

## Loeng 3. Arendusprotsess: Andmebaaside arendamine organisatsioonis

Loengu eesmärk:

- Anda ülevaade andmebaasi arendusprotsessist tervikuna, tema kohast infosüsteemi arendusprotsessist
- Tutvustada andmebaaside arendusmetodoloogiale esitatavaid nõudeid
- Tutvustada andmebaaside arendamise tegevusi, nende tegevuste objekte, vajalikke rolle andmebaaside arendusmetodoloogia rakendamisel konkreetses organisatsioonis
- Ülevaade andmebaaside arendusmetodoloogia valiku küsimustest

Sisu:

- Mis on andmebaaside arendusmetodoloogia?
- Andmebaaside arendusmetodoloogia valimine
- Andmebaaside arendusmetodoloogia komponendid:
  - Arendustegevused
  - Arenduse tulemused
  - Arendusorganisatsioon

## Mis on andmebaaside arendusmetodoloogia?

Andmebaaside arendusprotsessi organiseerimine konkreetsetes firmades / projektides nõuab kaasaegsete arendusmetodoloogiate kasutamist.

Andmebaaside arendusmetodoloogiad on alamtihulk infosüsteemi kogu elutsükli toetavatest metodoloogiatest, mida saab kasutada andmebaaside ehitamisel alates vajaduste defineerimisest/analüüsist läbi projekteerimise kuni füüsilise realiseerimiseni.

Selleks, et saavutada andmebaaside arendamisel edu, peab arendusmetodoloogia olema iteratiivne, kasutajat laialdaselt kaasahaarav, sisaldama taaskasutatavaid dokumendimakette ning näidistulemusi.

Kui otsitakse organisatsiooni jaoks andmebaaside arendusmetodoloogiat, tasub valida kergesti omandatav ja kasutatav ning elektrooniliselt realiseeritud protsess, mida arendusmeeskonnad saavad läbi võrgu kiiresti rakendada erinevates asukohtades.

Vaatamata uutele alternatiividele (universaalsed andmebaasid, objektiandmebaasid, multidimensionaalsed andmebaasid jne.) on relatsioonilised andmebaasid praegu ning lähitulevikus enamiku transaktsioonitöötuse (Online Transaction Processing) rakenduste ning ettevõtete infosüsteemide aluseks.

Relatsioonilise andmebaasilahenduse ehitamise keerukust saab vähendada kasutades projektides läbiproovitud andmebaaside arendamise metodoloogiaid. Sellised metodoloogiad annavad edasi heade projektimeeskondade parimaid kogemusi ja praktikaid, võimaldavad oluliselt vähendada projektiga seotud riske ning tähtaegu, tõsta projektitulemuste kvaliteeti.

Metodoloogia defineerib eduka projekti tegevused ja nende tegevuste tulemused. Metodoloogia kasutamine lihtsustab projekti kulgemise jälgimist/mõõtmist projektijuhi poolt. Metodoloogia annab näidisvormid / -tulemused arendusprotsessi kiireks alustamiseks.

## Andmebaasi arendusmetodoloogia valik

Hästidefineeritud andmebaaside arendamise faase sisaldavad arendusmetodoloogiad on turul üldiselt kättesaadavad. Paljud neist sisalduvad modelleerimise või projektijuhtimise vahendites (näit. ERWin CASE,..). Paljud suuremad konsultatsioonifirmad on välja arendanud oma metodoloogiad lähtudes oma firma projektide kogemustest ning parimast praktikast. Nad müüvad metodoloogiat eraldi või muude konsultatsiooniteenuste koosseisus.

Iga andmebaaside arendusmetodoloogia, mida tasub organisatsioonides rakendada, peaks sisaldama enamikku järgnevatest omadustest:

- Liides mõne täielikku elutsükli toetava infosüsteemi arendusmetodoloogiaga. Täielikku elutsükli omav metodoloogia toetab enamat kui andmebaasi arendamine. Valitav andmebaasi arendamise metodoloogia peaks olema täieliku elutsükli metodoloogia komponent või peaks omama liidest vähemalt ühega neist. Kui seda nõuet ei järgita, võivad tagajärjeks olla kokkusobimatud tehnikad või mittekasutatavad projektitulemused.
- Erinevate tehnikate kombineerimine tulemuste saavutamiseks. Traditsiooniline struktuurne lähenemine (“veelange” mudel) näeb ette tulemuste saavutamist järjestikuste tegevustega. Tulemust B ei tohi alustada enne, kui tulemus A on lõpetatud ja allkirjastatud. See lähenemine viib lõpptulemuseni, kuid liiga aeglaselt. Viimasel aastakümnel on saavutanud populaarsuse kiire rakenduse arendamise (RAD) lähenemine, mis annab tulemused palju väiksema ajaga, kasutates paralleeltööd, iteratsiooni ja prototüüpimist. RAD on kasutatud edukalt väiksemate projektide jaoks, kuid on problemaatiline suuremate projektide korral keerukate ärivajaduste tõttu. Suhteliselt uuem lähenemine kombineerib struktuurse ja RAD lähenemise parimaid omadusi ning on rakendatav ka suurtele projektidele.
- Tugi: Arendusmetodoloogia on toode, ükskõik kas organisatsioon on selle eest maksnud või mitte. Seepärast on tähtis metodoloogia toetamine tootja poolt tulevikus. Vastasel korral muutub metodoloogia kiiresti kasutuks.

