Tarkvara elutsükli mudelid

Tarkvaraarenduse üldised protsessimudelid.
Koskmudel, evolutsiooniline mudel, spiraalmudel, inkrementaalne mudel, ...

Elutsükli või protsessi mudel

Tarkvara loomise protsesside kirjeldamiseks kasutatakse erinevaid mõisteid:

- Tarkvara(arenduse) protsessi mudelid *(software process models)*.
- Tarkvara elutsüklimudelid (*software life cycle models)*

Protsessi ja elutsükli puhul on jutt tavaliselt samadest asjadest.

Tarkvaraprotsess

I. Sommerville'i *“Software Engineering”* ja R. Pressmann'i *“SE: A Practitioner's Approach”* järgi (sh joonised)

- **Tarkvaraprotsess** on omavahel seotud tegevused, et spetsifitseerida, kavandada, realiseerida ja testida tarkvarasüsteem.
- **Tarkvaraprotsess** peab tagama ratsionaalse ja tähtaega mahtuva arenduse.

Tarkvaraprotsessi tegevused

Tarkvaraprotsessi üldised tegevused olid:

- **tarkvara spetsifitseerimine** – nõuete väljaselgitamine, tarkvara tegevuste kirjeldame;
- **tarkvara realiseerimine** – teeme tarkvara valmis;
- **tarkvara valideerimine** – vaatame, kas tarkvara ikka teeb seda, mida tellija tahtis;
- **tarkvara edasiarendamine** – muudame tarkvara, et ta vastaks tellija muutunud nõuetele.

Taustategevused

Kogu protsessi vältel tegeletakse lisaks mitmete tegevustega, mis on kaudselt tarkvara loomisega seotud. Näiteks:

- Tarkvaraprojekti jälgimine ja juhtimine
- Riskide haldus
- Kvaliteedi kindustamine
- Tehnilised läbivaatused
- Tarkvara konfiguratsioonihaldus
- Korduvkasutuse haldus
- jne

Protsessimudel

Mudel on tegelikkuse lihtsustatud kirjeldus, abstraktsioon ehk üldistus kõige olulisemast.

Tarkvaraprotsessi mudel on tarkvaraprotsessi lihtsustatud kirjeldus, mis iseloomustab mingit vaatenurka protsessile, tuues sealjuures välja selle, mis antud protsessi juures oluline ja eriline on.

Protsessimudelid

Protsessimudelid loodi selleks, et tuua kord tarkvara arendusega seotud kaosesse.
Iga **mudel kirjeldab** protsessi kulgemist ja seda, kuidas protsessi elemendid on üksteisega seotud.
Iga mudel rõhutab protsessis erinevaid tegevusi ning esitab tegevused erinevalt järjestatuna.

Protsessimudelid

On välja toodud erinevaid.
Näiteks R. Pressmanni järgi on normatiivsed mudelid:

- Koskmudel e lineaarne mudel
- V-mudel (koskmudeli alaliik)
- Inkrementaalne mudel
- Evolutsioonilised mudelid: prototüüpimine ja spiraalmudel
- Komponentidel tuginev arendus (nn spetsialiseeritud mudel)

