

# Tarkvaratehnika: tarkvaraprojekti juhtimine (loeng I)

Asko Seeba

# Teemad

- Loeng I – sissejuhatus
- Loeng II – tarkvaraprojekti juhtimise kontseptsioonid
- Loeng III – tarkvaraprojekti juhtimise töövoog
- Loeng IV – tarkvaraarendusprotsessi rakendamine

# Juhtimise funktsioonid

- Planeerimine
- Organiseerimine
- Inimeste juhtimine
- Kontroll

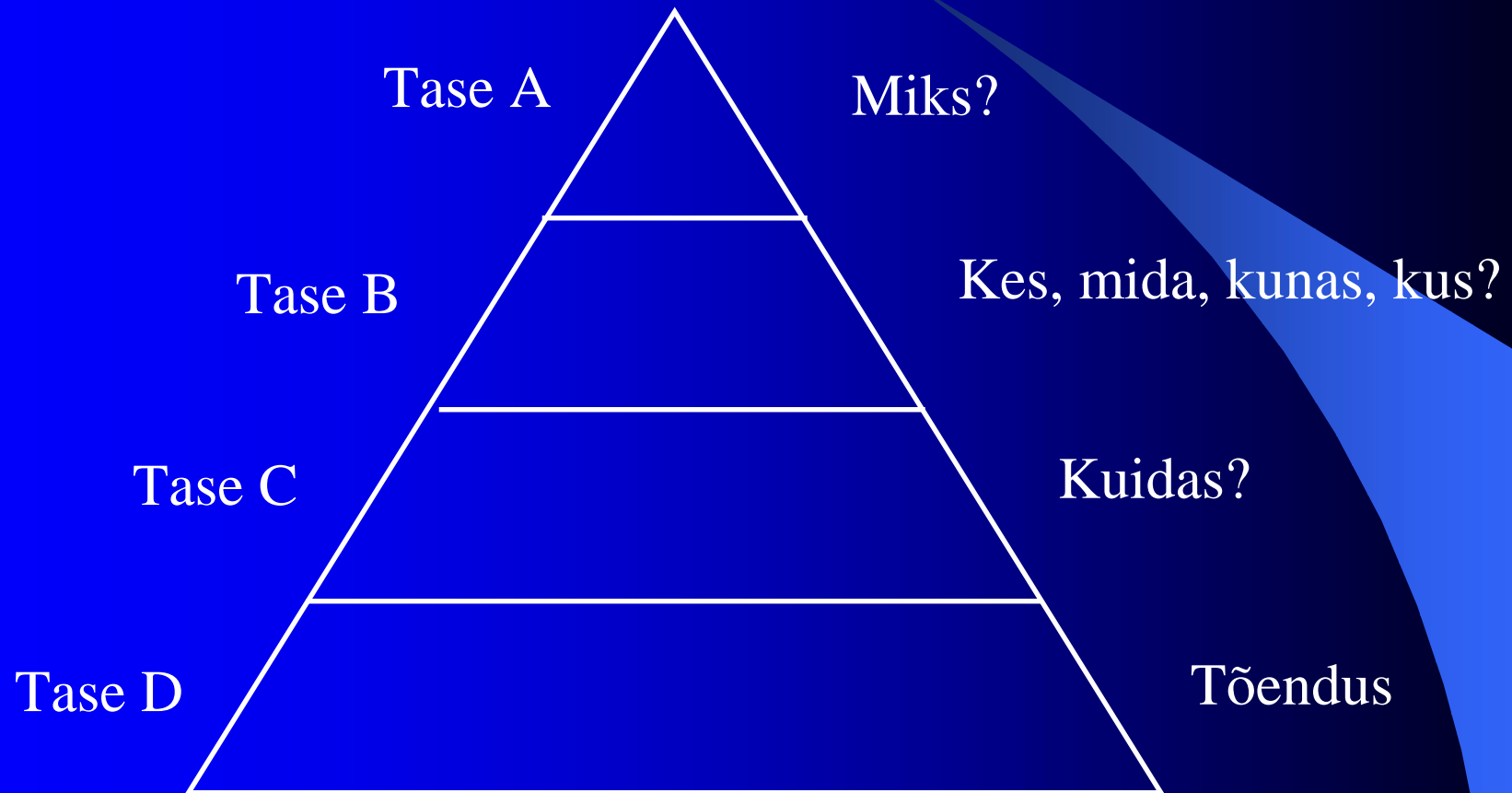
# Uuemad juhtimisteooriad – täielik kvaliteedijuhtimine (TQM)

- Organisatsiooni juhtimise meetod
- Kvaliteedi kriteeriumiks kliendi nõudmine
- ISO 9001:2000

# ISO 9001:2000 (I)

- Tase A – kvaliteedikäsiraamat
- Tase B – kvaliteediprotseduurid
- Tase C – töödokumendid
- Tase D – kvaliteediandmestik

# ISO 9001:2000 (II)



# Uuemad juhtimisteooriad – piirangute teooria (TOC)

- Eesmärk
  - Puhaskasum
  - Investeeringu tasuvus
  - Rahavoog
- Piirang e. pudelikael
  - Füüsiline
  - Turg
  - Ettevõtte poliitika

# Projekt

- Mingi ajutine, unikaalne püüdlus teatava eesmärgi saavutamiseks
  - Uue toote või teenuse väljatöötamine
  - Uue sõiduki projekteerimine
  - Poliitilise kampaania läbiviimine
  - Silla ehitamine
  - Marsile kulguri saatmine
  - Internetti e-kauplemise teenuse ülespanemine



# Projekti juhtimine

- Projektipõhine organisatsioon
  - Sobib seal, kus turg ei lase tekkida monotoonsetel/rutiinsetel protsessidel.
  - Turu rahuldamine käib projektide kaudu
  - Kliendi muutuvate ja keerukate probleemide lahendamiseks paindlikum
  - Vähem inimesi teeb rohkem asju

# Projekti juhtimise probleemid

- Projekti juhtimine – mingile projektile esitatud nõuete täitmiseks vajalike teadmiste, oskuste, vahendite ja tehnika rakendamine vajalike tegevuste kaudu
- Projekti meeskond peab toime tulema
  - vastuoluliste tingimustega: projekti ulatus, aeg, hind, riskid ja kvaliteet
  - erinevate vajaduste ja ootustega osanike rahuldamisega
  - täitma tuvastatud nõudmised

# Tarkvara väljatöötamise probleemide sümptomid

- Kasutaja või talitluse vajadused on rahuldamata
- Nõudmised logisevad
- Moodulid ei integreeru
- Raskesti hallatav
- Puuduste hiline avastamine
- Kehv kvaliteet või puudulik lõppkasutaja kogemus
- Kehv töövõime koormuse all
- Koordineerimata meeskonnatöö
- Valmimisjärgsed redaktsioonid

# Sümptomite algpõhjused

## Sümptomid

Rahuldamata vajadused  
Logisevad nõudmised  
**Moodulid ei sobi**  
Raskesti hallatav  
Hiline avastamine  
Kehv kvaliteet  
Meeskonna vastuolud  
Järelredaktsioonid

## Algpõhjused

Ebapiisavad nõudmised  
**Ebaseelge suhtlemine**  
Habras arhitektuur  
Liigne keerukus  
**Avastamata vastuolud**  
Puudulik testimine  
Subjektiivne hindamine  
Koskmudeli kasutamine  
Juhitamatud muutused  
Kehv automatiseeritus

## Parim tööpraktika

Arenda iteratiivselt  
Halda nõudeid  
Kasuta komponentarhitektuuri  
**Modelleeri visuaalselt (UML)**  
**Kontrolli pidevalt kvaliteeti**  
Halda muudatusi

