

Projektrapport
AUC - datalogistudiet

Omstilling fra SA til OOA

Vurdering af metoder og beskrivelse af omstillingsproces

Gruppetagere

Ole Dolriis
Bjørk Busch
Bjarne Larsen
Bruno Johansen

Forord.

Intet er uforanderligt. Dette kan passende stå som motto for dette projekt, der handler om ændringer af metoder til udvikling af edb-baserede systemer.

Datalogien har i sin forholdsvis korte historie været præget af forandringer i genstandsområdet, holdningen til datalogi og det omgivende samfund.

Forandringerne har været voldsomme, både hvad angår datamatens hardwareteknologi, styresystemer, standardsystemer og systemer til udvikling. Fremkomsten af forholdsvis billige, let anvendelige, globalt udbredte systemer har øget muligheden og interessen for anvendelse af edb, så udbredelsen går langt ind udviklingslandene. Den stigende udbredelse og kompleksitet medfører at konsekvenserne af måden systemer udvikles på spiller en større og større rolle.

Projektgruppen, der som en del af uddannelsen i datalogi, nærmere betegnet bachelor studiet, åben uddannelse ved Aalborg Universitet, gennemfører projektet vil gerne takke vejleder Keld Pedersen for inspirerende og konstruktivt samarbejde. Vi vil her påpege nytten af den direkte og letforståelige kritik, samt forslag til forbedringer vi er blevet forsynet med. Endvidere vil vi også takke for god inspiration fra et andet kursus i datalogisk filosofi, vi for tiden følger hos Peter Axel Nielsen, hvor vi tillige oplæres i den svære kunst at skrive essays. Vi håber, at vore anstrengelser hermed også vil kunne aflæses i denne rapport.

Ole Dolriis

Bjork Busch

Bjarne Larsen

Bruno Johansen

Indholdsfortegnelse.

Indledning.	3
Læsevejledning.	5
1. Grundsyn.	6
2. Perspektiver på taksonomi	8
2.1 Indledning	8
2.2 Flerfaglig disciplin	9
2.3 Brugerdeltagelse	11
2.4 Læreproces	15
2.5 Komplexitet og usikkerhed	17
2.6 Dokumentation	19
2.7 Opstilling af taksonomi	21
3. Kort præsentation af metoder	24
3.1 Struktureret Metode	24
3.2 Objekt Orienteret Analyse	28
4. Sammenligning	30
4.1 Sammenlign SA og OOA	30
4.2 Sammenfatning	36
4.3 Forandringsbehov	38
5. Omstilling fra SA til OOA	40
5.1 Omskoling af systemudviklere	41
5.2 Omstilling af organisation	46
6. Konklusion	51
7. Litteraturliste	54

Indledning.

I gennem en årrække har idealet været, at få indført strukturerede metoder med standarder og værktøjer, der dels skulle gøre systemudvikling til en standardiseret proces, som kunne rationaliseres og kvalitetssikres, dels gøre virksomhederne mere uafhængig af den enkelte systemudvikler.

Diskussionen har været intens mellem forskellige retninger, hvor nogle mener at metoden skal beskrive idealet og være en rettesnor, og andre mener at metoden skal beskrive, hvordan man faktisk gør i praksis. Faktum er under alle omstændigheder, at metoderne har ændret sig i praksis. Der inddrages prototyping og iterationer uden, at dette ses som et egentlig brud med de strukturerede metoder.

Dette projekt vil undersøge om den nyere metode objektorienteret analyse bedre skulle kunne understøtte udviklingen af kvalitetssystemer, og i givet fald hvorfor og hvordan. For at belyse dette har vi udvalgt nogle nøglepunkter, som vi nøjere vil vurdere. Afslutningsvis vil vi beskrive, hvorledes en omstilling fra struktureret analyse til objektorienteret kan finde sted, samt hvilke nødvendige forudsætninger den må have.

Vi er i en brydningstid, hvor det nye slagord hedder objektorientering, der enten kan ses som en naturlig forlængelse af den praktiske brug af de moderne strukturerede metoder eller som et paradigmeskift. Uanset opfattelsen er der ingen tvivl om, at det er den vej "vinden blæser". Når vi i vores projekt ender op med at beskrive en omskolingsplan til den objektorienterede metode kan dette således ikke være nogen overraskelse, idet man med rette kan sige, at konklusionen var givet på forhånd. Vi vil dog ikke afskære os fra, at anbefale at man ikke blot skrotter den strukturerede metode totalt, men at man medtager de gode elementer, der med fordel kan indarbejdes i den objektorienterede metode og berige den.

Gruppens medlemmer er til daglig beskæftiget som undervisere på datamatikeruddannelsen i Odense. Dette forhold har ikke kunnet undgå at påvirke vores fokusområde til de områder, vi finder fortjener at blive fremhævet. Vi tillader os således at fjerne fokus fra det rationelle ideal, der normalt præger edb-området, til det mere humanistiske ideal vel vidende, at set ud fra en snæver virksomhedsbetragtning ville andre områder kunne hævdes, at være ligeså relevante.

Vores målgruppe med projektet er dels studerende og dels lærere, der har beskæftiget sig med de strukturerede metoder og skal i gang med den objektorienterede metode, dels beslutningstagere og medarbejdere i en virksomhed, der skal ændre metode fra den strukturerede til den objektorienterede.

Læsevejledning.

For at skabe overblik for læseren, har vi foruden indholdsfortegnelsen udarbejdet denne læsevejledning, som forhåbentlig kan give oversigt og forståelse for rækkefølge og indhold af rapportens afsnit.

Da projektet går ud på at sammenligne to systemudviklingsmetoder, Struktureret Analyse og Objektorienteret Analyse, har vi valgt at foretage en kortfattet beskrivelse af de to metoder, baseret på kendte fortalere for metoderne. Denne beskrivelse udgør kapitel 3. I kapitel 4 sammenlignes metoderne på en række udvalgte punkter, som projektgruppen har beskrevet og argumenteret for i kap. 2.

Kapitel 4 afsluttes med en sammenfatning af resultatet samt en beskrivelse af forandringsbehov.

I kapitel 5 er beskrevet, hvorledes en forandring fra SA til OOA vil påvirke systemudviklerne og organisationer.

Som nævnt indeholder kapitel 2 en række kriterier for bedømmelse af de to metoder, en såkaldt taksonomi. Da mængden og forskellighederne på mulige taksonomier er stort, har vi i kapitel 1 valgt at redegøre for baggrunden for vores valg, baseret på menneskesyn, egne erfaringer etc..

I kapitel 6 vover vi at drage vores egne konklusioner, baseret på den indsigt og vejledning vi er kommet til del i dette projekt og i øvrige beslægtede moduler på bacheloruddannelsen.

