

Süsteemiarendus ja andmebaasi disaini koht selles

Teema 1

Andmebaasid II 2017 © Erki Eessaar

Mille kohta võiks infosüsteemide arendusmetoodika juhiseid anda?

- ◆ Arenduses osalejate **rollid**
- ◆ Arendajatelt oodatavad **oskused**
- ◆ **Arendusmeeskondade** loomise põhimõtted
- ◆ **Standardid**, mida arendamisel jälgida
- ◆ Kuidas tagada **kvaliteet?**
- ◆ Arenduses kasutada soovitatavad **töövahendid**
- ◆ **Tegevused**, mida tuleb arendamise käigus läbi viia
- ◆ **Tehised (artefaktid)**, mis arendamise käigus luuakse

14.12.2017 Teema 1 3

Andmebaasid II 2017 © Erki Eessaar

Metoodikate džungel



Juba 1990ndatel üle tuhande metoodika!

14.12.2017 Teema 1 5

Andmebaasid II 2017 © Erki Eessaar

Mis on infosüsteemi arendusmetoodika?

- ◆ **Kogum**
 - protseduure,
 - tehnikaid (praktikaid),
 - töövahendeid ja
 - dokumentatsiooni koostamise juhiseid,
- ◆ mis aitavad arendajaid infosüsteemi realiseerimise püüdluses.

Kui on tööl on tähtajad, mängus on raha, töö toimub rühmas, siis ei saa endale lubada üksildase kauboi stiilis puusalt tulistades improviserimist, tunnustatud reeglite eiramist ja oma reeglite tegemist (*cowboy coding*). 

14.12.2017 Teema 1 2

Andmebaasid II 2017 © Erki Eessaar

Väga palju erinevaid info-süsteemide arendusmetoodikaid

Pragmatic Programming	Scrumban	Beta's life cycle development methodology	The James Martin Rapid Application Development methodology
Adaptive Software Development	Dynamic Systems Development Method	Outside In Development	Enterprise Unified Process
Capability Maturity Model Integration	Extreme Programming	Agilefall Development	Unified Process
Enterprise Unified Process	Water-Scrum-Fall	Feature Driven Development	Oracle-CASE*
Kanban	Lean Development	Scrum	Method
Disciplined Agile Delivery	Agile Rational Unified Process	Open Unified Process	Cowboy coding
MODENA	Crystal Clear	Microsoft Solution Framework	Lisaks veel meetodidarendus – e sobiliku metodika kokkupanemine fragmentitest
* Esitatud metoodikate loetelu pole lõplik	Agile Unified Hybrid	Agile-Waterfall	Oracle Unified Method

14.12.2017 Teema 1 4

Andmebaasid II 2017 © Erki Eessaar

Moodne meetodikate kasutus




- ◆ Organisatsiooni ja projekti vajadustele kohandatud arendusmetoodikate loomine – **situatsiooniline metoodikaarendus**.
- Luuakse mingit *baasmetoodikat* (nt RUP) *kohandades*.
 - Paljud metoodikad on tegelikud *raamistikud*, mille põhjal konkreetse projekti jaoks sobiv metoodika välja töötada.
- Pannakse kokku *fragmentidest* (nagu LEGO).

14.12.2017 Teema 1 6

Andmebaasid II 2017 © Erki Eessaar

Moodne meetodikate kasutus (2)

- ♦ Järgitakse *üldjoontes* mingit metoodikat, kuid täitmine erineb küllaltki palju kirja pandud "idealist".
 - **Küsimus** – kuhu tömmata piir, mille ületamisel ei saa enam väita, et järgitakse metoodikat X?
- ♦ Ühes 2009. aasta uuringus ütlesid **30.6%** vastajatest, et nad ei kasuta ühtegi formaalset metoodikat.

14.12.2017 Teema 1 7

Andmebaasid II 2017 © Erki Eessaar

Lõppkasutaja poolne arendus (*end-user development*)

- ♦ Lõppkasutaja teeb ise valmis talle vajaliku tehise (nt andmebaasi+rakenduse).
 - Parima raamatupidamisprogrammi teksid raamatupidajad.
 - Vajalikud on sobivad töövahendid (vt Excel).
 - Arusaadavad ehitusplokid, intuitiivne kasutajaliides, viisandid, mallid, võimalus tulemust siluda ja testida.
 - Tegijal ei pruugi olla vastavat eriharidust; loodetavasti on aega ja motivatsiooni.

14.12.2017 Teema 1 8

Andmebaasid II 2017 © Erki Eessaar

Lõppkasutaja poolne arendus (*end-user development*) (2)

- ♦ Võimalik viis kasutaja individuaalse rakenduse loomiseks.
- ♦ Suuremate süsteemide loomisel võimalik metoodika komponent.
 - Lõppkasutaja kirjeldab enda *pädevusala* nõudeid.
 - Modelleerib.
 - Loob andmebaasi+tarkvara prototüübi.
 - Kohandab iseseisvalt talle valmis tehtud tarkvara.

14.12.2017 Teema 1 9

Andmebaasid II 2017 © Erki Eessaar

Metoodikad ei tee imet (ei ole hõbekuulid)



- ♦ Kui me ei saa aru, millist tarkvara me tahame (**puudulikud nõudmised**, ebarealsed ootused), siis pole vahet, millist metoodikat kasutada – tarkvara valmis ei saa.
- ♦ Metoodika järgimine aitab kehval arendajal oma tulemuste taset tõsta, kuid mitte luua erakordseid ja uuenduslikke süsteeme – selleks on vaja **andekaid inimesi**.

14.12.2017 Teema 1 10

Andmebaasid II 2017 © Erki Eessaar

Metoodikad ei tee imet (ei ole hõbekuulid) (2)

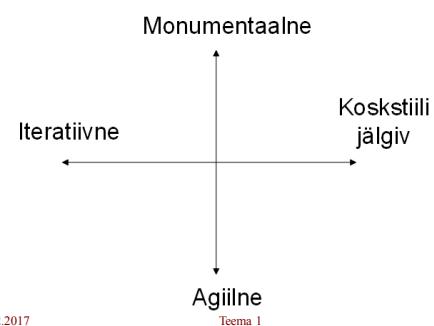


- ♦ Kui nõudmised on selged ja arendajad tasemel, siis sobilik metoodika parandab arenduse **efektiivsust ja kasumlikkust**.
- ♦ Kui metoodika kasutajad pole piisavalt pühendumud, ei järgi seda piisavalt, ei ürita võimalusel oma projekti / arendusmeeskonna järgi metoodikat kohandada, siis ei anna mistahes metoodika **loodetud kasu**.

14.12.2017 Teema 1 11

Andmebaasid II 2017 © Erki Eessaar

Võimalik raamistik metoodikate iseloomustamiseks



Iteratiivne	Monumentaalne
Agilne	Koskstili jälgiv

14.12.2017 Teema 1 12

Andmebaasid II 2017 © Erki Eessaar

Raamistiku metafoor – sõjavägi

- Monumentaalne – raskelt relvastatud, suure moonavooriga, liigub aeglaselt.
- Agiilne – kergelt relvastatud, liigub kiiresti, moona hangib kohapealt või heidetakse lennukist.

14.12.2017 Teema 1 13

Andmebaasid II 2017 © Erki Eessaar

Raamistiku metafoor – sõjavägi (2)

- Koskstil – kindluse vallutamine, seda süsteematiselt ette valmistades (plaanide uurimine, pommitamisega kaitserajatiste süsteematisline hävitamine) ja lõpuks üks kord tormi joostes.
- Iteratiivne – kindluse vallutamine korduvate rünnakukatsetega erinevatelt suundatelt, et rünnatavat nõrgestada ja segadusse ajada ning leida ründamiseks sobivad nõrgad kohad.

14.12.2017 Teema 1 14

Andmebaasid II 2017 © Erki Eessaar

Kuidas korraldada IS arendustegevuste läbiviimist?

- Kose (kaskaadse, veelange) mudeli (stiili) alusel
 - Näidis arendusmetoodikana vaatleme Oracle-CASE*Method
 - Veel näiteid: Beta's life cycle development methodology
- Iteratiivse mudeli (stiili) alusel
 - Näidis arendusmetoodikatena vaatleme Rational Unified Process (RUP), Scrum ja Extreme Programming (ekstreemprogrammeerimine).
 - Veel näiteid: Enterprise Unified Process, Essential Unified Process

14.12.2017 Teema 1 15

Andmebaasid II 2017 © Erki Eessaar

Kuidas korraldada IS arendustegevuste läbiviimist? (2)

- Need on **fundamentaalsed** mudelid.
- Mudelid, mida kirjanduses sageli nendega samal tasemel nimetatakse (V mudel, spiraalmudel, agiilne mudel, kiire prototüüpimine) on tegelikult nende **variatsioonid**.
- Metafooriks on põhivärvid**, mille kokkusegamisel saab kõik ülejää nud värvid.

14.12.2017 Teema 1 16

Andmebaasid II 2017 © Erki Eessaar

Kose mudel



14.12.2017 Teema 1 17

Andmebaasid II 2017 © Erki Eessaar

Kose mudel (2)

```

graph TD
    SA[Strateegiline analüüs] --> DA[Detail-analüüs]
    DA --> D[Disain]
    D --> E[Ehitamine]
    E --> R[Rakendamine]
    R --> H[Hooldamine]
    style aeg fill:none,stroke:none
    aeg -.-> H
  
```

14.12.2017 Teema 1 18

Andmebaasid II 2017 © Erki Eessaar

Kose mudeli iseloomustus

- Arendusprotsess on jaotatud **etappideks**.
- Nõuetekohased töötavat süsteemi **1–2 aastat**.
- **Nõuded fikseeritakse** projekti algul ja neist peetakse jäigalt kinni.
- Järgmist etapi ei alustata enne, kui eelmine on **lõppenud**.
- Iga etapi tulemusena koostatakse **detailne dokumentatsioon**.
- Pöhineb **sõjaväelisel** nägemusel süsteemide loomisest.
 - Standard: DoD STD 2167 "Defense System Software Development"

14.12.2017 Teema 1 19

Andmebaasid II 2017 © Erki Eessaar

Kose mudeli iseloomustus (2)

- Esimene kirjeldus.
 - Royce, W. W., 1970. *Managing the development of large software systems*. Proceedings of IEEE WESCON. Vol. 26. No. 8.
- Kasutusse läks teisel leheküljel olev joonis, kuid muu on unustusse vajunud.
 - Royce (1970): "**I believe in this concept, but the implementation described above is risky and invites failure.**"

14.12.2017 Teema 1 20

Andmebaasid II 2017 © Erki Eessaar

Royce (1970) pakutud parandused kose mudelile

- **Enne analüüsiga vii läbi esmane programmi disain**, et analüütik mõistaks piiranguid süsteemi töökiiruse, koormustaluvuse ja andmemahtude osas.
- **Tee kaks korda**. Esimese korraga äravivasatav prototüüp. Selle abil õpid kuidas teine versioon paremini teha. Kasutajale anna teine versioon.

14.12.2017 Teema 1 21

Andmebaasid II 2017 © Erki Eessaar

Royce (1970) pakutud parandused kose mudelile (2)

- **Kaasa klient**. Klient peaks andma sisendit nõuetekohasele, hindama esmase disaini ja "päris" disaini tulemusi ning testimaa programmi kui tervikut.
- Teadmised disaini kohta tuleb talletada **põhjalikes dokumentides**.
- Plaani, kontrolli ja jälgvi **testimist**.

Tänapäeva paindlikud soovitavad neist paljusid

14.12.2017 Teema 1 22

Andmebaasid II 2017 © Erki Eessaar

Oracle-CASE*Method

- Pöhineb kose mudelil
- Kasutab **struktuurse analüüsiga** ja disaini meetodit
- Süsteemi modelleeritakse mitmeaastisel ajal ...
- põhirõhk **funktionaalsel** vaatel – kirjeldatakse tegevuste hierarhia
- Süsteemi kirjeldus talletatakse CASE* Dictionary teadmuisbaasis

14.12.2017 Teema 1 23

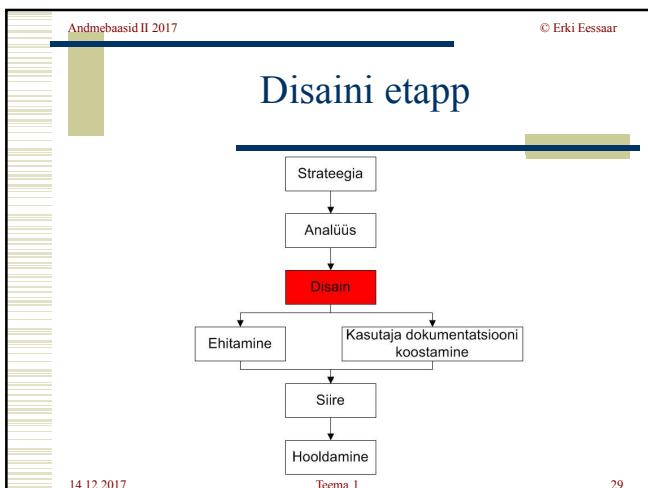
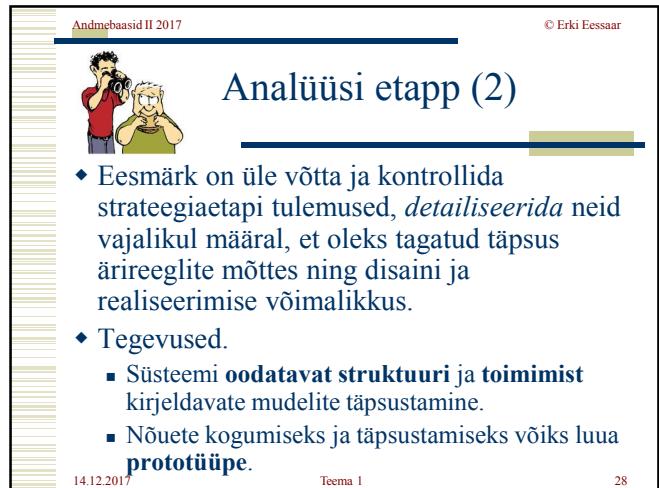
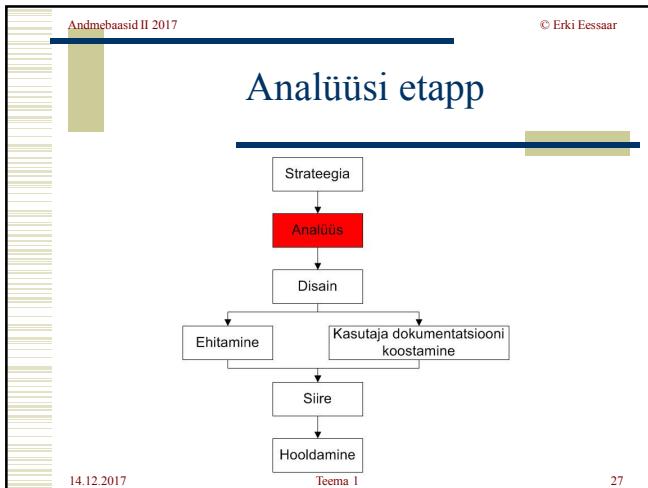
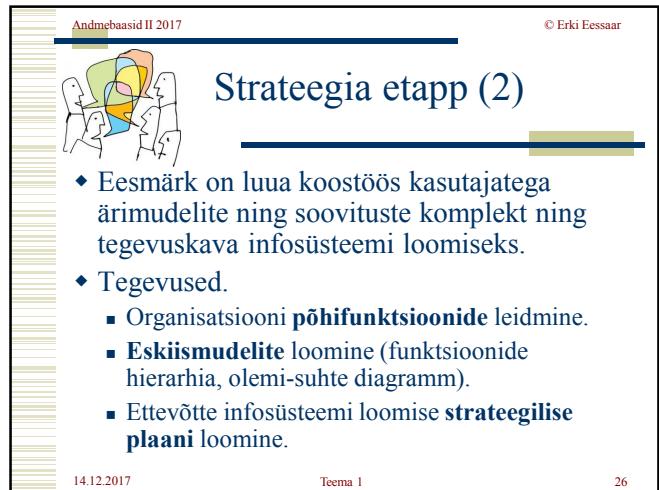
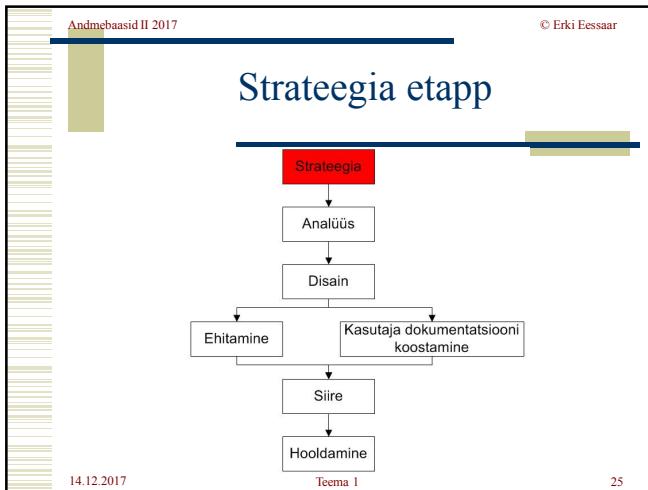
Andmebaasid II 2017 © Erki Eessaar

