



NTNU

Det skapende universitet

Systemutvikling (Software Engineering)

Professor Alf Inge Wang

Undervisningsmål og henvisning

- Målet med timen er:
 - Få kunnskap om hva systemutvikling er
 - Forstå hva en utviklingsprosess består av
 - Få kunnskap om ulike utviklingsprosessmodeller
- Henvisning:
 - Theory Book IT Grunnkurs, Software Engineering, side 241-276



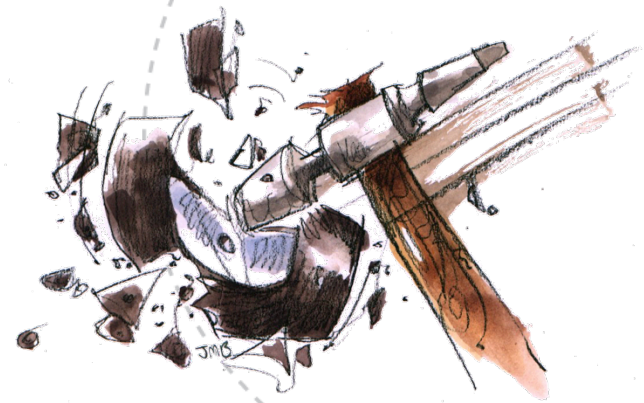
pages 243-245

SoftwARE EnGINEERING

Motivasjon for systemutvikling

- Nesten alt vi omgir oss med i samfunnet er avhengig av programvare og programvaresystemer:
 - Økonomi, handel, transport, utdanning, underholdning, statlige tjenester, kommunikasjon, helsetjenester, ...
- For at samfunnet skal fungere, må programvaren fungere!
- Systemutvikling (software engineering) omhandler alle teorier, metoder og verktøy for profesjonell utvikling av programvare fra start til slutt.
 - Sikre systemer som fungerer, som kan utvikles raskere, etc...

Suksess og fiasko i systemutvikling



- Norge, 1990: Rikstrygdeverkets informasjonssystem TRESS-90 ble ikke ferdig innen tidsfristen. Fem å senere etter flere utsettelse blir prosjektet stoppet uten at fungerende system levert.
- Denver, 1994: Programvare til styring av bagasjetransport utsettes 12 måneder.
- Fransk Guyana, 1996: Kommunikasjonssatellitten Adriane-5 stryter på grunn av feil i programvare.
- År-2000 problemet. Massive problemer
- Planeten Mars, 2000: NASA mister Mars Orbiter som skulle gå i bane rundt Mars på grunn av programmeringsfeil.

Viktige attributter for god programvare

- Vedlikeholdbarhet: Programvare skal skrives slik at systemet kan endres for å møte kundenes behov.
- Pålitelighet og sikkerhet: Pålitelighet til et system inkluderer sikkerhet, tilgjengelighet og safety.
- Effektivitet: Et system skal ikke sløse bort systemressurser som minne, diskplass, nettverk eller CPU.
- Akseptering: Systemet må kunne aksepteres av den type brukere det er designet for ved å være forståelig, brukervennlig, og kompatibelt.

sidene 247-274

SOFTWARE PROCESS

Programvareutviklingsprosessen

- Et strukturert sett av aktiviteter som trengs for å utvikle et programvaresystem
- En beskrivelse av en programvareutviklingsprosess inkluderer:
 - Aktiviteter: Det som skal utføres som en del av utviklingen
 - Produkter: Det som produseres som en del av utviklingen
 - Roller: Reflekterer ansvar ulike personer i prosessen har
 - Pre- og post-tilstand: Forutsetninger for for eksempel å sette i gang med en aktivitet

Fire fundamentale aktiviteter i en systemutviklingsprosess

- Programvarespesifikasjon, hvor kundene og ingeniørene definerer programvaren som skal produseres og operasjonelle begrensninger.
- Programvareutvikling, der designes og programmeres programvaren.
- Programvarevalidering, der sjekker man at programvaren møter kundenes krav.
- Programvareevolusjon, der programvaren endres for å reflektere endring i kunde- og markedskrav.