1. Grundsyn.

I mange virksomheder er man i gang med eller har netop afsluttet indførelse af struktureret analyse. I sådan en virksomhed er det nødvendigt, at have gode argumenter på hånden, hvis man vil foreslå indførelse af objektorienteret analyse.

Formålet med denne opgave er blandt andet at tilvejebringe sådanne argumenter. Det vil vi gøre ved først at opstille en taksonomi, som kan bruges, når man vil prøve at finde styrker og svagheder ved konkrete metoder. Når taksonomien er fastlagt, vil vi bruge den til at sammenligne struktureret analyse og objektorienteret analyse. Til slut vil vi forsøge at skitsere en relevant omskolingsplan som kan anvendes, hvis systemudviklere skal omskoles fra at arbejde med struktureret analyse til objektorienteret analyse.

Vi har grundlæggende en skeptisk holdning til taksonomier. Det har vi, fordi det er vores erfaring, at man meget let bevidst eller ubevidst kommer til at manipulere ved hjælp af taksonomien. Denne manipulation sker først og fremmest i forbindelse med udvælgelse og vægtning af de kriterier, der ligger til grund for taksonomien. Problemet er, at udvælgelsen af kriterierne i høj grad er dirigeret af ophavsmandens personlige, uddannelsesmæssige og erfaringsmæssige baggrund.

Ophavsmandens baggrund er ofte ikke udtrykt eksplicit i forbindelse med taksonomien; men efter vor mening bør man redegøre for sine egne grundholdninger til systemudvikling, hvis taksonomien skal være troværdig og anvendelig for andre. Vi vil derfor bruge resten af dette afsnit til at redegøre for vore grundholdninger til systemudvikling. Disse grundholdninger har været bestemmende for valget af de 5 kriterier, vi har udvalgt som værende de vigtigste for os i forbindelse med opstillingen af vor taksonomi.

Vor grundholdning til systemudvikling er sammenfaldende med Dahlbom og Mathiassens tanker om dialektisk systemtænkning, som er baseret på den idé, at verden er i evig forandring, og at man ikke kan forstå den, medmindre man kan forstå, hvad forandring er, og hvorfor den finder sted. Den dialektiske systemtæknings påstand er, at man må tænke ved hjælp af modsætninger for at kunne forstå, forklare og kontrollere forandring.¹

Den dialektiske systemtænkning er et forsøg på en vanskelig balancegang. På den ene side accepterer man den hårde systemtæknings ambition om at kortlægge verden. På den anden side accepteres den bløde systemtæknings ambition om at skabe en rationel debat mellem forskellige verdensanskuelser, selvom der rejses tvivl om den underliggende antagelse om harmoni, orden og

¹ Dahlbom, Bo og Mathiassen, Lars: *Computers in Context*, p. 59.

deltagernes fælles interesser. Ved at tilføje en dimension af konflikter, modsigelser, interesser og magtkampe kommer den dialektiske tænker til at se disse ambitioner med sund skepsis.²

Vi er også enige med dem, som mener, at god systemudvikling skyldes gode systemudviklere - ikke gode værktøjer eller metoder. Efter vor mening, er evnen til at kunne lave gode systemer primært afhængig af en række personlige kvalifikationer hos systemudvikleren. Booch udtrykker det på den måde, at systemudviklingen bør foregå som round-trip gestalt design. Hermed vil han fremhæve systemudviklingens inkrementale og iterative natur. Systemet udvikles gennem gentagne forfinelser af et antal forskellige, men konsistente logiske og fysiske syn på systemet som helhed.³

Booch refererer også et studium af et antal professionelle systemudviklere, som blev observeret, mens de udviklede systemer. Efter denne undersøgelse konstateres det, at systemudvikling ser ud til at bestå af et antal overlappende, iterative, løst koblede processer under opportunistisk kontrol. Gode udviklere arbejder på flere forskellige abstraktions- og detaljeringsniveauer samtidig.⁴

Også Bjarne Stroustrup fremhæver den afgørende rolle, den enkelte systemudviklers personlige evner spiller i forbindelse med systemudvikling. Ifølge Stroustrup findes der ingen kagebogsmetode, som kan erstatte intelligens, erfaring og god smag i forbindelse med systemudvikling.⁵

På basis af vores grundholdning til systemudvikling vil vi i kapitel 2 beskrive 5 kriterier, som vi anser for at være de vigtigste faktorer for en succesfuld systemudvikling. Disse kriterier går igen i den taksonomi, vi opstiller med henblik på at finde fordele og ulemper ved henholdsvis struktureret analyse og objektorienteret analyse.

² Dahlbom, Bo og Mathiassen, Lars: *Computers in Context*, p. 64.

³ Booch, Grady: *Object Oriented Design with Applications*, p. 188

⁴ Booch, Grady: *Object Oriented Design with Applications*, p. 189. Booch citerer her B. Curtis: *But You Have To Understand. This Isn't the Way We Develop Software at Our Company*. MCC Technical Report 1989.

⁵ Stroustrup, Bjarne: *The C++ Programming Language*, 2. ed., p. 362

2. Perspektiver på taksonomi

2.1. Indledning.

Da vi har til hensigt at vurdere metoder til systemanalyse, er det rimeligt at starte med at forholde os til hvad begrebet systemanalyse i det hele taget dækker over.

Starter vi med begrebet analyse har "The new encyclopædia britannica" forskellige beskrivelser for forskellige vidensområder. Vi kan derudaf konkludere, at der ikke er en entydig opfattelse af analyse som generelt begreb, men at opfattelsen følger genstands og videns området. Der er dog det fælles for alle opfattelserne, at det centrale formål med analyse er at skabe indsigt og viden, hvilket derfor også er formålet med systemanalyse.

Systembegrebet kan defineres uendelig bredt, men vi vil indsnævre det til at handle om systemer i virksomheder, hvor der inddrages såvel mennesker som edb-teknologi.

Systemanalyse har i den sammenhæng fokus på forandringer, hvor disse forventes at gavne organisationen. Det centrale formål med analysen bliver at afdække forandringsønsket og skabe et beslutningsgrundlag. Systemanalysen skal dog ikke ses isoleret men i sammenhæng med de øvrige processer i systemudviklingen.

At anvende en kortfattet taksonomi til bedømmelse af systemanalysemetoder er i virkeligheden en uretfærdighed, idet ikke alle aspekter af det omhandlede kan få sin rette sammenhæng. Vi vil her ikke desto mindre forsøge, at opstille en taksonomi baseret på vigtige egenskaber ved en analyse, idet vi ikke vil sprede os for meget, men derimod gå i dybden med de aspekter, der efter vores opfattelse virkelig betyder noget.