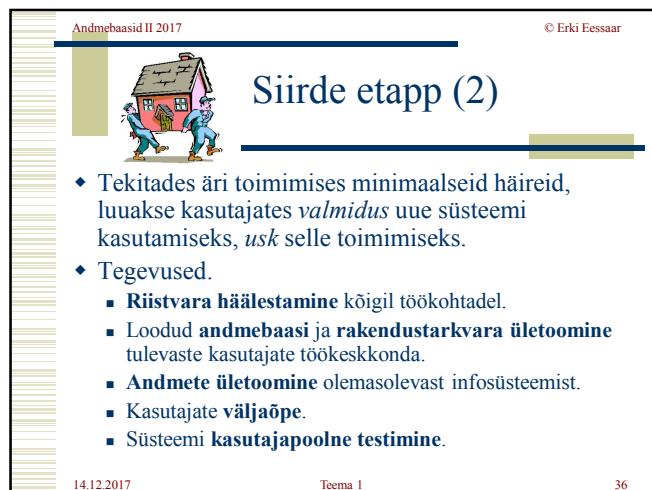
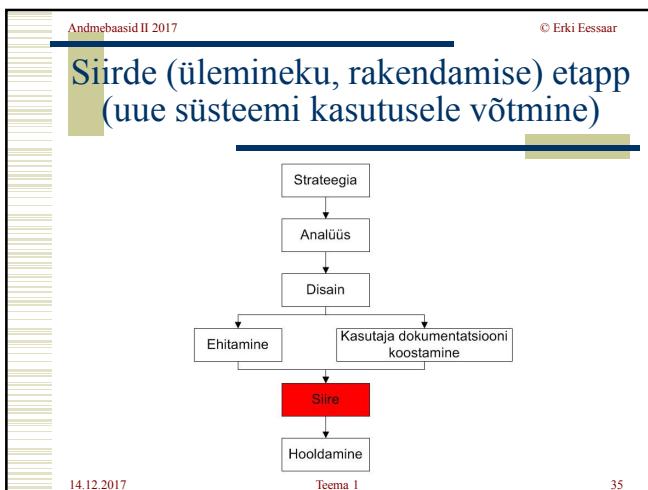
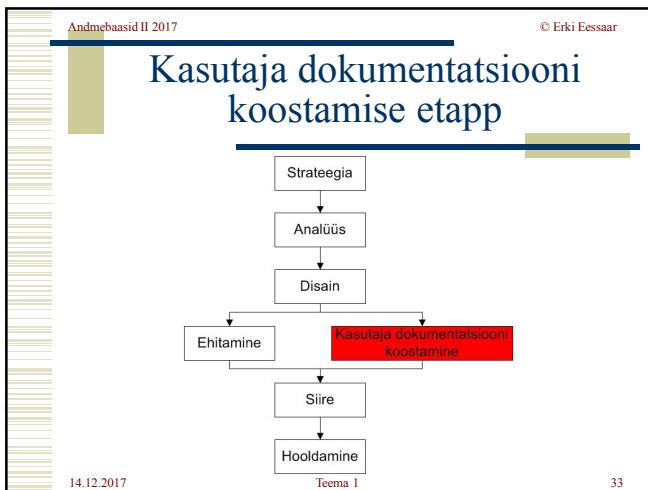
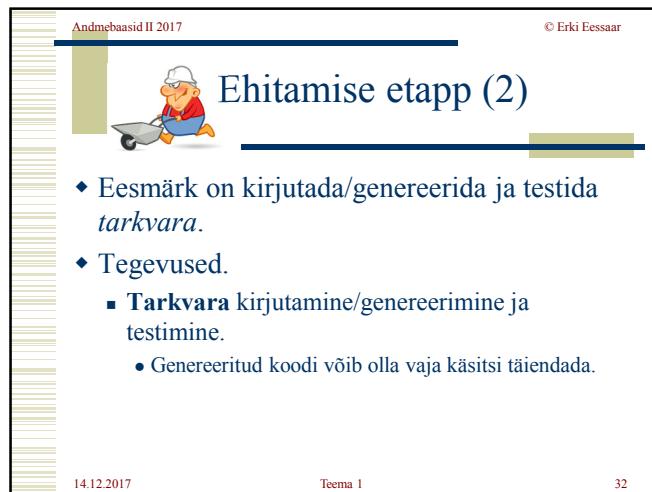
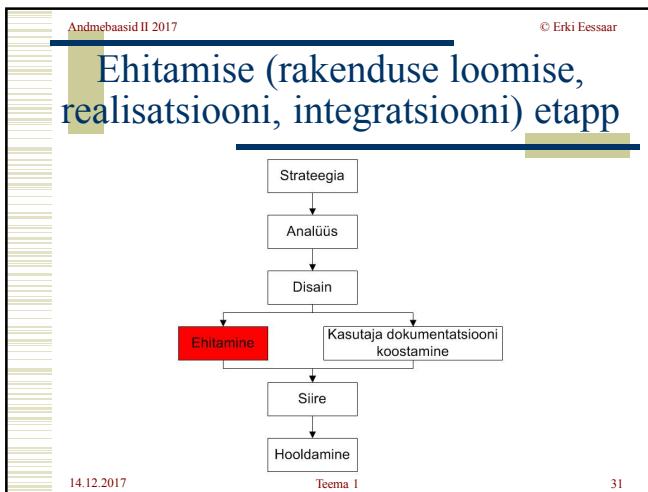
Üks analüüsiga põhitulem – andmevoodiagramm

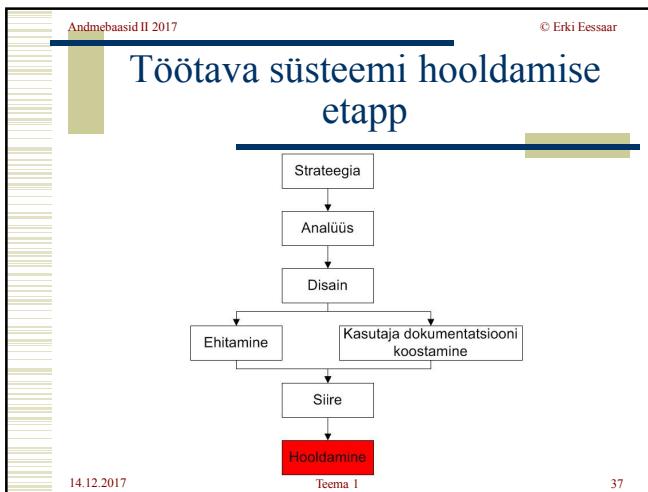
Ei kirjelda tegevuste/ protsesside/ andmevoogude toimumise järjekorda

terminator
 Üliõpilane
 Oppeaine
 Üliõpilane_oppaines
 Öppenud
 andmevoog
 andmehoidla andmebaasis peab olema kohast vastavate andmete hoidmiseks
 tegevus/protsess
 Ei kirjelda, kas andmeid töugatakse või tömmatakse.

14.12.2017 Teema 1 24







- Andmebaasid II 2017 © Erki Eessaar
- ## Hooldamise etapp (2)
-
- Eesmärk on tagada süsteemi *ladus toimimine* minimaalse sekumisega toetusmeeskonnalt.
 - Tegevused.
 - Süsteemi **häireteta funktsioneerimise** tagamine (nt andmete varundamine ja taastamine).
 - Süsteemi **töökiruse** näitajate pidev jälgimine.
 - Meetmete rakendamine probleemide kõrvaldamiseks **ebapiisava jõudluse** ilmnemisel.
 - **Muudatuste** sisseviimine infosüsteemis kooskõlas protsesside kulgemise loogika muutumisega aja jooksul.
 - Tarkvara hoolduskulud **50–80%** selle kogukuludest.
- 14.12.2017 Teema 1 38

- Andmebaasid II 2017 © Erki Eessaar
- ## Kose mudelil põhineva süsteemiarenduse võimalikud eelised
-
- Põhjalik ettevalmistus (analüüs ja disain) peaks tagama süsteemi võimalikult **probleemivaba** realiseerimise.
 - Loodetavasti paraneb tarkvaraarenduse **ennustatavus** (palju kulub aega ja raha).
 - Klient teab mida oma raha eest saab, nõudmiste kogumise lõppedes saab allkirjastada lepingu arenduse (disaini, ehitamise, rakendamise) läbiviimiseks.
- 14.12.2017 Teema 1 39

- Andmebaasid II 2017 © Erki Eessaar
- ## Ennustatavusest
-
- Avalik sektor peab arendusteenuse sisseostmiseks korraldama **riigihanke**.
 - Tarkvara riigihankes tuleb ära määrata **maksumus ja ootav funktsionaalsus**.
 - Nii tellija kui pakkujad on huvitatud **nõuetest detailsest arusaamisest** – soosib kose mudelite või vähemalt selle rakendamist süsteemianalüüsiks.
- 14.12.2017 Teema 1 40

- Andmebaasid II 2017 © Erki Eessaar
- ## Kose mudelil põhineva süsteemiarenduse võimalikud eelised (3)
-
- Teadmised süsteemi kohta säilivad põhjalikes **dokumentides**.
 - Põhjalik dokumentatsioon aitab tagantjärgi lihtsamalt selgeks teha õnnetuste ja vigade **põhjuse** (nt lennuõnnnetuste uurimisel).
 - Mudelist lihtne **aru saada, õpetada**.
- 14.12.2017 Teema 1 41

- Andmebaasid II 2017 © Erki Eessaar
- ## Kose mudelil põhineva süsteemiarenduse võimalikud probleemid
-
- **Nõuded vananevad**.
 - Võimalik **analüüsist tingitud paralüüs**.
 - Analüusi kulu (aeg, inimesed) ületab oodatavat tulu.
 - Analüusi protsess takerdub oma mahukuse tõttu.
 - Liiga varakult ja rangelt fikseeritud nõuded võivad panna **projekti venima või seisma**.
 - USA sõjaväes võtab praegu suure süsteemi loomine 15 aastat, 25 aastat tagasi 5 aastat.
- 14.12.2017 Teema 1 42

Andmebaasid II 2017 © Erki Eessaar

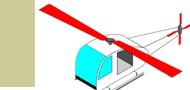
Kose mudelil põhineva süsteemiarenduse võimalikud probleemid (2)

- Seljataguse kindlustamiseks soovivad arenduses osalejad sõlmida võimalikult ruttu **lepingu** selle kohta, mida valmis tehakse.
- Parem oleks seda teha siis, kui kliendile saab ette näidata tehnilise kavandi (disaini).
- Kose mudeli loogika suunab lepingu sõlmimisele peale analüüs. Tulemuseks on suurema tõenäosusega lepingud, mida ei saa täita või mida on vaja hakata hiljem muutma.

14.12.2017 Teema 1 43

Andmebaasid II 2017 © Erki Eessaar

Näide



- Nõuded USA sõjaväe kergetele ründekopteritele – olgu **kerge ja lennaku üle Atlandi**. Pole võimalik mõlemat täita.
- Kuna nõudeid kogusid asja mitte jagavad bürokraadid, need lõödi lukku ning paremat lahendust (ostke projekti rahade eest rohkem transpordilennukeid) ei saanud bürokratilkel põhjustel kasutada, siis projekt soikus.

14.12.2017 Teema 1 44

Andmebaasid II 2017 © Erki Eessaar

Kose mudelil põhineva süsteemiarenduse võimalikud probleemid (3)

- Testimise ja ehitamiseni jõutakse hilja.**
 - Süsteemi seisukohalt saatluslikud probleemid avastatakse hilja.
 - Kasutaja saab valmivat süsteemi testida alles projekti hilises faasis.
 - Tulemus võib rahuldada algseid, kuid mitte hetke soove.
 - Tellija ja täitjad võisid saada nõuetest erinevalt aru.

14.12.2017 Teema 1 45

Andmebaasid II 2017 © Erki Eessaar

Probleemide näide

2014. septembri seisundi kulutati 2.1 miljardit USD.

- <https://www.healthcare.gov/> – USA ravikindlustuse vahendamise keskkond (ObamaCare).
- 55 alltöövõtjat.
- Üldine projekt kose mudeli järgi (alamprojektid ka *agiilised*).
- Süsteemi kui terviku testimine septembri kaks viimast nädalat.
- Süsteem läks käiku 1. oktoobril 2013.
- Kuni aasta lõpuni suured tehnilised probleemid.

14.12.2017 Teema 1 46

Andmebaasid II 2017 © Erki Eessaar

Kose mudel pole veel välja surnud



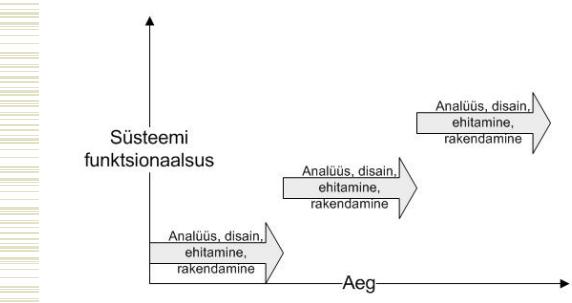
- Uuring 2009 – **13%** vastajatest kasutas kose mudelit.
- HP uuring 2015 – 601-st küsitletud IT spetsialistist **2%** kasutab puhas kose mudelit ja **7%** kalandub kose mudeli kasutamise suunas.
- <https://www.quora.com/Is-waterfall-design-still-widely-used>

Kasutamist võivad tingida **kultuurilised eripärad**.

14.12.2017 Teema 1 47

Andmebaasid II 2017 © Erki Eessaar

Iteratiivne mudel (öeldakse ka iteratiivne ja inkrementaalne mudel)



14.12.2017 Teema 1 48

Andmebaasid II 2017 © Erki Eessaar

Näide süsteemist, mida tuleks arendada iteratiivselt

- Uuriti 4.5 aasta jooksul kasutusel olnud 171 MediaWiki (<https://www.mediawiki.org/wiki/MediaWiki>) SQL-andmebaasi skeemi versiooni.
- Curino, C.A., Moon, H.J., Tanca, L., & Zaniolo, C. (2008). *Schema evolution in Wikipedia: toward a web information system benchmark*. In: ICEIS 2008.
- Tabelite arv kasvanud 17lt 34ni (kasv 100%) ja veergude arv 100lt 242ni (kasv 142%).

14.12.2017 Teema 1 49

Andmebaasid II 2017 © Erki Eessaar

Näide süsteemist, mida tuleks arendada iteratiivselt (2)

- 54.9%** muudatustest skeemi muudatused ning **40.3%** töökiiruse vajadustest tingitud indeksi/võtme muudatused.
- Algsete skeemi versioonide põhjal loodud päringutest töötab viimaste skeemi versioonide põhjal ainult umbes **22%** (st ülejäänud päringuid on olnud vaja süsteemi arengu käigus muuta).

14.12.2017 Teema 1 50

Andmebaasid II 2017 © Erki Eessaar

Rationali unifitseeritud protsess (RUP)

- Kasutab *objekt-orienteeritud* analüüs ja disaini meetodit.
- Süsteemi modelleeritakse mitmevaateliselt kuid ...
- põhiröhk on *objektivaatel*.

The diagram illustrates the Rational Unified Process (RUP) with four main phases: Inception, Elaboration, Construction, and Transition. It shows iterative cycles across these phases. Disciplines listed include Business Modeling, Requirements Analysis & Design, Implementation, Test, Deployment, Configuration & Change Mgmt, Project Management, and Environment. The diagram also shows the relationship between phases and iterations, with specific labels like Initial, Elab #1, Elab #2, Const #1, Const #2, Const #N, Const #1, and Tran #1, #2.

14.12.2017 Teema 1 51

Andmebaasid II 2017 © Erki Eessaar

RUP (2)

- Järgud (Phases)
 - Algatamine (Inception)
 - Detailimine (Elaboration)
 - Konstrueerimine (Construction)
 - Siire (Transition)
- Distsipliinid (Disciplines)
- Iteratsioonid (Iterations)
 - arenduse piisitsüklid – lühikesed (2–6 nädalat) ja kindlast ajaliselt piiratud (ajakarbi meetod)
- Arendustükk
- Tarkvara generatsioon

The diagram illustrates the RUP process flow across phases and iterations. It shows the flow from Inception through Elaboration, Construction, and Transition. The process is divided into iterations, with specific labels for initial phases and subsequent iterations (Elab #1, Elab #2, Const #1, Const #2, Const #N, Const #1, Tran #1, Tran #2). The diagram also includes a legend for disciplines: Business Modeling, Requirements Analysis & Design, Implementation, Test, Deployment, Configuration & Change Mgmt, Project Management, and Environment.