I. Sommerville lisab siia:

- Korduvkasutusele orienteeritud arendusmudel

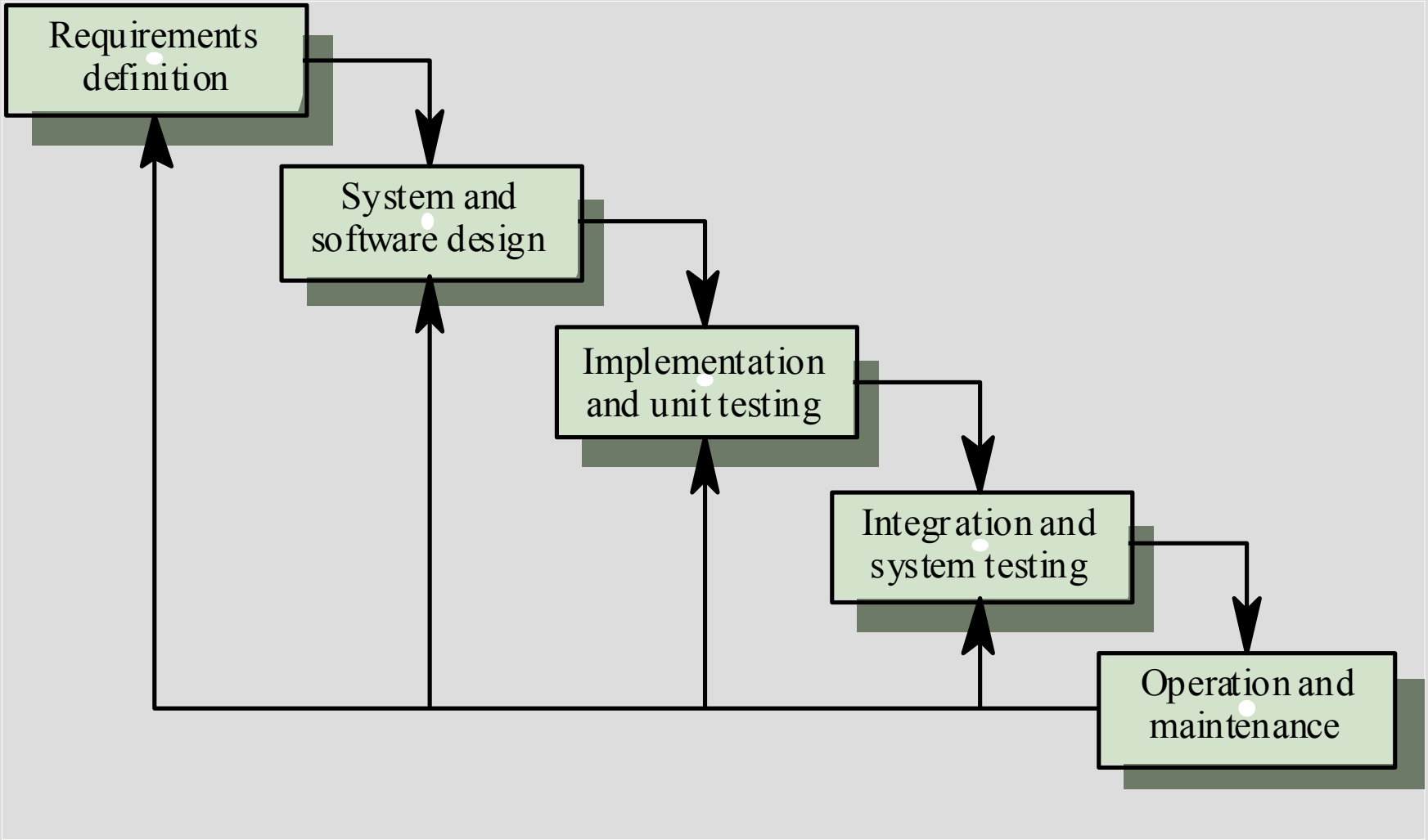
Koskmudel

- Ka klassikaline ehk jadamudel
  (Waterfall model, classic life cycle)
- Kirjeldas Winston W. Royce 1970

http://www.cs.umd.edu/class/spring2003/cmsc838p/Process/waterfall.pdf

- Jadamisi kulgevat protsessi kirjeldav mudel, tugineb tavalises tootearenduses kasutusel olnud protsessimudelile (inseneriteadus). Ühe etapi töö tehakse korraga ära ja liigutakse uude etappi.
- Royce'l olid lisatud tagasiside tsüklid, kuid praktikas pigem rangelt lineaarne.

Koskmudel.Skeem



Koskmudel.Etapid (1)

- Nõuete analüüs ja kirjeldamine:
  - süsteemianalüüs
  - tarkvaraanalüüs
- Süsteemi ja tarkvara kavandamine:
  - andmestruktuurid
  - tarkvara arhitektuur
  - liideste omadused
  - algoritmilised detailid
- Realisatsioon ja moodulite testimine:
  - luuakse programmiosad (funktsioonid, klassid)
  - luuakse moodulid (programmiosad ühendatakse)
  - kõike testitakse eraldi

Koskmudel.Etapid (2)

- Integratsioon ja süsteemi testimine:
  - programmid ja moodulid ühendatakse
  - testitakse süsteemi loogikat ja osade liidestamist
  - testitakse vastavust nõuetele (valideerimine)
- Kasutamine ja hooldus:
  - tarkvara kasutatakse
  - tarkvara muudetakse / täiendatakse, sest:
    - kasutajad avastavad vigu
    - ümbrus ja töökeskkond muutuvad
    - klient tahab lisafunktsionaalsuseid

Koskmudel.Probleemid

Puuduseks on ebamugavus muudatuste sisseviimine, kui protsess on käima läinud.

- Vajadus teha õigeid otsuseid varajases faasis
- Projekti paindumatu jaotus faasideks
- Klient näeb tulemust alles lõpus ja valesti mõistetud nõuded lõppevad katastroofiga
- Raske on arvestada kliendi uute soovidega