2.2 Flerfaglig disciplin.

Systemanalysen er en flerfaglig disciplin og hver disciplin har betydning for en vurderingen af systemanalysemetoderne.

Der kan inddrages mange flere aspekter end de efterfølgende, men vi har begrænset os til dem vi mener, der er de væsentligste i en sammenligning af systemanalysemetoder.

Filosofi.

Systemanalyse inddrager først og fremmest filosofi, da systemtanken og menneskesynet spiller en helt centralt rolle. Dette gælder både i forhold til forståelse af produktet og processen. Metoderne må vurderes i forhold til et mekanistisk, et romantisk eller dialektisk menneskesyn og tilsvarende hård system, blød system og dialektisk systemtænkning⁶.

Vores udgangspunkt er at mennesket skal placeres centralt og der skal anlægges et dialektisk perspektiv.

Organisationsteori.

Da systemanalyse sker i organisationer, hvor der er et samspil mellem opgaver, strukturer, teknologi og aktører⁷, bliver organisationsteorien også central, såvel i forhold til selve systemanalyseprocessen som konsekvenserne af de ændringer der sker i såvel de direkte som de indirekte brugernes situation ved indførelse af et nyt/ændret system. Vi mener at elementet af brugerdeltagelse er væsentligt og uddyber det senere.

Beslutningsteori.

Beslutningsteorien bliver også aktuel at inddrage, idet systemanalysen kan opfattes som en del af en beslutningsproces. Beslutningsteorien kan bidrage med modeller for beslutningsprocessen, der enten kan opfattes som forklarings-modeller eller som bevidste styringsmodeller. I beslutningsteorien kan analysen opfattes som problemløsning og dermed en læreproces⁸ hvilket indebærer at pædagogik også får en væsentlig betydning. Perspektivet om læreprocessen finder vi væsentligt og vi uddyber det senere.

⁶ Dahlbom, Bo og Mathiassen, Lars: *Computer in context*

⁷ Leavitt, H.J.: *Applied organizational change in industry: Structural, technological and humanistic approaches*

⁸ Enderud, Harald: *Beslutninger i organisationer - i adfærdsteoretisk perspektiv.*

Matematik.

Matematikken bliver aktuel at inddrage, dels indirekte i form af dens formalistiske tankegang, dels mere direkte i form af blandt andet relationel algebra, mængdelære og ikke mindst til vurdering af kompleksitet og usikkerhed. Forståelsen af kompleksitet og usikkerhed ved systemanalyse finder vi væsentligt og uddyber det senere.

Historie.

De arbejdsområder hvor der anvendes edb-teknologi har ændret sig hen over tiden og det samme har systemudviklingsmetoderne. Hvor man tidligere betragtede edb-teknologien som middel til automatisering er perspektivet ændret til også at omfatte middel til kommunikation og vidensformidling. Dette kan ikke betragtes isoleret, men må ses i sammenhæng med den generelle samfundsudvikling, hvor der også er sket markante skift i måden at betragte arbejdssituationen på. Dokumentation fra systemanalysen kan opfattes som et historisk dokument. Betydningen af dokumentationen er væsentlig og uddybes senere.

Psykologi og antropologi.

Psykologi og antropologi er også væsentlige at inddrage, idet forståelsen af disse områder har betydningen for at se menneskelige reaktioner i et bredere perspektiv og specielt for at kunne arbejde mere bevidst blandt andet gennem selvreflektion.

2.3 Brugerdeltagelse

Relevansen af brugerdeltagelse i forbindelse med systemudvikling vælger vi at betragte som indiskutabel, og derfor vil vi ikke argumentere nærmere herfor i denne fremstilling.

I historisk perspektiv har brugerdeltagelsen gennemgået en udvikling fra de aller tidligste edb-systemer til dem, der udvikles i nutiden. Dette hænger sammen med flere forhold, for det første manglende erfaring med systemudvikling og dermed manglende erkendelse af brugerdeltagelsens vigtighed. Dernæst har brugere i dag højere status i organisationen end tidligere, hvilket også hænger sammen med den almindelige demokratiseringsproces. Endelig har opgavernes natur også haft betydning for den historiske udvikling. De opgaver, som først blev automatiseret, var som oftest fast definerede funktioner afviklet i batchmode med dermed følgende minimal brugerinteraktion. Lønberegning er et eksempel herpå. Senere tiders applikationer har indeholdt stigende grad af brugerinteraktion hen imod nutidens sagsbehandlingssystemer med relativt komplekse dialoger og stigende grad af fokus på brugergrænseflader.

Ændringsstrategier

Blandt brugerne vil der altid være modstand mod forandringer⁹ - det kendetegner enhver organisation. Modstanden kan bunde i frygt for ikke at slå til, frygt for at miste prestige eller i den yderste konsekvens sit job samt almindeligt ubehag ved tanken om, at få lavet om på velkendte rammer. For at imødegå denne modstand er der udviklet en model for anvendelse af ændringsstrategier.

Modellen fokuserer menneske- og organisationsopfattelse. Menneskesynet kan være X eller Y, hvor X repræsenterer den gammeldags bureaukratiske menneskeopfattelse, ifølge hvilken mennesket er dovent, uvilligt til at påtage sig ansvar og stimuleres kun af økonomisk gevinst. I Y-opfattelsen er mennesket derimod arbejdsomt, kreativt, villigt til at påtage sig ansvar og stimuleres af andet (og mere) end penge. Organisationsopfattelsen kan være enten harmoni eller konflikt. I førstnævnte drives samarbejdet af interessefællesskab og ægte vilje og evne til at opnå enighed om fælles mål og midler. Konfliktopfattelsen, som støttes af filosofferne Hegel og Sartre, opfatter ethvert fremskridt som et resultat af en konflikt, dvs. resultaterne nås gennem uenighed og konkurrence.

Det er vigtigt at understrege, at disse fire opfattelser, bør betragtes som yderpunkter på to skalaer. For hver kombination af menneske- og organisationsopfattelse findes der en ændringsstrategi¹⁰. For kombinationen X og harmoni anvendes den rene ekspertstrategi, X og konflikt håndteres af fler-ekspertstrategien også kaldet det sociotekniske design, sidstnævnte er indgående beskrevet af Mumford¹¹. Tilsammen kaldes disse to for ekspertstrategier.

⁹ Neergaard, Peter: *Planlægning af ændringer*, kapitel I.3

¹⁰ Neergaard, Peter: *Planlægning af ændringer*, kapitel I.3

¹¹Mumford & Ward: *Edb, system og menneske*, kapitel 9

De to sidste kaldes under et for repræsentationsstrategier, og kombinationen Y og harmoni kaldes simpelthen repræsentationsstrategien, idet der her er enighed om målene og derfor kan brugerne 'repræsenteres'. Det kan de ikke i kombinationen Y og konflikt, hvorfor alle må involveres på en eller anden måde. Denne strategi kaldes deltagelses- eller magtstrategien.