# Arenda iteratiivselt

## Arenda iteratiivselt

Halda nõudeid

Kasuta komponentarhitektuuri

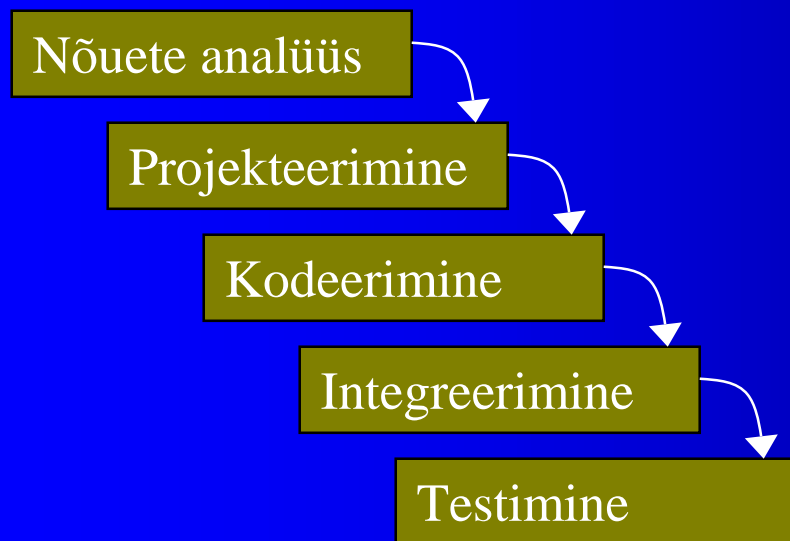
Modelleeri visuaalselt (UML)

