

# Vurdering af forskellige systemudviklingsmodeller med henblik på anvendelsen i et bibliotek

Af Edith Clausen

## 1. Indledning

Indførelsen af edb i de bibliotekstekniske processer har igennem årene været årsag til mange dybe frustrationer, utallige søvnløse nætter og uhyggeligt ukontrollerbare omkostninger. På trods af dette har de fleste biblioteker i dag dog i større eller mindre omfang fået integreret edb-teknikken i biblioteks-rutinerne. Men det er jo ikke gjort med valg eller udvikling af det *første* system; verden forandrer sig, nye systemer udvikles, kravene til biblioteket og dets informationssystem ændrer sig og vokser også ofte, og vi befinder os i en situation, hvor vi hele tiden skal afstemme systemerne med omverdenskravene. Hvordan gør vi det? Hvorfor var de første forsøg med implementering af edb-systemer så smertefulde? Og hvordan kan vi komme til at leve med løbende forbedringer af eksisterende systemer og udvikling af nye, uden at det sluger alle bibliotekets ressourcer - både personale- og materialekontiene? - Hvilken systemudviklingsmodel kan vi med fordel anvende til design af bibliotekets informationssystem?

Bibliotekets krav til modellen må være:

- data fra det eksisterende system skal kunne genanvendes
- delelementer af systemet skal kunne ændres uafhængigt af det totale system
- ændringer må kunne foretages hurtigt
- bibliotekets medarbejdere skal selv kunne arbejde med systemet, f.eks. med et 4. generationsværktøj

## 2. Systemudvikling

Når vi taler om systemudvikling er det næsten altid i forbindelse med edb-baserede systemer, selvom der intet er i vejen for at anvende systemudviklingsmodeller på manuelle rutiner.

*Systemudvikling* i forbindelse med et edb-baseret system, f.eks. et integreret bibliotekssystem, betegner den samlede proces, dvs. alle de aktiviteter, der er forbundet med udviklingen og indførelsen af systemet i organisationen/biblioteket.

Dette indbefatter et edb-system, plus andre maskiner og værktøjer, arbejdsmetoder samt de mennesker, der bidrager til løsningen af den bestemte opgave. Det begynder altså fra ideen opstår hos én eller flere medarbejdere, til systemet er i regelmæssig og helst næsten problemfri drift hos nogle forhåbentligt tilfredse brugere (lånere og personale).

Systemudvikling som fagområde kan siges at have en polyparadigmatisk karakter. For at udvikle edb-baserede systemer må man være i stand til at kombinere viden fra matematiske og teknisk-naturvidenskabelige fag (ingeniørvidenskab, organisationsteori) med en viden fra humanistiske (sprogfilosofi) og samfundsvidenskabelige fag (arbejdspsykologi, sociologi, pædagogik).

Inden for systemudvikling taler man om forskellige skoler (Bansler & Clausen 1989):

#### 1. *Den informationsteoretiske tradition*

Inden for den informationsteoretiske skole er maskinen i centrum, og alle problemer vil kunne løses, hvis blot man får adgang til de rigtige informationer. Det betyder, at mennesker i disse organisationer betragtes på lige fod med råvarer og maskiner, nemlig som produktionsfaktorer. Det er en moderne variant af Taylorismen.

#### 2. *Den sociotekniske tradition*

Inden for den sociotekniske skole er trivsel sat i højsædet. Det drejer sig om at øge produktiviteten igennem øget trivsel og tilfredshed blandt medarbejderne. Forskellen fra den informationsteoretiske skole er et andet menneskesyn: der er væsentlig forskel på mennesker og maskiner. Brugerindflydelse er et nøgleord.

#### 3. *Den fagpolitiske tradition*

Inden for den fagpolitiske skole er intentionen at bidrage til et øget demokrati på arbejdspladserne også ud fra antagelsen af, at et øget demokrati vil føre til udviklingen af bedre systemer. Kampen om magten og ressourcerne

spiller en fremtrædende rolle i den fagpolitiske skole.

### 2.1 Systemudviklingsprocessen

Selve systemudviklingsprocessen kan betragtes ud fra forskellige perspektiver (Bansler & Clausen 1989).

#### 1. *Systemudvikling som organisationsudvikling*

I dette perspektiv er systemudvikling en slags organisationsudvikling. Udvikling og implementering af edb-systemer er næsten altid forbundet med organisationsforandringer. Det kan dreje sig om fordelingen af ansvar og kompetence for de nye opgaver, det kan være udarbejdelse af nye regler, oplæring/uddannelse etc. Set ud fra dette perspektiv vil de spørgsmål, man stiller, bl.a. være: Hvordan gennemføres organisationsforandringer med indførelsen af edb-systemet? Hvilke konsekvenser får det for f.eks. arbejdsmiljø og arbejdsforhold?

#### 2. *Systemudvikling som arbejdsproces*

I dette perspektiv lægges vægten på, at systemudvikling også er en arbejdsproces, dvs. en målrettet, økonomisk aktivitet, hvor der stilles krav om koordinering og styring for at udnytte ressourcerne bedst muligt. Der fokuseres i dette perspektiv på udviklingsorganisationen. De spørgsmål, der stilles her, vil bl.a. være: Hvordan planlægges arbejdet? Og hvordan sikrer man sig, at tidsplaner og budgetter holdes?

#### 3. *Systemudvikling som design- og konstruktionsproces*

I dette perspektiv påpeges, at netop design- og konstruktionsaktiviteter spiller en væsentlig rolle i systemudvikling. Design-aktiviteter forudsætter kreativitet og fantasi; mens konstruktionsaktiviteter har krav til disciplin og systematik. I dette perspektiv fokuseres på edb-systemet og de aktiviteter, der er med til at udforme og implementere dette system. De spørgsmål, der stilles her, vil bl.a. være: Hvordan beskrives det nye system og dets anvendel-

se? Hvilke metoder og værktøjer skal der benyttes? Hvordan indkøres og afprøves systemet?

#### 4. Systemudvikling som politisk proces

I dette perspektiv er fokus på den politiske, sociale proces, hvor magtkampe, konflikter og modsætninger mellem forskellige interessegrupper er fremherskende. Det kan være konflikter

mellem edb-specialist og bruger eller mellem edb-specialist og ledelse. De spørgsmål, der stilles her, vil bl.a. være: Hvilke parter er involveret i systemudviklingen, og hvilke interesser har de? Hvilke strategier anvender de involverede parter?

Er der en sammenhæng imellem de tre nævnte skoler og de fire systemudviklingsprocesser? I Figur 1 er de to dimensioner sat op i en matrix.