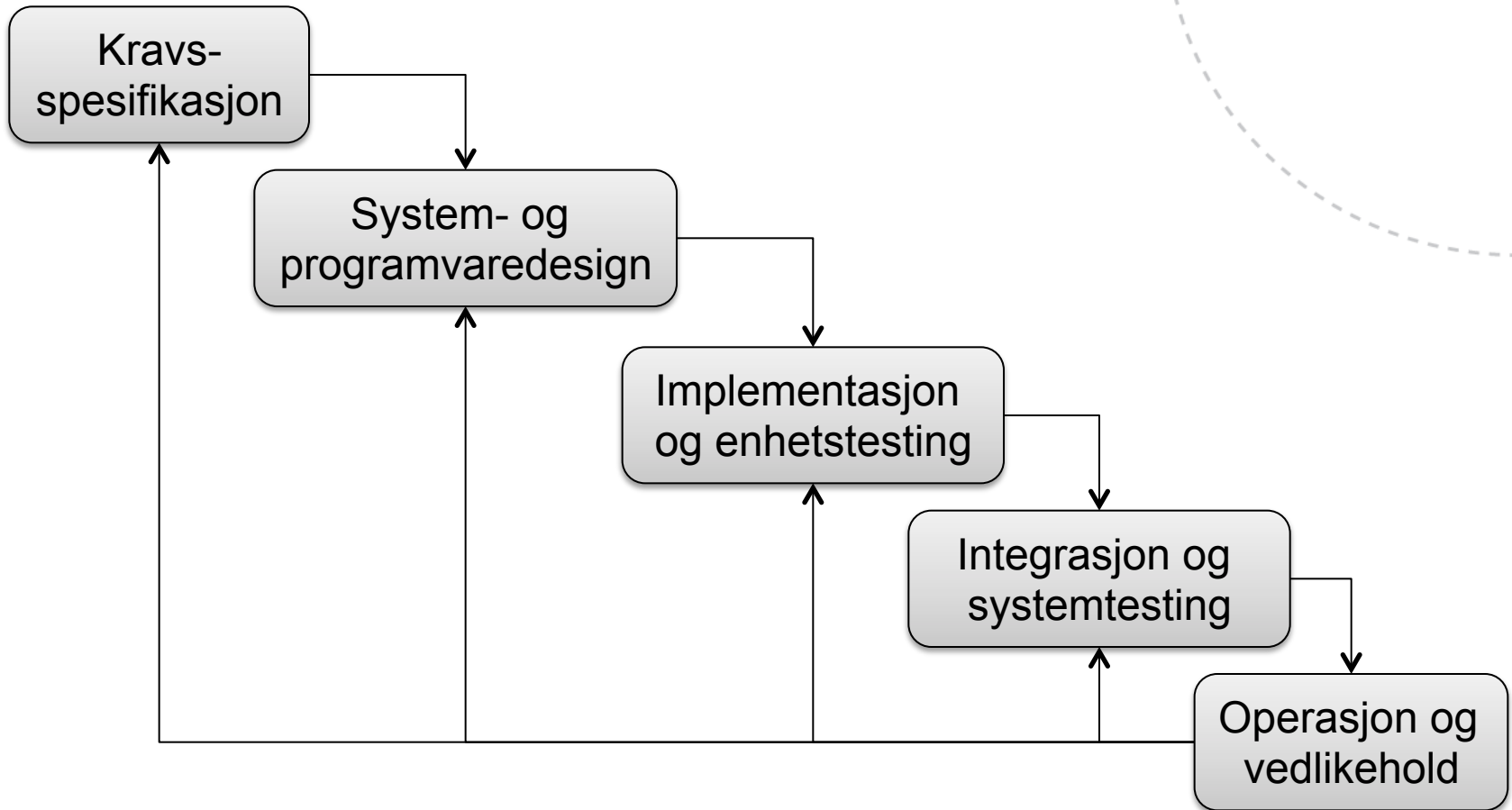
Plandrevet og smidige (*agile*) prosesser

- Plandrevet prosesser: Alle aktivitetene er planlagt på forhånd og framdrift blir målt mot denne planen.
- Smidige prosesser: Inkrementell planlegging som gjør det enklere å tilpasse prosess til endring av krav fra kunde
- I praksis vil utviklingsprosesser være en kombinasjon av plandrevet og smidig.
- Finnes ikke noe fasit på riktig og feil for programvareutviklingsprosesser.

Programvareprosessmodeller (software process models)

- Vannfallsmodellen:
 - Plandrevet modell: Deler opp prosessen i distinkte og separate faser av spesifisering og utvikling.
- Inkrementell utvikling:
 - Spesifisering, utvikling og validering er sammenflettet. Kan være både plandrevet og smidig.
- Gjenbruksorientert systemutvikling:
 - Systemet blir satt sammen av eksisterende komponenter. Kan være både plandrevet og smidig.
- I praksis utvikles de fleste store systemer i en prosess som inkluderer elementer fra alle tre ovenfor.