Kontrolli pidevalt kvaliteeti

Halda muudatusi

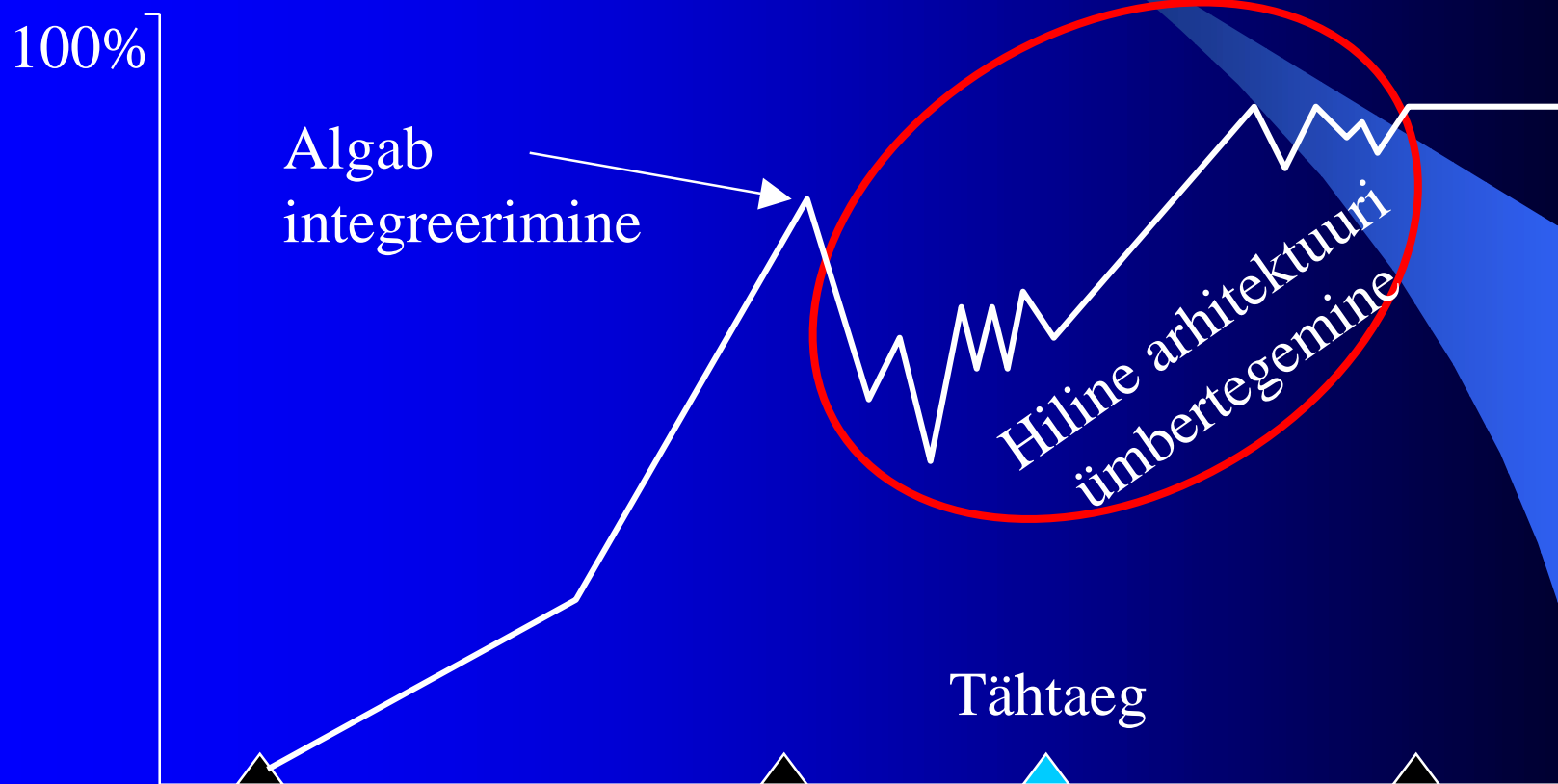
# Klassikaline koskprotsess

## Koskprotsess

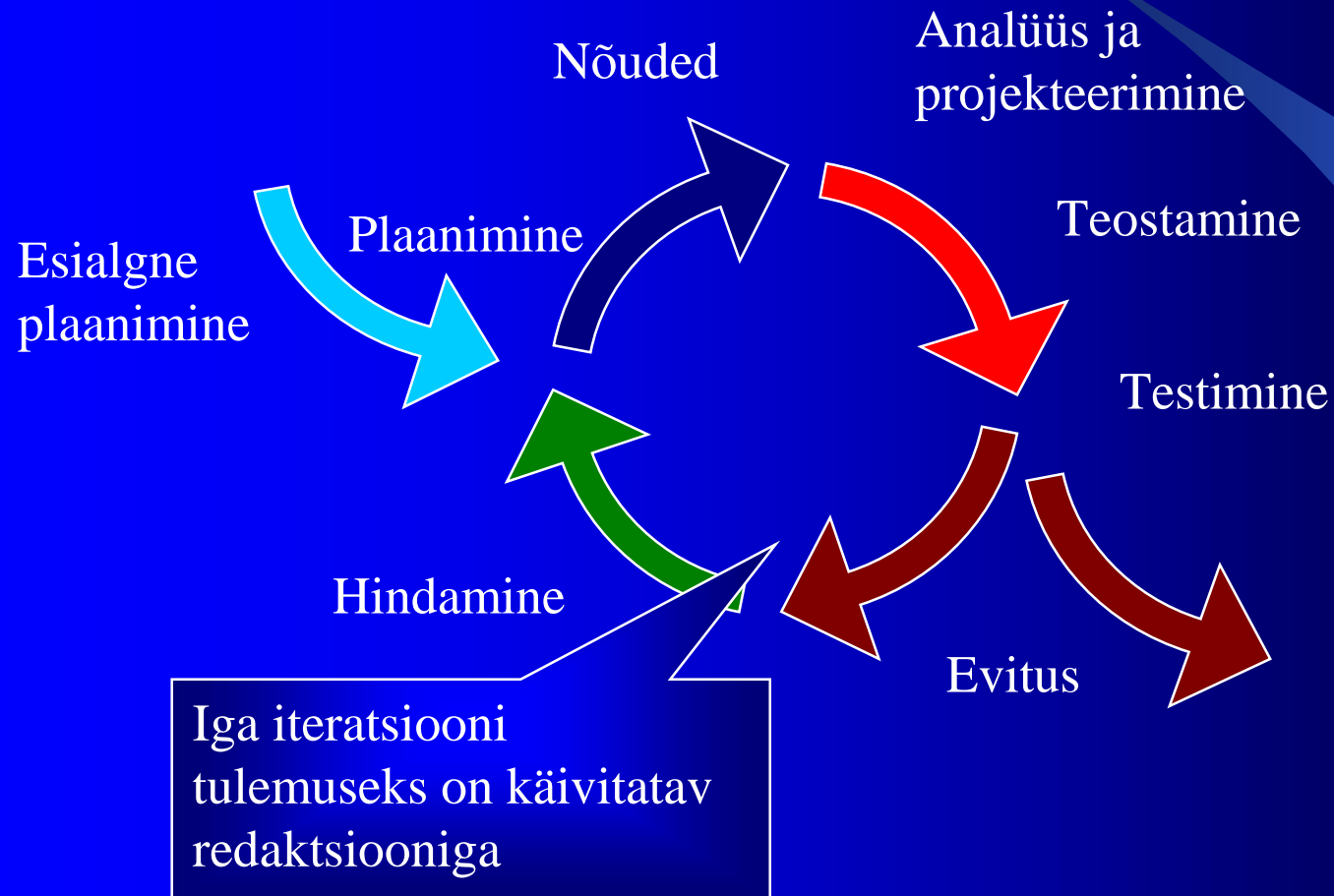


- Kriitiliste riskide lahendamine viibib
- Hindab progressi töö tulemuste pealt, mis on edasiste ajahinnangute andmiseks kehvad
- Integreerimine ja testimine viibib
- Välistab varajase evituse
- Sageli lõppeb suurte plaanimatute iteratsioonidega

# Projekti tavaline areng

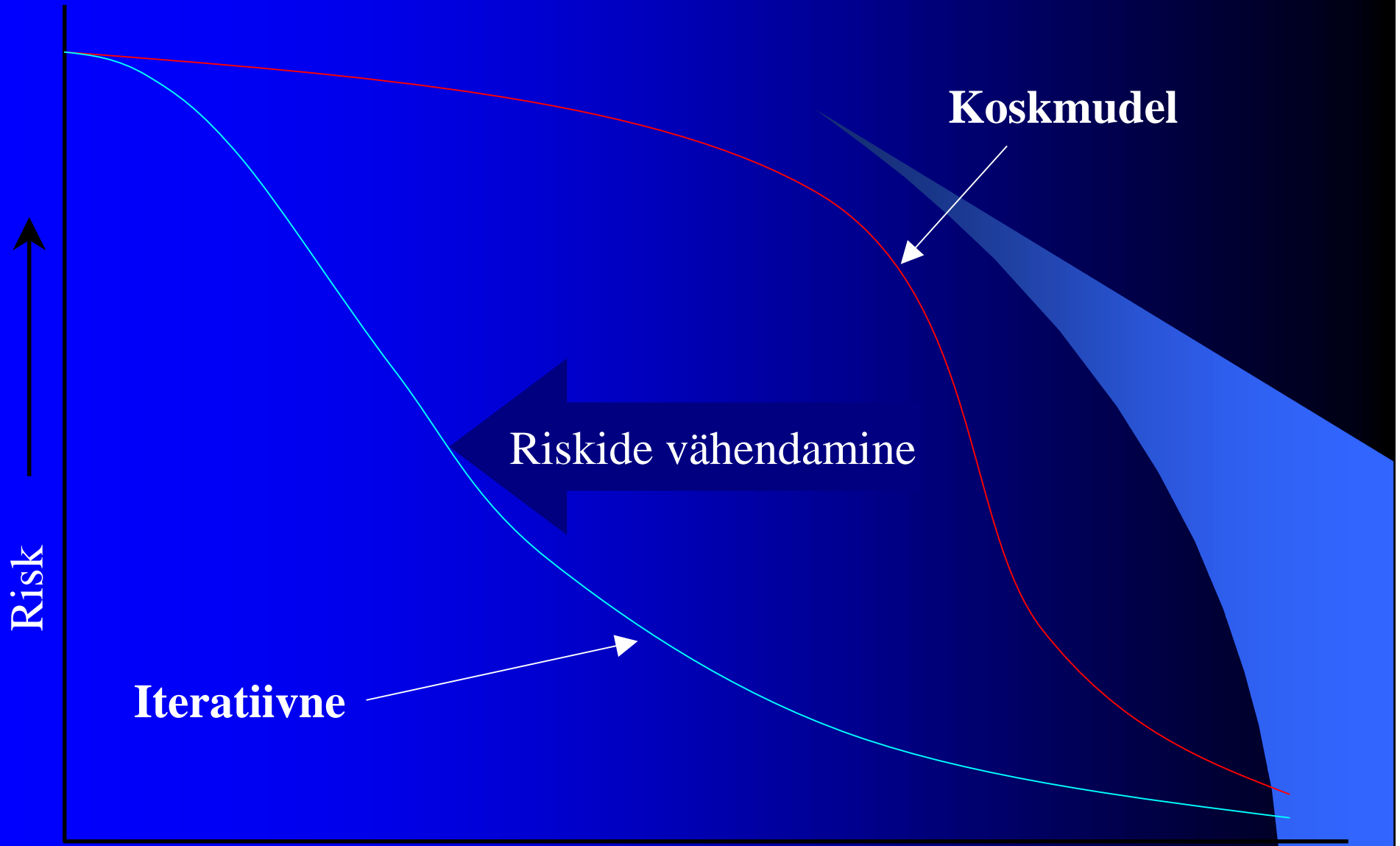


# Iteratiivne väljatöötamine





# Riskid vähenevad rütem



# Halda nõudeid

Arenda iteratiivselt

Halda nõudeid

Kasuta komponentarhitektuuri

Modelleeri visuaalselt (UML)