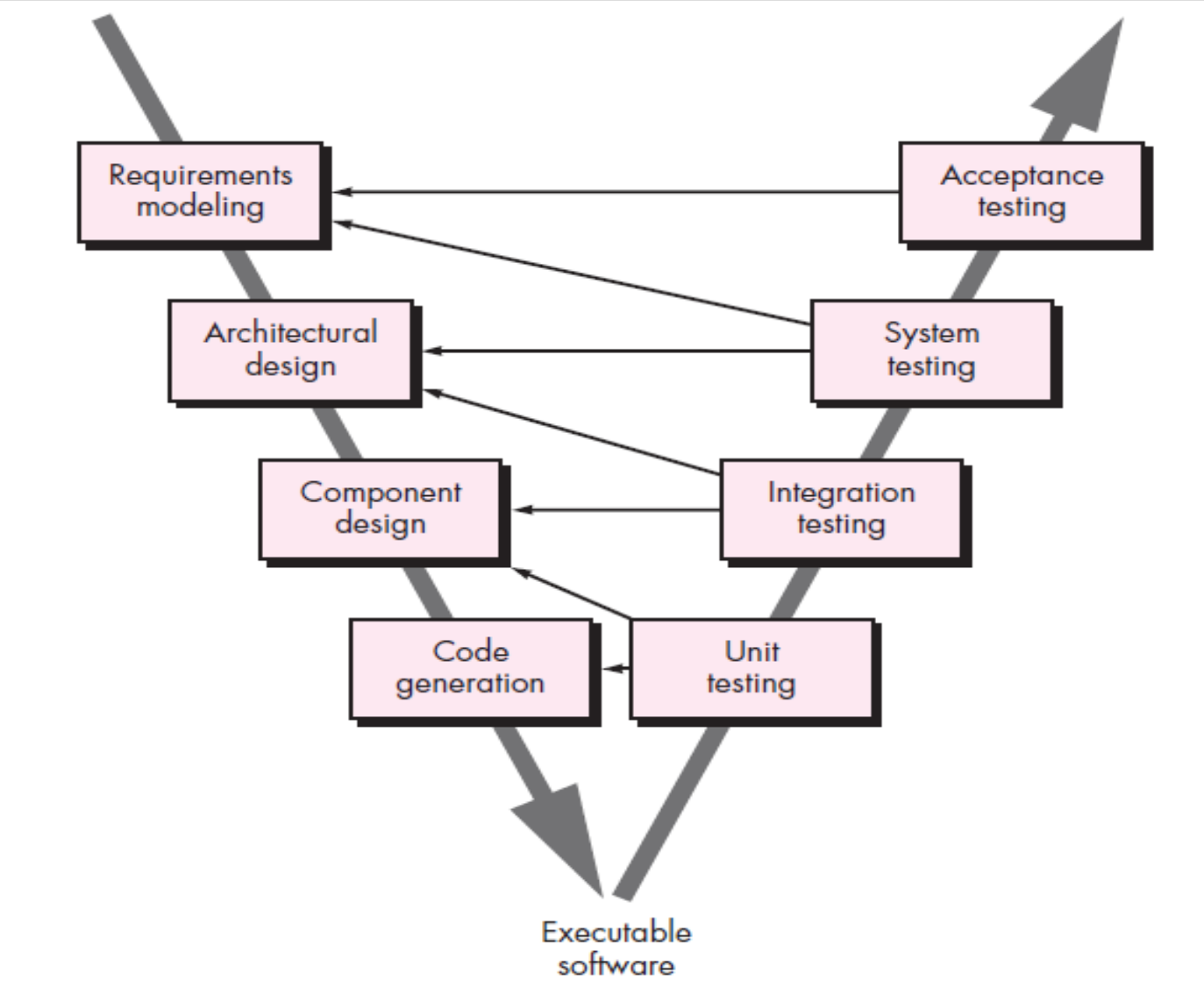
Mudel on **kasutatav,** kui

- probleemist saadakse aru ja nõudmised on selged
- tarkvara on suhteliselt väikesemahuline
- olemasolevasse süsteemi tehakse arusaadavaid muudatusi

V-mudel

- Olemuselt koskmudel, kus suhtlemine (nõuete kogumine), modelleerimine ja realiseerimine on seostatud kvaliteedi kindlustamisega
- Vasakult alla liikudes täpsustatakse nõuded kuni koodi genereerimiseni.
- Edasi paremalt poolt üles liikudes, kontrollib arendusmeeskond nii koodi kui mudelite kvaliteeti.
- Tegevused kulgevad jadamisi ning plussid ja miinused on samad koskmudeliga

V-mudel


Requirements modeling
Acceptance testing
Architectural design
System testing
Component design
Integration testing
Code generation
Unit testing
Executable software