14.12.2017 Teema 1 52

Andmebaasid II 2017 © Erki Eessaar

Dokumentide loomise järjekord RUPis

- s** – start – dokumendi loomise algus
- r** – refinement – dokumendi täpsustamine

Disciplina	Tulen	Algatamine It...	Detailimine El...En	Konstrueerimine Co...On	Siire Tr...Tj
All-modelleerimine					
Valdkonnamudel	s	r			
Kasutusjahtude mudel	s	r			
Vision	s	r			
Täisendav spetsifikatsioon	s	r			
Riskide nimelik ja riskide hinnanguline pähn	s	r			
Prototüüp	s	r			
Sõnastik	s	r			
Disain		s	r		
Tarkvara arhitektuurimudel		s	r		
Andmenemuudel		s	r		
Realiseerimine					r
Rakendusmuudel		s	r		r
Projektjätk		s	r	r	
Tarkvara arenduskeskkond					
Testmine			s	r	
Keskonna hälestamine			s	r	
Arenduskeskonna kirjeldus					

14.12.2017 Teema 1 53

Andmebaasid II 2017 © Erki Eessaar

RUPi parimad praktikad

- Iteratiivne arendus.
- Nõuete haldamine (kasutusjuhud).
- Visuaalne modelleerimine (UML).
- Kasuta komponentidel põhinevat arhitektuuri (taaskasutamine).
- Tarkvara kvaliteedi kontrollimine (testimine).

14.12.2017 Teema 1 54

Andmebaasid II 2017 © Erki Eessaar

Iteratiivse arenduse eelised

- ♦ Riskidega tegeletakse **varakult**.
- ♦ **Nähtav** progress.
- ♦ Pidev tellijate **tagasiside**.
 - Nõuete muutumine on võimalik ka projekti hilises faasis.
- ♦ Otsustada tuleb **kiiresti**.

14.12.2017 Teema 1 55

Andmebaasid II 2017 © Erki Eessaar

Iteratiivse arenduse eelised (2)

- ♦ Kliendid ja arendajad on **rahulolevamad**, kui näevad toimivat süsteemi.
- ♦ Kasutajatel on aega uue süsteemiga **harjuda**.

14.12.2017 Teema 1 56

Andmebaasid II 2017 © Erki Eessaar

Iteratiivse arenduse väljakutseid

- ♦ Kõrgendatud nõuded arendajatele.
 - Peab suutma luua võimalikult kvaliteetset koodi, mida oleks hiljem võimalikult lihtne täiendada.
 - Peab olema hea suhtlemisoskus.
- ♦ Koodi parandamiseks ei jäää piisavalt aega, mis viib **tehnilise võla** kuhjumiseni.
- ♦ Tuleb investeerida tarkvara **paigaldamise automatiserimisse**.

14.12.2017 Teema 1 57

Andmebaasid II 2017 © Erki Eessaar

Süsteemiarenduse osa sõltumata arendusmudelist ja metoodikast

Andmete disain

- ♦ Analüüs aitab mõista probleemi.
- ♦ Disain pakub välja probleemile tehniline lahenduse.
- ♦ Andmete disaini sammud.
 - Andmete loogiline disain.
 - Andmete füüsiline disain.
 - Disaini täpsustamine.
 - Andmebaasi häältestamine.

14.12.2017 Teema 1 58

Andmebaasid II 2017 © Erki Eessaar

Andmete disain kose mudelil põhineva arendamise korral

```

graph TD
    SA[Strateegiline analüüs] --> DA[Detail-analüüs]
    DA --> DD[Disain Andmete disain]
    DD --> E[Ehitamine]
    E --> R[Rakendamine]
    R --> H[Hooldamine]
    style DD fill:#ffff00,stroke:#000
    style E fill:#ffff00,stroke:#000
    style R fill:#ffff00,stroke:#000
    style H fill:#ffff00,stroke:#000
    style SA fill:#ffff00,stroke:#000
    style DA fill:#ffff00,stroke:#000
    
```

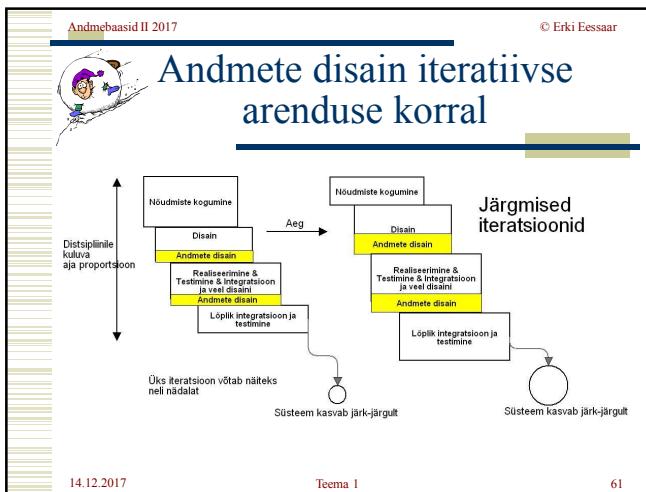
14.12.2017 Teema 1 59

Andmebaasid II 2017 © Erki Eessaar

Võimalikud tulemid (ajalises järjekorras)

- ♦ **Kontseptuaalne andmemudel** (strateegiline- ja detailanalüüs).
- ♦ **Loogilise disaini andmemudel** (disain).
- ♦ **Füüsilise disaini andmemudel** (disain).
- ♦ **Käufailid ja andmebaas** (ehitamine).

14.12.2017 Teema 1 60



Andmebaasid II 2017 © Erki Eessaar

Andmete disain iteratiivse arenduse korral (2)

- Iga iteratsiooni tulemusena võib andmebaasi struktuur muutuda
- Tabeli struktuuri muutmine on keeruline ülesanne.
 - Juba registreeritud andmed ei tohi kaotsi minna
 - Tuleb muuta kõiki sellest tabelist sõltuvaid süsteemi elemente – nii andmebaasi sees (nt vaated, rutinid) kui väljapoole (rakendused, testid, mudelid, ...)

14.12.2017 Teema 1 62

Andmebaasid II 2017 © Erki Eessaar

Andmete disain iteratiivse arenduse korral (3)

- Ankurmodelleerimine (vt "Andmebaasid I" teema 9 ja "Andmebaasid II" teema 14) on populaarsust koguv viis, kuidas lihtsustada skeemimuudatusi – olemasolevatesse tabelitesse veergude lisamise asemel on vaja lisada uusi tabeleid.
 - <http://www.anchormodeling.com/>
 - Peaaegu kõik tabelid on kuuendal normaalkujul.

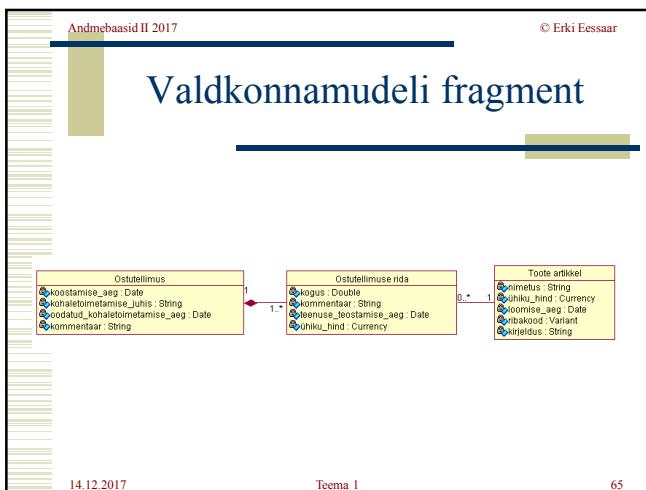
14.12.2017 Teema 1 63

Andmebaasid II 2017 © Erki Eessaar

Seotud tulemid RUPis

- Valdkonnamudel e domeenimudel (detailimine) (valmib samm-sammult).
- Tarkvara klassidiagramm (detailimine ja konstrueerimine) (valmib samm-sammult).
- Andmemudel (detailimine ja konstrueerimine) (valmib samm-sammult).
 - Luuakse tarkvara klassidiagrammi alusel. Iga püsivalvestust vajava tarkvara klassi kohta peab olema defineeritud andmebaasis vastav andmestruktuur.
- Käsufailid ja andmebaas (detailimine ja konstrueerimine) (valmib samm-sammult).

14.12.2017 Teema 1 64



Andmebaasid II 2017 © Erki Eessaar

Disaini klassidiagrammi fragment

Klass on prototüüp, mis defineerib kõigile mingit tüüpi objektidele ühtsed muutujad ja meetodid.

14.12.2017 Teema 1 66

Andmebaasid II 2017 © Erki Eessaar

Andmebaasi diagrammi fragment

```

classDiagram
    class Ostetellimus {
        ID: NUMBER(10, 0)
        koodarvuse_seg_DATE: DATE
        kohaletoimetamise_juh: VARCHAR2(1000)
        ostatud_kohaletoimetamise_seg_DATE: DATE
        kommentaar: CLOB
    }
    class Ostetellimuse_dba {
        Ostetellimus_ID: NUMBER(10, 0)
        Ostetellimus: ID: NUMBER(10, 0)
        logus: NUMBER(7, 0)
        teenuse_toimetamise_seg_DATE: DATE
        üliku_hind: NUMBER(10, 2)
    }
    class Toote_arikel {
        ID: NUMBER(10, 0)
        nimeitus: VARCHAR2(255)
        Üliku_hind: NUMBER(10, 2)
        loomine_seg_DATE: DATE
        ribaarv: NUMBER(6, 0)
        kirjeldus: CLOB
    }

    Ostetellimus "1..<<Identifying>>" Ostetellimuse_dba
    Ostetellimuse_dba "1..<<Identifying>>" Toote_arikel
    Ostetellimus "1..<<Identifying>>" Toote_arikel
  
```

Loodud kasutades SQL-andmebaaside projekteerimiseks mõeldud mudeliprofiili

14.12.2017 Teema 1 67

Andmebaasid II 2017 © Erki Eessaar

UML

- Unified Modeling Language (UML) üldotstarbeline standardiseeritud visuaalne modelleerimiskeel.
- UMLi võib kasutada *erinevate* süsteemiarenduse metodikate kooesseisus (nii kose mudelil kui iteratiivsel mudelil põhinevad).

14.12.2017 Teema 1 68

Andmebaasid II 2017 © Erki Eessaar

UML (2)

- Nii valdkonnamuudel kui ka andmebaasi diagramm luuakse *klassidiagrammi* baasil.
- UML pakub võimalusi luua keele laiendusi, luuves uusi *mudeliprofile*.
 - Mudeliprofiil SQL-andmebaaside projekteerimiseks.
 - http://staff.ttu.ee/~eessaar/SQL_profile/
 - Mudeliprofiil töeliselt relatsiooniliste andmebaaside projekteerimiseks.
 - http://staff.ttu.ee/~eessaar/ttm_profile/

14.12.2017 Teema 1 69

Andmebaasid II 2017 © Erki Eessaar

UML (3)

- UML laiendamine tähendab olemasolevatele keele elementidele uue tähenduse andmist, minemata vastuollu olemasolevaga.
- UML mudeliprofil kirjeldatakse kasutades:
 - stereotüübide,
 - sildi definitsioonid (stereotüibi omadused) ja siltväärtused,
 - piirangud.

14.12.2017 Teema 1 70

Andmebaasid II 2017 © Erki Eessaar

Andmebaasi diagrammi fragment

Stereotüp *Tabel* laiendab UMLi klassidiagrammi elemendi *Klass* tähendust.

```

classDiagram
    class Ostetellimus {
        ID: NUMBER(10, 0)
        koodarvuse_seg_DATE: DATE
        kohaletoimetamise_juh: VARCHAR2(1000)
        ostatud_kohaletoimetamise_seg_DATE: DATE
        kommentaar: CLOB
    }
    class Ostetellimuse_dba {
        Ostetellimus_ID: NUMBER(10, 0)
        Ostetellimus: ID: NUMBER(10, 0)
        logus: NUMBER(7, 0)
        teenuse_toimetamise_seg_DATE: DATE
        üliku_hind: NUMBER(10, 2)
    }
    class Toote_arikel {
        ID: NUMBER(10, 0)
        nimeitus: VARCHAR2(255)
        Üliku_hind: NUMBER(10, 2)
        loomine_seg_DATE: DATE
        ribaarv: NUMBER(6, 0)
        kirjeldus: CLOB
    }

    Ostetellimus "1..<<Identifying>>" Ostetellimuse_dba
    Ostetellimuse_dba "1..<<Identifying>>" Toote_arikel
    Ostetellimus "1..<<Identifying>>" Toote_arikel
  
```

Primaarvõit modelleeritakse tabeli veerge kujutavatele *klassi attribuutidele* lisatud siltväärtusega (*PK*) ja stereotüübiga <<PK>> *klassi operatsiooniga*, mille:

- nime abil kirjeldatakse primaarvõtme kitsenduse nimi,
- parametrite järgkord näitab veergude järjekorda primaarvõtmes.

14.12.2017 Teema 1 71

Andmebaasid II 2017 © Erki Eessaar

Andmebaasi arendamine – RUP (kasutusjuhtude keskne lähenemine)

- Nõuded => Kasutusjuhtude mudel => Selle analüüsimisel leitakse olemitüübide => Kasutusjuhtude mudeli alusel koostatakse klassidiagrammid ja valitud olemitüüpide seisundiagrammid => Klassidiagrammi alusel genereeritakse andmebaasi skeemi kirjeldus

14.12.2017 Teema 1 72

Andmebaasid II 2017 © Erki Eessaar

Võib läbida nii koskstilis kui iteratiivselt.

Andmebaasi arendamine – andmekeskne lähenemine

- ♦ Nõuded => Põhiolemitüüpide nimekiri => Igaühе kohta seisundidiagramm => Iga põhiolemitüübі kohta mustrite alusel esialgne klassidiagramm (kontseptuaalne andmemudel) => Kasutusuhtude mudel luuakse seisundidiagrammide, kontseptuaalse andmemudeli ja aruandluse vajaduste alusel => Kontseptuaalsest andmemudelist genereeritakse andmebaasi skeemi kirjeldus

14.12.2017 Teema 1 73

Andmebaasid II 2017 © Erki Eessaar

RUPi probleemid

- ♦ On küll iteratiivne, kuid peetakse liiga mahukaks, *monumentaalseks* (tardmetoodika).

14.12.2017 Teema 1 74

Andmebaasid II 2017 © Erki Eessaar

Kuidas täiustada kose mudelit?

- ♦ V mudel.
 - Nagu kose mudel, kuid analüüs ja disaini etappidega paralleelselt valmistatakse ette teste (ühiktestid, integratsiooni testid).
 - Testijad tutvuvad varakult projekteerimise tulemustega.
 - Peale realiseerimise etappi läbitakse kavandatud testid.

14.12.2017 Teema 1 75

Andmebaasid II 2017 © Erki Eessaar

Kuidas kombineerida kose ja iteratiivset mudelit?

- ♦ Võimaldada liikuda vajadusel tagasi eelmisesse etappi (nt disain => analüüs).
- ♦ Teha igas etapis tegevusi iteratiivselt, etapid võivad olla ajalise ülekattega.
 - Agifall metoodika.

14.12.2017 Teema 1 76

Andmebaasid II 2017 © Erki Eessaar

Kuidas kombineerida kose ja iteratiivset mudelit? (2)

- ♦ Läbida ehitamise etapp ja võimalik, et ka järgmised etapid iteratiivselt, võimalik et paralleelse projektidega – vesiagiilsus.
 - Sobib järk-järguliseks üleminekuks paindmetoodika kasutamisele.
 - Miks mitte riigihangetega projektides – tehakse süsteemi kui terviku analüüs. Süsteemi edasine arendus jagatakse väiksemateks osadeks ja seda võivad teha erinevad ettevõtted.

14.12.2017 Teema 1 77

Andmebaasid II 2017 © Erki Eessaar

Kuidas kombineerida kose ja iteratiivset mudelit? (3)

- ♦ Spiraalmudel.
 - Süsteemiarendus koosneb iteratsioonidest (spiraalidest).
 - Igas iteratsioonis toimub arendus kose mudeli alusel (läbitakse järjest neli järku).
 - Iga iteratsiooniga valmib süsteemi uus ja täiuslikum versioon.