Kontrolli pidevalt kvaliteeti

Halda muudatusi

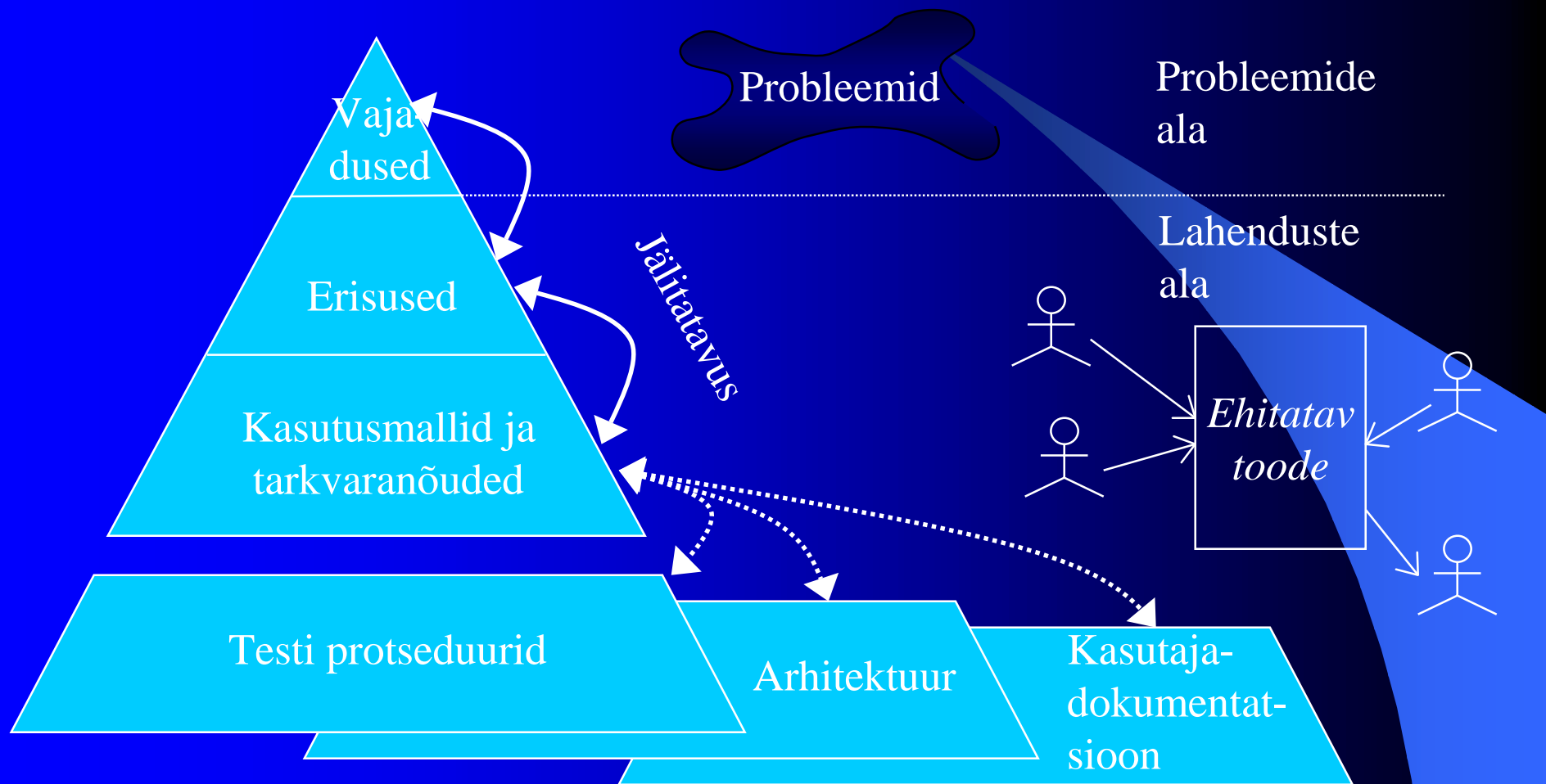
# Nõuetealdus

- Tagab, et
  - lahendatakse õiget probleemi
  - ehitatakse õige süsteem
- kasutades süsteemi muutuvate nõuete süstematiseeritud
  - kogumist
  - organiseerimist
  - dokumenteerimist
  - ja haldamist.

# Nõuetealduse tegevused

- Probleemi analüüsimine
- Kasutaja vajadustest arusaamine
- Süsteemi defineerimine
- Süsteemi ulatuse haldamine
- Süsteemi definitsiooni täpsustamine
- Õige süsteemi ehitamine

# Tegevuste kaardistus



# Kasuta komponentarhitektuuri

Arenda iteratiivselt  
Halda nõudeid

## Kasuta komponentarhitektuuri

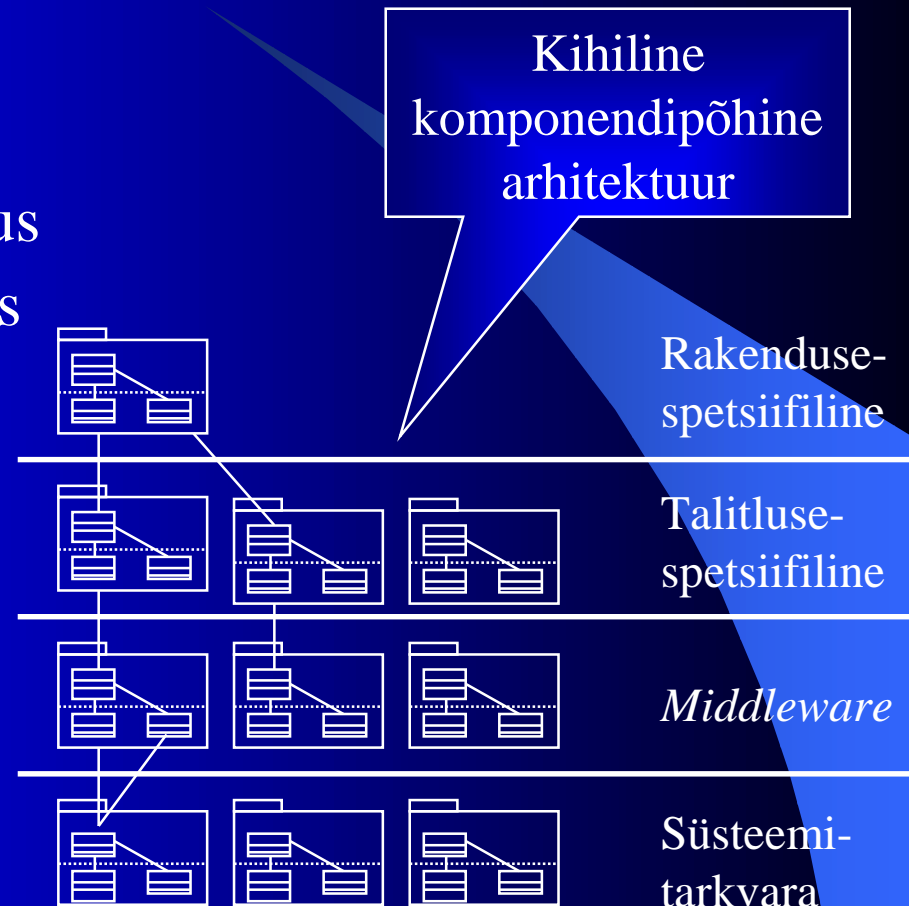
Modelleeri visuaalselt (UML)  
Kontrolli pidevalt kvaliteeti  
Halda muudatusi

# Paindlik, komponendipõhine arhitektuur

- Paindlik
  - Vastab jooksvatele ja tulevastele nõuetele
  - Parandab laiendatavust
  - Võimaldab korduvkasutust
  - Kapseldab süsteemi sõltuvused
- Komponendipõhine arhitektuur
  - Korduvkasutab või kohandab komponente
  - Kasutab kommertsiaalselt saadaolevaid komponente
  - Arendab olemasolevat tarkvara inkrementaalselt

# Komponendipõhise arhitektuuri otstarve

- Korduvkasutuse alus
  - Komponendi korduvkasutus
  - Arhitektuuri korduvkasutus
- Projektijuhtimise alus
  - Plaanimine
  - Personal
  - Üleandmine
- Intellektuaalne juhtimine
  - keerukuse haldamine
  - kooskõlalisuse hoidmine





# Modelleeri visuaalselt (UML)

Arenda iteratiivselt

Halda nõudeid

Kasuta komponentarhitektuuri

Modelleeri visuaalselt (UML)