Vannfallsmodellen



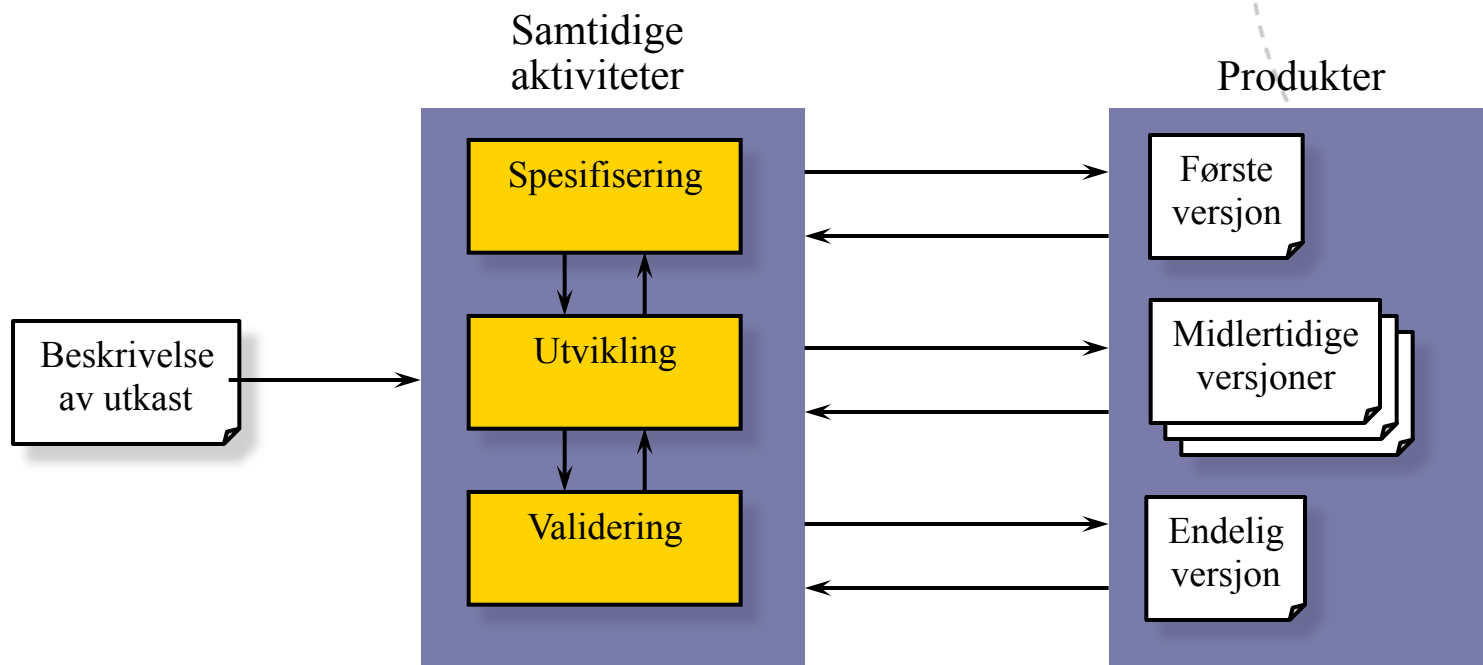
Vannfallsmodellen

- Kravspesifikasjon (Hva): Spesifisere hva kunden ønsker i form av funksjonelle, ikke-funksjonelle og kvalitetskrav.
- System- og programvaredesign (Hvordan): Designe programvarearkitekturen og hvordan systemet skal implementeres (struktur, algoritmer, databaser)
- Implementasjon og enhetstesting: Skrive og teste kode.
- Integrasjon og systemtesting: Teste integrering av ulike deler av systemet og systemet i sin helhet.
- Operasjon og vedlikehold: Installere og drifte system hos kunde. Gjøre nødvendige tilpasninger og endringer.

Egenskaper til vannfallsmodellen

- Deler inn prosessen i veldefinerte faser
- Enklere prosjektledelse (veldefinert prosess)
- Ulempe: Vanskelig å håndtere endringer underveis. Må vanligvis avslutte en fase før fortsette til neste.
- Lite fleksibel oppdel av prosjekt gjør det vanskelig å håndtere endring av krav fra kunde.
 - Egner seg til prosjekter der krav er veldefinerte på forhånd
 - Få forretningssystemer har stabile krav
- Vannfallsmodellen blir mest brukt til store prosjekter som utvikles på flere plasser (distribuert)

Inkrementell utvikling



- **Motivasjon for inkrementell utvikling:**
 - + Innfri øyeblikkelige krav fra kunder
 - + Inkrementell spesifisering av systemet
 - + Systemet reflekterer en gradvis bedre forståelse av bruker