Inkrementaalne arendus

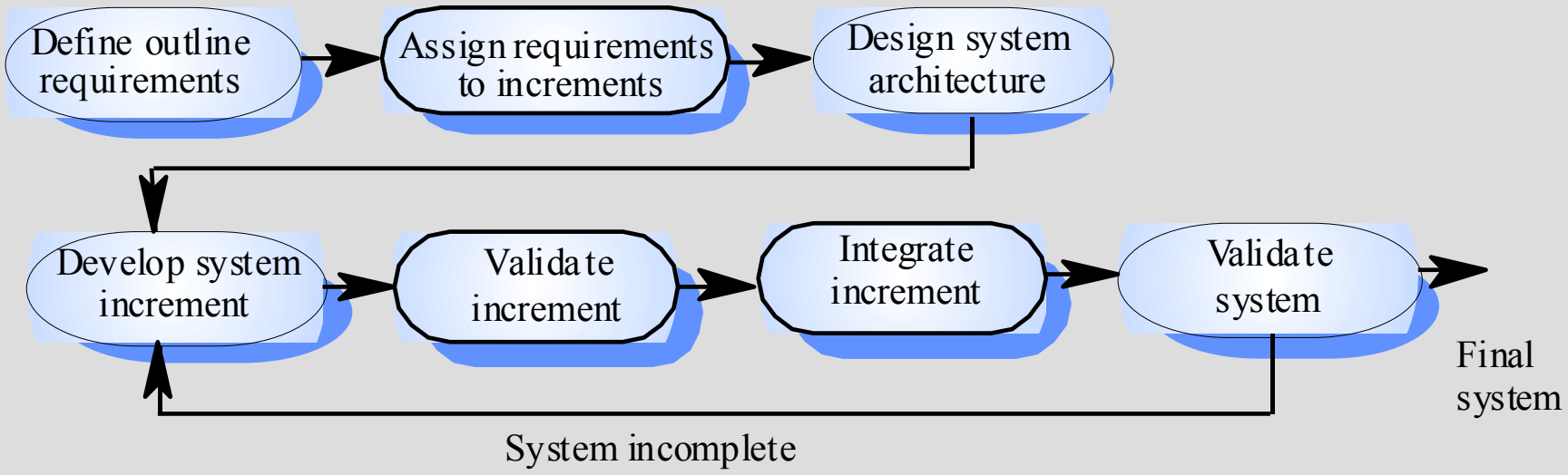
Inkrementaalse arendusmudeli *(Incremental model)* sobivad eeltingimused:

- Esmased nõuded tarkvarale võivad olla hästi kirjeldatud, kuid lineaarse arendusmudeli kasutamist takistab see, et ei teata kogu tarkvara ulatust.
- Vajalikuks võib osutuda piiratud funktsionaalsusega tarkvara võimalikult kiire üleandmine ja võimalus täiendavad funktsionaalsused hiljem lisada.

Inkrementaalne arendus. Iseloomustus

- Arendus on tsükliline.
- Tarkvarasüsteemi ei tarnita korraga, vaid osade *(increment)* kaupa, iga osa lisab süsteemi mingi funktsionaalsuse.
- Kasutajate soovid järjestatakse, kõrgema prioriteediga soovid täidetakse varem.
- Kui ühe osa arendamist alustatakse, külmutatakse selle osa nõuded. Teiste osade nõudeid võib samal ajal muuta.
- Iga osa realiseerimine võib toimuda lineaarse mudeli kohaselt.

Inkrementaalne arendus.Skeem



Inkrementaalne arendus.Eelised

Eelised

- Klient saab hakata tarkvara olulisemaid funktsionaalsusi varem kasutama.
- Esimesed osad on prototüübi eest, mis aitavad järgmiste osade nõudeid selgemaks saada.
- Projekti läbikukkumise risk on väiksem.
- Kõrgema prioriteediga (ja riskantsemad) süsteemiosad saavad rohkem testitud, kuna on varem valmis ja kasutuses.

XP – extreme programming

- **XP** on iteratiivsele mudelile tuginev agiilne arendusmeetod
- K. Beck 1999
- *Suhteliselt* uus meetod – programm luuakse ja tarnitakse hästi väikeste tükkidena
- Kliendid on tihedalt programmeerijatega seotud.
- Rakendatakse paarisprogrammeerimist.
- Ühine koodiomandus
- Testjuhitud arendus.

Evolutsioonilised mudelid

Eeltingimused:

- nõuded tarkvarale muutuvad arenduse vältel;
- lühikesed tähtajad ei võimalda kogu tarkvara korraga valmis teha;
- keskne osa nõuetest võib olla selge, kuid vajalikke lisasid on raske esialgu määrata.

Kirjeldatud olukorras ei sobi kasutada koskmudelit, sest kõiki tingimusi ei saa arendustegevuse alguses paika panna.