Den rene X-opfattelse hører fortiden til og derfor burde udviklingen forlængst have overflødiggjort anvendelsen af ekspertstrategier. At dette ingenlunde er tilfældet må tilskrives uvidenhed eller kortsigtede økonomiske betragtninger. Man kan således godt minimere selve udviklingsomkostningen ved at anvende en ekspertstrategi. Problemet er så bare, at man må forvente store efterfølgende omkostninger til rettelser af fej og almindeligt vedligehold. Neergaard har formuleret fænomenet i sin såkaldte K-lov¹², som lidt provokerende slår fast, at omkostningerne ved at indføre systemet er konstante uanset, hvilken strategi man vælger. De falder bare på forskellige tidspunkter.

I den ideelle verden er der således kun repræsentationsstrategierne, da disse begge repræsenterer det aktive menneskesyn (y-opfattelsen) og skelnen mellem repræsentationsstrategien og magtstrategien sker på baggrund af organisationsopfattelsen.

At vælge den rigtige strategi er dog ofte forbundet med visse vanskeligheder¹³. For at den enkelte strategi kan give succes skal projektet eller ændringen besidde bestemte karakteristika, og langt fra alle typer af projekter kan 'presses ned i skabelonen'. Dette betyder, at man må gå på kompromis med sin menneske- eller organisationsopfattelse - til skade for brugerne. Et eksempel: lad os forestille os et flyselskab, som skal have nyt reservationssystem til front-desk personalet. Menneskeopfattelsen er Y og organisationsopfattelsen er konflikt (det kunne være SAS). Eneste anvendelige strategi ifølge teorien er magtstrategien, som kræver deltagelse af alle brugere, og den vil i praksis efter al sandsynlighed afskrives af økonomiske, tidsmæssige og praktiske hensyn, og i stedet vil man vælge det socio-tekniske design og dermed gå på kompromis med sin menneskeopfattelse

Opsummerende kan man sige, at holdningerne til mennesker og organisation skal være samstemt med den valgte ændringsstrategi og dermed med brugerdeltagelsen. Hvis man vælger forkert, begrænser man på forhånd sine muligheder for succes. Dette valg er som regel en ledelsesbeslutning, og der kan gives mange eksempler på at metoden efterfølgende har fået skylden for det pauvre resultat; men sandheden er, at ingen metode kan reparere de skader, som en forkert valgt ændringsstrategi forårsager. Det mest almindelige er at vælge repræsentationsstrategien, idet mange opfatter den som en slags helgardering, uden at den er alt for dyr. Det, der ofte sker, hvis der er konflikter i brugerorganisationen, er at brugerrepræsentanterne ender i en slags ingenmandsland, fordi deres basisorganisation ikke længere betragter dem som sine egne, men snarere som edb-folk.

I de fleste virksomheder er konfliktopfattelsen den fremherskende¹⁴, og det betyder at metoden også skal måles på i hvor høj grad den er i stand til at håndtere konflikter mellem brugerne.

¹²Neergaard, Peter: *Planlægning af ændringer*, s. 107-8

¹³Neergaard, Peter: *Planlægning af ændringer*, kapitel II.1

¹⁴Dahlblom & Mathiassen: *Computers in context*, s. 111

Der findes andre måder at anskue samarbejdsformen med brugerne på. For eksempel anviser N.E. Andersen et al. en trinvis metode til valg af form¹⁵.

Med hensyn til at imødegå modstanden mod forandringer er det nok begrænset hvad metoden kan bidrage med, hvis den rigtige strategi er valgt; men man kan sige at den i sin form og værktøjer, skal være acceptabel for brugerne, således at den i sig selv ikke bidrager til yderligere modstand.

Brugere i projektorganisationen

Ved systemudvikling af et vist omfang anses projektorganisationen som oftest og med rette for at være den mest velegnede. Det er imidlertid langt fra altid, at den implementeres som anvist i lærebøgerne¹⁶. I mange større virksomheder og koncerner (Told*Skat, A.P. Møller, m.fl.) er det således, at projektlederen vælges fra brugerorganisationen eller man arbejder med 2 projektledere: en fra brugerorganisationen og en fra edb-leverandøren. Denne organisationsform vil ofte vise sig uhensigtsmæssig, da projektlederen fra brugerorganisationen ofte mangler viden og erfaring om projektarbejde, og vigtigere har begrænset viden om edb-systemers muligheder og begrænsninger. Hvis denne organisationsform skal lykkes, er det en afgørende forudsætning, at brugerne også er villige til at sætte sig ind i edb-folkenes "forretning" - det nytter ikke noget, at det kun er edb-folkene der skal være imødekommende.

Problemer ved brugerdeltagelse

Brugerinvolvering i analysearbejdet er på ingen måde ukompliceret. Parnas og Clements nævner blandt andet fire forhold, som kræver særlig opmærksomhed, idet de kan vanskeliggøre samarbejdet¹⁷:

- Brugere ved ikke hvad de vil have og/eller formulerer sig upræcist.
- Designændringer sent i forløbet kan ikke undgås, selv om kravene er kendte.
- Kravene ændrer sig over tiden pga. nye brugere og nye konflikter.
- Mennesker begår fejl.

Det må altså kræves, at metoden i videst muligt omfang er i stand til at håndtere disse vanskeligheder.

Med hensyn til brugernes uafklarethed vil en metode, som tillader eksperimenter, være at foretrække frem for en som ikke gør det. En måde at imødekomme de øvrige problemer på, vil være at man kører forløbet iterativt, således at fejl og ændrede krav kan samles op i senere gennemløb.

Et velkendt problem for alle metoder er: hvorledes præsenteres analysedokumentation i en form, således at brugeren kan forstå og dermed have forudsætninger for at godkende den? Eftersom alt andet bliver ligegyldigt, hvis forståelsen mangler, må dette kriterium blive det vægtigste i taksonomiens brugerdeltagelses-del.