14.12.2017 Teema 1 78

Andmebaasid II 2017 © Erki Eessaar

Monumentaalsed (tard) metoodikad



- ◆ **Monumentaalne** – raske, range, plaanist juhitud.
- ◆ Edu saavutatakse läbi etteennustatava tegevuse arenedes seejuures tasapisi perfektse tulemuseni.
- ◆ Iseloomulikud omadused:
 - ulatuslik planeerimine,
 - kõikehõlmav dokumenteerimine ja prognoosimine,
 - töövahenditele orienteeritus,
 - koondatud ja hierarhilise otsustamine.
- ◆ Näited: Oracle-CASE*Method, RUP.

14.12.2017 Teema 1 79

Andmebaasid II 2017 © Erki Eessaar

Paindmetoodikad (agiilsed, väledad metoodikad)



- ◆ **Agiilne** – väle, kerge, paindlik.
- ◆ Iseloomulikud omadused:
 - inimestele orienteeritus,
 - muutustega kohandumine,
 - lihtsus; tee mida klient vajab, aga mitte rohkem,
 - YAGNI – *You ain't Gonna Need It* – realiseeri ajad alles siis, kui neid vajatakse. Ara realiseeri käesoleval hetkel ajus, mida arvad tulevikus võib-olla vaja minevat.
 - KISS – *Keep It Simple and Stupid*
 - jagatud otsustamine,
 - vajadusel uesti tegemine.
- ◆ Näited: Scrum, Ekstreemprogrammeerimine (XP).

14.12.2017 Teema 1 80

Andmebaasid II 2017 © Erki Eessaar

Paindmetoodikad (2)

- ◆ Väärtustavad **isiksust ja suhtlemist rohkem** kui protsesse ja töövahendeid.
- ◆ Väärtustavad **töötavat tarkvara rohkem** kui põhjalikku dokumentatsiooni.
- ◆ Väärtustavad **koostööd kliendiga rohkem** kui läbirääkimisi lepingu üle.
- ◆ Väärtustavad **muutustele avatust rohkem** kui plaani järgimist.

Need on agilise tarkvaraarenduse manifesti põhipunktid
<http://agilemanifesto.org/>

14.12.2017 Teema 1 81

Andmebaasid II 2017 © Erki Eessaar

Kuidas teha "midagi" agiisel viisil?

- ◆ Saa aru, milline on hetkeseis.
- ◆ Liigu väikeste sammudega eesmärgi poole.
- ◆ Korrigeeri oma arusaama vastavalt eesmärgi poole liikudes õpitule.
- ◆ Korda.
- ◆ Kui oled valikute ees ja alternatiivid on pakutava väärtsuse seisukohalt võrdväärsed, siis vali selline, mis muudab hiljem muudatuste tegemise lihtsamaks.

14.12.2017 Teema 1 82

Andmebaasid II 2017 © Erki Eessaar

Paindmetoodikad (3)

- ◆ Kogum väärtsuseid, printsiipe ja praktikaid, mis juhivad tarkvarasüsteemi arendamist.
- ◆ Sellistena on need samuti **raamistikud** (nagu ka RUP).
 - Konkreetsesse projekti jaoks tuleb teha valik – milliseid väärtsuseid, printsiipe ja praktikaid järgida.

14.12.2017 Teema 1 83

Andmebaasid II 2017 © Erki Eessaar

Paindmetoodikad (4)

- ◆ Alistair Cockburn pakub, et neid võiks nimetada mitte metoodikateks, vaid "**eneseanalüüs kaudu paremaks saamise raamistikeks**" (*Reflective Improvement Frameworks*).
 - Pakuvad välja soovitavad tulemused, kuid mitte detailset protsessi, kuidas nendeni jõuda.
 - Sarnane Jaapani mõistele "Kaizen", mis tähdab pidevarengut: <https://en.wikipedia.org/wiki/Kaizen>

14.12.2017 Teema 1 84

Andmebaasid II 2017 © Erki Eessaar

Paindmetoodikad (5)

- ◆ "Agile" tõeline lõeksõna, selle teemalised konsultatsioonid ja sertifitseerimine uus tööstusharu.
- ◆ Paljud, kes sõnades "agiilsed" ei muuda mõtteviisi, vaid *kõnepruuki ja tseremooniaid*.
- ◆ Hakkab teenima kriitikat:
 - <http://www.halfarsedagilemanifesto.org/>
 - <https://effectivesoftwaredesign.com/2014/03/17/the-end-of-agile-death-by-over-simplification/>
 - <https://pragdave.me/blog/2014/03/04/time-to-kill-agile.html>

14.12.2017 Teema 1 85

Andmebaasid II 2017 © Erki Eessaar

Paindmetoodikad (6)

Gartneri haubitsükkel

VISIBILITY

TIME

Technology Trigger

Peak of Inflated Expectations

Slope of Enlightenment

Trough of Disillusionment

Plateau of Productivity

14.12.2017 Teema 1 86

Paindmetoodikad on ilmselt ülespuhutud ootuste tipu ületanud!

Andmebaasid II 2017 © Erki Eessaar

Raamat, mida soovitan

- ◆ Meyer, B., 2014. *Agile! : The Good, the Hype and the Ugly*, Springer. 170 p.
http://www.esther.ee/record=b4604870*est
 - Autorist: https://en.wikipedia.org/wiki/Bertrand_Meyer (arvutikeeled, Eiffel, OO programmeerimine, lepingprojekteerimine)
- ◆ Paindmetoodikate metaanalüüs, mis toob välja autori arvates nende tugevad, ülehinnatud ja nõrgad küljed.

14.12.2017 Teema 1 87

Andmebaasid II 2017 © Erki Eessaar

Traditsioniliste paindmetoodikate (Scrum, XP) iseloomustus

- ◆ Keskenduvad *tarkvara* (mitte infosüsteemi) *ehitamisega* (vaid üks osa arendustusklist) seotud tegevustele.
- ◆ Eesmärk produtseerida *regulaarselt kvaliteetset* tarkvara, mida kölbaks *kliendile üle anda*.
- ◆ Väikesed (vähem kui 15 liiget) lähestikku asuvad iseorganiseeruvad, koostöö orienteeritud arendusmeeskonnad .
- ◆ *Projektijuht* toetab, mitte ei kamanda.

14.12.2017 Teema 1 88

Andmebaasid II 2017 © Erki Eessaar

Traditsioniliste paindmetoodikate (Scrum, XP) iseloomustus (2)

- ◆ Rakendamiseks *kõige sobivamad* projektid:
 - ei ole väga suure kriitilise tähtsusega,
 - nõuded muutuvad sageli,
 - arendajate hulk ei ole väga suur,
 - arendajatel suured kogemused ja hea enesedistsipliin.
 - Näiteks idufirma, mis tahab oma tooteaga (või selle prototüübiga) kiiresti turule tulla ja kasutajate tagasisidega operatiivselt arvestada.

14.12.2017 Teema 1 89

Andmebaasid II 2017 © Erki Eessaar

Scrumi mõningad põhimõisted

Sprindide tagasisiiale kohutamine

Sprindide planeerimise kohutamine

Päevane kohutamine

Sprindide läbipaistmise graafik

Sprindide nõuete nimikiri

Nõue nimikirjas

Definition of Done

Valitlasse

Töode

Töode mõõtute nimikiri

Omanik

Töote emanik

Arvendaja

Meeskond

Scrum-mester

Scrum of Scrums

Valjakas planeerimise kohutamine

14.12.2017 Teema 1 90

Andmebaasid II 2017 © Erki Eessaar

Scrumi mõningad põhimõisted (2)

- Iga iteratsioon (**sprint**) kestab 1–4 nädalat.
- Iseorganiseeruv arendajate meeskond** (optimaalselt 5–9 liiget), ilma ühe kindla liidritähtaega.
- Meeskonda toetavad toote omanik ja ScrumMaster.
- Kui meeskond on **auto**, siis **toote omanik** on **auto juht** ning **ScrumMaster** on **peamehaanik**, kes tagab auto tõrgeteta töö.

14.12.2017 Teema 1 91

Andmebaasid II 2017 © Erki Eessaar

Scrumi mõningad põhimõisted (3)

- Suletud akna reegel** – sprindi käigus ei hakata käsitlema uusi nõudmisi – selleks on järgmised sprindid.
 - Jääb veidi aega rahuneda, mõtted koguda.
- Ajakarbi meetod** – kõigile tegevustele on nähtud ette maksimaalne lubatud pikkus.
 - Kui on näha, et kõike planeeritud ei jõuta valmis, siis **vähendatakse skoopi**, mitte ei muudeta tähtaega.

14.12.2017 Teema 1 92

Andmebaasid II 2017 © Erki Eessaar

Alternatiivkasutusjuhtude mudelile Kasutusloo võimalik formaat

- Mina kui **lõppkasutaja roll** tahan **[soov]** selleks, et **[põhjendus]**.
- Mina kui üliõpilane tahan **esitada õpingukava**, selleks, et deklareerida ülikoolile oma huvi käesoleval semestril teatud õppeaineid õppida.
- Mina kui **õppejõud** tahan **näha käesoleval semestril minule** õppeaine deklareerinud üliõpilaste nimkirja selleks, et planeerida oma tööprotsessi ja mõista eesootavat töökoormust.

14.12.2017 Teema 1 93

Andmebaasid II 2017 © Erki Eessaar

Sprindi läbi-põlemise graafik

- Kirjeldab tehtud ja tegemata töö hulka.
 - X teljel on aeg (päevade kaupa).
 - Y teljel veel teha oleva töö hulk tundides.
- Ideaalis graafik sprindi lõppedes nullis – kõik tööd said tehtud.
- Graafik on kõigile osalistele nähtaval kohal, et anda infot sprindi hetkeseisust.

<http://www.methodsandtools.com/archive/scrumburndown.php>

14.12.2017 Teema 1 94

Andmebaasid II 2017 © Erki Eessaar

Ekstreemprogrammeerimise (XP) mõningad põhimõisted

14.12.2017 Teema 1 95

Andmebaasid II 2017 © Erki Eessaar

XP mõningad põhimõisted (2)

- Iteratiivne arendamine** – iga iteratsioon kestab 1–3 nädalat.
- Kasutuslood** – kirjutab klient kasutades oma terminoloogiat. Kasulik arenduseks kuluva aja hindamisel, vastuvõtutestide kirjutamiseks.
- Väljalaske plaani planeerimiskohumine** – millised kasutuslood millises väljalaskes.
- Iteratsiooni planeerimiskohumine** – mida teha konkreetse iteratsiooni käigus.

14.12.2017 Teema 1 96

Andmebaasid II 2017 © Erki Eessaar

XP mõningad põhimõisted (3)

- ♦ *Lihitne kavand* – ära lisa süsteemi seda, mida pole küsitud.
- ♦ *Metafoor* – lühike ja kõikidele arusaadav lugu süsteemist ning selle tööst on süsteemi arendustegevust koos hoidvaks võtmeks.
 - Näide: Labor kui ladu.
- ♦ *Testimine (!!)*
 - Test on käivitatav spetsifikatsioon.
- ♦ Kasvata iteratsiooni käigus süsteemi ja siis lihtsusta disaini seda vajadusest refaktoreerides.

14.12.2017 Teema 1 97

Andmebaasid II 2017 © Erki Eessaar

Testid-kõigepealt tehnika

- ♦ Kirjuta test (mis eeldab süsteemilt mingit käitumist).
- ♦ Veendu, et see ebaõnnestub (kuna funktsionaalsust pole realiseeritud).
- ♦ Realiseeri funktsionaalsus, et test õnnestuks.
- ♦ Korda.

Guernsey III, M., 2013. *Test-Driven Database Development. Unlocking Agility*. Addison-Wesley. 315 p.
<http://proquestcombo.safaribooksonline.com/book/databases/database-design/9780132776486>

14.12.2017 Teema 1 98

Andmebaasid II 2017 © Erki Eessaar

XP mõningad põhimõisted (4)

- ♦ Refaktoreerimine (ingl *refactoring*)
- ♦ Paarisprogrammeerimine
 - <http://i.imgur.com/Q1icg6U.gif>
- ♦ Kood on kollektiivses omanduses
- ♦ Pidev integratsioon
- ♦ 40 tunnине töönädal
- ♦ Kliendi esindaja on arendajatele pidevalt kättesaadav
- ♦ Kodeerimisstandardid

14.12.2017 Teema 1 99

Andmebaasid II 2017 © Erki Eessaar

Modelleerimine paindmetoodikate kasutamise korral

- ♦ Paindmetoodika ei välista modelleerimist!
- ♦ Süsteemi modelleeritakse mitmevaateliselt
- ♦ Mudeleid luuakse paralleelselt
- ♦ Lihtsus ja järk-järguline loomine
- ♦ Dubleerimise asemel viita
- ♦ Mudeli ühisvaldamine
- ♦ Grupitöö
- ♦ Kliendi esindaja on arendajatele pidevalt kättesaadav

Modelleerimine ei tohi tähendada (aja, raha, inimeste) **raiskamist**. Mudeleid peab vaja minema!

14.12.2017 Teema 1 100

Andmebaasid II 2017 © Erki Eessaar

Modelleerimine paindmetoodikate kasutamise korral (2)

- ♦ Mudel kui abivahend
 - Ei esitata triviaalset infot
- ♦ Modelleerimisel käepärased vahendid
 - Tahvel, märkmelehed, ...
- ♦ Standardite ja mustrite kasutamine, et tagada kiire loomine ning hea kvaliteet
- ♦ Ka mudeleid saab refaktoreerida
- ♦ <http://www.agilemodeling.com/>

14.12.2017 Teema 1 101

Andmebaasid II 2017 © Erki Eessaar

Projektide edukuse statistika 2011–2015

Suurus	Stiil	Edukas	Väljakutsetega (probleemne)	Ebaõnnestunud
Kõik projektid	Paind (agiilne)	39%	52%	9%
	Kosk	11%	60%	29%
Suured projektid	Paind	18%	59%	23%
	Kosk	3%	55%	42%
Keskmine suurusega projektid	Paind	27%	62%	11%
	Kosk	7%	68%	25%
Väikesed projektid	Paind	58%	38%	4%
	Kosk	44%	45%	11%

Üle 10000 projekti; Standish Group 2015 Chaos Report
<https://www.infoq.com/articles/standish-chaos-2015>

14.12.2017 Teema 1 102

Andmebaasid II 2017 © Erki Eessaar

Paindmetoodikate sobivus teiste arendusmetoodikatega

- Paindmetoodikate põhimõtted võib kasutada erinevates arendusmetoodikates – näiteks RUP.
- Populaarseimad praktikad (2016) – iteratsioonide planeerimine, igapäevased kohtumised, iteratsioonile tagasivaatamise (ja sellest õppimise) kohtumised, iteratsiooni tulemuste ülevaatamine, lühikesed iteratsioonid, väljalasete planeerimine, terve meeskonna kaasamine edenemise planeerimisse ja hindamisse, toote omanik.
- RUP on üldisema metodika – UP (Unified Process)
 - detailne ja laialdaselt kasutatav täpsustus.

14.12.2017 Teema 1 103

Andmebaasid II 2017 © Erki Eessaar

Sobivus teiste arendusmetoodikatega (2)

- UPi agiilsed derivaadid:
 - Agile Unified Process (AUP),
 - OpenUP,
 - Essential Unified Process (EssUP).

14.12.2017 Teema 1 104

Andmebaasid II 2017 © Erki Eessaar

Sobivus teiste arendusmetoodikatega – vesiagiilus

Soblik esimeseks sammuks üleminekul paindmetoodika kasutamisele

Suurema töenäosusega tahavad seda kasutada suured organisatsioonid

14.12.2017 Teema 1 105

Andmebaasid II 2017 © Erki Eessaar

Paindmetoodikate praktikate kombineerimine

- Disciplined Agile Delivery (DAD)
 - Üks põhiautor Scott Ambler (IBM)
 - Arendusprotsessi raamistik (nagu RUP)
 - Kombineerib erinevaid paindmetoodikaid
 - Laiendab Scrumi praktikatega ennast töestanud metodikatest nagu Agile Modeling, Extreme Programming, Unified Process, Kanban, Lean Software Development, Outside In Development.
 - Skaleeritav suurtele ja keerulistele projektidele
 - Saab kohandada näiteks vastavalt meeskonna suurusele, osalistele geograafilisele paiknemisele, regulatiivsetele nõuetele, kultuurilistele eripäradele või tehnilisele keerukusele.