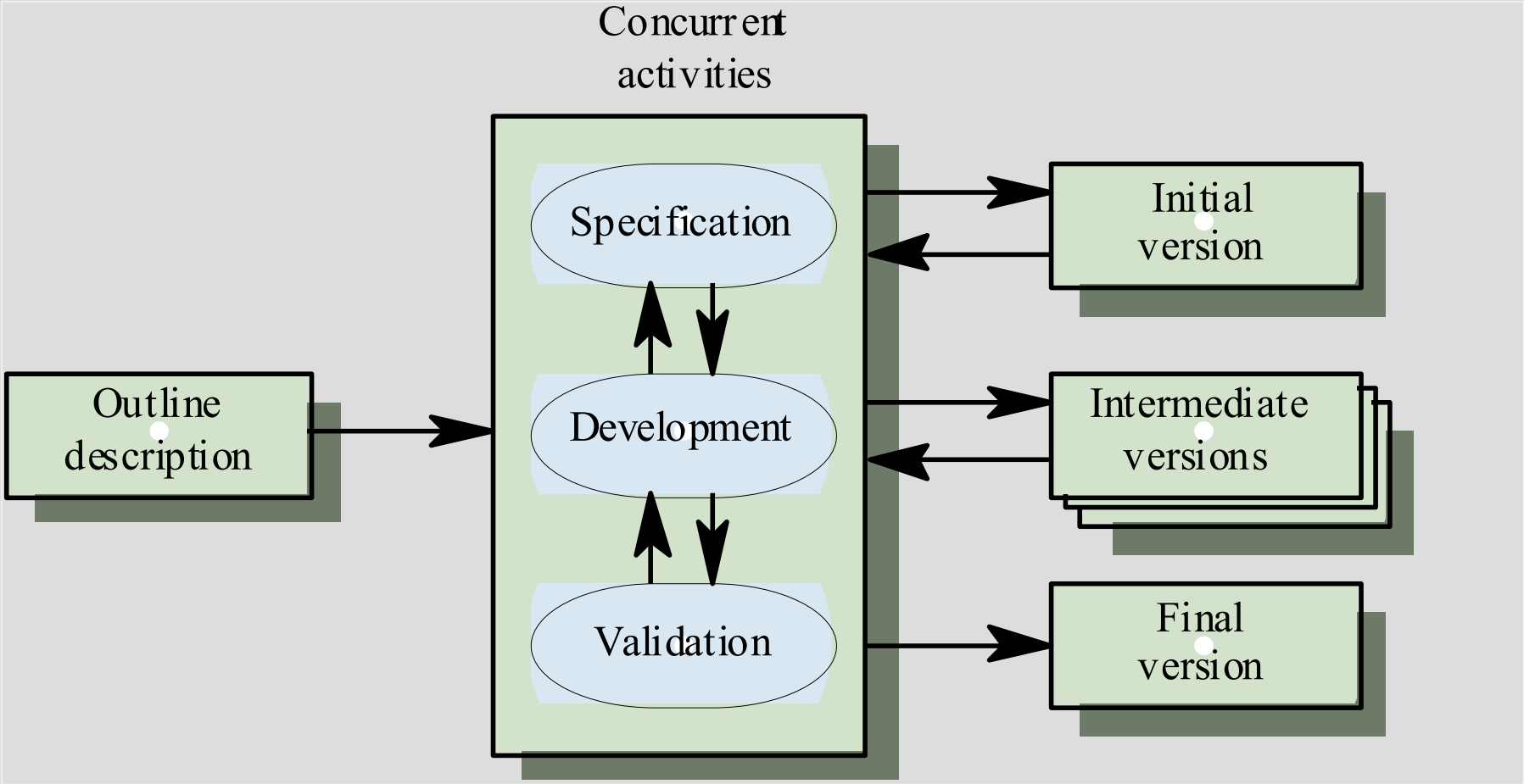
Evolutsioonilised mudelid

Evolutsioonilised mudelid *(evolutionary model)* on iteratiivsed, mis tähendab, et tarkvara ja kõik temaga seonduv muutub iga versiooniga täiuslikumaks.

Kaks näidet:

- Prototüüpimine *(Prototyping)*
- Spiraalmudel *(Spiral Model)*

Evolutsiooniline mudel.Skeem



Evolutsiooniline mudel.Probleemid ja kasutatavus

- Probleemid
  - Protsess pole nähtav.
  - Tekkiv tarkvara võib olla halvasti struktureeritud
  - Võib olla vaja erilisi oskuseid, vahendeid (prototüüpimise keel jms)
- Kasutatavus
  - Sobib väikeste ja keskmiste interaktiivsete süsteemide jaoks
  - Suurte süsteemide osade jaoks (kasutajaliides)
  - Lühikese elueaga süsteemide jaoks, sest ei pruugi olla hea struktuuriga

