OOM Loeng 2: Ülevaade UML diagrammitehnikatest ja iteratiivsest arendusprotsessist.

Eesmärgid

1. Aine põhiraamistike tutvustamine ja analüüs

Kava

1. UML diagrammitehnikad
2. Ühtne arendusprotsessi raamistik

NB! Antud loengus kasutatud skeeme leiate Rational Rose lingi alt, mille püüan siia homme lisada

1. UMLi diagrammitehnikad

UML mudelite arhitektuur

UML ühendab kolme varem iseseisvalt eksisteerinud modelleerimiskoolkonda:
Objektmodelleerimine (OMT, Rumbough),
Use Case (ehk rollikeskne, funktsionaalne) modelleerimine (OOSE, I. Jakobson),
Äriprotsesside modelleerimine.

Diagrammide liigitus

1. Staatika diagrammid:

Klassidiagramm (Class Dgm):

Sarnaneb klassikalisele Olemi-Suhte diagrammile
Klassid, atribuudid, operatsioonid
Assotsiatsioonid, agregatsiooni- ja üldistus-pärimishierarhiad
Kontseptuaalne tase (Analüüsis), tarkvaraklassid (Disainis)

Use Case diagramm:

Süsteemi funktsionaalne "kontekst":
Tegutsejarollid (actors) ja nende osalemine Süsteemi protsessides (kes mida teeb?). Süsteemi kui "musta kasti" funktsioonid / teenused väljapoole (Tegutsejatele).
Funktsioonide-protsesside (use case'id) omavahelised seosed.
Kirjeldused ja stsenaariumid. Seos dünaamikadiagrammidega.

2. Dünaamikadiagrammid:

Jadadiagramm (Sequence Dgm):

Protsesside (use case'id) sündmusstsenaariumid ajas. Use case'i stsenaarium. Ajatelg, objektide elujooned, objektide sõnumivahetus. Sündmus kui sõnumivahetus kahe objekti vahel. Operatsioonide käivitamine. Alamstsenaariumid. Jadadiagramm kui analüüsi vahend.

Koostöödiagramm (Collaboration Dgm)

Loogilise disaini põhivahend. Sarnasus jadadiagrammiga. Esitab objektide sõnumivahetust (sündmusi) ruumis. Ajatelge otseselt pole. Sõnumitel (hierarhilised) järjenumbrid. Võimalikud kasutada programmi konstruktsioone (valik, kordus,..). Koostöödiagramm luuakse operatsiooni või use case'i kohta.

Jada ja koostöödiagrammi "ühine nimetaja" on interaktsioonidiagramm (suhtlusdiagramm).

Olekudiagramm (State Dgm):

Väljendab objekti elutsüklit. Nagu klassikaline State-Transition diagramm, ainult palju rohkem detaile ja täpsust. Ooteseisundid ja tegevusseisundid.

Tegevusdiagramm (Activity Dgm)

Väljendab protsessi elutsüklit. On olekudiagrammi erijuhtum, kus kõik seisundid on tegevusseisundid. Sarnaneb klassikalisele programmi plokkskeemile. Väljendab töö kulgemist ajas ja organisatsioonis. Sobib hästi äriprotsesside modelleerimiseks.

Oleku- ja tegevusdiagrammi ühisnimetajaks on "elutsükli diagramm"

Komponentdiagramm (Component Dgm):

Programmeerija-ehitaja tööriist. Tarkvarakomponendid, nende liidesed ja sõltuvused. Komponent kui "kest" (liideste hulk) tarkvaraobjektidele.

Rakendusdiagramm (Deployment Dgm):

Süsteemi administraatori ning arhitekti tööriist. Võrgu sõlmed (Nodes) ja ühendused (Connections). Nende stereotüübid ja ikoonid. Komponentide lohistamine sõlmede vahel.

Komponendi ja rakendusdiagrammid kokku on realisatsioonidiagrammid

Iteratiivse arendusprotsessi raamistik: Unified Process

Klassikaline jadaarendus (Oracle CASE Meetod). Kose mudel. Selle puudused: Ülipikk arendustsükkel (poolteist kuni kaks aastat). Pikad järjestikused etapid, mis ei anna Tellija jaoks "käegakatsutavat" tulemust. Tagasiside saabub liiga hilja. Süsteem on kogu aeg "jalad õhus".

Iteratiivne inkrementaalne arendamine:

Toote/Teenuse (edaspidi Süsteem) kogu elu sünnist surmani sisaldab Toote/Teenuse erinevaid variante / versioone.
Iga versioon läbib nn. "makroprotsessi", mille faasideks on:

- Visiooni loomine
- Visiooni arendamine
- Visiooni realiseerimine (konstrueerimine)
- Rakendamine

Iga faas viiakse läbi N iteratsiooniga. Iteratsioonide arv ei ole protsessi käivitamisel teada. Iga iteratsioon realiseerib ning

integreerib terviksüsteemi ühe Tellijale/Kasutajale olulise omaduse/väärtuse ehk inkremendi. Igas iteratsioonis viiakse läbi planeerimise, analüüsi, disaini, arhitektuuri, programmeerimise, testimise jne. tegevusi. Nii saavutatakse arendusprotsessi maksimaalne juhitavus, kiire tagasiside, kuna ühe iteratsiooni kestus on kahest nädalast kahe kuuni. Iteratsioonid võivad toimuda)osaliselt) paralleelselt.

Meie kasutame iteratiivse arendusprotsessi pisut lihtsamat varianti, kus makroprotsessi kaks esimest faasi (planeerimine ja täpsustamine) on kokku pandud:

1. Planeerimine ja täpsustamine (visioon) (ühe iteratsiooniga)
2. Ehitamine: arendustsüklid (N iteratsiooni, igaüks mingi use case'i arendamiseks ja realiseerimiseks)
3. Rakendamine (use case'i töölepanek Tellija keskkonnas)