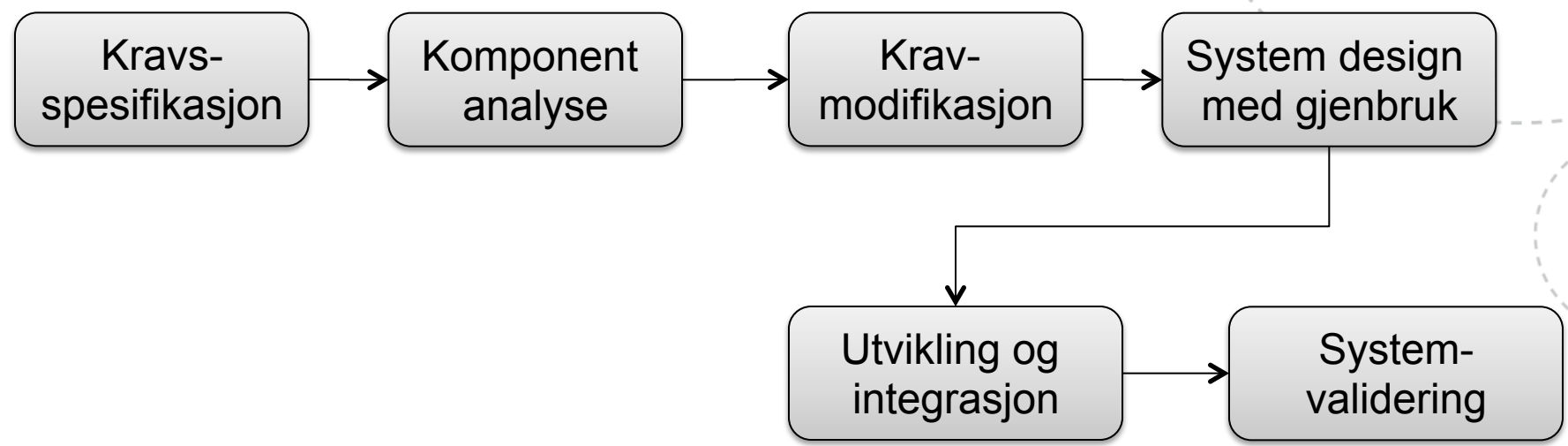
Fordeler med inkrementell utvikling

- Reduserer kostnader ved endring av brukerkrav.
 - Mindre omarbeiding av dokumentasjon enn vannfallsmodellen.
- Enklere å få tilbakemelding fra kunder på utvikling som er gjort underveis.
 - Kundene kan kommentere på demonstrasjoner av systemet og hele tiden se hvor mye som er implementert.
- Raskere levering og kortere tid til å ta i bruk fungerende deler av systemet for kunden.
 - Kunden kan bruke og få verdi fra systemet tidligere enn vannfall

Ulemper med inkrementell utvikling

- Prosessen er ikke synlig
 - Prosjektledere må styre leveranser for å måle framdrift. Hvis systemet utvikles raskt, er det ikke kost-effektivt å produsere dokumenter som reflekterer hver versjon av systemet.
- Strukturen i systemet (arkitekturen) har en tendens til å forringes for hver gang et inkrement blir lagt til.
 - Hvis ikke tid og penger brukes på ”refactoring” for å forbedre programvaren, vil det lede til ødelegge strukturen i programvaren.
 - Dette kan igjen føre til at det blir vanskelig og dyrt å legge til funksjonalitet eller gjøre endringer.

Gjenbruksbasert systemutvikling



Egne steg for gjenbruk

- Komponentanalyse: Undersøke hvilke komponenter som kan brukes og fyller bruker- og systemkrav.
- Kravmodifikasjon: Endring av krav basert på de komponenter som skal brukes.
- Systemdesign med gjenbruk: Designe systemet slik at det kan integreres med eksisterende komponenter.
- Utvikling og integrasjon: Mye av kodingen vil gå ut på å sy sammen systemet med ulike komponenter.