¹⁵ N.E. Andersen et al., *Professionel systemudvikling*, s. 179 ff

¹⁶ Andreas Munk-Madsen: *Strategisk projektledelse*, kapitel VI

¹⁷ Parnas & Clements: *A rational design proces: how and why to fake it*, s. 346-47

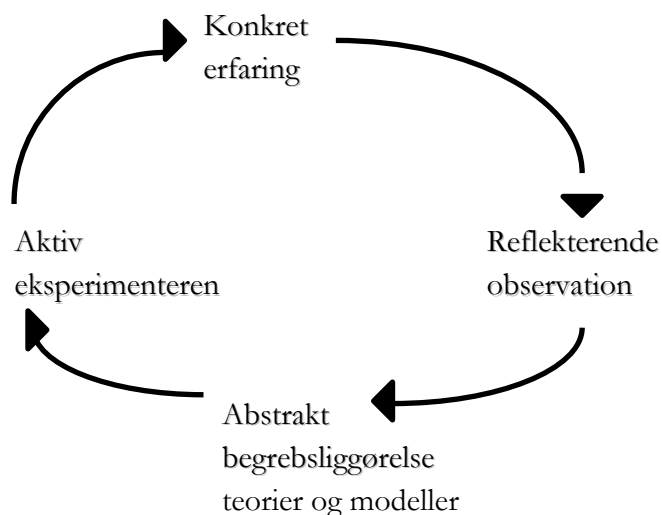
Det bliver således af stor betydning, at syntaksen for dokumentationen ikke lægger bånd på brugerens muligheder for at beskrive sin forretning, så systemfolk kan forstå den, og omvendt må systemfolkene ikke tvinge brugerne til at anvende beskrivelsesværktøjer, som de er usikre på.

2.4 Systemudvikling som en læreproces.

Erfaringspædagogik og problemløsning/systemudvikling.

Når man som systemudvikler læser lærebøger om pædagogik, bemærker man gang på gang den lighed, der eksisterer mellem pædagogiske teorier og teorier om problemløsning/systemudvikling. Det bliver specielt tydeligt indenfor den gren af pædagogikken, der går under betegnelsen erfaringspædagogik. Grundantagelsen i erfaringspædagogikken er, at læring er mere givende, hvis den tager udgangspunkt i og/eller inddrager de studerendes egne praktiske erfaringer. David Kolb har udviklet en læringsmodel (experiential learning: Relating to or derived from experience), hvori læring først og fremmest ses som en proces.

Kolb definerer læring som en proces, hvor viden bliver skabt gennem transformationen af erfaring. Det er umiddelbart klart, at en sådan definition må placere personen og det personlige element centralt i læringen. Kolb har illustreret det ved hjælp af sin såkaldte læringscirkel¹⁸:



I det følgende skal man altså være opmærksom på, at der kan sættes lighedstegn mellem læringsprocessen og systemudviklings- eller problemløsningsprocessen. Læring (problemløsning) kan tage udgangspunkt i et hvilket som helst af de fire punkter, men ofte er det sådan, at man starter med en konkret erfaring.

Konkrete erfaringer gøres, når man har kontakt med det, man kalder den virkelige verden. Systemudvikling og den problemløsning, som systemudviklingen forudsætter kræver en nøje kontakt med de organisationer og de mennesker, som skal benytte systemet (se også brugerdeltagelse).

¹⁸ Kolb, D.A.: *Experiential learning: Experience as the source of learning and development*

På basis af de indhøstede erfaringer foretager man (ofte ubevidst) en række *observerende refleksioner*. Man overvejer, hvad problemet består i, man danner hypoteser, der forklarer problemet, og man tester hypoteserne gennem flere observationer. Et åbent spørgsmål i denne sammenhæng er, om det er muligt at uddanne bedre systemudviklere, hvis man kan finde frem til en måde at undervise i observerende refleksion.

Det er nu muligt for den studerende (systemudvikleren) at skabe en sammenhæng med eksisterende *abstrakte teorier og modeller*, eller man bliver måske i stand til selv at udlede sådanne. Herved opnås en bedre (dybere og/eller bredere) forståelse for problemet, som kan danne grundlaget for næste skridt: *aktiv eksperimenteren*.

Et andet interessant forhold, som kan udledes af denne model, er at læring (systemudvikling) slet ikke kan forstås uafhængigt af den lærende (problemløseren/systemudvikleren). Hele processen og oplevelsen af den er stærkt subjektivt. Det bliver således også klart, at hver enkelt studerende (systemudvikler) har sin egen helt specielle måde at arbejde og løse problemer på - den personlige stil. Et vigtigt element i undervisningen bliver derfor at hjælpe hver enkelt studerende med at finde frem til sin optimale arbejdsform, men også at forbedre denne, ved at blive bevidst om sine stærke og svage sider. På tilsvarende måde må uddannelse af systemudviklere tage udgangspunkt i, at den enkelte systemudvikler skal lære at finde frem til sin egen personlige stil.

Det er let at se parallellen mellem Kolbs læringscirkel og f.eks. Boehms spiralmodel.

2.5 Komplexitet og usikkerhed.

Systemudvikling er en overordentlig kompliceret proces, som således stiller krav til metoden om kunne håndtere kompleksiteten, uden at gå på kompromis med de øvrige krav. Hermed tænkes på udfordringen, der ligger i samspillet mellem situationer, arbejdsformer og betingelser. På den ene side præger den anvendte arbejdsform de situationer, der opstår og på den anden side er mange situationer en god anledning til at påvirke arbejdsformen.

Situationerne kan veksle mellem rutine, problemløsning og problemdefinering. Metoden skal i problemdefineringsituationen kunne afdække problemet i en atmosfære af stor usikkerhed. Vi skal i problemløsning kunne eksperimentere os frem til den rette arbejdsform, der bringer os i en rutinepræget situation. Giver metoden ikke tilstrækkelig frihed og fleksibilitet og indeholder metoden ikke arbejdsformer, der kan støtte i ovenstående situationer, vil vi ikke let kunne se fejl og mangler, men kun arbejde ud fra standarder og normer¹⁹.

Et andet aspekt er, hvordan design håndteres i systemudviklingen. Det der gør det svært er, at så mange faktorer virker ind på processen og at de gensidig påvirker hverandre i et turbulent miljø. Foruden at forstå edb-teknikken skal man kunne indfri brugernes krav og ønsker (som ikke altid er entydige). Mathiassens og Stages svar herpå er princippet om begrænset reduktion.

Princippet slår fast, at der ikke er nogle genveje i systemudviklingen og princippet er, at den kompleksitet, der er i systemerne skal ses i sammenhæng med den tilgængelige relevante information. Hvis man analytisk abstraherer informationsmængden ned, bliver der brug for eksperimenter, som så kaster ny kompleksitet af sig. Systemudvikleren bør altså ikke falde i hverken den analytiske eller den eksperimentelle grøft; men benytte analytiske tiltag i forbindelse med eksperimenter og foretage eksperimenter, som man bearbejder analytisk.

Parnas og Clements ser problemet på en anden måde, idet systemudvikling i praksis ikke kan udføres som en rationel proces, men indeholder elementer af analyse, design og implementering. Tidligt i forløbet (hvor vi har begrænset indsigt) introduceres fejl, som skal rettes senere. Brugere/køberne af systemer ved ikke præcist hvad de vil have og efterhånden, som de opdager det, ændres deres krav. Parnas og Clements råder til, ikke at lade som om vi kan gøre det sekventielt, men i stedet for organisere resultatet, som om det var fremkommet i denne rækkefølge²⁰.