Populaarsus 2016

- Scrum 58%
- Scrum/XP Hybrid 10%
- Hübrid erinevatest praktikatest (custom hybrid) 8%

14.12.2017 Teema 1 106

Andmebaasid II 2017 © Erki Eessaar

Paindmetoodikate praktikate kombineerimine (2)

- Näiteks näeb see metodika ette, et arendus algab *algatamise faasiga*, mis on tüüpiliselt ajaliselt pikem kui üks iteratsioon.
- Selle käigus toimub muuhulgas ka *esialgse arhitektuuri visiooni loomine*.
 - Äriarhitektuur
 - Rakenduste arhitektuur
 - Tehniline arhitektuur

Raamatut saab lugeda Safari andmebaasist:
<http://proquestcombo.safaribooksonline.com/book/software-engineering-and-development/agile-development/9780132810098>

14.12.2017 Teema 1 107

Andmebaasid II 2017 © Erki Eessaar

Veel mõtted

- Agiilsus ei tähenda distsipliineerimatus.
- Bürokratia, tseremooniad ei tähenda distsipliini.
- Paindlikus vs. monumentaalsus ei ole absoluutsed määragud.
- IT projektide õnnestumine/ebaõnnestumine sõltub kõige rohkem **inimestest**, kes neis osalevad (sh nende koostöövõimest).

14.12.2017 Teema 1 108

Andmebaasid II 2017 © Erki Eessaar

Veel mõtteid (2)

- ♦ (Agiilsete) IT projektide õnnestumiseks peaks osalevaid inimesi/meeskondi iseloomustama
 - enesedistsipliin
 - võetakse vastu tööd mida suudetakse ära teha ning tehakse tööd nii efektiivselt kui võimalik
 - iseorganiseeruvus
 - võimekus tööd planeerida, tulemuslikkust hinnata ning koostöös planeeritud tööd teostada
 - eneseteadlikkus
 - võimekus õppida vigadest, leida enda puhul kõige paremini töötavad meetodid ning tehnikad

14.12.2017 Teema 1 109

Andmebaasid II 2017 © Erki Eessaar

Veel mõtteid (3)

- ♦ Tarkvara on liiga keeruline, et oleks võimalik vigu/eksimusi täielikult vältida.
- ♦ Žöpitakse just vigadest/eksimustest!
- ♦ Paindmetoodikad pakuvad raamistiku kuidas vigu/eksimusi võimalikult kiiresti avastada ja parandada.
 - **Võrreldes koskstiliga saab ebaõnnestuda varem ja taasalustada kiiremini.**

14.12.2017 Teema 1 110

Andmebaasid II 2017 © Erki Eessaar

Meyer, B., 2014. *Agile! : The Good, the Hype and the Ugly*, Springer. 170 p.

The Ugly

- ♦ Täielik loobumine planeerimisest, dokumentatsioonist, „suurest pildist“ (arhitektuur).
- ♦ Testidel põhinev arendamine.
 - Tunnelnägemine.
- ♦ Arendusmeeskonnaga koos töötav klient.
 - Ainult ühe inimese vaade; neil, kes tõesti kasulikud pole selleks aega.

14.12.2017 Teema 1 111

Andmebaasid II 2017 © Erki Eessaar

Enamasti karm reaalsus

The Ugly (2)

Soov

Liidetav keerukus –
süsteemi uusi osi saab lisada suhteliselt sõltumatult



Korrutatav keerukus –
süsteemi uute osade lisamisel tuleb arvestada kasvava hulga sõltuvustega



„Suur pilt“ aitab vähendada sõltuvusi ja vältida otsuseid, mida hilisemate pisiparandustega on võimatu korrigeerida.

14.12.2017 Teema 1 112

Andmebaasid II 2017 © Erki Eessaar

The Ugly (3)

- ♦ "These days we do not program software module by module, we program software feature by feature." (B. Meyeri vahendatud paindmetoodika eestkõneleja tsitaat)
- ♦ SQL keel ja standard on näide, mis on sellise lähenemise tulemus.

14.12.2017 Teema 1 113

Andmebaasid II 2017 © Erki Eessaar

Meyer, B., 2014. *Agile! : The Good, the Hype and the Ugly*, Springer. 170 p.

The Ugly (4)

- ♦ Kasutuslood.
 - Liiga vähe detaile. Raske leida ärireegleid (nõuded, mis peavad valdkonna reeglitest lähtuvalt olema alati täidetud).
- ♦ Ei arvestata piisavalt funktsionaalsuste vaheliste sõltuvustega, ei jälgita neid.
- ♦ Projektijuhi asemel paindmetoodika „guru“, kes peale rääkimise midagi ei tee.

14.12.2017 Teema 1 114

Andmebaasid II 2017 © Erki Eessaar

Meyer, B., 2014. *Agile! : The Good, the Hype and the Ugly*, Springer. 170 p.

The Brilliant and the Good

- ♦ Lühikesed iteratsioonid
- ♦ Pidev integratsioon ja testimine
- ♦ Suletud akna reegel
- ♦ Ajakarbi meetod
- ♦ Toote omanik
- ♦ Rõhk töötava tarkvara saavutamisele
- ♦ Edenemise jälgimine
- ♦ Refaktoreerimine
- ♦ Lühikesed päevased kohtumised
- ♦ Meeskonnasisese koostöö olulisuse rõhutamine
- ♦ Takistuste kõrvaldamine, raiksamise vähendamine

14.12.2017 Teema 1 115

Andmebaasid II 2017 © Erki Eessaar

Andmebaasi arendamine paindmetoodika kasutamise korral

- ♦ Andmebas luuakse *iteratiivselt* ja *inkrementaalselt* – valmib järk-järgult.
- ♦ Ei nähta ette eraldi *andmebaasi arendaja* rolli.
 - Arendusmeeskond peab need oskused ära katma.
- ♦ Tihe *koostöö* rakenduse arendajate, andmebaaside arendusmeeskonna (kui selline leidub) ja andmebaaside administraatorite vahel.

14.12.2017 Teema 1 116

Andmebaasid II 2017 © Erki Eessaar

Andmebaasi arendamine paindmetoodika kasutamise korral (2)

- ♦ Andmebaasi struktuur rahuldagu *hetke nõudeid* (KISS, YAGNI).
- ♦ Andmebaasi ja seda kasutavate rakenduste sage ja põhjalik *testimine*.
- ♦ Andmebaasi refaktoreerimine – *väike muudatus* andmebaasi skeemis, et muuta *disaini paremaks*. Refaktoreerimise tulemusena ei tohiks muutuda andmete tähdus ega süsteemi funktsionaalsus.

14.12.2017 Teema 1 117

Andmebaasid II 2017 © Erki Eessaar

Refaktoreerimisest



- ♦ *Disaini* või *koodi halb lõhn* on disaini või koodi sügavama probleemi – **tehnilise võla** – **sümpтом**.
- ♦ See probleem ei takista hetkel süsteemi kasutamist, kuid muudab **edasise arenduse** raskemaks.
- ♦ Halvast lõhnast vabanemiseks tuleb disaini või koodi **refaktorerida**.

14.12.2017 Teema 1 118

Andmebaasid II 2017 © Erki Eessaar

Tehniline võlg



- ♦ "Tarkvara-intensiivsetes süsteemides on tehniline võlg kogum disaini või realisatsiooni konstruktsioone, mis lühikeses perspektiivilis on sobivad. Paraku loovad need tehnilise konteksti, mis teeb tulevikus süsteemi muutmise kulukamaks või võimaluks. Tehniline võlg on tegelik võimalik kohustus, mille mõju on piiratud süsteemi sisemiste omadustega – eeskätt hallatavuse ja edasiarendatusega."

Allikas Izurieta, C., Ozkaya, I., Seaman, C. B., Kruchten, P., Nord, R. L., Snipes, W., Avgeriou, P., 2016. Perspectives on Managing Technical Debt: A Transition Point and Roadmap from Dogma. In QuASoQ/TDA@ APSEC. pp. 84–87

14.12.2017 Teema 1 119

Andmebaasid II 2017 © Erki Eessaar

Tehnilise võla (kui metafoori) metafoor



metafoor – "liik könekujundeid: ülekanne, ühe sõna väljendi tähinduse teisele ülekandmine sarnasuse alusel." (EKI Võõrsõnade leksikon)

- ♦ Tehniline võlg on nagu **otsetee üle muru**.
- ♦ Saab kiiremini kohale, aga pikas perspektiivilis on ümbrus kole ja laiali veetud porist räpane; riite pesule kulub aega ja raha; tolmuosakesed tungivad hingamisteedesse ja teevald hageks.

14.12.2017 Teema 1 120

Andmebaasid II 2017 © Erki Eessaar

Tehnilise võla (kui metafoori) metafoor (2)

Mida rohkem maad tallata, seda **kulukam on **taastamistöö** ja lõpuks ei jäää muud üle kui pinnas eemaldada ja **otsast alata**.**

- Rada üle muru ühes kohas näitab **halba eeskjuu** ja sellised rajad teikavad ka mujale.

14.12.2017 Teema 1 121

Andmebaasid II 2017 © Erki Eessaar

Tehnilise võla ilmnemine

♦ Tehniline võlg võib ilmneda mistahes süsteemiarenduse **tehisest**, sh:

- mudelid,
- testid,
- arhitektuuri kirjeldus,
- andmebaasi disain,
- nt andmete liiasus, kitsenduste mitte jõustamine andmebaasi tasemel
- lähtekood.**

Samuti neid ümbritsevas keskkonnas, sh riistvara ja operatsioonisüsteem.

14.12.2017 Teema 1 122

Andmebaasid II 2017 © Erki Eessaar

Tehnilise võla näide

♦ Süsteem ei ole normaliseeritud süsteem, st selle elementide vahel on **kombinatoorsed efektid** – muutes ühte tuleb muuta ka teisi.

♦ Mida suuremaks süsteem kasvab (kasv on paratamatus), seda rohkem edasikanduvaid muudatusi ja seda kulukam on neid teha.

♦ Samal põhjusel on tehniliseks võlaks ka **kontrollimatu andmete liiasus**. Vt "Andmebaasid I", teema 9

14.12.2017 Teema 1 123

Andmebaasid II 2017 © Erki Eessaar

Tehnilise võla näide (2)

♦ SQL-andmebaasis **deklareerimata välisvõtme kitsendused**.

- Jõustamine tuleb realiseerida kõigis rakendustes (ja üha uesti nende uutes versioonides) ja ka siis saab nendest mööda minna, muutes andmeid väljapoole neid rakendusi.
- Realisatsioon võib olla poolik või vigane ja andmetesse teikavad ikkagi viidete terviklikkuse vead.
- Andmebaasi kasutajatel (andmebaasisüsteem, rakendused, arendajad, administraatorid) jääb saamata oluline informatsioon, mis vajalik näiteks töö optimeerimiseks.

14.12.2017 Teema 1 124

Andmebaasid II 2017 © Erki Eessaar

Tehnilise võla mõõtmine

♦ **Põhiosa** – kulu, mida tuleb kanda võlaga tehise (nt lähtekood, mudel) võlast vabastamiseks.

♦ **Intressid** – probleemidest tulenev täiendav kulu süsteemi hooldamise või edasiarendamise käigus.

♦ Kulu võib mõõta ajas või rahas.

14.12.2017 Teema 1 125

Andmebaasid II 2017 © Erki Eessaar

Tehniline võla dünaamika

♦ Kuni põhiosa tagasi ei maksa, seni intressid jooksevad.

♦ Põhiosa kasvades/kahanedes kasvavad/kahanevad ka intressid.

♦ Kuna süsteemide suurus ja keerukus ajas kasvavad, siis juhul kui võlga tagasi ei maksta, kasvavad ka võla põhiosa ning intressid.

14.12.2017 Teema 1 126

Andmebaasid II 2017 © Erki Eessaar

Tehnilise võla põjhused

- ♦ Arendusprotsess ei pööra vältimisele tähelepanu.
- ♦ Teadlik otsus (nt äravisatavad prototübid).
- ♦ Tegevus või tegevusetus (nt ei lepita omavahel kokku tehnilise võla talutavas tasemes; ei teavitata üksteist leitud või tekitatud võlast).
- ♦ Sündmus:
 - pealesuruv tähtaeg,
 - asjatundliku inimese puudumine,
 - teadmatus tehnilise vahendi võimalustest ja piirangutest,
 - uued tehnilised võimalused seoses uuele platvormile üleminekuga
 - (nt vana andmebaasisüsteem ei toetanud välisvõtmete kontrolli, uus totab).

14.12.2017 Teema 1 127

Andmebaasid II 2017 © Erki Eessaar

Tehnilise võla mõju

- ♦ Süsteemi **väärtuse langemine**.
- ♦ Tulevaste muudatuste **maksumuse kasvamine**.
- ♦ Tulevaste muudatuste **ajakava pikenemine**.
- ♦ Süsteemi **hoolduskulude suurenemine**.
- ♦ Väljalasete hilinemise **ennustatavuse vähinemine**.
- ♦ Süsteemi **kvaliteedi langemine**.

14.12.2017 Teema 1 128

Andmebaasid II 2017 © Erki Eessaar

Veel üks tehnilise võla klassifikatsioon

	Hoolimatu	Mõistlik
Tahtlik	Meil pole andmebaasi disainiks aega	Meil on vaja kiiresti midagi valmis saada, tagajärgedega tegeleme hiljem
Tahtmatu	Mis on normaliseerimine?	Nüüd me teame, kuidas seda tulevikus teha

14.12.2017 Teema 1 129

Andmebaasid II 2017 © Erki Eessaar

Tehniline võla tulu ja kulu

- ♦ Lühiajaline kasu sellest, et süsteemi esialgne versioon valmis kiiresti ja ei pidanud kulutama aega ja raha selle paremini
- ♦ Ühekordne makse.

Kui võlg pole liiga suur ja seda hallata, siis aktsipäriteav!

- ♦ Korduvad maksed.
- Süsteemi edasiarendamine edeneb aeglasmalt, süsteemis on vead ja kvaliteet langeb.

14.12.2017 Teema 1 130

Andmebaasid II 2017 © Erki Eessaar

Tehnilise võlaga kooseksisteerimine

- ♦ Tuleb leida **taskaal tehniliste huvide**
 - tööetika ei luba kehva tööd
 - tulevikus peab tehtut hooldama/edasi arendama
- ♦ ja **ärihuvide**
 - tähtajad suruvad peale
 - raske mõõda tehnilise võla vältimise majanduslikku kasu
- ♦ vahel.

14.12.2017 Teema 1 131

Andmebaasid II 2017 © Erki Eessaar

Tehnilise võlaga kooseksisteerimine (2)

- ♦ **Tunnista** võlga, **jaga** infot arendusmeeskonna sees ja väljapoole, **logi** võla vähendamiseks tehtud ja tegemata töid.
- Kui infot ei jagata (ei suhelda), siis on tulemuseks lisaks ka **sotsiaalne võlg**.
- ♦ Kasuta erinevaid vältimise meetodeid, millest küll ükski **pole hõbekuul**.

14.12.2017 Teema 1 132

Andmebaasid II 2017 © Erki Eessaar

Tehnilise võlaga kooseksisteerimine (3)

• Mõnikord võib olla **majanduslikult kasulikum** võla tagasimaksimise (probleemide kõrvaldamise) asemel

- Võlg **üüberstruktureerida**. Võlast täieliku lahtisaamise asemel asendatakse see teistsuguse võlaga, mille eest tuleb maksta vähem intresse.
- **Maksta intresse ja mitte maksta võla põhiosa.**
- **Minna pankrotti, saada amnestia** (võlga ei pea maksma) ja alustada süsteemiga puhtalt lehelt.

14.12.2017 Teema 1 133

Andmebaasid II 2017 © Erki Eessaar

Tehtavate tööde logi võimalik struktuur

Nähtav	Nähtamatud
Positiivne väärthus	Lisamist ootav funktsionaalsus
Negatiivne väärthus	Arhitektuuri/struktuuri parandused
	Parandamist ootavad vead
	Tagasimaksmist ootav tehniline võlg

Kruchten, P., Nord, R. L., Ozkaya, I., 2012. Technical debt: From metaphor to theory and practice. *IEEE Software*, Vol. 29, No. 6, pp. 18–21.

14.12.2017 Teema 1 134

Andmebaasid II 2017 © Erki Eessaar

Tehnilise võla vältimise meetodeid (pole hõbekuuli)

• Mustrite ja taaskasutatavate mudelite kasutamine.

- Ka kokaraamatu järgi tehes võib supi üle soolata.

• Koodigeneraatorite kasutamine.

- Sisendmudelis olev võlg kandub üle lähtekoodi.
- Kasutatav generaator/teisendus tekitab uut võlga.

• Testimine, testidel põhinev arendamine.

- Ei pruugi kõike leida (nt arhitektuuri probleeme).

14.12.2017 Teema 1 135

Andmebaasid II 2017 © Erki Eessaar

Tehnilise võla vältimise meetodeid (pole hõbekuuli) (2)

• Refaktoreerimine.

- Mõnikord refaktoreerimine enam ei aita.
- Poolik refaktoreerimine võib võlga lisada.

• Veel tegevusi paindmetoodikatest.

- *Definition of Done* kontrollnimekiri.
- Pidev integratsioon.
- Paarisprogrammeerimine.