Gjenbruksbasert systemutvikling

- Basert på systematisk gjenbruk hvor systemer er integrert fra eksisterende komponenter eller COTS.
- Gjenbruk er nå standardtilnærning for bygging av de fleste typer forretningssystemer.
- Typer av komponenter:
 - Web-tjenester (web-services) utviklet i følge standarder
 - Samling av objekter utviklet som en pakke til å integreres i komponentrammeverk som .NET eller J2EE
 - Frittstående programvaresystemer (COTS) som konfigureres til bruk i spesifikke programvareomgivelser

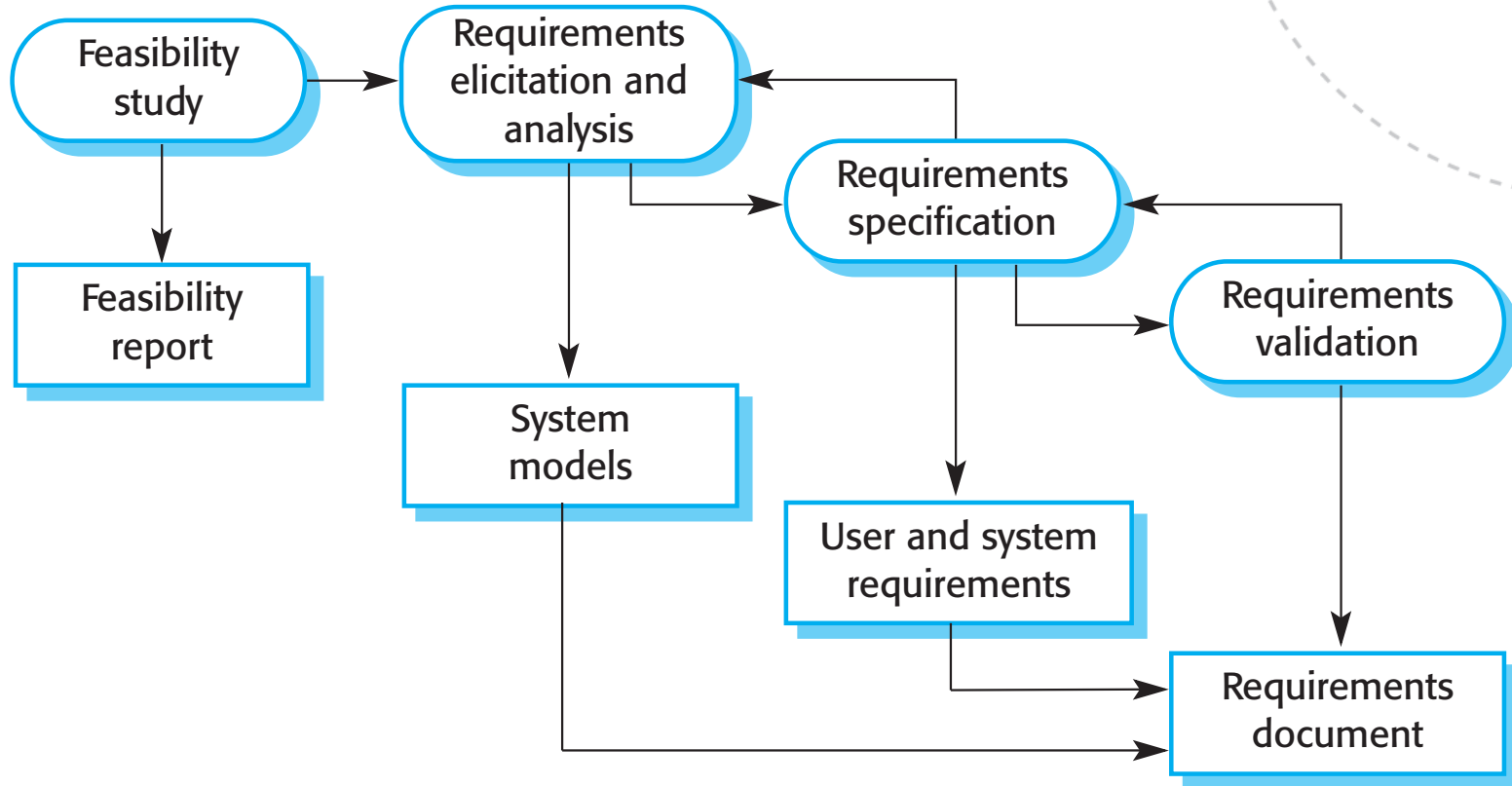
Aktiviteter i en systemutviklingsprosess

- **Programvarespesifikasjon**
- Programvareutvikling
- Programvarevalidering
- Programvareevolusjon

Programvarespesifikasjon

- Prosessen å etablere tjenestene som er påkrevd og begrensninger på systemets operasjon og utvikling.
- Requirement engineering prosess:
 - Gjennomførbarhetsstudie (feasibility study)
 - Er det teknisk og økonomisk forsvarlig å bygge systemet?
 - Kravinnhenting og analyse
 - Hvilke krav og forventninger har systemets interessehavere?
 - Kravspesifisering
 - Definere/Dokumentere kravene i detalj
 - Validering av krav
 - Sjekke validiteten (gyldigheten) til kravene