Skoler Processer	Informationsteoretiske skole	Sociotekniske skole	Fagpolitiske skole
Organisationsudvikling	X	(X)	
Arbejdsproces	(X)	X	
Design- og konstruktionsproces		X	
Politisk proces			X

Figur 1. Matrix over sammenhængen imellem skoler og processer inden for systemudvikling. X: væsentligste sammenhæng; (X): en vis sammenhæng imellem skoler og processer

Som det fremgår af matrixen, er det i den *informationsteoretiske tradition*, at systemudviklingen får form af organisationsudvikling. Der fokuseres på strukturer og processer i den organisation, hvor det udviklede system skal anvendes. En omstrukturering af en organisation i forbindelse med implementeringen af et edb-system vil nødvendigvis også se på forløbet af hele arbejdsprocessen. Den *sociotekniske tradition* derimod har som mål, at systemudvikling både er en arbejdsproces med effektivitet, styring og kontrol; men også en udviklende proces med design- og konstruktionsaktiviteter. Socioteknikerne opstiller anvisninger for samarbejde mellem de involverede parter. Men i følge den sociotekniske skole består en virksomhed af to systemer: det tekniske system og det sociale system, som påvirker hinanden. Derfor er

spørgsmål om arbejdsmiljø og arbejdsforhold også vigtige spørgsmål i forbindelse med organisationsændringer i denne tradition. I den *fagpolitiske tradition* er det hensigten at bruge systemudviklingen som en politisk proces i konflikterne imellem de forskellige interessegrupper. Målet er demokrati i arbejdslivet.

Hvorledes forholder Jørgen Banslers og Hasse Clausens (1989) teorier sig til de fire paradigmer for systemudvikling, som Rudy Hirschheim og Heinz K. Klein fremfører (1989); og hvilke processer hører til disse paradigmer? Figur 2 udbygger Figur 1's matrix.

Skoler Processer	Informationsteo- retiske skole	Sociotekniske skole	Fagpolitiske skole	
	Funktionalisme	Social relativisme	Radikal struk- turalisme	Neohumanisme
Organisa- tionsudvikling	X	(X)		
Arbejdsproces	(X)	X		
Design- og kon- struktionsproces		X		(X)
Politisk proces			X	
Kommunikativ proces				X

Figur 2. Udbygningen af Figur 1's matrix med Hirschheim og Kleins paradigmer. X: væsentligste sammenhæng; (X): en vis sammenhæng imellem skoler, processer og H&K's paradigmer

Det paradigme, som Hirschheim og Klein kalder *funktionalisme*, indbefatter meget godt den informationsteoretiske skole. Systemudviklingen udvikler sig udefra med anvendelse af formelle begreber og rationalistiske metoder og værktøjer. Fokus er her på organisatoriske procedurer, og det er eksperten, der udvikler. Paradigmet *social relativisme* indbefatter den sociotekniske skole. Her udformes systemudviklingen indefra ved at forbedre den subjektive forståelse gennem en interaktion med brugerne. Her er det vigtigt, at alles meninger bliver hørt. Udvikleren optræder som katalysator. Paradigmet *radikal strukturalisme* indbefatter den fagpolitiske skole, hvor systemudviklingen påvirkes udefra ved at vække den ideologiske bevidsthed gennem organiserede politiske aktioner. Det er de økonomiske klasseinteresser, der er i fokus, og udvikleren optræder nærmest som partisan (forkæmper for sociale fremskridt). Det fjerde paradigme *neohumanisme* finder vi ikke hos Bansler og Clausen, og vi finder det ikke særlig ofte i praksis endnu. Tanken er, at med alle aktørerne på banen

og individualiteten sat i højsæde, så er man ikke bundet af stramme systemer; men enhver kan deltage i udviklingen af det ideelle system. Udvikleren her er "befrieren" eller socialterapeuten. Og systemudviklingsprocessen fremtræder som en kommunikativ proces. Det betyder, at der i dette perspektiv lægges vægt på dialogen i en evolutionær udvikling. Dialogen imellem alle parter, der er deltagere i systemudviklingen - og edb-specialisten understøtter denne dialog.

## 2.2 Systemudviklingsfaser

Udviklingen af et edb-baseret system er ofte uhyre kompliceret, og man vil derfor traditionelt opdele systemudviklingen i flere faser. Blandt de fleste forfattere (Abrahamsen 1989; Mathiassen 1981; Dalby 1992; Zych & Mathiasen 1992; Soergel 1985; Burch & Grudnitski 1989; PC'eren på arbejde 1993; Professionel systemudvikling 1991) arbejder man med tre hovedfaser:

- analyse
- design
- implementering

I *analysefasen*, som indeholder idé- og opgaveformuleringen og forskellige forstudier, spørger man hvorfor og hvad (infologisk analyse). Er det en god idé? Skal den realiseres? Hvilke opgaver skal løses og af hvem? Hvordan ser organisationsstrukturen ud i forhold til opgaverne, og hvilken teknologi skal anvendes? Hvilke krav skal stilles til det nye system? Altså hvorfor skal vi have et system, og hvad skal det kunne.

I *designfasen* (systemkonstruktionen) - som også indeholder selve programmeringen af systemet, hvis man selv udvikler - spørger man hvordan (datalogisk analyse). Hvordan skal systemet udformes for at opfylde de kravspecifikationer, analysefasen fastlagde? (Sundgren 1992).

I *implementeringen* (realiseringen, systemigangsætning og drift) bliver summen af de tre ovenfor stillede spørgsmål lig med sådan (hvorfor + hvad + hvordan = sådan). Her sker også en test af systemet med henblik på at se, om systemet lever op til de stillede krav. Denne test fører ofte til justeringer af dele af systemet.

Den enkelte fase skal ikke nødvendigvis være afsluttet, inden detailplanlægningen af den næste går i gang. Nogle af aktiviteterne bør foregå parallelt, således at erfaringer, der er gjort i én fase, ikke nødvendigvis skal efterprøves igen i en anden fase.

### 3. Systemudviklingsmodeller

Der findes mange forskellige systemudviklingsmodeller, ja man kan nogle gange fristes til at tro, at hvert nyt projekt udvikler sin helt egen model. Det, der adskiller de enkelte modeller, er bl.a., hvilke faser modellen deles i, hvilke værktøjer der anvendes, og principperne for organiseringen; men naturligvis som noget meget vigtigt for modellen:

hvilket perspektiv ses systemudviklingen ud fra, jvf. Figur 1 og Figur 2.

Nogle af de tidlige systemudviklingsmodeller bygger på begrebet Systems Life Cycle (SLC), som kendetegnes ved, at der er en trinvis gennemgang af et antal faser. I hver fase beskrives det *nøje*, *hvad* der skal gøres; men ikke *hvordan* man skal gøre det. Som et værktøj til at strukturere selve systemudviklingsprocessen i forbindelse med projektstyring er SLC velegnet.

Det, Systems Life Cycle modellerne mangler, er forbindelsen mellem brugernes behov og det system, der er under udvikling og dermed også ofte, at det udviklede system bliver tilfredsstillende. Dette problem blev taget op i 1970'erne, og de modeller, der blev udviklet, kaldes under ét: Systems Development Methodologies (SDM). I disse modeller sker der en grundig analyse af brugernes behov, og alternative løsninger på problemerne er genstand for nøje vurdering. Som værktøj anvendes f.eks. data-flow-diagrammer (Parker & Benson 1988).