I bedømmelse af en systemudviklingsmetode vil det være særdeles relevant at vurdere, hvordan den håndterer kompleksitet samt hvordan den håndterer usikre informationer. Komplexiteten i en given situation kan dels stamme fra selve problemet, dels i håndteringen af mængden af de nødvendige informationer til løsning af opgaven, mens usikkerheden betinges af informationens pålidelighed og tilgængelighed.

¹⁹ N.E. Andersen et al: *Professionel Systemudvikling*

²⁰ Parnas, D.L. og Clements, P.C.: *A rational design process: How and why to fake it.*

For at kunne bedømme metoderne i forhold til kompleksitet og usikkerhed, vil vi derfor se på muligheden for, at balancere mellem analytiske tiltag og eksperimenter. Ligeledes vil vi bedømme metodens evne til at håndtere forandring.

2.6 Dokumentation.

Dokumentationen der udarbejdes i forbindelse med systemanalyse tjener efter vores opfattelse flere formål.

1. Beslutningsgrundlag for et projekts videre forløb. Dokumentation kan tjene som en kontrakt.
2. Arbejdsgrundlag for resten af projektføreløbet.
3. Arbejdsgrundlag for senere vedligeholdelse / ændring af systemet.
4. Arbejdsredskab til at fastholde ideer, synspunkter og aftaler for deltagerne selv under selve processen.
5. Erfaringsgrundlag for nye deltager der "springer" på projektet.
6. Arbejdsredskab til projektledelsen, så denne kan følge et projekts fremdrift og dermed styre projektet.
7. Erfaringsgrundlag for senere projektarbejder.

For formålene 1, 2 og 3 er dokumentationen meget produktorienteret og det kan svare sig at den er opbygget struktureret, som om den var fremkommet ved en rationel proces²¹. Det er samtidig af stor betydning at den er så kort som mulig, tilstrækkelig og korrekt. Samtidig skal den være let at tilrette ved senere revision.

For formål 2 er det af stor betydning, at der er en tydelig og enkel overgang til resten af projektet. Der bør i hele projektarbejdet kunne anvendes samme terminologi og dokumentationsform.

For formål 3 er det af stor betydning, at dokumentationen kan genanvendes når der sker ændringer i systemet eller der opstår en fejlsituation.

For såvel formål 2, 3 og 5 gælder det, at dokumentationen skal gøre det muligt for andre end de der har deltaget i analysearbejdet, at sætte sig ind i problemområdet. Alle valg, der er foretaget skal fremgå af dokumentationen. Dokumentation kan selvfølgelig ikke erstatte selve deltagelsen i det forudgående analysearbejde, men være et væsentligt bidrag til at forstå forløbet.

For formålet 4 er det vigtigt, at dokumentationsværktøjerne giver mulighed for kreativitet. Dokumentation skal kunne udvikles løbende sammen med processen selv om processen ikke er en rationel proces.

For formålet 6 er det vigtigt, at dokumentationen beskriver opfyldelsen af milepæle og faktorer, der kan have indflydelse på projektets ressourceforbrug og tidsplaner.

For formålet 5 og 7 er det vigtigt, at selve procesforløbet kan dokumenteres på en måde, der giver andre mulighed for at sætte sig ind i projektets arbejdsmetode og handlingsforløb. Denne del af dokumentationen er i praksis ofte en mangelvare, men utrolig vigtigt hvis nye deltagere hurtigt skal kunne involveres og hvis man skal kunne høste erfaringsmateriale til senere projektarbejder. Værdien af denne dokumentation ved vi er meget nyttig fra vores erfaringer med projektdagbøger.

Dokumentationen tjener, som beskrevet i det ovenstående, mange formål. Da formen af dokumentationen vil afspejles af formålet, vil den ikke kunne defineres entydig. Der bliver tale om flere former for dokumentation, der på den ene side må være formel som f.eks. et kontraktgrundlag og på den anden side mindre formel som f.eks. beskrivelsen af hvordan processen har forløbet i forbindelse med frembringelsen af kontraktgrundlaget.

²¹ Parnas, D.L. og Clements, P.C.: *A rational design process: How and why to fake it.*

2.7 Opstilling af taksonomi

Som tidligere nævnt finder vi, at det er meget vigtigt, at den, der stiller sig den opgave at opstille en taksonomi for et eller andet, også samtidig stiller sig den opgave at beskrive sine egne grundholdninger, fordi disse ikke kan undgå at påvirke de kriterier, man vælger at prioritere i sin taksonomi. Vi har derfor i opgavens indledning redegjort for vore grundholdninger til systemudvikling.

I forbindelse med denne opgave har vi set på nogle forskellige taksonomier med henblik på at finde ud af, om der eventuelt var én, vi umiddelbart kunne benytte. Herved opdagede vi, at det er typisk, at taksonomiens ophavsmand glemmer at redegøre for sine grundholdninger. Det gør som nævnt, at det bliver vanskeligt at afgøre, om taksonomien kan bruges i andre sammenhænge.

Vi blev også opmærksomme på, at taksonomier kan have mange forskellige formål. Således viste det sig, at en taksonomi for software udviklingsmetoder af Bruce Blum²² havde som sit formål at finde en passende udviklingsmetode i en given situation.

En anden type taksonomi anvendes af Shable og Cohen i forbindelse med en undersøgelse af fordele og ulemper ved to forskellige objektorienterede udviklingsmetoder i en bestemt situation.²³ Formålet med denne taksonomi er at opstille et system af målekriterier, som kan bruges til at afgøre, hvilken af flere metoder, der giver det bedste resultat i en given situation. Selv om vi er meget skeptiske overfor denne type af taksonomier, er den pågældende undersøgelse gennemført på en kompetent måde. Man har anvendt en række neutrale kriterier fra et skrift af Chidamber og Kermerer²⁴, som man har suppleret med kriterier, som ville formodes at tilgodese enten den ene eller den anden af de to undersøgte metoder.

En tredje type taksonomi er eksemplificeret ved en artikel af Monarchi og Puhr. Artiklen forsøger at udvikle en ramme for evaluering og sammenligning af aktuel forskning indenfor objektorienteret analyse og design. Rammen vil samtidig kunne danne basis for en samling af forskellige forskningsresultater til en sammenhængende og komplet metode indenfor objektorienteret analyse og design. Der er meget litteratur tilgængeligt om objektorienteret systemudvikling; men der er kun begrænset enighed eller standardisering imellem de forskellige nye udviklingsteknikker. Det kan skyldes, at man netop mangler "mekanismer" eller rammer til syntetisering eller evaluering af det udførte forskningsarbejde. Desværre når Monarchi og Puhr ikke frem til dette mål i deres artikel - i stedet får de opstillet en slags bruttoliste over summen af de faciliteter, der forekommer i de over

²² Blum, Bruce I: *A Taxonomy of Software Development Methods*. Comm. of the ACM Nov. 1994 p. 82.