Kontrolli pidevalt kvaliteeti

Halda muudatusi

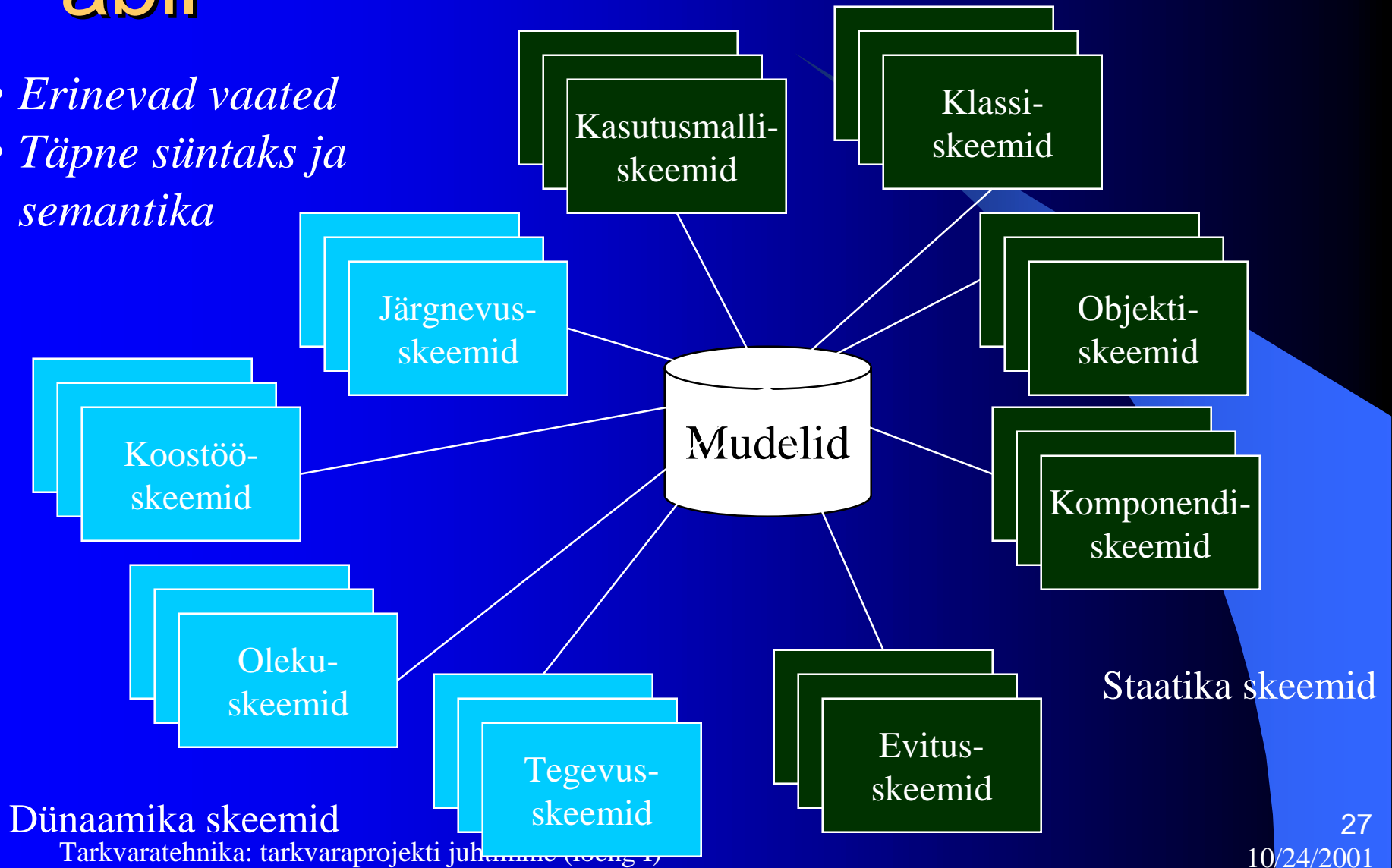
# Miks modelleerida visuaalselt?

- Aitab

- kujutada süsteemi struktuuri ja käitumist
- näidata, kuidas süsteemi elemendid kokku sobivad
- hoida projekteerimine ja teostamine kooskõlas
- vastavalt vajadusele varjata või tõsta esile detaile
- suhelda üheseltmõistetavalt
  - UML (*Unified Modeling Language* – unifitseeritud modelleerimise keel): üks keel kõigi osalejate jaoks

# Visuaalne modelleerimine UML abil

- *Erinevad vaated*
- *Täpne süntaks ja semantika*

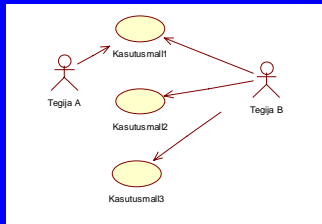


Dünaamika skeemid

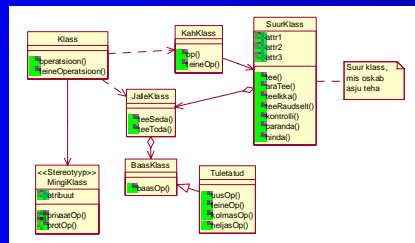
Tarkvaratehnika: tarkvaraprojekti juhtimine (loeng 1)

# Skeemid visuaalsele modelleerimisel

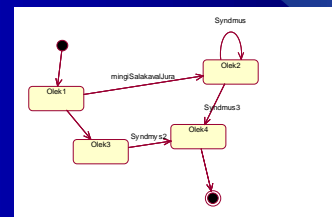
Kasutusmalli-  
skeem



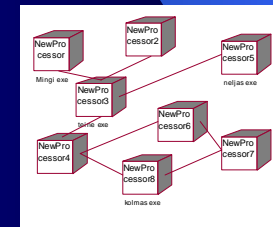
Klassiskeem



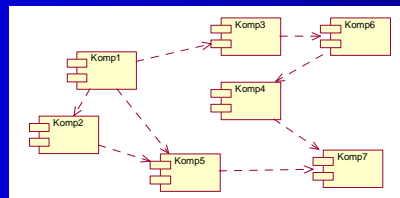
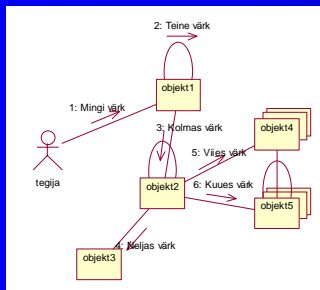
Olekuskeem