I de følgende afsnit vil jeg beskrive tre forskellige systemudviklingsmodeller, som alle baserer sig på SLC (den projektadministrative fasemodel). Alle har de imidlertid en forskellig perspektivering på systemudvikling, og derfor anvender de også forskellige metoder og værktøjer.

#### 3.1 Dagobert Soergel

Soergels systemudviklingsmodel (1985), som han kalder "systems analysis process", deles i fem faser: systemanalyse, systemdesign, systemimplementering, systemdrift og systemvedligehold (se Bilag 1). Vi kan her også tale om en vandfaldsmodel, men skal være opmærksom på, at der ofte inden for de forskellige funktioner vil forekomme cykliske forløb.

I den første fase *systemanalysen* skal problemet formuleres. Men inden selve systemanalysen sættes i gang, skal som den første funktion systemets

grænser afsættes, dvs. domænet skal nøje overvejes og beskrives, og de overordnede mål fastsættes. Soergel anviser ikke nogle værktøjer til at foretage denne afgrænsning, men viser i nogle eksempler, at det er vigtigt at fokusere på det "environment", systemet skal fungere i/for. - Næste funktion i analysen er en beskrivelse af det kommende system. Først defineres de overordnede mål, som dernæst specificeres i mere operationelle mål, og der opstilles kriterier, som et kommende system skal vurderes efter, bl.a. pris, tidsplan, risiko. Disse kriterier skal udformes i overensstemmelse med organisationens politik og danne grundlag for det endelige valg af system. Igen forsøger Soergel ved eksempler at give nogle ideer om, *hvad* man skal se på, men ikke *hvordan* (værktøjer og metoder). - Og tredje funktion i analysefasen er en vurdering af det nuværende informationssystems status og de elementer, der ikke kan ændres, set i relation til det nye system. Her kommer brugerundersøgelser ind i billedet, og Soergel anviser forskellige metoder til denne dataindsamling: materialer fra eksisterende system, observationer, interviews, spørgeskemaer, logning af aktiviteter.

I *designfasen* skal der findes en plan for løsningen af det opstillede problem. Designfasen består af to funktioner, dels udformning af en række systemforslag, dels valg af en løsning ud fra de opstillede krav. Ved udformningen af systemforslag anbefaler Soergel, at kreativiteten får lov til at blomstre, selvfølgelig med skyldig hensyntagen til resultaterne fra analysefasen, og at ideer dels tages fra den eksisterende litteratur, dels fra hvad man selv kan finde på (det er her, at kreativiteten er vigtig!). De fundne systemforslag skal checkes i forhold til de stillede krav, og de forslag, der opfylder disse krav, skal evalueres for at finde det endelige system. To modeller for evaluering nævnes: *optimizing* (alle forslag vurderes, og det bedste vælges) og *satisficing* (ét forslag ad gangen vurderes, og det, der først opfylder kravene, vælges). Her skal man igen vurdere omkostningerne ved de to metoder set i forhold til projektet. Designfasen afsluttes med en rapport. Som nævnt vil de første fem funktioner foregå i cykliske forløb.

*Systemimplementeringen* omfatter funktionerne konstruktion af systemet inklusive specifikationer og tests og installation af systemet herunder også brugeropklæring.

Og modellen slutter med en driftsfase og en vedligeholdelsesfase, hvor målformuleringerne holdes op imod systemet, og hvor kvaliteten kontrolleres. Disse funktioner giver naturligt feedback til det valgte system, som så evt. må ændres/forbedres.

Soergel opfatter systemudvikling blot som en metode til bedst muligt at opfylde de overordnede mål, som er formuleret i en politisk proces. Det er den politiske proces, der sætter rammerne/målene, og det er systemudviklingsmodellen, der sikrer, at de politisk fastsatte mål opfyldes optimalt. Altså en formel tilgang til problemløsning og beslutningstagning.

Soergels model hører til den informationsteoretiske skole og det funktionalistiske paradigme. Det er ledelsen/politikerne, der fastsætter systemets grænser og bestemmer målene, samt vælger løsningen (funktion 1, 2 og 5). Og selvom Soergel nævner, at medarbejderne også skal deltage i beslutningsprocessen, overlader han de øvrige funktioner primært til systemeksperten (s. 79).

### 3.2 John Burch og Gary Grudnitski

Burch og Grudnitskis systemudviklingsmodel (1989) hører til de såkaldte SDM-modeller (se ovenfor). Modellen er delt i fem faser: systemanalyse; generel systemdesign; systemevaluering; detaljeret systemdesign; og systemimplementering. Dertil kommer yderligere to faser: politik og planlægning; og systemledelse, som indrammer de nævnte fem faser og sætter systemudviklingen ind i virksomhedens strategiske informationssystemplanlægning (SISP) (se Bilag 2). Denne model kaldes vandfaldsmodellen. Den løber oppefra og ned, trin for trin (top-down). Karakteristisk for vandfaldsmodellen er, at den lægger op til, at systemet skal specificeres i form af kravspecifikationer, før systemet kan opbygges, dvs. krav

før design. Samtidig lægger modellen op til, at de enkelte deles samspil nøje skal planlægges, før de enkelte dele bygges, dvs. design før kodning. Vandfaldsmodellen som systemudviklingsmodel passer fint til en målrettet projektstyringsmodel.

Første fase i selve systemudviklingsmodellen er *analysefasen*. Her skal systemets mål og omfang klarlægges, ligesom brugernes ønsker og behov skal afdækkes. Det er den vigtigste fase i systemudviklingen; men der er delte meninger om, hvor stor en del af den samlede tid, der skal være til rådighed til analysen (fra 10% til 40%). Analysen skal munde ud i en rapport, som brugerne skal godkende, inden næste fase påbegyndes. - De værktøjer, der anvendes til indsamling af information i denne fase, er: interviews, gruppeanalysemetoder, spørgeskemaer, observationer, stikprøver og interne og eksterne dokumenter. Man blander her rationalistiske og sociotekniske elementer i processen.

Generel systemdesign, systemevaluering og detaljeret systemdesign kan under ét kaldes *designfasen*. Og det er her de kravspecifikationer, som blev opstillet i analysefasen, skal oversættes til flere løsningsforslag, som præsenteres for brugerne. I en iterativ proces mellem designere og brugere udvælges to eller tre løsningsforslag. I denne proces foretages der hele tiden en evaluering af systemerne; men desuden skal der ske nogle tekniske valg og cost-effectiveness analyser, inden det detaljerede designarbejde går i gang på det system, der vælges. - De værktøjer, der bruges i denne fase, er: en model med design forces, vægtning og byggeblokke, skitser på papir, prototyping, CASE-værktøjer, data-flow-diagrammer, entitets/relationsdiagrammer (E/R diagrammer), HIPO diagrammer (HIPO = Hierarchy plus Input Process Output), flowcharts og mange andre diagramtyper, som skal tydeliggøre systemet for brugerne. - I denne fase udarbejdes også en generel systemdesign rapport, som danner grundlaget for udbud til leverandører af både hardware og software. Tilbuddene evalueres/sorteres ud fra 8 kriterier.