Requirement engineering prosessen



Aktiviteter i en systemutviklingsprosess

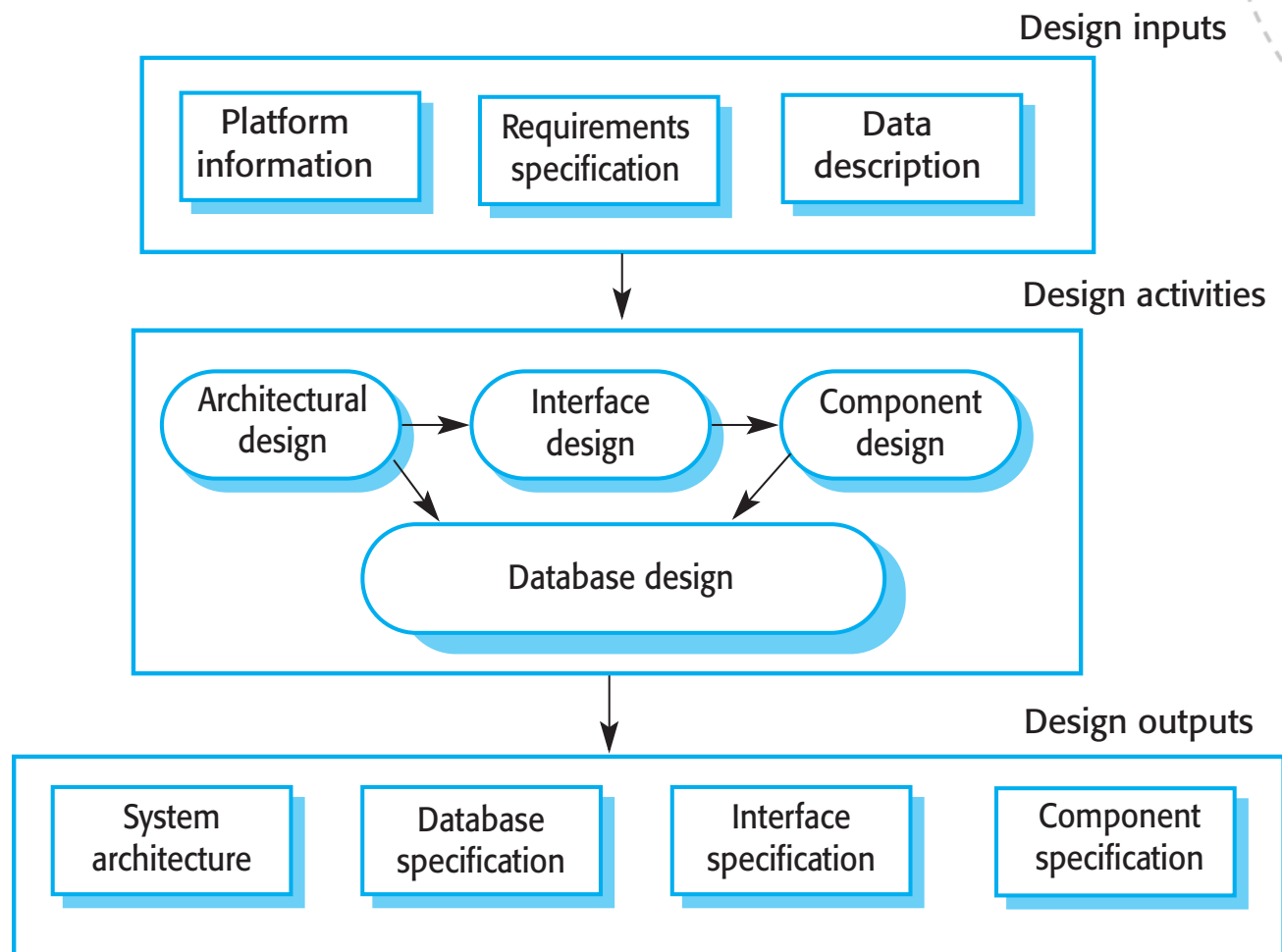
- Programvarespesifikasjon
- **Programvareutvikling**
- Programvarevalidering
- Programvareevolusjon

Programvareutvikling:

Programvaredesign og -implementasjon

- Prosessen å konvertere systemspesifikasjon til et kjørbart system
- Programvare design
 - Designe programvarestrukturen som realiserer spesifikasjonen
- Implementasjon
 - Oversette strukturen til en kjørbart program
- Aktivitetene design og implementasjon er tett relaterte og kan også utføres om hverandre

En modell for designprosessen



Design aktiviteter

- Arkitekturdesign: Hovedstrukturen til systemet bestående av komponenter, deres eksterne attributter og relasjonene mellom komponentene.
- System-interface design: Definerer grensesnitt mellom systemkomponenter.
- Komponentdesign: Designer hvordan hver systemkomponent skal fungere.
- Databasedesign: Designe hvordan datastrukturen for systemet skal være og hvordan det skal representeres i en database.