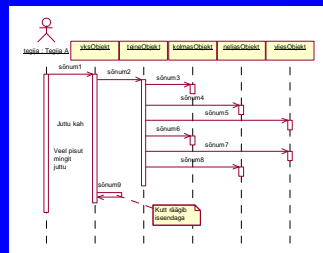
Evituskeem



Koostöö-  
skeem



Komponendiskeem



Järgneviskeem

1. Järgneviskeem. tarkvaraprojekti juhtimine (loeng I)

Pärikonstrueerimine ja pöördkonstrueerimine



Ehitatav süsteem

# Kontrolli pidevalt kvaliteeti

Arenda iteratiivselt

Halda nõudeid

Kasuta komponentarhitektuuri

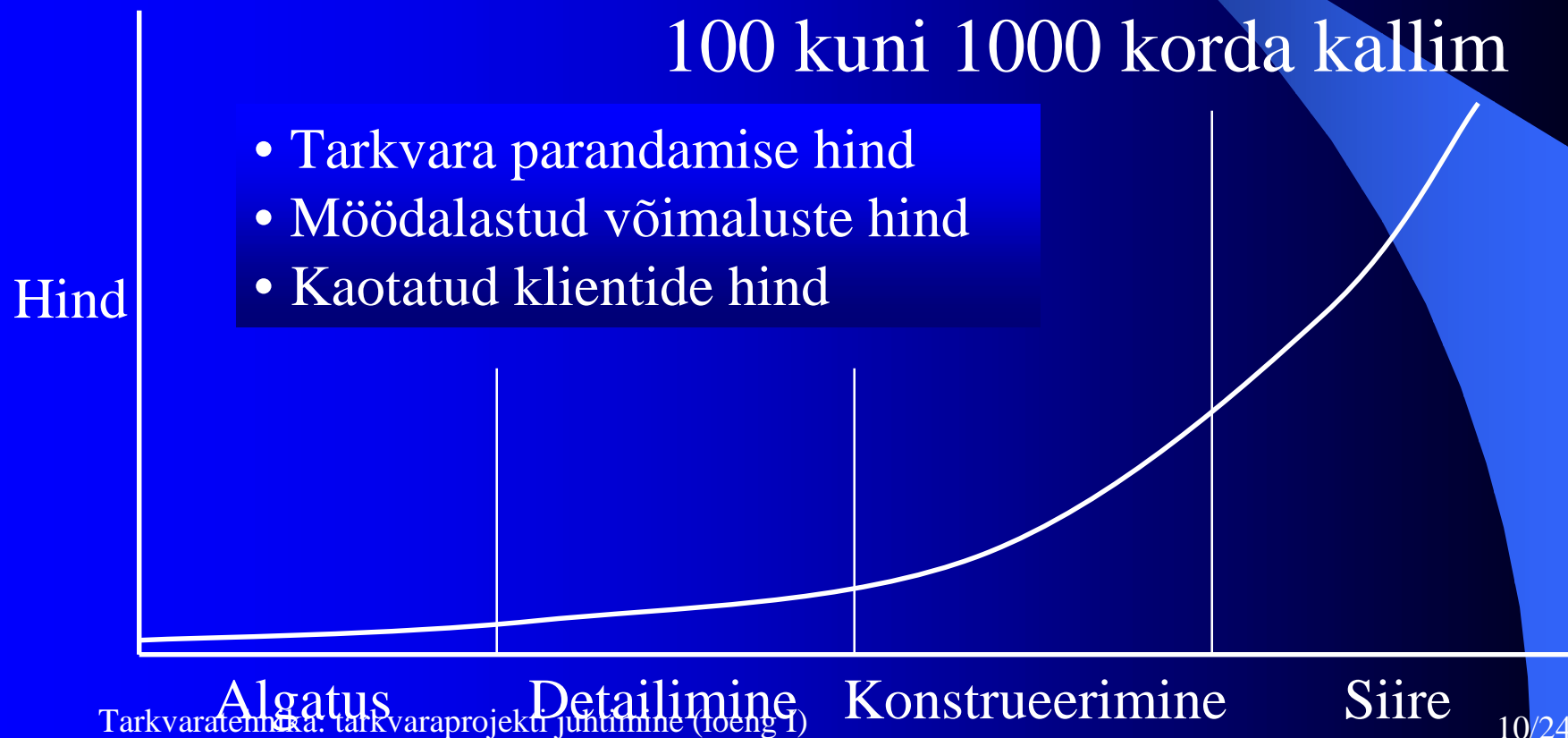
Modelleeri visuaalselt (UML)

Kontrolli pidevalt kvaliteeti

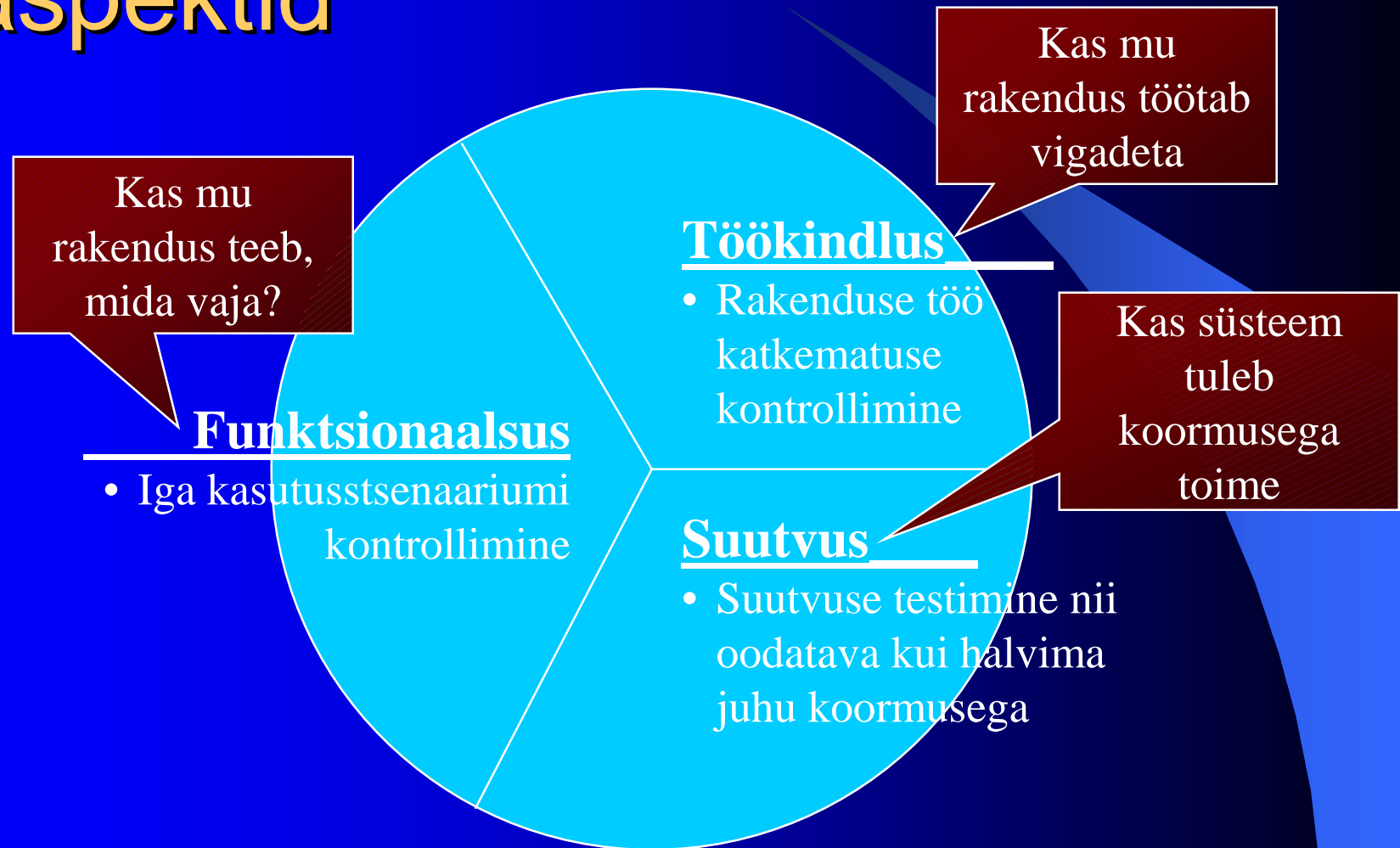
Halda muudatusi

# Tarkvara kvaliteedi pidev kontrollimine

Tarkvaraprobleemide leidmine ja parandamine on peale evitust 100 kuni 1000 korda kallim