Og så skal systemet *implementeres* i organisationen. Her udfærdiges en implementeringsplan, og et vigtigt element i denne fase er uddannelse og oplæring af brugerne. Der udføres forskellige tests på de forskellige elementer, og der sker naturligvis en del fejlretning i denne fase.

De to faser, der omkranser systemudviklingsmodellen, skal sikre, at det system, der nu bruges mange ressourcer på at udvikle, er i overensstemmelse med organisationens strategiske mål og informationspolitik. Her er det topledelsens opgave at definere informationssystemets *mål*, at finde og prioritere projekter, og at vurdere ressourcerne og kapaciteten af informationssystemet. - Og efter systemimplementeringen skal systemet *styres* i en driftsfase.

Burch og Grudnitskis meget omfattende model for systemudvikling bygger på en filosofi om en proaktiv informationspolitik, hvor informationssystemet har status på linie med jord, bygninger og produktionsmidler. Informationssystemet bliver organisationens strategiske, offensive våben, der sikrer en konkurrencemæssig fordel til gavn både for bund- og toplinien.

Burch og Grudnitskis model hører til den informationsteoretiske skole og det funktionalistiske paradigme. Vi må dog ikke overse, at Burch og Grudnitski også inddrager prototyping og netop i et flowchart viser muligheden af at anvende prototyping, når brugerkrav vanskeligt lader sig definere. Burch og Grudnitskis systemudviklingsmodel kan siges at høre til to paradigmer - funktionalisme og social relativisme - men med hovedvægten på funktionalismen.

### **3.3 A.T. Wood-Harper, Lyn Antill og D.E. Avison**

Wood-Harper, Antill og Avison (1985) har udviklet en systemudviklingsmodel, som de kalder Multiview for at markere, at i forbindelse med systemudvikling må man anlægge en helhedsbetragtning, således at de forskellige processer

(tekniske, strategiske, fagligt-arbejds-mæssige) identificeres, og forbindelserne imellem dem klargøres. Multiview er en metode til at strukturere de opgaver, eksperterne og brugerne skal arbejde med i en analyse- og designfase i en systemudviklingsproces. Det er netop kun analysefasen og designfasen, de tre forfattere beskæftiger sig med. Man kunne kalde denne model for springvandsmodellen (bottom-up).

De har struktureret deres model i fem trin (se Bilag 3): analyse af de menneskelige aktiviteter, analyse af entiteter, funktioner og hændelser (information modelling), analyse og design af det socio-tekniske system, design af menneske-computer interfacet, design af de tekniske subsystemer. Som det ses, falder trin 1, 2 og en del af trin 3 i analysefasen, og resterne af trin 3 samt trin 4 og 5 hører til designfasen.

I *analysefasen* undersøges i det første trin organisationens aktuelle status, hvilke mål og problemstillinger "the analyst" skal arbejde med; hvem er informationssystemets potentielle interessenter, og hvilke konflikter skal der evt. tages stilling til. Denne analyse skal føre til en model for hvilke aktiviteter (arbejdsopgaver), der er nødvendige for at følge organisationens målsætning. Som værktøjer til denne afdækning kan anvendes Rich Picture, som er et billede af miljøet, der gerne skulle give overblik over organisationen, de vigtigste personer og opgaver, den vigtigste omverden, samt relevante relationer imellem billedets elementer. Disse hårde facts skal suppleres med de bløde, som kan være potentielle konflikter og personernes bekymringer. Dernæst kan anvendes en checklisteteknik kaldet CATWOE (Client, Actor, Transformation, Weltanschauung, Owner, Environment), som specificerer, *hvem gør hvad for hvem*, og *hvem* er de ansvarlige overfor, *hvilke* antagelser gøres, og i *hvilket* domæne sker det? Og alt dette skulle resultere i opstillingen af en begrebsmæssig model for det kommende system. - Det følgende trin i analysefasen er en analyse af entiteter og funktioner. Der skal udvikles en funktionel model og en model af entiteter. De værktøjer, der an-

vendes på dette trin, er strukturdiagrammer, data-flow-diagrammer og E/R diagrammer. - Trin tre, der som nævnt både indeholder en analyse- og en designfase, ser på de sociale/arbejds-mæssige forhold og de tekniske forhold. Til begge disse grene opstilles alternativer. Outputtet er socio-tekniske systemløsninger. Disse er sammen med resultatet af informationsanalysen i trin 2 input til trin 4.

Trin 4 og 5 hører til *designfasen*. Først fastlægges dialogformen menneske-maskine (brugergrænsefladen) for det kommende informationssystem; dernæst udformes så systemets tekniske specifikationer, som danner udgangspunkt for enten standard-programmelindkøb eller udvikling af eget programmel. Denne formel er omvendt af de traditionelle udviklingsmodeller, hvor databasestruktur kommer *før* grænsefladedesign.

De fem trin i Multiview-modellen behøver ikke nødvendigvis at tillægges samme vægt i systemudviklingen; nogle kan fremhæves, andre nedtones eller helt udelukkes. *Men* rækkefølgen er vigtig: fra det generelle til det specifikke, fra det begrebs-mæssige til de hårde facts, og fra det problemorienterede til det joborienterede.

Multiview-modellen ser således på informations-systemer som sociale systemer, der mere eller mindre anvender informationsteknologien i designfasen til forskel fra de "klassiske" modeller, der er tekniske løsninger, som så får konsekvenser for processer og mennesker. Multiview tilhører den sociotekniske skole og det social relativistiske paradigme. I dette paradigme lægges vægten på de menneskelige relationer, og "the analyst" skal arbejde ud fra brugernes synsvinkler, og som en katalysator være med til at finde de for dem mest optimale løsninger. Multiview anvender naturligt prototyping i interaktionen mellem brugere og systemfolk.



### 3.4 Sammenfatning af de tre modeller

Alle tre modeller er faseopdelt og har som udgangspunkt en Systems Life Cycle model. Faserne kan være mere eller mindre opdelte; men indeholder de tre grundelementer - *analyse* - *design* - *implementering* (NB: implementeringen tages ikke op af Multiview). Dataindsamlingen, der ligger til grund for analyserne, sker ved interviews af relevante personer i organisationen. Alle modellerne arbejder med udformningen af kravspecifikationer som grundlag enten for udbud eller konstruktion af et informationssystem.

Soergel er den mest restriktive mht. medindflydelse for systemets kommende brugere. Her er det ledelsen, der afgrænser og bestemmer og eksperter, der udformer. Hos Burch og Grudnitski er systemudvikling en del af organisationens informationssystempolitik, og modellen inddrager derfor brugerne ved design af de 6 byggeblokke (building blocks). I Multiview er brugerne udgangspunktet for systemudviklingen.

I Soergel og Burch og Grudnitskis modeller er det det tekniske system, der fastlægges/designes først, dernæst kommer udformningen af brugergrænsefla-

den. Altså den klassiske model: først databasen, dernæst brugergrænsefladen. I Multiview designes human-computer interfacet først, dernæst designes det tekniske system. (Brugergrænsefladen -> databasen).