²³ Shable, Robert C. & Cohen, Samuel S.: *The Object-Oriented Brewery: A Comparison of Two Object-Oriented Development Methods*. ACM SIGSOFT Software Engineering Notes Apr. 1993 p. 60.

²⁴ Chidamber, S. R. & Kermerer, C. F.: *Towards a Metrics Suite for Object Oriented Design*. SIGPLAN Notices, Oct. 1991 p. 197.

20 objektorienterede metoder, der indgår i undersøgelsen. I en tabel er der krydset af i en kolonne for hver metode, hvilke faciliteter den pågældende metode rummer. Krydserne er så talt sammen og antallet bruges som et for kvaliteten.²⁵

Også taksonomier af denne art er vi yderst skeptiske overfor. Det siger nemlig ikke noget om metodens kvalitet, at der kan sættes et eller andet antal krydser i en bestemt kolonne i en tabel. Det eneste sådanne taksonomier kan bruges til, er til at give et overblik over mange metoder.

Ud fra denne lille oversigt over taksonomier, kan man konstatere, at det vi har brug for til vort formål, er en taksonomi af den anden type: en taksonomi, der kan bruges til at afdække fordele og ulemper ved forskellige metoder. Desværre har vi ikke fundet en taksonomi af denne type, der netop passer til vores formål. Vi vil derfor i det følgende opstille vor egen taksonomi baseret på de 5 kriterier, som vi allerede har fremhævet som dem, vi anser for de vigtigste under hensyntagen til vores grundholdning til systemudvikling.

Herefter følger en gennemgang af de 5 kriterier med fokus på de aspekter, som vi vil vurdere i de to metoder.

Flerfaglig disciplin

Vi har argumenteret for, at man opfatter systemudvikling som en flerfaglig disciplin. For taksonomien betyder det, at systemudviklingsmetoden skal rumme nogle anvisninger eller opfordringer til systemudvikleren om at inddrage forskellige aspekter i forbindelse med systemudviklingen.

Brugerdeltagelse

Der kan peges på flere mål med brugerdeltagelse i systemudviklingen. For det første kan brugerens deltagelse i processen være nødvendig, da det er brugeren, der sidder inde med viden om hvordan systemet skal anvendes. Desuden kan brugeren bidrage til opbygning af brugergrænseflader, brugervejledninger samt bidrage til en realistisk test. Udover det konkrete bidrag giver brugerinddragelsen også mulighed for at lære systemet at kende parallelt med udviklingen. Disse forhold bidrager alle til en øget accept af det nye system og er i god overensstemmelse med et demokratisk syn på brugersamarbejdet.

Systemudviklingsmetoden skal rumme nogle anvisninger eller anbefalinger om at inddrage brugerne i systemudviklingsprocessen.

Læreproces

Systemudviklingen kan betragtes som en læreproces både for systemudvikleren og for den deltagende bruger. Systemudviklingsmetoden skal være iterativ og gerne baseret på en spiralmodel.

²⁵ Monarchi & Puhr: *A Research Typology for Object Oriented Analysis and Design*. Comm. of the ACM Sept. 1992

Der skal lægges vægt på, at der løbende sker akkumulation af viden baseret på de erfaringer, man har opnået i hvert gennemløb eller hver iteration.

Kompleksitet og usikkerhed

Kompleksitet og deraf følgende usikkerhed er en iboende egenskab ved de systemer, der fremstilles i dag. For at kunne overvinde denne iboende kompleksitet, skal systemudvikleren kunne veksle mellem en analytisk og en eksperimentel arbejdsform. På tilsvarende måde skal systemudvikleren kunne veksle mellem statiske og dynamiske synsvinkler på systemet. Systemudviklingsmetoden skal give systemudvikleren disse muligheder.

Dokumentation

Det drejer sig om at få fremstillet den rigtige dokumentation. I praksis betyder det, at systemudvikleren skal kunne anvende en bred vifte af værktøjer i forbindelse med dokumentationen. Systemudviklingsmetoden skal være åben og uformaliseret på dette punkt.

3 Kort præsentation af systemudviklingsmetoder

3.1 Struktureret analyse

Struktureret analyse er en del af de strukturerede metoder, som er betegnelsen for et antal metoder, som er udviklet over en fælles idé: den strukturerede top-down nedbrydning.

Den følgende beskrivelse af de strukturerede metoders historiske udvikling er baseret på James A. Kowals "Analyzing Systems".²⁶

Udviklingen af de strukturerede metoder tog sin begyndelse i midten af 1960'erne. Det startede med struktureret programmering, som var et revolutionerende paradigme på sin tid. Paradigmet er baseret på en teori fremsat af C. Bohm og G. Jacopini²⁷ og videreudviklet af Dijkstra²⁸.

Teorien går ud på, at enhver programspecifikation kan nedbrydes til tre elementære konstruktionselementer:

- sekvenser
- iterationer
- selektioner

I starten af 1970'erne blev struktureret design tilføjet. Med struktureret design følger en række diagrammeringsværktøjer, som er skabt med henblik på fremstilling af strukturerede programmer.²⁹

Før struktureret design dokumenteredes problemløsninger ved hjælp af logiske rutediagrammer (flow-charts), beslutningstabeller eller lignende. Målet med struktureret design er at frembringe løsningsmodeller med følgende egenskaber:

- Designet skal afspejle virkeligheden, dvs. problemets struktur skal afspejles i løsningens struktur.
- Designet skal sikre et brugervenligt system.
- Designet skal være enkelt at modificere (vedligeholde), så systemet til stadighed kan leve op til omgivelsernes krav.
- Designet skal sikre, at ressourcerne udnyttes effektivt.
- Designet skal sikre en høj grad af pålidelighed (en stabil driftssituation).

²⁶ Kowal, James A.: "Analyzing Systems"; Prentice-Hall 1988.

²⁷ Bohm, C & G. Jacopini: "Flow Diagrams, Turing Machines and Languages with Only Two Formation Rules"; Comm. of the ACM No. 5 1966, p. 13 - 25.

²⁸ Dijkstra, E.W.: "Cooperating Sequential Processes" i F. Genuys (ed.): "Programming Languages"; Academic Press, 1968.

²⁹ Yourdon, E. & L. L. Constantine: "Structured Design: Fundamentals of a Discipline of Computer Program and Systems Design"; Prentice-Hall 1978.