- Hind: Metodoloogia hinda tuleb arvestada tervikuna ja osade kaupa ning kaaluda oodatava tulemuse suhtes. Arvestada tuleb ühekordseid kulusid, koolituskulusid, uuendamise (upgrade) kulusid, aasta litsensi maksumusi, kulutusi kasutaja kohta, häälestamise / kohandamise maksumust, riistvara/tarkvara toe maksumust, kulutusi tulevastele versioonidele/väljalasetele.
- Tootja: Tuleb arvestada metodoloogia tootja stabiilsuse ja turupositsiooniga. Stabiilsed ja suuremat turuosa omavad tootjad täiendavad suurema tõenäosusega oma metodoloogiat uute tehnikatega tulevikus.
- Läbiproovitud edukus: Kõige kindlam tee metodoloogia valikul on kontrollida sarnaseid organisatsioone, kes seda edukalt on kasutanud oma arendusprojektides.
- Elektrooniline kättesaadavus: Metodoloogia peaks olema kättesaadav elektrooniliselt läbi Interneti või CD-ROM-i. Samuti peaks ta olema kirjas paberkujul.
- Maketid / Näidistulemused: Taaskasutatavad maketid (Templates) ja näidistulemused võimaldavad projektimeeskonnal kiiresti alustada arendusprojekti täitmist. Kui selliseid näidismakette / -tulemusi Kliendile ei anta, peaks viimane need ise arendama pilootprojekti käigus. Ehkki see aeglustab pilootprojekti kulgu, kulgevad järgnevad projektid siis efektiivsemalt.
- Liides uuemate arhitektuuridega: Metodoloogia peaks toetama ühendusi moodulitega, mis toetavad andmevaramuid, objektitehnoloogiat, elektronkommertsit, Web arhitektuure. Soovitav on metodoloogia laiendatavus nimetatud omadustega.
- Lihtne õppimine ja kasutamine: Koos vastavate koolituskursustega.

## Metodoloogia komponendid:

Järgnevalt kirjeldatud arendustegevuste loetelu on piiratud andmebaasispetsiifiliste tegevustega. Kirjeldatud alamprotsessid sobituvad suuremasse täieliku elutsükliga infosüsteemi arendusmetodoloogiasse, mis käsitleb andmebaasi arendusmetodoloogia raamistikust välja jäävaid arendustegevusi (n. organisatsiooni ja meeskondade ehitamine, kasutajaliidese arendamine, rakenduse disain, tehnoloogia valik,...)

### **Äri vajaduste defineerimine / anaslüüs**

Äri vajadused defineeritakse igasuguse süsteemiarenduse korral. Neid kasutatakse ka loogilise andmemudeli ehitamiseks. Määratakse (muuhulgas) olemite arv, atribuutide nimed ja andmetüübid. Sageli jaotatakse need ainevaldkondadesse (Subjekt Area), millest võivad saada (kuid ei pea saama) iseseisvad andmebaasid.

### **Andmemudeli hankimine või loomine**

Kui ollakse aru saanud äri vajadustest, on kasulik otsida turult andmemudelit, osta sobiv ning kohandada oma organisatsioonile.

### **Loogilise andmemudeli ehitamine**

Loogilist andmemudelit ehitatakse iteratiivselt. Esimene vaade tavaliselt kõrgel tasemel, alustades ainevaldkonna või kontseptuaalse andmemudeliga. Järgnevad tasemed sisaldavad enam detaile. Ka normaliseerimist rakendatakse sellel sammul. Välisvõtmeid ja potentsiaalseid indekseid ei käsitleta siin. Ei arvestata veel andmebaasi jõudlust, füüsilise taseme probleemid jäävad hilisematesse sammudesse.

### **Andmemudeli õigsuse kontrollimine**

Loogilist andmemudelit kontrollitakse iteratiivselt koos kasutajatega, kasutajaliidese väljadega ning protsessimudelitega. See võib viia andmemudeli muutmisele.

### **Andmearhitektuuri ehitamine**

Andmearhitektuur defineeritakse füüsilise andmekeskonna kontekstis. Tehakse otsustusi andmebaasiserveri, jaotuse, komponentide ja tükelduse kohta.