Soergels model giver slet ikke mulighed for at anvende prototyping. Hos Burch og Grudnitski gives der en mulighed for at inddrage prototyping, nemlig når brugerkrav er vage og uformulerede, så det vil være vanskeligt at udforme kravspecifikationer. Multiviews udvikling af det tekniske system baserer sig på det tætte samarbejde, der har været imellem "the analyst" og brugerne med anvendelse af prototyping.

De tre modeller viser det skred, der er sket i systemudviklingen igennem 10-15 år: Fra Soergels rigide og stereotype model, over mere komplekse tiltag som Burch og Grudnitski til Wood-Harper, Antill og Avisons mere dynamiske, brugerrelaterede model.

De tre ovennævnte systemudviklingsmodeller samt en fjerde, som omtales i afsnit 4.2, er i Figur 3 sat ind i en videnskabsteoretisk sammenhæng.

Paradigmer Modeller	Funktionalisme	Social relativisme	Radikal strukturalisme	Neohumanisme
Soergel	X			
Burch & Grudnitski	X	(X)		
Multiview		X		
Evolutionær systemudvikling				X

Figur 3. De 4 systemudviklingsmodellers tilhørsforhold til Hirschheim og Kleins paradigmer. X: væsentligste tilhørsforhold; (X): et vist tilhørsforhold til H&K's paradigmer

#### 4. Systemudviklingsmodel for et bibliotek

Hvilken strategi skal man anlægge for at sikre, at det system, der udvikles til det enkelte bibliotek, opfylder netop dette biblioteks ønsker og behov? Og hvilken systemudviklingsmodel - om nogen - vil føre til det ønskede resultat?

Arne Munck (Munck 1993) foreslår følgende hovedopdeling af strategierne: 1. Ekspertstrategi, 2. Socio-teknisk strategi, 3. Repræsentationsstrategi og 4. Deltagerstrategi. Ekspertstrategien kendetegnes ved, at det er edb-eksperter, der varetager hele systemudviklingen. Den socio-tekniske strategi er også en form for ekspertstrategi; men her er det eksperter både på edb og de sociale og arbejdsmæssige områder; der sker altså både sociale og tekniske overvejelser i systemplanlægningen. Repræsentationsstrategien har i nogen grad brugerstyring f.eks. ved nedsættelse af arbejdsgrupper med repræsentanter både fra ekspert/lederside og bruger/medarbejderside, hvorved brugersynspunkter afspejler sig i beslutningsprocessen. I deltagerstrategien er det i højere grad brugerne, der styrer systemudviklingsforløbet naturligvis med fornøden brug af faglig bistand.

Soergels systemudviklingsmodel baserer sig på en ekspertstrategi, som vi må afvise som velegnet til implementering af et (nyt) edb-system i et bibliotek, idet det primært er edb-eksperterne, der udvikler systemet uden at gøre sig brugernes behov klart. Burch og Grudnitskis model hører hjemme i den socio-tekniske strategi. Her inddrages andre end edb-eksperter i systemudviklingen, og der gives mulighed for at anvende prototyping, hvis kravene til systemet er svært definerbare. Denne model vil være anvendelig, hvis brugerne intet kender til edb. En model som Multiview falder primært inden for repræsentationsstrategien. Brugere er deltagere i processen; men dog ikke styrende for den, som i en deltagerstrategi, og denne vil være den bedst egnede af de tre modeller for implementering af et edb-system på et bibliotek. De må dog alle karakteriseres som traditionelle systemudviklingsmodeller, idet de også i deres måde

at anvende prototyping på arbejder med faseopdelt systemudvikling og opstilling af kravspecifikationer for udformningen af det endelige system.

Hvilke metoder kan da anvendes for at sikre, at vi får en deltagerstyret systemudvikling, hvor bl.a. et af de store problemer ved traditionelle systemudviklingsmodeller, at systemerne er fyldt med fejl, som først opdages, når systemerne (ofte meget forsinkede) tages i brug, kan undgås?

##### 4.1 Prototyping

Andre problemer i de traditionelle systemudviklingsmodeller er en ringe forståelse og accept af systemerne fra de kommende brugere. Det skyldes dels, at brugerne ikke inddrages tilstrækkeligt i processen, dels at de modeller og diagrammeringsteknikker, der anvendes af systemfolkene, ikke forstås i tilstrækkelig grad af brugerne. Det fører til misforståelser og frustrationer over, at man ikke får det, man forventer. Det er yderst kompliceret at opbygge en fælles referenceramme for systemudviklerne, der ser på teknologien, datastrukturen og systemarkitekturen, og brugerne, der bekymrer sig om indholdet, indflydelsen på deres eget arbejde og arbejdssituation. - Den traditionelle faseopdelte systemudvikling er tidkrævende. Den for systemet så vigtige analysefase skal afdække *alle* problemer og klarlægge *alle* informationsbehov, og der må ikke være fejl i analyserne, da de så vil gå igen i systemet. Det tager tid, og mens tiden går, ændrer verden sig, således at det informationsbehov, man tror sig i færd med at afdække, bliver gårsdagens og ikke fremtidens informationsbehov. I litteraturen er utallige historier om, hvor galt det kan gå, når den nye teknologi erobrer organisationen.

Derfor er det nærliggende at lave en simpel model - en prototype - af systemet, for at komme udenom nogle af de nævnte problemer, vise den for brugerne, få reaktioner på den, og så udvikle den næste prototype, få reaktioner på den etc., indtil systemet kan godkendes af brugerne. Det er altid lettere at se, hvad der er forkert i et system; men det kan være svært at sige, hvad der er det rigtige. Det er

ofte først, når man ser systemet, at det fuldt og helt går op for en, hvad edb-folkene egentlig mente.

*Prototyping* defineres (Prototyping 1992) som en indfaldsvinkel til softwareudvikling, der har følgende træk: en operationel systemversion udvikles på et tidligt tidspunkt; relevante problemer afklares ved at arbejde med modellen; prototyperne giver en fælles basis for diskussioner mellem systemudviklerne og brugerne.

I de traditionelle systemudviklingsmodeller, som vi ser dem f.eks. hos Burch og Grudnitski og i Multi-view, benyttes prototyping for at opnå en god kravspecifikation. Den laves af systemfolkene i nært samarbejde med brugerne, men den anvendes kun til at lave kravspecifikationen. Når kravspecifikationen godkendes af brugerne, udvikles systemet på traditionel vis. Denne form for prototyping kaldes *eksplorativ prototyping*.

*Eksperimentel prototyping* fokuserer på den tekniske implementering af de opstillede mål. Ved at eksperimentere får brugerne som sagt mulighed for nøjere at specificere deres ønsker, og systemudviklerne får indsigt i brugernes arbejdssituation og miljø.

Den tredje type af prototyping kaldes *evolutionær prototyping*. Her er prototyping ikke blot et værktøj, der anvendes i systemudviklingsarbejdet, men en kontinuerlig proces i forbindelse med udvikling af systemer i en evigt foranderlig verden. Her bliver systemudviklerne tekniske konsulenter, der arbejder tæt sammen med brugerne for at forbedre systemet.