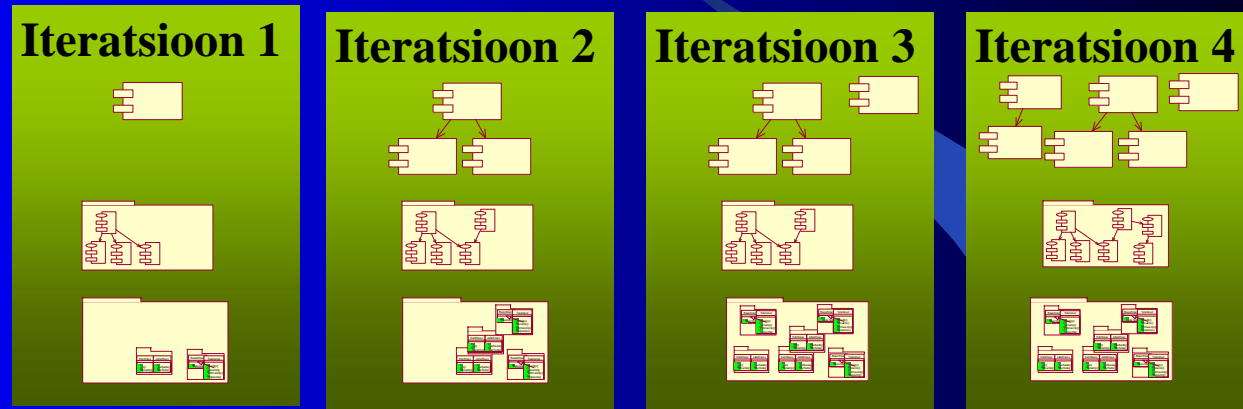


# Tarkvarakvaliteedi erinevad aspektid

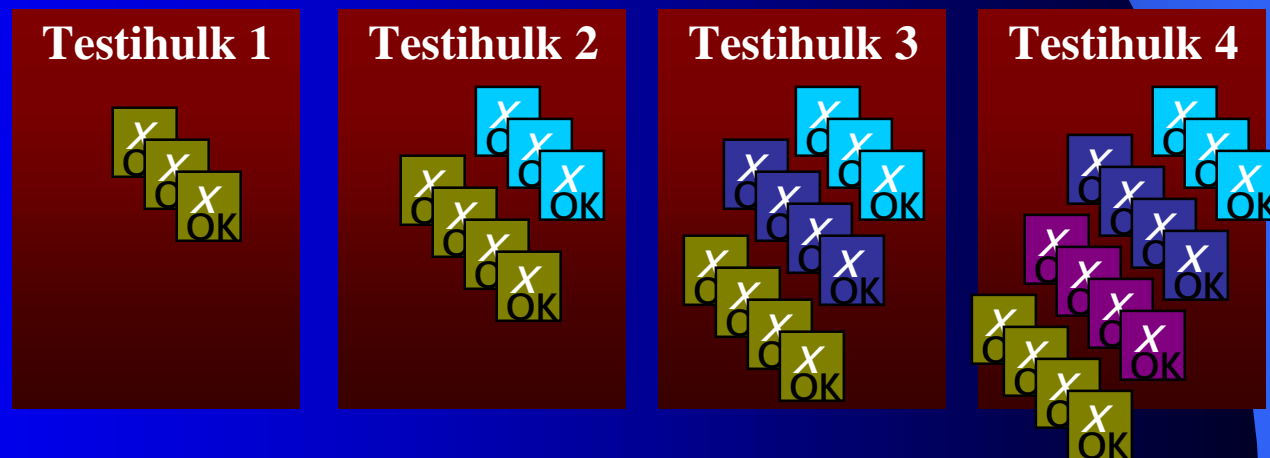


# Testi igas iteratsioonis

UML mudel  
ja teostus



Testid





# Halda muudatusi

Arenda iteratiivselt

Halda nõudeid

Kasuta komponentarhitektuuri

Modelleeri visuaalselt (UML)

Kontrolli pidevalt kvaliteeti

**Halda muudatusi**

# Mida on vaja juhtida?

- Muudatusi, et itereerimine oleks võimalik
- Privaatne töökoopia iga arendaja jaoks
- Automatiseeritud teostuse/integreerimise haldus
- Paralleelne arendus

Konfiguratsioonihaldus on rohkem, kui lihtsalt *check-in* ja *check-out*

# Konfiguratsioonihaldussüsteemi aspektid

- Muutmistaotluste (*Change Request* – CR) haldamine
- Konfiguratsiooni seis hindamine
- Konfiguratsioonihaldus (CM)
- Muudatuste jälitamine
- Versiooni jälgimine
- Tarkvaraline keskkond

# Unifitseeritud muutmisjuhtimine (UCM)

- Kirjeldab, kuidas juhtida muutusi
  - nõuetes
  - projekteerimismudelites
  - dokumentatsioonis
  - komponentides
  - testjuhtumites
  - lähtekoodis
- Unifitseerib projekti plaanimise ning projekti arengu ja muutuvate tehiste jälgimise

# Parima tööpraktika osad võimendavad teineteist

## Parim tööpraktika

Arenda iteratiivselt

Halda nõudeid

Kasuta komponentarhitektuuri

Modelleeri visuaalselt (UML)

Kontrolli pidevalt kvaliteeti

Halda muudatusi

Tagab kasutajate kaasamise  
ja nõuete arenemise

Võimaldab varakult arhitektuuri  
otsuseid vastu võtta

Võimaldab projekteerimise/teostamise  
keerukust inkrementaalselt lahata

Kvaliteeti mõõdetakse varakult ja tihti

Arendusalus areneb inkrementaalselt

# Unifitseeritud tarkvaraarenduse protsess

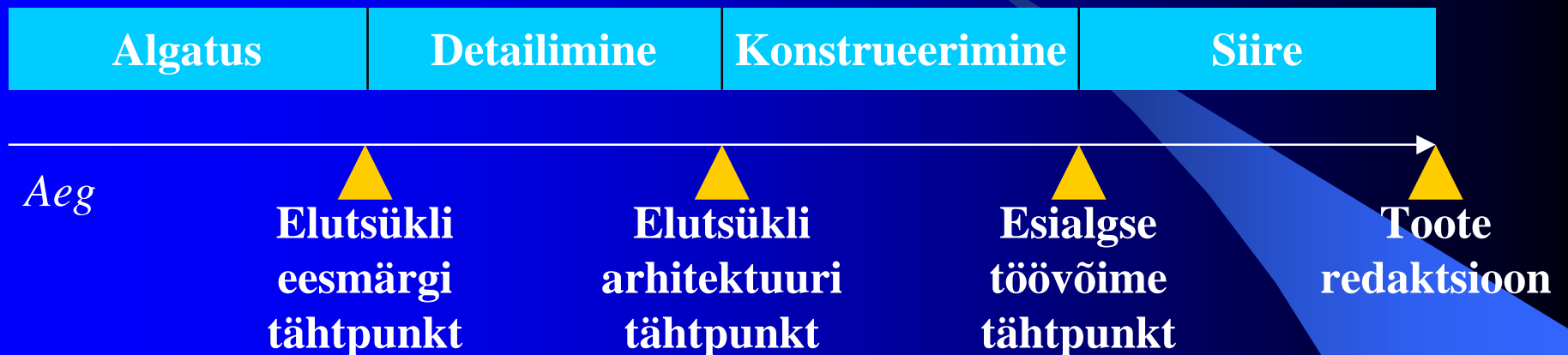
- Protsessi eesmärk
  - Koosneb juhenditest kvaliteetse tarkvara efektiivseks väljatöötamiseks
  - Vähendab riske ja suurendab ennustatavust
  - Soodustab üldise töökultuuri tekkimist
- Unifitseeritud tarkvaraarenduse protsess
  - Üldine objektorienteeritud tarkvara väljatöötamiseks
  - Rakendab parimat tööpraktikat

# Protsess üldises mõttes

Protsess defineerib, kes teeb **mida**, **millal** ja **kuidas**, et saavutada teatavat eesmärki.



