INTERFETELOR

OCL =)

CURS 10

Modele si transformari

Transformari la nivelul modelului – modificari pe model

Refactorizari – modificari pe cod sursa

Inginerie directa – din model obiectual -> fragment de cod

Inginerie inversa – cod sursa -> model

Fiecare transformare trebuie sa vizeze optimizari din perspectiva unui singur criteriu

Fiecare transformare trebuie sa fie locala

Fiecare transformare trebuie aplicata izolat de alte schimbari

Fiecare transformare trebuie urmata de validari aferente

+ idk ceva detalii transformari, asocieri, exemple cod, ocl

CURS 11

TESTAREA SISTEMELOR SOFT

Testarea = procesul de identificare a diferenetelor dintre comportamentul dorit al sistemului si cel observat

Testarea – unitara, structurala, functionala, performantei

ALTE (SEMINARII + STUFF)

Scenariu = o secventa de pasi/evenimente, care descrie o interactiune tipica a unui utilizator cu sistemul

Caz de utilizare = o multime de secvente de evenimente, care descriu toate interactiuniile posibile intre un utilizator si sistem, in scopul realizarii unei anumite functionalitati; multimea tuturor scenariilor ptr o functionalitate ~