Men hvornår vil det være relevant at vælge prototyping som metode? Roland Vonk (Pries-Heje 1993) konkluderer, at anvendelsen af prototyping er afhængig af 4 kriterier: *Omkostninger kontra fordele*. Hvis ikke den udviklede prototype kan anvendes, bør man kun anvende eksplorativ prototyping i forbindelse med afklaring af kravene til systemet. *Tilgængelighed til automatiserede værktøjer*.

Prototyping og CASE-værktøjer (Computer Aided Software Engineering) hører naturligt sammen, det vil næsten være uigennemførligt med prototyping uden CASE-værktøjer. *Systemtype, der skal udvikles*. Da de 4. generationsværktøjer, der eksisterer, har visse begrænsninger, vil prototyping især være velegnet til systemer, der er dataorienterede (ikke procesorienterede), der lægger stor vægt på brugergrænsefladen, eller der har en stærk online karakter. *Brugere og systemudviklere*. Brugerne skal være repræsentative for organisationen og være i stand til/have beføjelser til at træffe beslutninger. Systemudviklerne skal have viden inden for informationsanalyse, teknisk systemdesign og programmering samt være meget brugerorienterede.

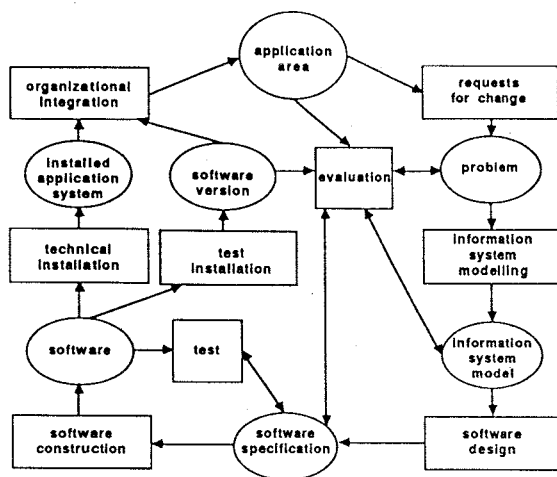
#### 4.2 Evolutionær systemudvikling

Gældende for evolutionær systemudvikling er, at kommunikationen mellem systemudviklere og brugere finder sted under hele systemudviklingsprocessen - ikke kun i analyse og implementeringsfasen; at systemudvikling er en gensidig læreproces for alle, der er involveret i den; at systemet udvikles i små enheder, som gør det lettere at rette og ændre i systemerne; at specifikationerne og implementeringen er komplementære aktiviteter, som ikke kan udføres separat; og at prototyper udvikles så langt, som det er muligt, for at fremme evalueringen i så tidlig en fase af systemudviklingen som muligt, se Figur 4.

Den evolutionære systemudviklingsmodel er i dag mulig og nødvendig, da der er billige PC'er med stor datakraft og udviklet "standard" software på markedet, da brugerne er langt mere vidende og krævende mht. udvikling af systemerne, og da der er en klar tendens til interaktiv programmering.

Samtidig er det en fornuftig model at anvende, da edb-systemerne videreudvikles fra eksisterende systemer, og sjældent konstrueres from scratch, da det ligger i tiden, at genbrug er et must, og da vi arbejder med forskellige teknologier, hvor vi via

forsøg kan bygge brugergrænseflader, der bruger disse forskellige teknologier.



Figur 4. En procesmodel for evolutionær systemudvikling (Prototyping 1992, s. 71)

De umiddelbare fordele både for systemfolk og brugere ved evolutionær systemudvikling er indlysende: medbestemmelse, hurtig udvikling, decentralisering, reducerede omkostninger, mv. Men der kan også være problemer ved denne metode. Dialogen mellem systemudvikler og bruger kan være omfattende og tidkrævende, især hvis parterne står meget langt fra hinanden. Metoden er meget velegnet til at løse brugernes problemer; men kan den enkelte bruger overskue, om alle problemer defineres og løses? Det kræver i hvert tilfælde, at store opgaver brydes ned i mindre enheder, som derved kan overskues, uden at man helt må miste overblikket. Den iterative proces med udvikling af prototyperne kan hyppigt føre til, at den bagvedliggende dokumentation ikke kan følge med eller helt opgives, og det kan resultere i, at samme problem behandles mere end én gang.

Generelt kan man sige, at evolutionær systemudvikling bygger på et positivt menneskesyn. Slutbrugeren får ansvar, indflydelse og magt over det, der skal blive vedkommendes daglige arbejdsredskab.

Som tidligere nævnt er alle former for prototyping ikke realistiske, hvis ikke man kan benytte et computer-baseret værktøj (Pries-Heje 1993). Der kan nævnes to: CASE (Computer Aided Software Engineering) og CSCW (Computer Supported Cooperative Work). De væsentligste fordele ved CASE-værktøjer er knyttet til analyse- og designaktiviteterne. Problemet med dokumentation kan også ofte løses, da en del CASE-værktøjer har indbygget dokumentationsstandarder og blanketter. Der findes mange forskellige CASE-værktøjer på markedet i dag, dels nogle som understøtter en større eller mindre del af systemudviklingen, dels andre der understøtter alle dele af systemudviklingsprocessen. Denne sidste type har den fordel, at informationerne først én gang lagt ind i systemet, kan de bruges over hele systemet. - CSCW-værktøjet blev lanceret i 80'erne. Det er et værktøj, der støtter samarbejde om systemudvikling, hvor samarbejdet forstås som flere personers fælles handlinger, som retter sig mod det fælles mål. Til fremme af dette samarbejde er møder velegnede, og forskellige mødetyper specificeres, selvom mange møder naturligvis kan indeholde elementer af flere typer: 1. Udforskning og brainstorming; 2. Informationsudveksling eller præsentationer; 3. Problemløsning eller beslutningstagning; 4. Forhandlinger og ressourceallokering; 5. Moralsk oprustning eller social strukturering. Disse mødetyper skulle sikre en større bredde i systemudviklingen. Ved disse møder kan de mere traditionelle værktøjer så som data-flow-diagrammer, struktur-diagrammer, en data dictionary, Rich Pictures og rapportgenerator anvendes.

## 5. Konklusion

En gennemgang af tre forskellige, men traditionelle systemudviklingsmodeller viser, at brugerne nok kan få indflydelse/medbestemmelse, men ikke være bestemmende. De traditionelle systemer, der inddrager brugerne, inddrager dem primært i analysefasen for at få afklaret og beskrevet kravspecifikationerne. Til denne afklaring anvender to af modellerne prototyping; men der er som sagt tale om den eksplorative prototyping.

De traditionelle modellers byggen på kravspecifikationen forudsætter, at en komplet og korrekt beskrivelse af informationsbehovet er mulig. Alle, der har arbejdet med edb-systemer, ved, at dette ikke lader sig gøre.

De traditionelle modellers opbygning i faser, forudsætter, at én fase kan afsluttes, inden det næste trin tages. Dette lader sig ikke gøre, da den ene fase hyppigt griber ind i den næste.

De traditionelle modeller har som forudsætning, at de formelle specifikationer er forståelige for både brugere og udviklere. Dette er langt fra tilfældet.