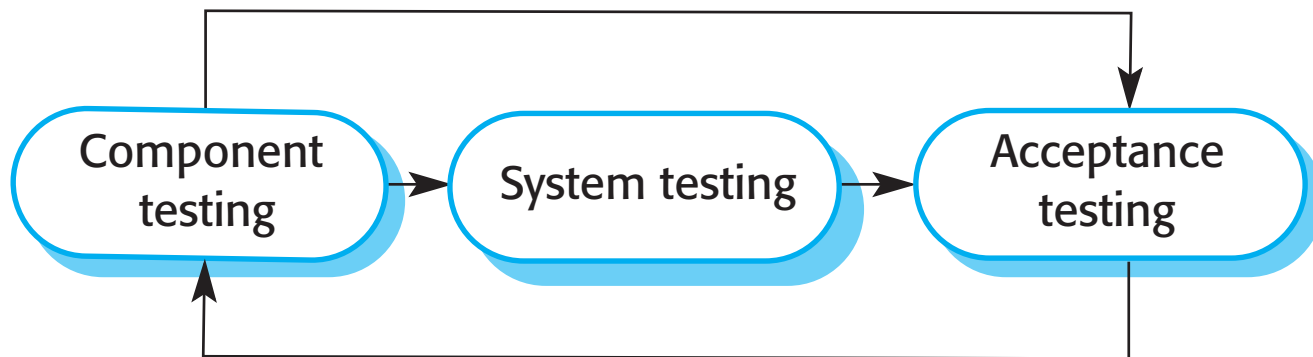
Aktiviteter i en systemutviklingsprosess

- Programvarespesifikasjon
- Programvareutvikling
- **Programvarevalidering**
- Programvareevolusjon

Programvarevalidering

- Verifikasjon og validering (V&V) skal vise at et system stemmer med spesifikasjonen og kundekrav
- Involverer sjekking- og gjennomsynsprosesser (reviews) og systemtesting.
- Systemtesting involverer virkelige brukertester av systemet nedarvet fra spesifikasjon med virkelig data.
- Testing er den mest vanlige V & V aktiviteten.