## **Füüsilise andmemudeli ehitamine**

Loogiline andmemudel teisendatakse füüsiliseks andmemudeliks sõltuvalt konkreetsest kasutatavast andmebaasist. Füüsiline andmemudel varieerub sõltuvalt andmebaasisüsteemi ja vahendite valikust. Füüsiline andmemudel sisaldab indekseid, välisvõtmeid, trigereid, vaateid (view), kasutaja-defineeritud andmetüüpe. Füüsiline andmemudel optimeeritakse jõudluse eesmärgil ning tavaliselt denormaliseeritakse seetõttu. Denormaliseerimine võib tekitada andmeliiasust, kuid võib parandada süsteemi jõudlust. Füüsilise andmemudeli ehitamine pole ühekordne protsess. Lõplik versioon saavutatakse paljude katsetega.

## **Andmemudeli täiustamine**

Füüsilist andmemudelit täiustatakse ja testitakse pidevalt. Samaaegselt hooldatakse ka loogilist andmemudelit (loogiline ja füüsiline mudel ei tohi olla vastuolus)

## **Transaktsioonianalüüsi lõpuleviimine**

Süsteemi transaktsioonid vaadatakse üle selleks, et täiustada / optimeerida füüsilist andmemudelit süsteemi jõudluse tõstmiseks. Transaktsioonianalüüs omab mõtet, kui ärivajadused ja süsteemidisain on tehtud hästi. Transaktsioonianalüüs annab statistikat andmebaasi tabelite kasutussageduste, andmemahtude ja ajahinnangute kohta.

## **Andmete sisseviimine**

Kui andmebaasi struktuur on paigas, viiakse sisse andmed, kasutades vastavaid andmeskripte, rakendusi või andmeteisendusi. Selle tegevuse detailid kirjeldatakse täieliku elutsükliga metodoloogiates.

## **Lõplik testimine**

Andmebaasi testimine toimub rakenduste kontekstis ja käsitletakse täieliku elutsükliga metodoloogiates. Testimiseks kasutatakse spetsvahendeid. Testimine võib viia füüsilise andmemudeli täiustamisele.

## Arendamise tulemused:

Loetleme mõningad olulisemad tulemused, mida luuakse alates andmebaasi projekti algatamisest kuni füüsilise andmebaasi loomiseni. Kasulik oleks luua andmebaas, mis sisaldab kõikide nende tulemuste näidiseid, et arendusmeeskond teaks, mida nad ehitama hakkavad.

- Vajaduste (Analüüsi) dokument: *See on ärivajaduste ja –eesmärkide seadistus arendatava rakenduse jaoks. Ta võib sisaldada vabas vormis tekste ning suvalisel arvul mudeleid või prototüüpe, mida on tarvis ärivajaduste hõlmamiseks/esitamiseks.*
- Kontseptuaalne mudel / Subjektipiirkonnad: *See on kõrgtaseme vaade ärisubjektide probleemvaldkondadest / pädevusaladest, mis kuuluvad andmemudeli skooopi (n. raamatupidamine, administreerimine, arvete maksmine, tootearendus)*
- Loogiline andmemudel: *Sisaldab olemeid, atribuute, ärireegleid subjektipiirkondade sees. Samuti seoseid olemite vahel. Identifitseeritakse ka võtmeväljad ning välisvõtmed.*
- Transaktsioonianalüüs: *see on loetelu süsteemi poolt toetatavatest transaktsioonidest, olemitest (ja võimalik, et ka väljadest), mida transaktsioon kasutab, ja nende kasutamissagedusest. Ka CRUD (Create, Read, Update, Delete) matriks on kasulik selle analüüsi abistamiseks.*
- Füüsiline andmemudel: *Loogilise andmemudeli denormaliseeritud versioon, mis on optimeeritud toimimiseks spetsiifilises tehnilises keskkonnas ning täiustatud transaktsioonianalüüsi tulemuste läbi. Füüsilist mudelit täiustatakse tavaliselt kogu elutsükli vältel ega lõpetata enne realiseerimist. Füüsiline andmemudel sisaldab füüsilisi objekte, nagu tabelid, väljad, indeksid, välisvõtmed, primaarvõtmed, vaated, kasutaja-defineeritud andmetüübid, reeglid.*
- Objektmudel: *Objektmudel toetab loogilist andmemudelit. Ta moodustab sageli vahekihi objektipõhise kasutajaliidese ja relatsioonilise alusandmebaasi vahel.*
- Kontrolli mudelid: *Need on mudelitevahelised ristseosed, näiteks protsessimudelitest loogilisse andmemudelisse, et tõestada viimase õigsust. Sageli sisaldab see teisendust loogilise andmemudeli ja kasutajaliidese vahel ning aruandeid puuduste leidmiseks.*