De traditionelle modeller udelukker brugerne fra afgørende aktiviteter, hvilket betyder, at implementeringen og driften af systemet bliver en omfattende og ressourcekrævende proces.

Skal derfor de i indledningen stillede krav: genbrug af data og erfaring, mulighed for at ændre dele af systemet, hurtig gennemførelse af ændringer og medarbejdernes mulighed for "hands on" kunne opfyldes, må vi pege på utraditionelle systemudviklingsmetoder, her den *evolutionære systemudviklingsmodel* som brugbar i et bibliotek.

Konkluderende kan vi sige, at der med den evolutionære systemudviklingsmodel er sket et paradigmeskifte i følge Hirschheim og Klein (Hirschheim & Klein 1989). Den evolutionære systemudviklingsmodel befinder sig i nederste højre felt i deres Figur 1 (side 1202), feltet der karakteriseres ved subjektivitet og konflikt: *neohumanismen*, (se Figur 3). Udvikleren er beskrevet som befrier eller socialterapeut. Systemudviklingen sker indefra ved en forbedring af den menneskelige kommunikation og forståelse, og ved en befrielse fra uønsket tvang. Den gensidige forståelse befordres af hermeneutikken, som studerer de principper, der kan anvendes for at skabe forståelse for situationer og tekster, der kan være vanskeligt forståelige. I evolutionær systemudvikling konfronteres udvikleren med et hermeneutisk problem, når han/hun skal fortolke systemkrav, idet faglig og metodisk

erkendelse er sekundær i forhold til forståelsen af mennesker i deres daglige arbejdsliv. Den menneskelige viden, de erhvervede erfaringer er forudsætninger for den faglige viden. Den evolutionære systemudviklingsmodel kan karakteriseres som en "blød" systemanalysemetode (Christensen 1994), som har mange paralleller til det, der betegnes som den filosofiske hermeneutik.

## 6. Litteratur

Abrahamsen, John. Edb : systemudvikling og bruger. - 3. udg. - København : Udgivervirksomheden, 1989.

Bansler, Jørgen & Hasse Clausen. Fire perspektiver på systemudvikling. *DIKU / Rapport* ; nr. 89/15.

Burch, John, Gary Grudnitski. Information systems : theory and practice. - 5. ed. - New York : John Wiley, 1989.

Christensen, Marianne L. Hermeneutik - fortolkning og forståelse. *Biblioteksarbejde*, 1994, vol. 15, nr. 41, s. 25-40.

Dalby, Joakim. Database-håndbogen : fra design til brug : den relationelle database. København : Samfundslitteratur, 1992.

Hirschheim, Rudy and Heinz K. Klein. Four paradigms of information systems development. *Communications of the ACM*, 1989, vol. 32, nr. 10, s. 1199-1216.

Mathiassen, Lars. Systemudvikling og systemudviklingsmetode. *DAIMI PB-136/1981*.

Munck, Arne. Systemudvikling og projektarbejde. - 5. udg. - Frederiksberg : Danmarks Forvaltningshøjskoles Forlag, 1993.

Parker, Marilyn M., Robert J. Benson with H.E. Trainor. Information economics : linking business

performance to information technology. Englewood Cliffs, N.J. : Prentice Hall, 1988.

PC'eren på arbejde : administrative systemer og slutbrugeren / Povl Rostgård Andersen ... [et al.]. - 2. udg. - Risskov : Forlaget Tenakel, 1993.

Pries-Heje, Jan. Metoder & værktøjer til udvikling af edb-systemer. København : Handelshøjskolen i København, 1993.

Professionel systemudvikling : erfaringer, muligheder og handling / Niels Erik Andersen ... [et al.]. - 5. opl. - København : Teknisk Forlag, 1991.

Prototyping : an approach to evolutionary system development / Reinhard Budde ... [et al.]. Berlin : Springer, 1992.

Soergel, Dagobert. Organizing information : principles of data base and retrieval systems. London : Academic Press, 1985.

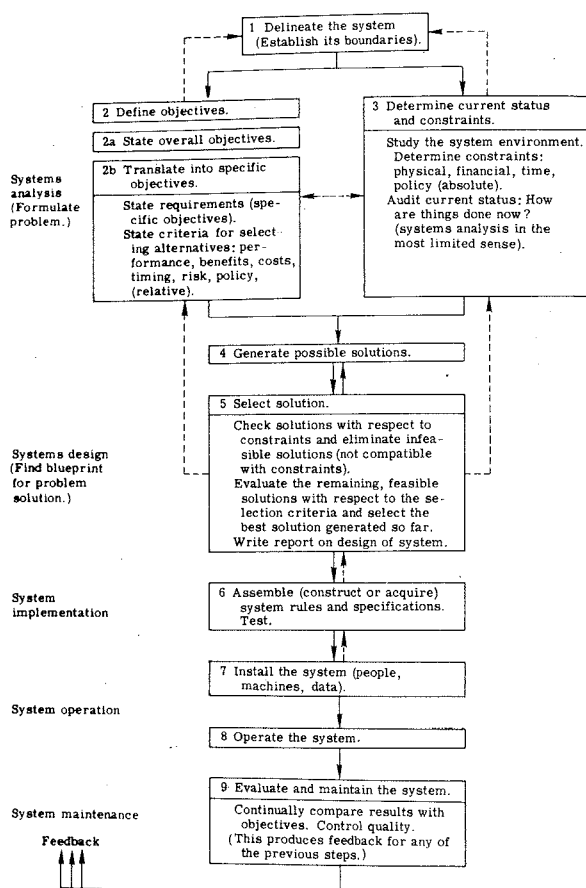
Sundgren, Bo. Databasorienteret systemudvikling. Lund : Studentlitteratur, 1992.

Wood-Harper, A.T., Lyn Antill and D.E. Avison. Information systems definition : the Multiview approach. London : Blackwell, 1985.

Zych, Palle, Helle Mathiasen. Struktureret systemudvikling - med nye beskrivelsesværktøjer. - 2. udg. - Hørsholm : Forlaget /uvlib, 1992.