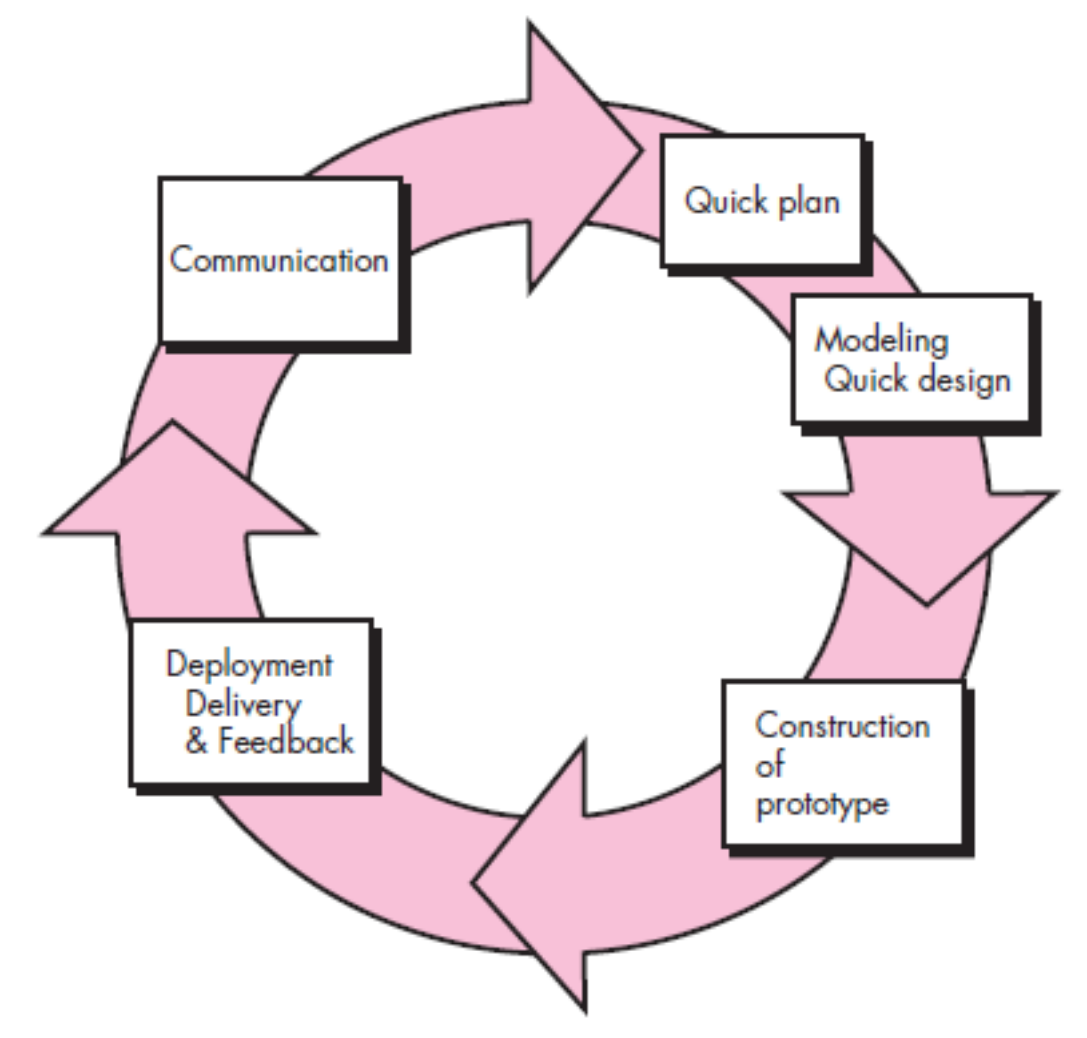
Prototüüpimine

- Kliendil võib olla üldine arusaam sellest, mida ta tahab, kuid ta ei oska tuua välja konreetset funktsionaalsust jt nõudeid.
- Arendaja ei ole kindel, kas valitud viis arvuti ja kasutaja suhtlemiseks on mõistlik (liides).
- Prototüüpimist võib käsitleda kui eraldi protsessimudelit.
- Prototüüpimist võib samas kasutada suvalise arendusmudeli osana.
- Kasulik, kui nii kasutajal kui ka arendajal on detailid segased

Prototüüpimine. Etapid

- Nõuete kogumine, kommunikatsioon
  Üldisemalt + mida tuleb kindlasti täpsustada
- Kiire kavandamine → prototüüp
  Keskendub süsteemi nähtavale osale
  (kasutajaliides, väljundvormid, ...)
- Prototüüpi näidatakse kliendile
  Klient hindab prototüüpi ja täpsustab oma soove,
  tagasiside alusel minnakse “uuele ringile”
- Iteratsioon
  Kõik kordub algusest peale, kuid uuel tasemel

Prototüüpimise skeem



Prototüüpimise viisid

- Prototüüpimisega määratakse nõuded tarkvarale.
- Prototüüpimisel võib kasutada juba tegeliku tarkvara fragmente (evolutsiooniline, prototüüp areneb tarkvaraks)
- Prototüübi võib täielikult “ära visata” ja alustada nullist uuega *(throwaway)*. St prototüüp on tihti tehtud spetsiaalsete prototüüpimise vahenditega, mitte programmeerimise vahenditega.