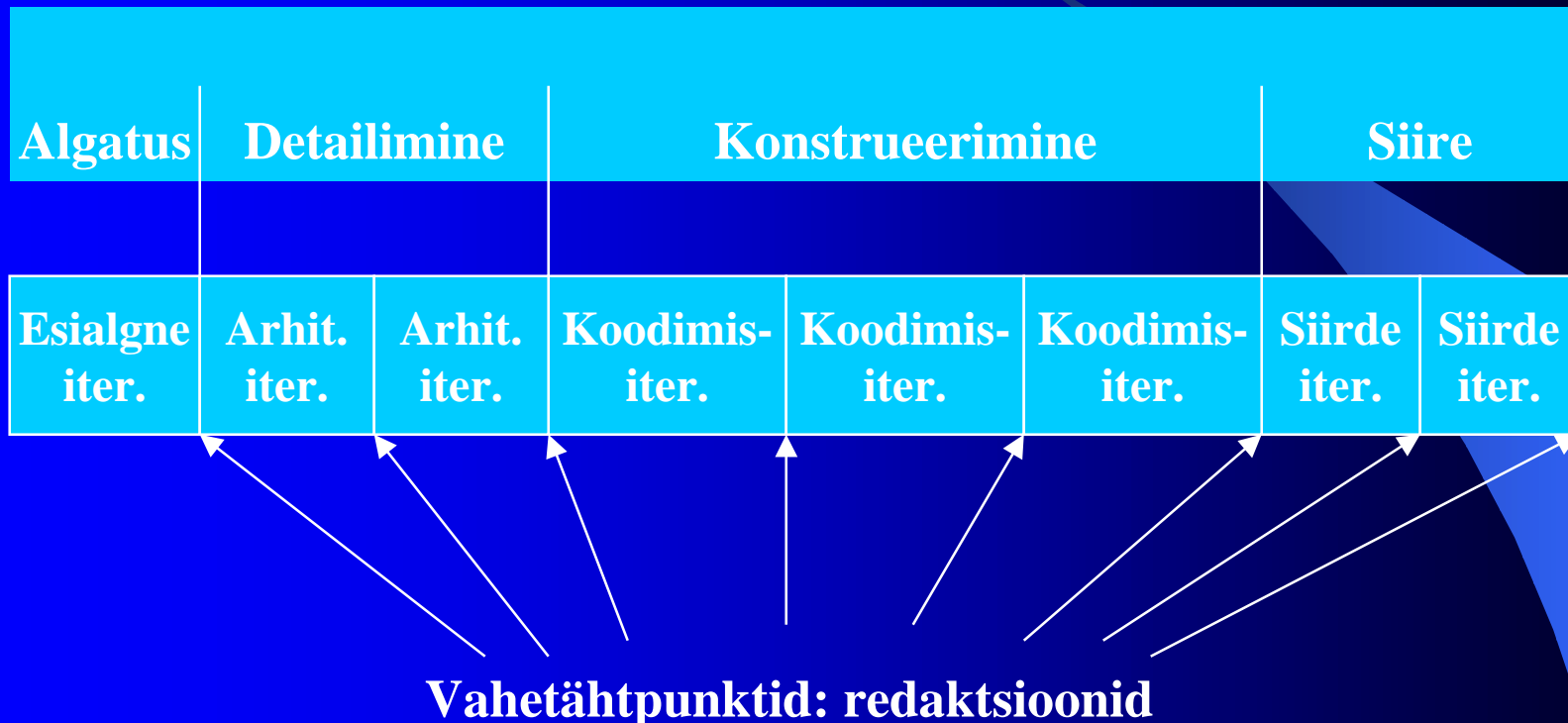
# UPi struktuur – faasid ja peamised tähtpunktid



- UPis on neli faasi:
  - **Algatus** – defineerib projekti ulatuse
  - **Detailimine** – planeerib projekti, spetsifitseerib erisusi, loob arhitektuuri arendusaluse
  - **Konstrueerimine** – valmistab toote
  - **Siire** – kannab toote lõppkasutajateni



# Iteratsioonid ja faasid



**Iteratsioon** on kinnitatud plaani ja hindamiskriteeriumite järgi toimiv eristatav tegevuste jada, mille tulemuseks on **demonstreeritav redaktsioon**.

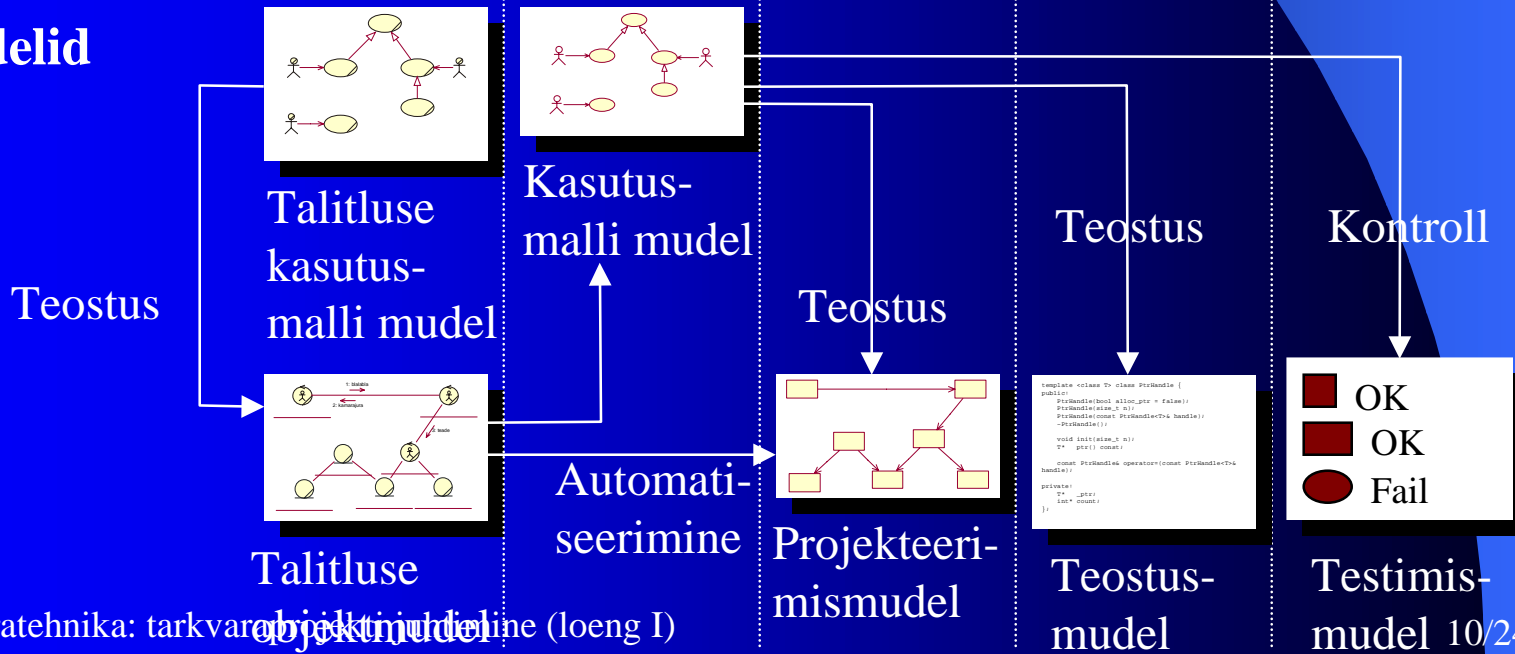
Tarkvaratehnika: tarkvaraprojekti juhtimine (loeng I)

# Töövood

**Protsessi põhivood**



**Mudelid**



# Kõik koos