## 7. Bilagsfortegnelse

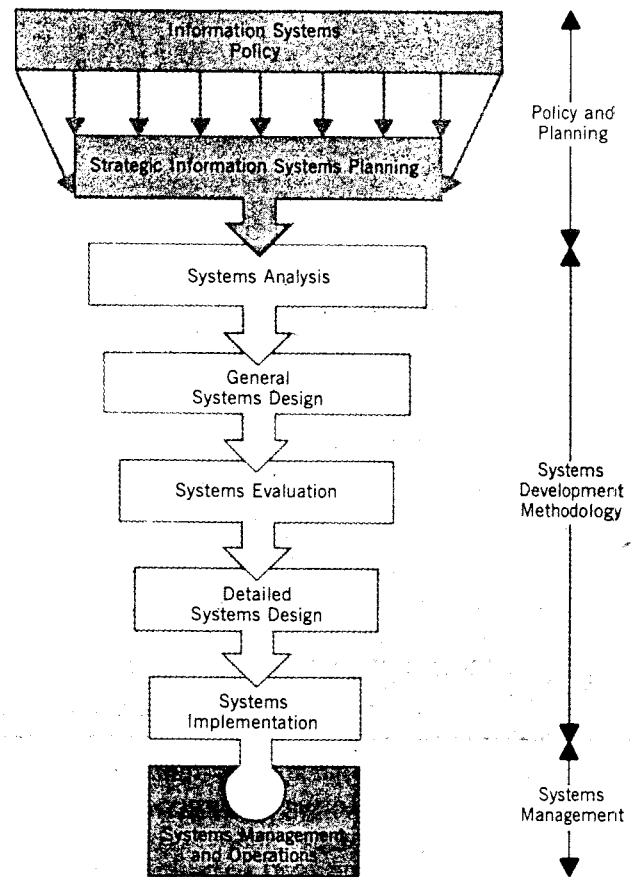
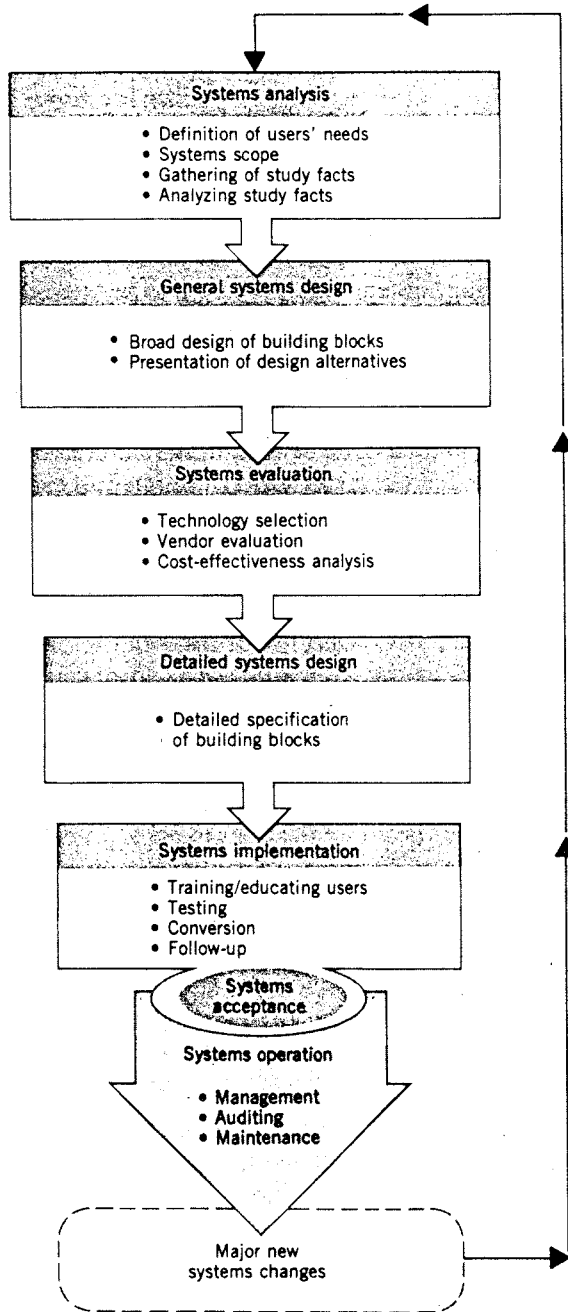
1. Soergels systemudviklingsmodel
2. Burch og Grudnitskis systemudviklingsmodel
3. Multiview metodologien



Bilag 1:  
Soergels systemudviklingsmodel

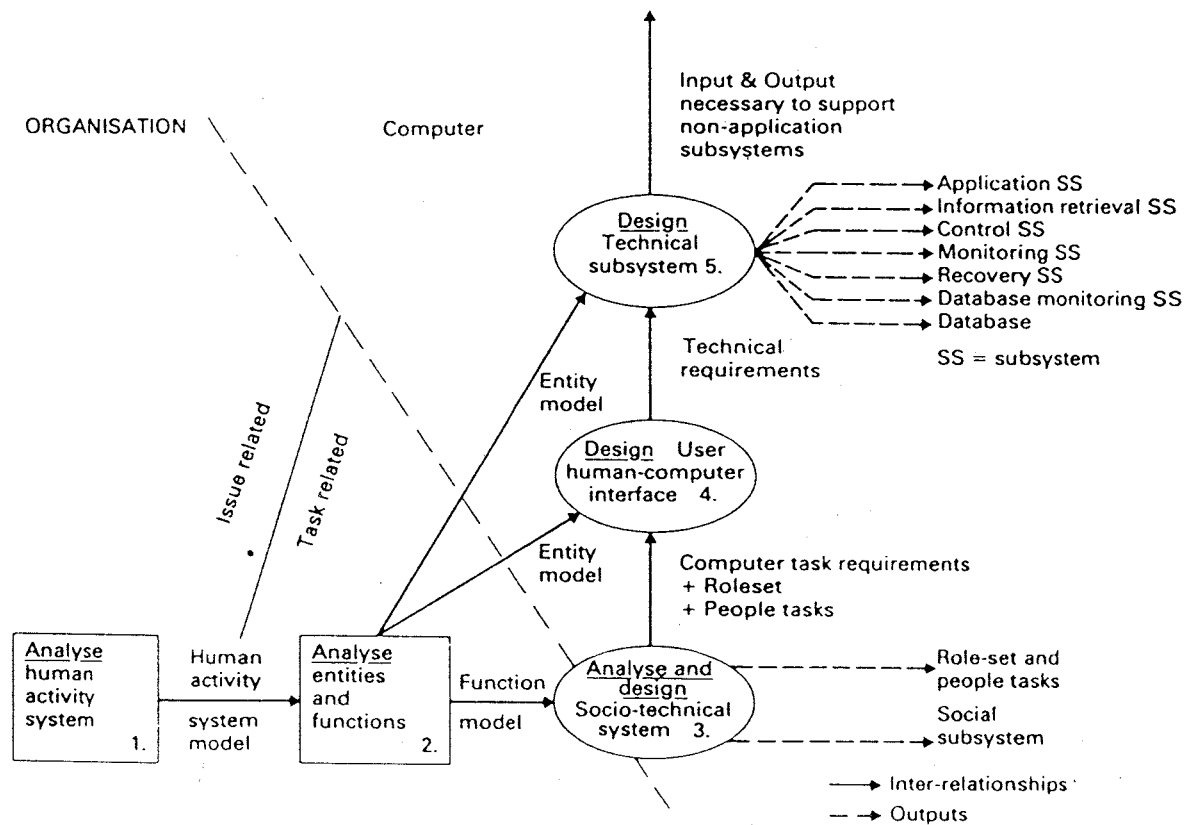
Soergel 1985, s. 72.

**Bilag 2:**  
Burch og Grudnitskis systemudviklingsmodel



Burch & Grudnitski 1989, s. 55 og 532.

**Bilag 3:**  
Multiview metodologien



Wood-Harper, Antill & Avison 1985, s. 17.