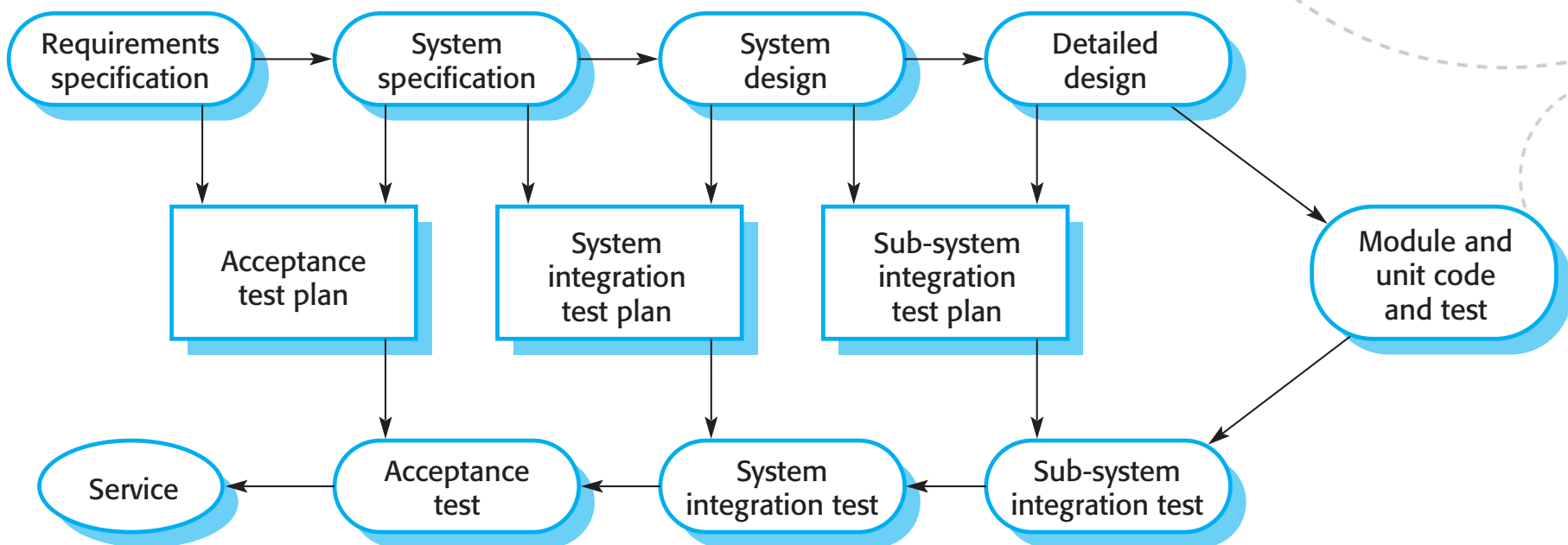
Faser i testing



Testfaser

- **Utvikling- og komponenttesting**
 - Individuelle komponenter testes uavhengig
 - Komponenter kan være funksjoner, objekter eller grupper av disse.
- **Systemtesting**
 - Testing av systemet i sin helhet. Testing av framtidige egenskaper er spesielt viktig (stresstesting).
- **Akseptansetesting**
 - Testing med kundedata for å sjekke om systemet møter kundenes behov.

Testfaser i en plandrevet programvareprosess



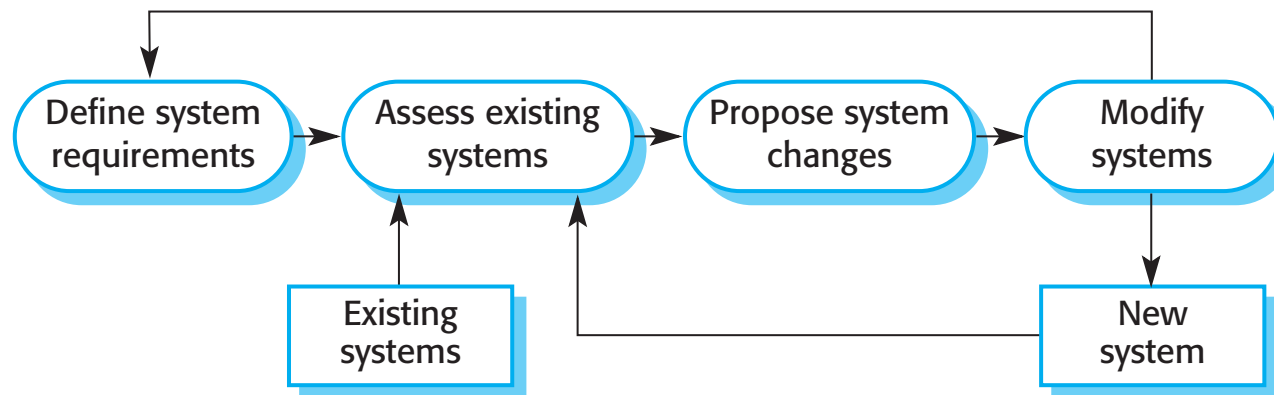
Aktiviteter i en systemutviklingsprosess

- Programvarespesifikasjon
- Programvareutvikling
- Programvarevalidering
- **Programvareevolusjon**

Programvareevolusjon

- Programvare er per def. fleksibel og kan endres.
- Programvaren må endres ettersom kravene endres pga. av endringer i omgivelsene (utføringsmiljøet).
- Skillet mellom utvikling og vedlikehold (evolusjon) blir mer og mer irrelevant ettersom færre og færre systemer blir utviklet fra bunnen av (helt nye).

Programvareevolution



Oppsummering

- Systemutvikling omhandler alle teorier, metoder og verktøy for profesjonell utvikling av programvare fra start til slutt.
- Skiller mellom planmessige og smidige utviklingsprosesser
- Aktiviteter i programvareutvikling:
 - Programvarespesifikasjon, programvareutvikling, programvarevalidering, og programvareevolusjon.
- Forskjell på kravanalyse og design:
 - Kravanalyse ser på HVA system skal gjøre og levere
 - Design ser på HVORDAN system skal oppfylle kravene

Spill igjennom quizzen på Kahoot!

- <https://play.kahoot.it/#/k/d85d54ba-e6ee-4e04-bfe3-aa3b3afb6bbf>
- Spill mot andre:
- <https://play.kahoot.it/#/?quizId=d85d54ba-e6ee-4e04-bfe3-aa3b3afb6bbf&gameMode=ghost&startTime=1447085936006>