14.12.2017 Teema 1 136

Andmebaasid II 2017 © Erki Eessaar

Tehniline võlg – lisalugemine

• "Autori hinnangul on suurim tehniline võla kontrolli all hoidmise eeldus sellise **töökultuuri ja -keskkonna loomine**, mis aitab tehnilist võlga vähendada ja juhtida läbi üldise teadlikkuse tõstmise tehnilise võla fenomenist."

- Paide, R., 2017. *Juhumiuring tehnilisest võlast Eesti IKT sektori ettevõtetes*. Magistritöö. TTÜ Tarkvarateaduse instituut. [WWW] <https://digi.lib.ttu.ee/i/?8865>

14.12.2017 Teema 1 137

Andmebaasid II 2017 © Erki Eessaar

Refaktoreerimisest (2)

• Robert C. Martini "**puhta koodi**" (*clean code*) lähenemine propageerib samuti halva lõhnata koodi loomist.

• Andmebaasi arendamine on programmeerimine.

• Halva lõhnata ja puhta koodi mõiste on ühitemoodi oluline nii rakenduse kui andmebaasi arendamisel!!

14.12.2017 Teema 1 138

Andmebaasid II 2017 © Erki Eessaar

Andmebaasi arendamine paindmetoodika kasutamise korral (3)

- Burns, L., 2011. *Building the agile database : how to build a successful application using agile without sacrificing data management*, Technics Publications. 276 p. [WWW] http://www.estet.ee/record=b4425682*est
 - Toetab kontseptuaalse andmemudeli ning loogilise ja füüsiline disaini andmemudeli loomist ning andmebaasi kapseldamist andmebaasi avaliku liidese taha.

14.12.2017 Teema 1 139

Andmebaasid II 2017 © Erki Eessaar

Andmebaasi arendamine paindmetoodika kasutamise korral (4)

- YAGNI – üldine andmete arhitektuuri kirjeldus (“suur pilt”) jääb pigem loomata, sest (on moes arvata, et) see võtab liiga palju aega, ei anna kohest kasu ja nõuded süsteemile pole lõpuni selged.
- Andmebaasi disaini käigus valitakse **esimene sobiv lahendus**, mitte ei otsita parimat lahendust.
- Probleemidega tegeletakse **käigu pealt**.
- Võivad tekkida probleemid (*architecture breakers*), mida ainult **refaktoreerimisega ei paranda**.

14.12.2017 Teema 1 140

Andmebaasid II 2017 © Erki Eessaar

Miks peaks analüüs (sh kontseptuaalne andmemudel) olema tasemel?

- Nõuete kogumise (analüüsi) faasis tehtud vigade parandamiseks kuluv pingutus kasvab iga järgmise süsteemiarenduse faasiga **kolm korda**.
 - Disaini faasis on neid **3**,
 - realiseerimise faasis **9** ja
 - testimise faasis **27** korda kulukam parandada kui nõuete kogumise faasis.

Jaakkola, H., Hemo, J., Welzer-Druzovec, T., Thalheim, B., Mäkelä, J., 2016. Why Information Systems Modelling Is Difficult. In: *SQAMIA*. pp. 29–39.

14.12.2017 Teema 1 141

Andmebaasid II 2017 © Erki Eessaar

Andmebaasi arendamine paindmetoodika kasutamise korral (5)

- Hästi kavandatud andmebaas lihtsustab oluliselt andmebaasi kasutavate programme loomist.
- Kuidas tagada **kvaliteetne** andmebaasi disain olukorras, kus süsteemi tellijad ja paindmetoodikat järgivad arendajad soovivad töötava tarkvara **võimalikult kiiret** valmimist?

14.12.2017 Teema 1 142

Andmebaasid II 2017 © Erki Eessaar

Mida on vaja tarkvara (ja andmebaasi) loomiseks?



- Tarkvara kirjutamiseks on vaja **nõudeid**.
- Nõuete esitamiseks on vaja tunda **valdkonda**, mille jaoks tarkvara luuakse.
- Selle mõttekäigu autoriks on taani teadlane Dines Bjørneri ning seda mõttekäiku nimetatakse *tarkvara triptühhooniaks*.

14.12.2017 Teema 1 143

Andmebaasid II 2017 © Erki Eessaar

Üks võimalik lähenemine!

Andmebaasi agilne mustripõhine arendamine

- Tüüpilisest organisatsiooni andmemudelist:
 - üle **50%** üldised osad, mida saab kasutada enamikes organisatsionides,
 - **umbes 25%** on tööstusharu-spetsiifiline (nt tervishoid, kindlustus),
 - **umbes 25%** on organisatsiooni-spetsiifiline.
- **Lahendus** – kasuta *mustreid/taaskasutatavaid andmemudeliteid*, et luua kiiresti esialgne andmete arhitektuuri kirjeldus (“suur pilt”) ja seda iteratsioonide käigus täpsustada.

14.12.2017 Teema 1 144

Andmebaasid II 2017 © Erki Eessaar

Mustrite raamatud TTÜ raamatukogus

- ♦ Silverston, L., 2001a. *The Data Model Resource Book. A Library of Universal Data Models for All Enterprises*. Revised Edition. Vol. 1. Wiley Computer Publishing, 2001. 540 p. http://www.esther.ee/record=b1918808*est
- ♦ Silverston, L., 2001b. *The Data Model Resource Book. A Library of Universal Data Models by Industry Types*. Revised Edition. Vol. 2. Wiley Computer Publishing, 2001. 556 p. http://www.esther.ee/record=b1918820*est

14.12.2017 Teema 1 145

Andmebaasid II 2017 © Erki Eessaar

Mustrite raamatud TTÜ raamatukogus (2)

- ♦ Silverston, L., Agnew, P. 2009. *The Data Model Resource Book*. Vol. 3, Universal Patterns for Data Modeling. Wiley, 2009. 606 p. http://www.esther.ee/record=b4577685*est
- ♦ Fowler, M., 1997. *Analysis Patterns: Reusable Object Models*. Addison Wesley. 357 p. http://www.esther.ee/record=b1686448*est

14.12.2017 Teema 1 146

Andmebaasid II 2017 © Erki Eessaar

Mustrite raamatud TTÜ raamatukogus (3)

- ♦ Piho, G., 2011. *Archetypes Based Techniques for Development of Domains, Requirements and Software. Towards LIMS Software Factory*. Doktoritöö, Tallinna Tehnikaülikool. 170 p. <https://digi.lib.ttu.ee/i/?636>
- ♦ Blaha, M., 2010. *Patterns of Data Modeling*. CRC Press. 245 p. <http://ebookcentral.proquest.com/lib/tuee>

14.12.2017 Teema 1 147

Andmebaasid II 2017 © Erki Eessaar

Integratsiooni muster (mustrite muster)

Allikas: Giles, J., 2012. *The Nimble Elephant. Agile Delivery of Data Models Using a Pattern-based Approach*. Technics Publication, LLC. 238 p.

Integratsiooni muster esitab komponent-mustrite seostamise raamistiku.

14.12.2017 Teema 1 148

Andmebaasid II 2017 © Erki Eessaar

Komponentmustri Party (osapool) variatsioon

• Üldistusseoste kirjeldamisel lahtuti juhenditest, mis esitatakse artiklis: Guizzardi, G., Wagner, G., Guarino, N., Sinderen, M. (2004). An Ontologically Well-Founded Profile for UML Conceptual Models. In Proceedings of CAiSE04, LNCS 3084, pp. 112-126.

14.12.2017 Teema 1 149

Andmebaasid II 2017 © Erki Eessaar

Andmebaasi agilne mustripõhine arendamine (2)

- ♦ Lihtsustan süsteemi spetsiifikat arvestades integratsiooni mustrit.
- ♦ Leian lihtsustatud integratsiooni mustri põhjal põholemitübid (põhiobjektid) ja nendevahelised seosed.
 - Igale komponentmustriile vastab eraldi põholemitüp.
 - Igale komponentmustrite vahelisele seosele vastab põholemitüpide vaheline seos.

14.12.2017 Teema 1 150

Andmebaasid II 2017 © Erki Eessaar

Andmebaasi agiline mustripõhine arendamine (3)

- Igale põhiolemitüübile vastab süsteemi äriarhitektuuris eraldi **register**.
- Iga registrit haldab (loob/loeb/muudab/kustutab) üldjuhul eraldi **funktionaalne allsüsteem**.
- Iga sellise registri detailsemaks kirjeldamiseks saab võtta aluseks vastava **komponentmustri**.
 - Esimese kiire tulemuse jaoks kasutatakse mustri standardset versiooni
 - Seejärel kohandatakse mustrit konkreetse süsteemi spetsiifikat arvestades.

14.12.2017 Teema 1 151

Andmebaasid II 2017 © Erki Eessaar

Mustrite kasutamise näide – päästeteenindus



- Tuleb säilitada infot **sündmuste** (tulekahju, keemiaõnnetus jne) kohta, millest igaüks toimub mingis **asukohas**.
- Sündmus tingib hulga **ülesannete** täitmise, milleks läheb vaja **ressurssse/vara** (autod, raadiosaatjad jne) ning mingites **rollides osapooli**.

14.12.2017 Teema 1 152

Andmebaasid II 2017 © Erki Eessaar

Mustrite kasutamise näide – päästeteenindus (2)

"Programmeerimise asemel pane kokku" - Douglas McIlroy, NATO tarkvaraarenduse konverents, 1968.

Päästeteenistuse eripära arvestades lihtsustatud integratsiooni muster, mis on aluseks päästeteenistuse infosüsteemi äriarhitektuuri koostamisele (*Rough Design Up Front*).

14.12.2017 Teema 1 153

Andmebaasid II 2017 © Erki Eessaar

Mustrite kasutamise näide – päästeteenindus (3)

Esialgne äriarhitektuuri fragment

- Funktionaalsed allsüsteemid**
 - Sündmuste arvestus
 - Asukohtade arvestus
 - Ülesannete arvestus
 - Ressursside/vara arvestus
 - Osapoolte arvestus
 - Klassifikaatorite arvestus
- Registrid** (igal registril on üks seda haldav funkts. allsüsteem)
 - Sündmuste register
 - Asukohtade register
 - Ülesannete register
 - Ressursside/vara register
 - Osapoolte register
 - Klassifikaatorite register

14.12.2017 Teema 1 154

Andmebaasid II 2017 © Erki Eessaar

Ettevõtte arhitektuur

ArchiMate Framework (pildil) määrab ära, milliseid arhitektuuri mudeleid võiks luua.

ArchiMate on ettevõtete arhitektuuri modelleerimise keel (<https://en.wikipedia.org/wiki/ArchiMate>)

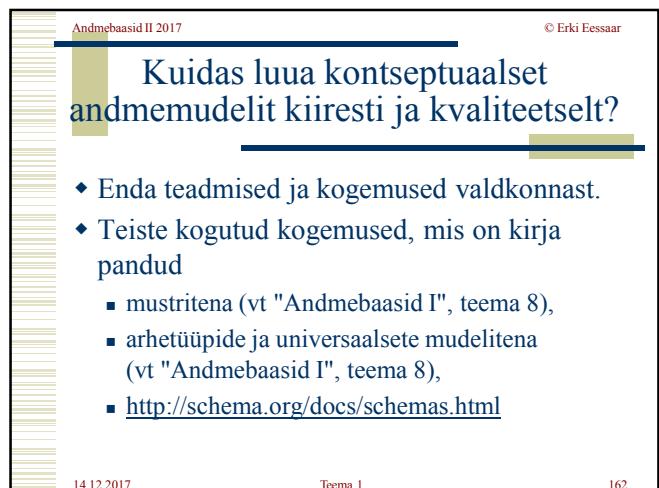
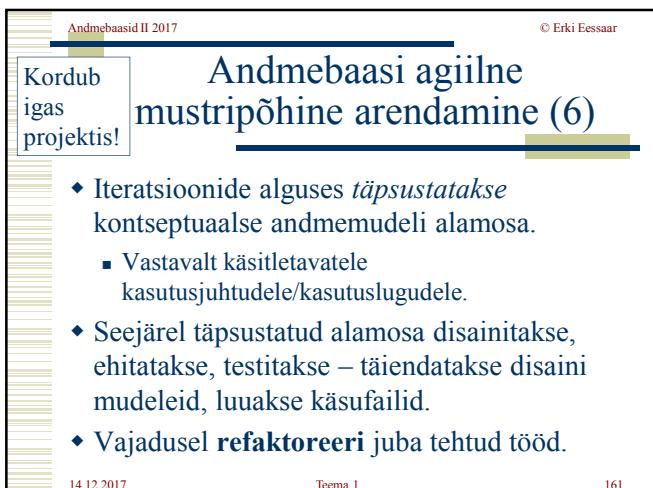
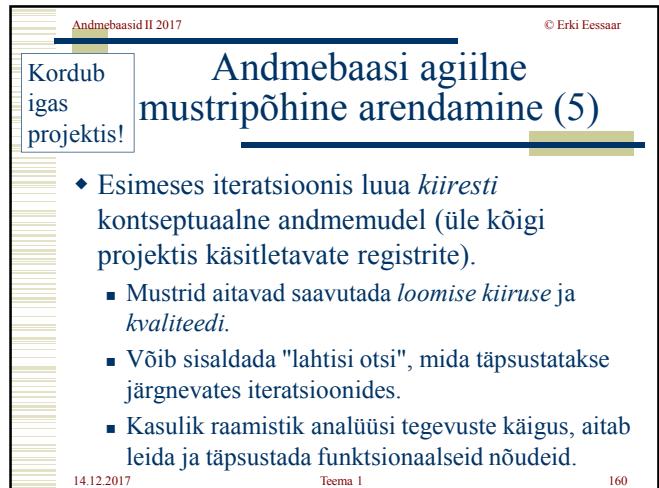
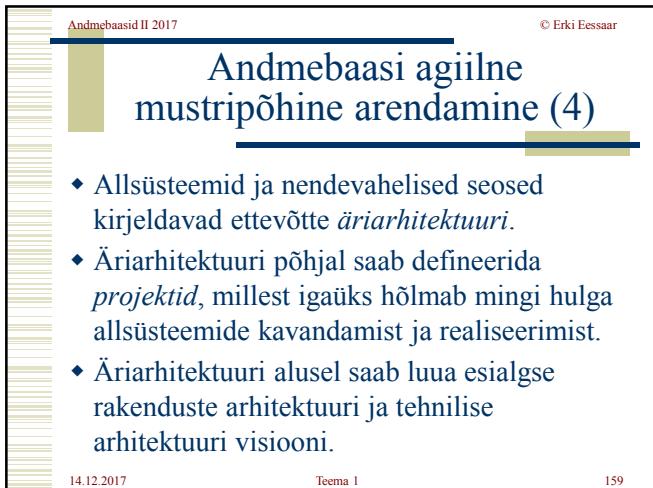
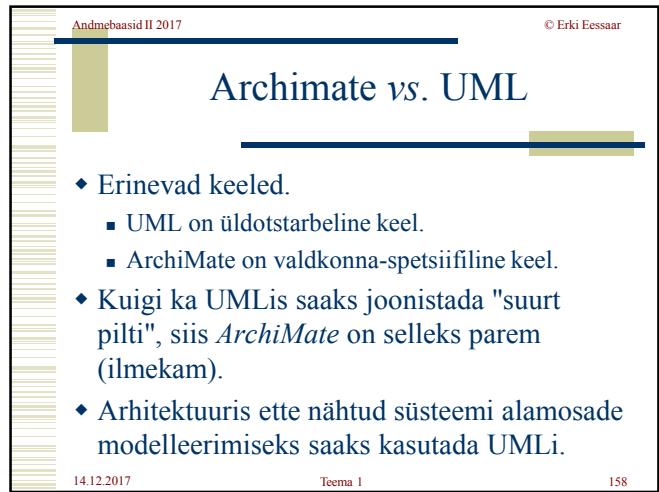
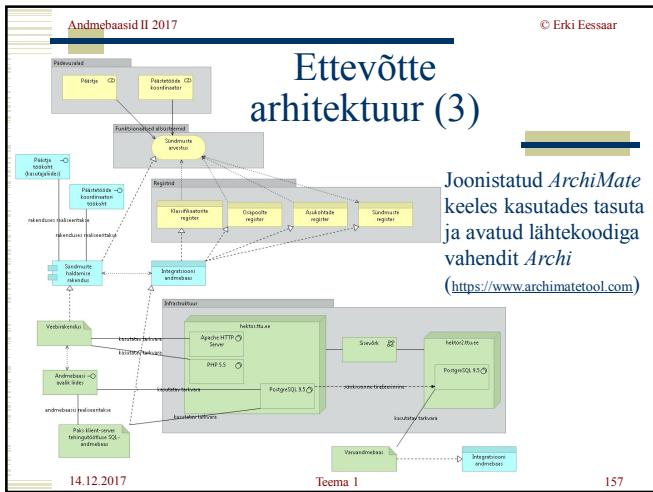
14.12.2017 Teema 1 155

Andmebaasid II 2017 © Erki Eessaar

Ettevõtte arhitektuur (2)

Ettevõtte äriarhitektuur, mille järgi infosüsteem koosneb kolme liiki allsüsteemidest on sellega kenasti kooskõlas.