Prototüüpimine.Probleemid

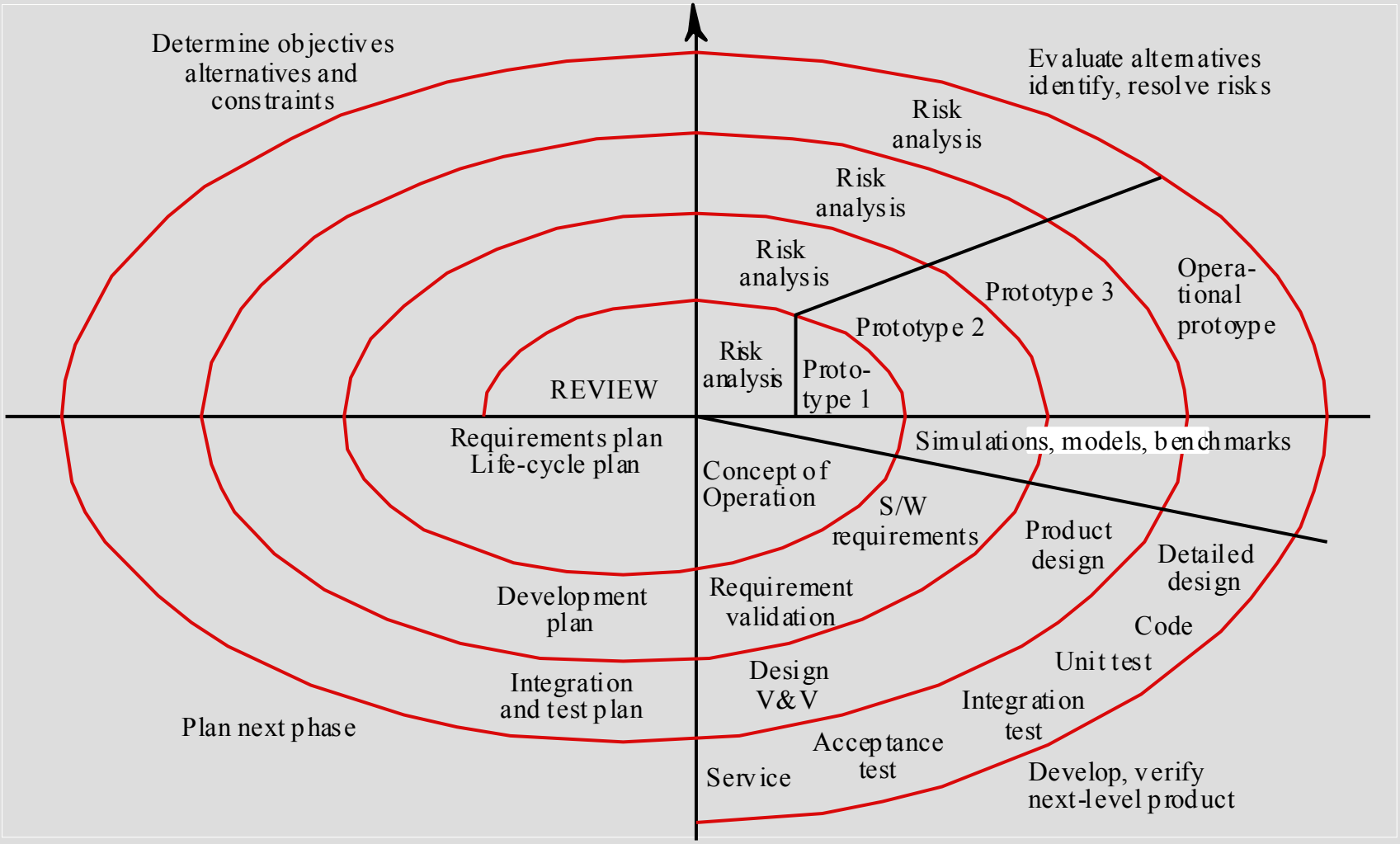
Probleemid

- Tarkvara seisab esialgu koos “kleeplindi ja nätsu” abil. Kvaliteetse tarkvara jaoks tuleb kõik ringi ehitada. Klient ei saa aru, et paari päevaga ei teki prototüübist kvaliteetset süsteemi. Ja arendaja allub kliendi “virisemisele”.
- Arendaja kasutab alguses mugavusest mitte kõige optimaalsemaid vahendeid. Lõpuks need muutuvad harjumuspäraseks ja jäävadki süsteemi.

Spiraalne arendus

- Boehm 1988 a
- Protsessi kujutatakse spiraalina, mitte omavahel suhtlevate tegevuste jadana.
- Iga kordus kujutab mingit arendusfaasi (kuid mitte omaette etapi mõttes).
- Faasid ei ole fikseeritud (spetsifitseerimine, kavandamine), vaid tehakse seda, mida parasjagu vaja.
- Suur tähelepanu riskide hindamisel ja maandamisel iga korduse alguses.

Spiraalne arendus.Skeem



Spiraalmudeli sektorid

- Eesmärkide seadmine
  - Määratakse selle faasi eesmärgid, alternatiivsed võimalused
- Riskide hindamine ja maandamine
  - Analüüsitakse riske ja leitakse teed nende maandamiseks (sh kasutatakse prototüüpimist)
- Arendus ja valideerimine
  - Valitakse (riskidest lähtudes) selle faasi jaoks sobiv arendusmudel ja tehakse midagi valmis.
- Järgmise etapi planeerimine
  - Projekt vaadatakse üle ja otsustakse, mis saab edasi, planeeritakse järgmine kordus.