- Andmeülekande strateegiad: *See on strateegia seadistus, mida kasutatakse andmete teisendamiseks uude rakendusse. Detailsusaste võib oluliselt varieeruda, alates kõrgtaseme printsiipidest kuni detailsete teisendusskriptideni.*

## Vahendid

Modelleerimisvahendid on kriitilised andmebaasi arendusprotsessis. Modelleerimisvahendid peavad toetama nii andmete kui ka protsesside modelleerimist. Võiksid toetada objektmudeleid, samuti tagasiteisendust (reverse engineering) füüsilistest andmebaasidest loogiliseks andmemudeliks, eriti kui andmestruktuure muudetakse tihti (näiteks ettevõtete ühinemise tõttu).

Vajalikud CASE vahendid, teisendamise, testimise ning andmebaasi serveri vahendid.

## Organisatsioon

Nimetame andmebaasiprojektides vajalikud rollid. Nimetatavad rollid on spetsiifilised andmebaasiprojektidele. Täielikes realisatsiooniprojektides on neid sageli vaja täiendada teiste rollidega.

- Projekti sponsor: *Kõrvaldab takistusi ja kindlustab projektimeeskonnale sellise toetuse, mis eduks vajalik.*
- Projektijuht: *Projektijuht on vastutav kogu projekti eest, kaasa arvatud andmebaasid.*
- Ärikasutaja (Business User): *Ärikasutaja annab rakenduse jaoks ärireeglid, mida kasutatakse andmesalvestuseks vajalike olemite ja atribuutide tuletamiseks.*
- Ärianalüütik (Business Analyst): *Ärianalüütik võimaldab kriitilise ühenduse ärikasutaja ja andmearhitekti vahel, mõistes ärivajadusi ja tõlkides need tehnilisse keelde.*
- Andmearhitekt: *Andmearhitekt vastutab andmearhitektuuri defineerimise eest. Selleks võib olla hajutatud, tsentraalne, eraldatud (stand-alone), või integreeritud keerukama koguarhitektuuriga.*

- *Andmeanalüütik: Andmeanalüütik töötab koos ärianalüütikuga kõikide andmeelementide kooskõlalise vaate ehitamiseks. See isik saab aru ühendusest äri ja üksikute andmeelementide vahel.*
- *Andmete modelleerija: Andmete modelleerija töötab koos andmearhitektiga loogilise relatsioonilise andmemudeli ehitamiseks ja võib olla kaasatud ka loogilise andmemudeli teisendamisse füüsiliseks andmemudeliks.*
- *Objektmodelleerija: Objektmodelleerija osaleb projektides, milles ehitatakse objektumudelit, mis sisaldab sõnumeid ja meetodeid. See isik võib olla vastutav ka objektumudeli teisendamise eest ettevõtte andmemudeliks.*
- *Andmebaasi administraator: Andmebaasi administraator realiseerib füüsilist andmebaasi, hooldab ja optimeerib füüsilist keskkonda, jagab andmebaasi erineva taseme kasutajatele õigusi, annab nõu arendusmeeskonnale loogilise andmemudeli teisendamisel füüsiliseks andmemudeliks, vastutab andmebaasikeskkonna igapäevase toimimise eest.*
- *Võrguadministraator: Võrguadministraator hooldab füüsilist võrku, vastutab andmekeskonda toetava füüsilise keskkonna terviklikkuse eest, toimib operatsioonisüsteemi ja riistvara tasemel. Näiteks lisab kõvakettaid suurema andmebaasi toetamiseks.*
- *Arendaja: Arendaja kasutab andmebaase rakenduse arendamiseks.*

## Väärkasutused

Metodoloogia vale kasutamine võib mõjutada negatiivselt projekti tähtaegu. On üsna tavaline, et organisatsioonid kasutavad metodoloogiaid kui protsessidiagramme või retsepte ilma ühtegi tegevust sisuliselt läbi mõtlemata. See võib lõppeda tohutu ajaraiskamisega, kui tehakse tegevusi ja toodetakse tulemusi mõistmata nende seost terviklahendusega. Metodoloogiaid on vaja häälestada konkreetsetele projektidele. Mittevajalikud tegevused ja tulemused tuleb kustutada projektiplaanist.

Metodoloogiad, mis on liiga keerukad kasutada ja õppida, ei leia kuigi palju kasutamist projektimeeskondade poolt. Mõningad metodoloogiad

sisaldavad tuhandeid projekti pisidetaile. Tihedate projekti ajagraafikute korral heidetakse sellised metodoloogiad kõrvale.

Oluline on uuendada metodoloogiaid ajas. Uued projektikogemused ja parimad praktikad tuleb lülitada metodoloogiasse teatud ajavahemike tagant.