14.12.2017 Teema 1 156



Andmebaasid II 2017 © Erki Eessaar

Kontseptuaalne andmemudel

- ♦ Saab olla aluseks funktsionaalse nõuete täpsustamisele.
- ♦ Küsimused stiilis:
 - Kas tellimuse puhul tuleb registreerida, et tegemist on kingitusega?
 - <http://schema.org/Order>
 - Kas on õige, et tellimus tekitab arve ja see tekib koos tellimusega
 - [Tellimus]-1-----1-[Arve]

14.12.2017 Teema 1 163

Andmebaasid II 2017 © Erki Eessaar

Juhtumianalüüs

- ♦ Toomast, K., 2016. *Ühe valdkonna modelleerimine kasutades erinevaid arhetüüpide hulkasid*. Bakalaureusetöö. TTÜ Informaatikainstituut. [WWW] <https://digi.lib.ttu.ee/i/?8766>
- Modelleerib hotelli infosüsteemi alamosa kontseptuaalset andmemudelit kasutades erinevaid arhetüüpide hulkasid.

14.12.2017 Teema 1 164

Andmebaasid II 2017 © Erki Eessaar

Juhtumianalüüs – järeldus – (ükski) metodika pole hõbekuu!

- ♦ "Tööst selgus veel, et arhetüüpidel põhinev kontseptuaalse andmemudeli loomine ei ole algajale üldsegi lihtne. See viis nõuab samuti aega ja pikemat läbitöötlemist. Näiteks on keeruline põhiobjektide ja arhetüüpide vastavuse leidmine. Kindlustamaks antud meetodi kasutamise hea lõppulemus tuleb enne modelleerima hakkamist rohkelt eeltööd teha."

14.12.2017 Teema 1 165

Andmebaasid II 2017 © Erki Eessaar

Andmebaasi parendamine (refaktoreerimine)

- ♦ Andmebaasi parendamine on praktika, mida võib (ja sageli on vaja) kasutada sõltumata kasutatavast arendusmetoodikast.
- ♦ Uute süsteemide loomisel tehke kohe nii, et oleks vaja *võimalikult vähe* refaktoreerida.
 - Paindmetoodikad soovitavad üldiselt vastupidist – arendus koosnegu lühikestest *proovi-testi-paranda* iteratsioonidest.

14.12.2017 Teema 1 166

Andmebaasid II 2017 © Erki Eessaar



Andmebaasi parendamine (2)

- ♦ Igasugune hilisem parandamine on **kulu** vörreledes sellega, et asi kohe hästi teha.
 - Tarkvaral omadus muutuda ajas **keerulisemaks** (vt tarkvara evolutsiooni seadused) – mida keerulisem, seda raskem hiljem parandada.
- ♦ Halvasti tehtud asi (andmebaas, rakendus jne) on **tehniline võlg** (*technical debt*) – loodame, et suudame selle hiljem ära maksta, kuid praktikas hakkab võlg sageli kuhjuma.

14.12.2017 Teema 1 167

Andmebaasid II 2017 © Erki Eessaar

Näide pärandsüsteemide halva kvaliteedi kohta

- ♦ 2001
 - Uuritud 35 andmebaasist 75% olid jõustatud primaarvõtmed ja 10% olid jõustatud välisvõtmel.
- ♦ 2015
 - Uuritud kahes vaba tarkvara SQL-andmebaasis olid ühes primaarvõtmel 97% tabelitest ja teisel 78% tabelitest. Välisvõtmel olid ühes üldiselt jõustatud, teises üldse mitte.
- ♦ Lapõnin, A., 2015. *Andmebaasi loogilise ja füüsilise disaini antimistrite esinemine mõnedes vaba tarkvara poolt kasutatavates SQL-andmebaasides*. Bakalaureusetöö. TTÜ Informaatikainstituut. <https://digi.lib.ttu.ee/i/?3459>

14.12.2017 Teema 1 168

Andmebaasid II 2017 © Erki Eessaar

Andmebaasi parendamise vajalikkus ning tulemused

- ♦ Miks on andmebaasi parendamist vaja?
 - Pärandsüsteemide disaini kvaliteedi parandamine.
 - Andmebaaside evolutsiooni toetamine.
- ♦ Oodatavad tulemused.
 - Olemasoleva disaini paranemine (nt operatsioonide töökiiruse parandamine ja andmete liiasuse vähendamine).
 - Disaini arusaadavuse suurenemine.
 - Uue funktsionaalsuse lisamise lihtsustamine
 - Arenduskiiruse ja -mugavuse tõstmine.

14.12.2017 Teema 1 169

Andmebaasid II 2017 © Erki Eessaar

Andmebaasi parenduste näiteid

- ♦ Tagamine, et samasisulised andmed on samade andmetüüpide ja väljapikkustega veergudes.
 - Nt kõigis tabelites veerg *kommentaar* on tüüp *VARCHAR(2000)*
- ♦ Veeru ümbernimetamine.
 - Nt kõigi veergude nimed eesti keeles.
 - Nt *first_name => given_name, last_name => surname*
- ♦ Veerule kitsenduse lisamine.

14.12.2017 Teema 1 170

Andmebaasid II 2017 © Erki Eessaar

Andmebaasi parenduste näiteid (2)

- ♦ Klassifikaatorite tabeli lisamine.
 - Nt *Töötaja* tabelis CHECK (amet IN ('insener', 'sekretär', 'direktor')) => Tabel *Amet(amet_kood nimetus)* ja *Töötaja* tabelis välivõtme kitsendus (*amet_kood*) Viitab *Amet(amet_kood)*
- ♦ Ühe veeru asendamine mitmega.
 - Nt nimi => eesnimi, perenimi
 - Näide: "Pertel" ja "Lukas" võivad olla nii eesnimi kui perenimi

14.12.2017 Teema 1 171

Andmebaasid II 2017 © Erki Eessaar

Andmebaasi parenduste näiteid – rutiinid

- ♦ Suur parameetrite hulk, juhparameetrid, mittekohustuslikud parameetrid viitavad, et rutiinil on rohkem kui üks ülesanne.
 - Sellist rutiini on keeruline kasutada.
 - *Otstarbe lahususe* printsibi järgimisel peaks igal rutiinil olema üks ülesanne.
- ♦ Asenda üks suur rutiin mitme väiksema rutiiniga, millel igaühel üks ülesanne.

14.12.2017 Teema 1 172

Andmebaasid II 2017 © Erki Eessaar

SQL-andmebaasi parendamise kirjandust

- ♦ Ambler, S. W, Sadalage, P. J., 2006. *Refactoring Databases: Evolutionary Database Design*. Addison Wesley Professional. <http://proquestcombo.safaribooksonline.com/0321293533>
- ♦ Ambler, S., Catalog of Database Refactorings. [WWW] <http://www.agiledata.org/essays/databaseRefactoringCatalog.html>

14.12.2017 Teema 1 173

Andmebaasid II 2017 © Erki Eessaar

Parendus, mis pole refaktoreerimine

```

classDiagram
    class Lepingu {
        PKlepingu_nr: INTEGER
    }
    class Lepingu_seos {
        PKlepingu_seos_id: INTEGER
        PKlepingu_nr: INTEGER
    }

    Lepingu "*" --> "0..1" Lepingu_seos : PKlepingu_nr
    Lepingu "*" --> "0..1" Lepingu_seos : PKlepingu_seos_id
  
```

- ♦ Allikas: antimuster "Sümmeetrliline seos", mis on kirjeldatud raamatus:
Blaha, M., 2010. *Patterns of Data Modeling*. CRC Press. <http://ebookcentral.proquest.com/lib/tuee>

14.12.2017 Teema 1 174

Andmebaasid II 2017 © Erki Eessaar

Parendus, mis pole refaktoreerimine (2)

- Andmebaasis on vaja registreerida sümmeetrilisi seoseid – kui leping 3 on seotud lepinguga 4, siis on leping 4 seotud lepinguga 3.
- Algse lahenduse korral pole selge, kuidas registreerida seos kahe lepingu vahel.

14.12.2017 Teema 1 175

Andmebaasid II 2017 © Erki Eessaar

Parendus, mis pole refaktoreerimine (3)

- Kas nii (**variant A**)?
 - ROW (leping1 3, leping2 4)
 - Aga mille alusel valida, milline leping on *leping1* ja milline leping on *leping2*?
- Või hoopis nii (**variant B**)?
 - ROW(leping1 3, leping2 4)
 - ROW(leping1 4, leping2 3)
- Uus lahendus võimaldab ka registreerida seoseid, kus osaleb üle kahe lepingu.
- Leping saab olla maksimaalselt ühes seoses.

14.12.2017 Teema 1 176

Andmebaasid II 2017 © Erki Eessaar

Miks algne lahendus ei sobi?

- Algne disain võimaldab kasutada nii varianti **A** kui ka **B**.
- Tekib oht, et osade seoste puhul kasutatakse ühte ja osade puhul teist – tulemuseks segadus andmete kasutamisel.

14.12.2017 Teema 1 177

Andmebaasid II 2017 © Erki Eessaar

Parendus, mis oleks refaktoreerimine

- Säilitada tabelite struktuur, kuid jõustada **variant B**.
- PostgreSQLis.
 - `CREATE TABLE Lepingu_seos (leping1 INTEGER NOT NULL, leping2 INTEGER NOT NULL);`
 - `CREATE UNIQUE INDEX pk_lepingu_seos ON Lepingu_seos (Least(leping1,leping2), Greatest(leping1,leping2));`

14.12.2017 Teema 1 178

Andmebaasid II 2017 © Erki Eessaar

Andmebaasi muutmine on keeruline ülesanne

- Andmebaas on infosüsteemi **tuum**.
- Andmebaasi struktuuri/käitumise muutmisel võib olla vaja teha muudatusi nii andmebaasi sees kui ka väljaspool andmebaasi (rakendused, testid, dokumentatsioon).
- Väljaspool andmebaasi vajalikke muudatusi aitab vähendada andmebaasi *kapseldamine virtuaalse andmete kihiga*.

14.12.2017 Teema 1 179

Andmebaasid II 2017 © Erki Eessaar

Virtuaalne andmete kiht kui refaktoreerimise lihtsustaja

Andmebaasid II 2017 © Erki Eessaar

- Osa parendusi sellised, mis tehakse baas-andmestruktuurides ja sellest tulenevalt virtuaalses andmete kihis, kuid mille töttu pole vaja muuta andmebaasi kasutavaid programme.

14.12.2017 Teema 1 180

Andmebaasid II 2017 © Erki Eessaar

Virtuaalne andmete kihtrik kui refaktoreerimise lihtsustaja (2)

- Ilma virtuaalse andmete kihita – mida rohkem programme andmebaasi kasutab, seda rohkem võtab muudatuste tegemine aega.
- Muudatusteks kuluv aeg on määratud programmiga, mille muutmine võtab kõige rohkem aega.

14.12.2017 Teema 1 181

Andmebaasid II 2017 © Erki Eessaar

Refaktoreerimise käigus tehtavad tegevused

- Andmebaasi **mudelipõhise spetsifikatsiooni** muutmine.
- Andmebaasi **skeemi** muutmine.
- Andmete** uutesse struktuuridesse **ülekandmine**.
- Andmebaasi poole pöördumiseks kasutatava **koodi muutmine** (sh rakenduste kood ja andmete laadimiseks kasutatav kood).
- Andmebaasi **testide** muutmine.
- Sage ja põhjalik **testimine**.

14.12.2017 Teema 1 182

Andmebaasid II 2017 © Erki Eessaar

Andmebaasi parendamise (refaktoreerimise) võimalik protsess

- Autor: Scott W. Ambler: <http://www.agiledata.org>
- Eeltingimused.
 - Üks või rohkem arendaja/arendajate paari andmebaasi.
 - Keskne arenduse andmebaas.
 - Tööbaas.
- Parendamist viib läbi andmebaasi administraator koostöös rakenduste arendajatega.

14.12.2017 Teema 1 183

Andmebaasid II 2017 © Erki Eessaar

Andmebaasi arenduskeskkonna võimalik ülesehitus

14.12.2017 Teema 1 184

Andmebaasid II 2017 © Erki Eessaar

Andmebaasi parendamise võimalik protsess (2)

- Parenduse vajalikkuse hindamine
 - Päringud süsteemikataloogist, et leida võimalikke disaini vigu
<http://apex.ttu.ee/queryexecution/web/app.php/et/login>
http://staff.ttu.ee/~eessaar/files/Design_flaws_queries.pdf
- Parenduse mõju hindamine (milliseid teisi objekte tuleb muuta)
 - Hecateus programm:
<http://www.cs.uoi.gr/~pvassil/projects/hecataeus/>

14.12.2017 Teema 1 185

Andmebaasid II 2017 © Erki Eessaar

Andmebaasi parendamise võimalik protsess (3)

OWNER	NAME	TYPE	REFERENCED_OWNER	REFERENCED_NAME	REFERENCED_TYPE
C#TUD1	PACK_EMP	PACKAGE	C#TUD1	EMP	TABLE
C#TUD1	PACK_EMP	PACKAGE	C#TUD1	EMP	TABLE
C#TUD1	GET_EMP_REC	PROCEDURE	C#TUD1	EMP	TABLE
C#TUD1	EMP_MANAGER	VIEW	C#TUD1	EMP	TABLE

- Muudatuste mõju hindamiseks andmebaasi piires saab kasutada päringuid süsteemikataloogi põhjal.

14.12.2017 Teema 1 186

Andmebaasid II 2017 © Erki Eessaar

Andmebaasi parendamise võimalik protsess (4)

- ◆ Kõige sobivama parendamise viisi valimine.
 - Kuidas esitada isikunimesid, klassifikaatoreid?
 - Kas kasutada domeene või mitte? ...
- ◆ Muudatuse sisseviimine arendajate andmebaasidesse.
- ◆ Üleminekuperiood – rakenduste koodis muudatuste tegemine.
- ◆ Vana lahenduse eemaldamine ja vajadusel andmete ülekanne uutesse struktuuridesse.

14.12.2017 Teema 1 187

Andmebaasid II 2017 © Erki Eessaar

Andmebaasi parendamise võimalik protsess (5)

- ◆ Andmebaasi loomise skripti täiendamine.
- ◆ Andmebaasi ja rakenduse koos testimine.
- ◆ Muudatuste dokumenteerimine.
- ◆ Muudatuste ülekanne kesksesse arenduse andmebaasi.
- ◆ Andmebaasi ja rakenduste koos testimine.
- ◆ Muudatuste ülekanne tööbaasi.

14.12.2017 Teema 1 188

Andmebaasid II 2017 © Erki Eessaar

Andmebaasi skeemi muutmise mõju rakenduste lähtekoodile

- ◆ Põhilised atomaarsed skeemimuudatused:
 - tabeli lisamine,
 - veeru lisamine,
 - veeru andmetüübi muutmine.
- ◆ Skeemimuudustute põhilised põhjused:
 - funktsionaalsuse lisamine,
 - struktuuri ja andmekvaliteedi refaktooreerimine.

Uuring kümmne populaarse avatud lähtekoodiga andmebaasirakenduse põhjal.

14.12.2017 Teema 1 189

Andmebaasid II 2017 © Erki Eessaar

Automatiserimine

- ◆ Kui eesmärgiks on saavutada tulemus **kiiresti ja kvaliteetselt ning kaotada (aja) raiskamine**, siis on automatiserimisest palju abi.
- ◆ Automatiserimisest on kasu igasuguse arendusmetoodika korral, kuid kindlasti on see kooskõlas paindmetoodika vaimuga.

14.12.2017 Teema 1 190

Andmebaasid II 2017 © Erki Eessaar

Mida saab näiteks andmebaasi arenduses automatiseerida?

- ◆ Mudelipõhise arenduse korral mudelist uute mudelite, koodi, dokumentatsiooni, testide genereerimine ning sünkroniseerimine.
- ◆ Refaktoreerimise vajaduste leidmine.
- ◆ Testide läbiviimine.
- ◆ Muudatuste paigaldamine.
- ◆ Versioonikontroll ja muudatuste haldus.
 - <http://www.liquibase.org/>

14.12.2017 Teema 1 191

Andmebaasid II 2017 © Erki Eessaar

Evolutsioon



- ◆ Evolutsioon on loomulik, jätkjärguline muudatuse protsess, mille käigus muutub sotsiotehnilises süsteemis nii sotsiaalne kui ka tehniline pool.