WINWIN spiraalmudel

- Spiraalmudeli täiustatud variant nö vastastikku kasulik mudel (Boehm 1998).
- Rõhk on arendajate ja klientide vahelistel läbirääkimistel.
- Tegutsetakse selle nimel, et mõlemad pooled võidaksid *(win)* – arendamise kolmnurk!!
- Klient saab süsteemi, mis enamusi tema soove rahuldab.
- Arendaja võidab realistliku eelarve ja tähtaegade osas.

WINWIN faasid

Võrreldes tavalise spiraalmudeliga on lisasektorid järgmised:

- Määrata kliendipoolne süsteemi või süsteemi osa huvipool *(stakeholder).*
- Määrata huvipoole “võidu tingimused” *(win conditions)* – mida peab klient saama
- Läbirääkimised huvipooltega, et nende soovid saaksid vastavusse arendajate soovidega ja et mõlemal poolel oleks võit.

Spiraalmudeli pos ja neg

- Ideaalis peab riskide hindamine andma võimaluse nende avastamiseks ja maandamiseks.
- Kui oluline risk jääb õigeaegselt märkamata, ei ole teiste riskide maandamisest kasu.
- Kliendid ei pruugi olla nõus (eriti lepingu tingimustes) pidevaid uusi läbirääkimisi pidama ja selline mudel ei tundu neile juhitav.

Komponentidele tuginev mudel

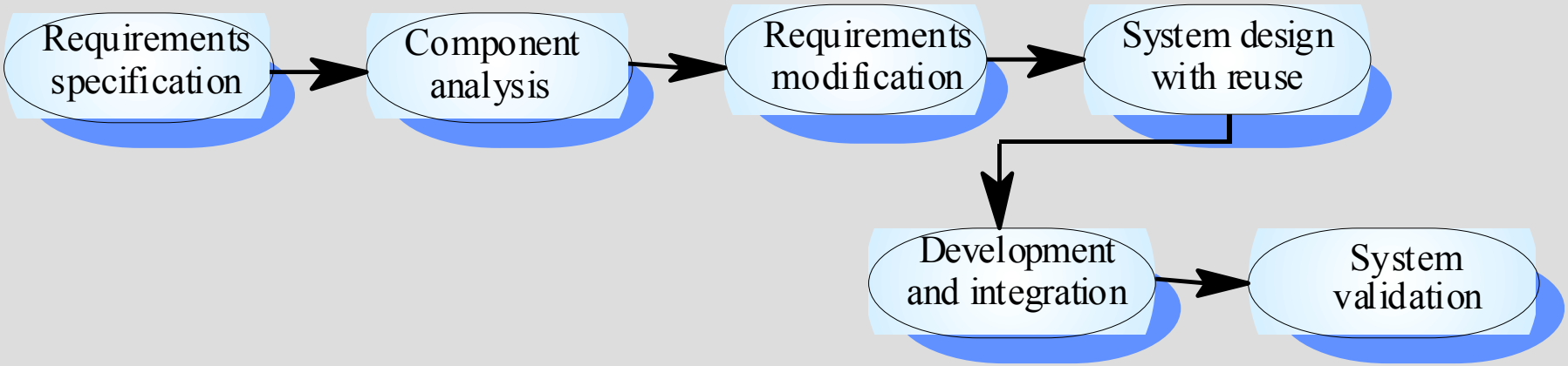
Component based model – spetsiifiline protsessimudel, mis eeldab varem loodud komponentide intensiivset taaskasutamist.

- Tugineb süstemaatilisele korduvkasutamisele *(reuse)*
- Tarkvara integreeritakse valmis komponentidest *(COTS - Commercial Off-The-Shelf komponendid).*
- Vajalik suur komponentide baas. Komponentide all võib mõista väga erineva suurusega osi.

Komponendid.Protsessi faasid

- Nõuete analüüs
- Komponentide analüüs
  Nõuetest lähtudes uuritakse komponentide baasi sobivate komponentide leidmiseks
- Uuritakse komponentide integreerimise võimalusi.
- Projekteeritakse tarkvara arhitektuur kooskõlas leitud komponentidega.
- Realiseeritakse uued komponendid.
- Testitakse, et veenduda töökindluses ja funktsionaalsuste olemasolus.

Komponendid.Skeem



Komponendid.Eelised ja puudused

- Eelised
  - Vähem on vaja programmeerida
  - Vähendab maksumust ja riske
  - Enamasti valmib kiiremini
- Puudused
  - Komponendid pole sobivad ja süsteem võib saada veidi vale
  - Kaob kontroll, kui ei hallata komponentide baasi, st baasi haldamine on lisatöö

Kokkuvõte

- Tarkvaraprotsess on tegevuste kogum tarkvara loomiseks. Protsessimudel on protsessi abstraktne, lihtsustatud esitus.
- Protsessid sisaldavad tarkvara spetsifikatsiooni tegemist, kavandamist ja realisatsiooni, valideerimist ja muudatuste tegemist.
- Üldised protsessimudelid kirjeldavad tarkvara protsessi ülesehitust.
- Iteratiivsed protsessimudelid esitavad tarkvara protsessi kui tegevuste kordust.