14.12.2017 Teema 1 192

Andmebaasid II 2017 © Erki Eessaar

Mis käivitab infosüsteemi evolutsiooni?

- ◆ Tehnilised uuendused.
- ◆ Muutused ärikeskkonnas.
- ◆ Muutused seadusandluses.
- ◆ Avastatakse programme uusi kasutusvõimalusi, kuid selleks on vaja neid veidi kohendada.
- ◆ Osapoolte paranenud arusaam süsteemist ja selle valdkonnast.
- ◆ Soov pakkuda paremat teenust.

14.12.2017 Teema 1 193

Andmebaasid II 2017 © Erki Eessaar

Lehmani tarkvara evolutsiooni seaduseid

Tarkvara ja andmebaasid on osa infosüsteemi "tehnilisest pooltest".

- ◆ Tarkvara arengu käigus **suureneb selle keerukus**, kui just ei tehta pingutusi keerukuse kontrolli all hoidmiseks või vähendamiseks.
- Keerukuse kasv muudab tarkvaras muudatuste tegemise raskemaks.

Lehman, M.M., 1996. *Laws of Software Evolution Revisited*. Proceedings of 5th European Workshop, EWSPT '96 Nancy, France, October 9-11, 1996. LNCS Vol. 1149, Springer, Berlin, pp. 108-124.

14.12.2017 Teema 1 194

Andmebaasid II 2017 © Erki Eessaar

Lehmani tarkvara evolutsiooni seaduseid (2)

- ◆ Tarkvara pakutavat **funktionaalsust** tuleb kogu selle eluea jooksul **suurendada**, et säilitada kasutajate **rahulolu** sellega.
- ◆ Kui tarkvara pidevalt ei **hallata** ega **kohandata** keskkonnas toimuvate muudatustega, siis hakkab selle **kvaliteet langema**.

Tarkvara vajab andmebaase. Tarkvara muutmine võib tingida vajaduse muuta andmebaase.

14.12.2017 Teema 1 195

Andmebaasid II 2017 © Erki Eessaar

Andmebaasi evolutsioon

- ◆ *Andmebaasi skeemi evolutsioon* – muudatuste tegemine andmebaasi struktuuris ja või käitumises (kitsendused).
- *Andmebaasi refaktoreerimine*, mille käigus säilitakse andmebaasis talletatava info sisu.

14.12.2017 Teema 1 196

Andmebaasid II 2017 © Erki Eessaar

Andmebaasi evolutsioon (2)

- ◆ *Semantiline evolutsioon*, mille käigus muutub andmebaasis talletatava info sisu (näiteks otsustatakse mingite andmete registreerimisest loobuda või siis vastupidi, hakata registreerima täiendavaid andmeid).

14.12.2017 Teema 1 197

Andmebaasid II 2017 © Erki Eessaar

Andmebaasi evolutsioon (3)

- ◆ *Andmebaasi migratsioon*, mille korral minnakse üle ühelt andmebaasisüsteemilt teissele.
- ◆ *Andmebaaside integratsioon*, et muuta erinevates andmebaasides olevad andmed kättesaadavaks kõigile rakendustele.

14.12.2017 Teema 1 198

Andmebaasid II 2017 © Erki Eessaar

Andmebaaside skeemi evolutsiooni seaduspärasused

- Avatud lähtekoodiga tarkvara põhjal on tehtud mitmeid uuringuid evolutsiooni seaduste paikapidavusest andmebaasides.
 - Võimalik võrrelda suurt hulka sama tarkvara jaoks mõeldud SQL-andmebaaside versioone.
 - Näide:* Skoulis, I., Vassiliadis, P., Zarras, A. V., 2015. Growing up with stability: How open-source relational databases evolve. *Information Systems*, 53, pp. 363–385.

14.12.2017 Teema 1 199

Andmebaasid II 2017 © Erki Eessaar

Skoulis et al. (2015) artikli tulemused

Andmebaaside skeemi evolutsiooni seaduspärasusi (2)

- Suurem osa tarkvara evolutsiooni seaduseid **kehtib** ka andmebaaside korral.
- Erinevat tüüpi faktide hulk, mida andmebaasis saab registreerida, ajas kasvab.
- Skeemi suurus (tabelite/veergude arv) kord kasvab, kord kahaneb (andmebaasi refaktoreerimine, nt liiasuse eemaldamine), kuid üldine **tendents on kasvu suunas**.

14.12.2017 Teema 1 200

Andmebaasid II 2017 © Erki Eessaar

Skoulis et al. (2015) artikli tulemused

Andmebaaside skeemi evolutsiooni seaduspärasusi (3)

- Andmebaasi skeemi suuruse kasv toimub **hüpetega**.
 - Kiirele kasvule järgnevad stabiilsed, väikese muudatuste arvuga/muudatusteta perioodid.
 - Vahepeal võib skeem isegi väiksemaks muutuda.
 - Ei järgi** regulaarsete **südamelöökide** mustrit.
- Skeemi kasvu tingib vajadus täiendada tarkvara **funktionaalsust**.

14.12.2017 Teema 1 201

Andmebaasid II 2017 © Erki Eessaar

Skoulis et al. (2015) artikli tulemused

Andmebaaside skeemi evolutsiooni seaduspärasusi (4)

- Skeemimuudatuste arv väheneb koos tarkvara "**küpsemaks**" saamisega.
- Skeemi kasvu kiirus **jääb alla** süsteemi funktsionaalsuse kasvu kiirusele.
- Andmebaasi skeem muutub aja jooksul **lihtsamaks** (refaktoreerimisse suhtutakse tõsiselt ja see aitab).

14.12.2017 Teema 1 202

Andmebaasid II 2017 © Erki Eessaar

Kuidas hoida andmebaaside keerukust kontrolli all?

- Leia äriarhitektuur (sh registrid).
- Kavanda registrid lähtuvalt andmete modelleerimise mustrites/universaalsetest mudelitest.
 - Tagab muutuvatele nõuetele hästi vastupidavad andmestruktuurid.
- Järgi häid andmebaasi disaini praktikaid (ning väldi halbu – SQL disaini antimustrid).
- Vajadusel refaktoreeri andmebaasi.

14.12.2017 Teema 1 203

Andmebaasid II 2017 © Erki Eessaar

Näiteid evolutsiooni toetusest (Oracle)

- Change Management Pack* – töövahend skeemide võrdlemiseks ja ühelt skeemilt teisele ülemineku skripti loomiseks.
- Online table redefinition* – tabeli struktuuri muudatus koosneb hulgast tegevustest, millega enamiku käigus saab tabel olla endiselt kasutuses.
 - Vanemates andmebaasisüsteemides ei saanud andmebaasiobjekti muuta, kui keegi seda kasutab.
 - Andmebaasi kasutavate rakenduste töö tuli muudatustesse viimiseks katkestada.

14.12.2017 Teema 1 204

Andmebaasid II 2017 © Erki Eessaar

Näiteid evolutsiooni toetusest (Oracle) (2)

- Väljaande-põhine (*edition-based*) ümber-definierimine (alates Oracle 11g Release 2) võimaldab ühes skeemis hoida ja hallata erinevaid versioone:
 - PL/SQL objektidest (paketid, protseduurid, funktsionid, triggerid),
 - sünontüümist,
 - tüüpidest,
 - vaadetest
- ning selle abil kiirendada andmebaasi ja rakenduste täiustatud versioonide kasutuselevõttu.
 - Andmebaasisüsteemile öelda, et vaikimisi väljaandeks on uus loodud väljaanne ning andmebaasi kasutajad hakkavad seda uut väljaannet kasutama.

14.12.2017 Teema 1 205

Andmebaasid II 2017 © Erki Eessaar

Näiteid evolutsiooni toetusest (Oracle) (3)

- Alates Oracle Database 11g Release 1 ei põhjusta andmekirjelduse operatsioonid nagu tabelisse veeru lisamine enam tabeli lukustamist.
- Alates Oracle Database 11g Release 1 saab *Enterprise Editionis* luua käsuga CREATE INDEX ONLINE uue indeksi ilma, et see blokeeriks samal ajal tabelis andmete lugemise ja muutmise.

14.12.2017 Teema 1 206

Andmebaasid II 2017 © Erki Eessaar

Näiteid evolutsiooni toetusest (Oracle) (4)

- Oracle võimaldab definierida *sünontüüme*.
 - Sünontüm on alternatiivne nimi tabelile, vaatele, arvujada generaatorile, operaatorile, protseduurile, funktsioonile, paketile, materialiseeritud vaatele, kasutaja-definieritud objektitüübile või teisele sünontüümile.
- Neid saab kasutada andmebaasi refaktoreerimise käigus, kui soovitakse ühtlustada andmebaasi-objektide nimesid.

14.12.2017 Teema 1 207

Andmebaasid II 2017 © Erki Eessaar

Minimeeri koodi maksimeeri andmeid

- Kõige paremini programmeerib see, kes programmeerib **vähe**.
- Leia tarkvaras osad, mis võivad ajas muutuda/täieneda ning muuda need **andmetega juhitavaks**.
- Lihtsustab süsteemi haldamist, evolutsiooni.
- Annab kasutajatele suurema kontrolli süsteemi üle.
- <http://database-programmer.blogspot.com/2008/05/minimize-code-maximize-data.html>

14.12.2017 Teema 1 208

Andmebaasid II 2017 © Erki Eessaar

Minimeeri koodi maksimeeri andmeid – näide

- Sotsiaalkindlusameti infosüsteemi toetuste väljamaksete algoritmid.
 - Söltuvad poliitilisest tahtest.
 - Muudatusi võib olla vaja teha äkki ning kiiresti.
 - Kui algoritmid saaks teha andmetega juhitavaks, siis ei peaks arendajad muutma tarkvara (paranda lähtekoodi, testi, installeeri), vaid süsteemi lõppkasutajad (ametnikud) peaksid muutma andmeid.

14.12.2017 Teema 1 209

Andmebaasid II 2017 © Erki Eessaar

Minimeeri koodi maksimeeri andmeid – näide

- Tabelid *Kaubaseisundi_liik* ja *Kaup*.
- Trigerid kontrollivad, et *kaubaga* seotud seisundi muutus vastab seisundiDiagrammis lubatule.

```

graph TD
    Active((Active)) -- "Kaubasseisundi_liik ja Kaup" --> Kubutuvad((Kubutuvad))
    Kubutuvad -- "Kaubaseisundi_liik ja Kaup" --> Active
    Kubutuvad -- "Kaubamüük otsustatakse väljaku kauba seisundit" --> Mitteniedre((Mitteniedre))
    Mitteniedre -- "Kaubamüük otsustatakse väljaku kauba seisundit" --> Kubutuvad
    Kubutuvad -- "Kaubamüük otsustatakse väljaku kauba seisundit" --> Active
    Kubutuvad -- "Kaubamüük otsustatakse väljaku kauba seisundit" --> Kubutuvad
  
```

14.12.2017 Teema 1 210

Andmebaasid II 2017 © Erki Eessaar

Traditsiooniline lähenemine – käitumise mutumiseks muuda koodi

```
--Triger, mis kontrollib, et kaup luuakse õiges seisundis
CREATE OR REPLACE TRIGGER kaup_i BEFORE INSERT ON Kaup
FOR EACH ROW
WHEN (NOT (NEW.kauba_seisundi_liik_kood=1))
BEGIN
    RAISE_APPLICATION_ERROR(-20002, 'Kaup luuakse ebakorrektses seisundis!');
END;
/
--Triger, mis kontrollib seisundimuudustuste korrektsust
CREATE OR REPLACE TRIGGER kaup_u BEFORE UPDATE OF kauba_seisundi_liik_kood ON Kaup
FOR EACH ROW
WHEN (NOT ((OLD.kauba_seisundi_liik_kood=NEW.kauba_seisundi_liik_kood) OR
(OLD.kauba_seisundi_liik_kood=1 AND NEW.kauba_seisundi_liik_kood=2) OR
(OLD.kauba_seisundi_liik_kood=2 AND NEW.kauba_seisundi_liik_kood=1) OR
(OLD.kauba_seisundi_liik_kood IN (1,2) AND NEW.kauba_seisundi_liik_kood=3)))
BEGIN
    RAISE_APPLICATION_ERROR(-20003, 'Kauba ebakorrektna seisundimuudatus!');
END;
/
Näide realiseeritud Oracle 12c Enterprise Edition Rel. 1 süsteemis.
```

14.12.2017 Teema 1 211

Andmebaasid II 2017 © Erki Eessaar

Andmetega juhitav lahendus – lubatud seisundimuudatuste tabel

```
/*Reeglite tabel, milles kirjeldatakse lubatud seisundimuudatused*/
CREATE TABLE Kauba_lubatud_seisundimuudatus (
algseisund NUMBER() NOT NULL,
loppseisund NUMBER() NOT NULL,
CONSTRAINT unik_kauba_seisundimuudatus PRIMARY KEY (algseisund, loppseisund));
INSERT INTO Kauba_lubatud_seisundimuudatus(algseisund, loppseisund)
VALUES (1, 2);
INSERT INTO Kauba_lubatud_seisundimuudatus(algseisund, loppseisund)
VALUES (2, 1);
INSERT INTO Kauba_lubatud_seisundimuudatus(algseisund, loppseisund)
VALUES (1, 3);
INSERT INTO Kauba_lubatud_seisundimuudatus(algseisund, loppseisund)
VALUES (3, 1);
INSERT INTO Kauba_lubatud_seisundimuudatus(algseisund, loppseisund)
VALUES (2, 3);
COMMIT;
```

Näide reeglite andmetena esitamisest!
Neid saab muuta ärikasutaja ilma arendajate sekkumiseta.

14.12.2017 Teema 1 212

Andmebaasid II 2017 © Erki Eessaar

Andmetega juhitav lahendus – triger

```
--Triger, mis kontrollib seisundimuudustuste korrektsust
CREATE OR REPLACE TRIGGER kaup_iu
BEFORE INSERT OR UPDATE OF kauba_seisundi_liik_kood ON Kaup
FOR EACH ROW
WHEN (NOT (OLD.kauba_seisundi_liik_kood=NEW.kauba_seisundi_liik_kood))
DECLARE
    m_tulemus NUMBER(10);
BEGIN
    SELECT Count(*) arv INTO m_tulemus
    FROM Kauba_lubatud_seisundimuudatus
    WHERE algseisund=coalesce(:OLD.kauba_seisundi_liik_kood,0)
        AND loppseisund=coalesce(:NEW.kauba_seisundi_liik_kood,0);
    /*Coalesce funktsiooni abil tehakse önnekodne väljund, kui ei ole midagi*/
    IF m_tulemus=0 THEN
        RAISE_APPLICATION_ERROR(-20003, 'Kauba ebakorrektna seisundimuudatus!');
    END IF;
END;
/
14.12.2017 Teema 1 213
```

Andmebaasid II 2017 © Erki Eessaar

Muutub lubatud seisundimuutuste hulk

```
/*Kaup koodiga 1 on seisundis koodiga 3*/
UPDATE Kaup SET kauba_seisundi_liik_kood=1 WHERE kauba_kood=1;
/*ORA-20003: Kauba ebakorrektna seisundimuudatus!
ORA-06512: at "C##TUD1.KAUP_IU", line 10
ORA-04088: error during execution of trigger 'C##TUD1.KAUP_IU'*/
INSERT INTO Kauba_lubatud_seisundimuudatus(algseisund, loppseisund)
VALUES (3,1);
COMMIT;

UPDATE Kaup SET kauba_seisundi_liik_kood=1 WHERE kauba_kood=1;
--OK
COMMIT;
```

14.12.2017 Teema 1 214

Andmebaasid II 2017 © Erki Eessaar

Kuidas luua uut modelleerimiskeelt või arendusmetoodikat?

- Võtta aluseks olemasolev, seada sellele piirangud.
 - Näiteid.
 - Võtta mõni raamistikuks olev metoodika (RUP, Scrum) ja teha valik selle poolt pakutavatest tehtitest, tegevustest, rollidest jne. Ise midagi uut ei lisa.
 - OO programmeerimises või andmebaasis andmetüüpide loomisel **pärimine kitsendamise teel (specialization by constraint)** (vt teema 10).

14.12.2017 Teema 1 215

Andmebaasid II 2017 © Erki Eessaar

Kuidas luua uut modelleerimiskeelt või arendusmetoodikat? (2)

- Võtta aluseks olemasolev, lisada elemente või muuta olemasolevate tähdendust.
 - Näiteid.
 - UML profiliide loomine.
 - Scrumi laiendamine ekstreemprogrammeerimise vötetega.
 - OO programmeerimises või andmebaasis andmetüüpide loomisel **pärimine laiendamise teel (specialization by extension)** (vt teema 10).

14.12.2017 Teema 1 216