Integreret systemanalyse er en analysemetode fra slutningen af 1980'erne. Integreret systemanalyse bygger på Tom DeMarco's strukturerede analysemetode til administrative systemer fra 1978.³⁰

DeMarco introducerede begrebet logisk model. En logisk model er en model, der er struktureret og navngivet ud fra brugerens funktioner og ikke ud fra de tekniske løsninger.

Videreudviklingen af den traditionelle strukturerede analyse blev foretaget af McMenamin og Palmer,³¹ som har indført begreberne essentiel systemanalyse og essentielle modeller, idet de bygger videre på DeMarcos tankegang og uddyber retningslinierne for, hvorledes intentionerne kan nås.

De vigtigste forskelle i forhold til den tidlige strukturerede analyse er:

- Hændelsesopdeling af modeller og systemer.
- Entitetsopdeling af data via informationsanalyse.
- Tilstandsdiagrammer til beskrivelse af entiteters tilstande og tilstandsændringer.
- En alternativ strategi, der minimerer arbejdet med beskrivelse af det eksisterende system, og letter projektstyringen.

Struktureret analyse gør op med de tidligere anvendte verbale former for beskrivelser, og indfører data-flow diagrammer til beskrivelse af systemets funktioner. Problemet med de verbale beskrivelser er det naturlige sprogs tvetydige karakter. I stedet tilstræbes den højere grad af formalisme, som et standardiseret diagrammeringsværktøj giver.

Det er vigtigt, at man forstår det gamle og det nye systems virkninger på organisationen, inden implementering af det nye system iværksættes. Denne specifikation er derfor det vigtigste mål for analysen. Det er således et afledt mål, at design og implementering ikke må komme til at lide under forvirring og ubeslutsomhed, hvad angår systemets virkemåde.

Et andet afledt mål er, at man forpligter sig til i den strukturerede analyse at skaffe sig al viden om et system, inden programmering påbegyndes, og at man forpligter sig til at dokumentere denne viden. Man erkender, at denne indsamling og dokumentation af viden tager lang tid; men det antages, at denne investering betaler sig i form af markant faldende vedligeholdelsesomkostninger.

Et centralt punkt i den strukturerede analyse er at finde en passende balance mellem modeller, der beskriver funktioner, og modeller, der beskriver data.

I den oprindelige strukturerede analyse tog man udgangspunkt i det eksisterende systems funktioner, der ofte var manuelle. Senere blev man klar over, at virksomhedernes informationer

³⁰ DeMarco, Tom: "Structured Analysis and System Specification"; Prentice-Hall, 1978.

³¹ McMenamin, Stephen M. & John F. Palmer: "Essential Systems Analysis"; Yourdon Press, 1984.

repræsenterer et væsentligt aktiv, både som et selvstændigt produkt og som kilde til udformning af forbedrede systemer.

Den moderne strukturerede analyse drager nytte af begge muligheder, idet metoden lægger vægt på, at systemudviklerne udmønter deres forståelse af informationsbehandlingen i en struktur, der følger problemet fremfor den hidtil anvendte teknologi.

Datagrundlaget spiller således en integreret rolle med det ønskede funktionsomfang, der kan fordre nye informationer lagret eller behandlet i det nye system.

Metoderne beskæftiger sig meget med værktøjsanvendelse og dette har affødt et udbud af edb-baserede værktøjer (CASE) ligesom værktøjsbenyttelsen fylder godt i de fleste bøger om struktureret analyse.

Endvidere spiller psykologi og organisation en vis rolle, idet man er meget bevidst om aktørernes betydning for processen.

Vedligeholdelsen er udnævnt til at være den enkeltfaktor, som udgør den største årsag til softwarekrisen, idet vedligeholdelse beslaglægger flere og flere ressourcer, som kunne have været udnyttet til konstruktion af nye profitable systemer.

Struktureret analyse prøver bevidst at skabe en situation, som gør det nemmere for systemudviklerne at vedligeholde dokumentationen, og samtidig gør organisationen uafhængig af den enkelte systemudvikler.

I den oprindelige strukturerede analyse var der tale om, at man beskrev fire modeller af systemet, der skulle forandres.

Gammel fysisk model og gammel logisk model er milepæle i forståelsen af, hvorledes man teknologisk og organisatorisk havde løst opgaven i det gamle system, og de viser samtidig, at udvikleren har genopdaget det gamle systems indre logik.

Den nye logiske model viser først og fremmest krav og ønsker til det nye system. Det nye systems logik skal til slut i den nye fysiske model ikklædes en fysisk løsning som tilfredsstillende aktørernes ønsker.

McMenamin og Palmer indførte essentiel modellering og hændelsesmodellering, hvorved man frigjorde sig fra teknologiske bindinger og fokuserede på de essentielle aktiviteter, der er reaktionen på en hændelse. De data, der lagres som resultat af hændelsen, kaldes hændelsesresultatet, og de indgår derfor også i informationsmodellen.

Essentiel modellering er et forsøg på at formindske arbejdsbyrden, idet den essentielle model gøres teknologisk neutral, d.v.s. at det væsentlige i et system ikke forældes eller ikke skal omdokumenteres på grund af ydre omstændigheder.

3.2 Objektorienteret Analyse.

Den beskrevne metode er hentet fra bogen *Objektorienteret analyse* af Lars Mathiassen, Andreas Munk-Madsen, Peter Axel Nielsen, Jan Stage³².

OOA opererer med to begreber, anvendelsesområdet og problemområdet.³³

“Et edb-systems anvendelsesområde omfatter edb-systemet, dets brugere og deres måde at anvende edb-systemet på i deres arbejde. Anvendelsesområdet er altså en del af brugerorganisationen. Et edb-systems problemområde betegnet, som edb-systemet handler om. Det omfatter den del af virkeligheden, som edb-systemet bruges til at administrere, styre eller overvåge. Problemområdet er det, som giver brugerorganisationens arbejde mening.”

Objektsystemet beskrives i OOA i en model, som skal kunne anvendes gennem hele systemudviklingsforløbet. En vigtig egenskab ved modellen er desuden, at den skal fungere som fælles systemopfattelse for såvel brugere som systemudviklere.

Objektmodellen bliver et udtryk for brugerens systemopfattelse og bliver herved også rammen for en kravspecifikation, hvor de funktionelle og formmæssige krav udtrykker hvordan edb-systemet relaterer sig til brugerens arbejdsopgaver i anvendelsesområdet.

For at sikre en dækkende aftale om en ønsket edb-løsning gennemføres et systemvalg, der udtrykker relevanskriterierne for modellen og kravspecifikationen. Systemvalget resulterer i en kortfattet beskrivelse af en løsning i naturligt sprog, kaldet systemdefinition.

For at sikre at systemdefinitionen er velformuleret og har de relevante elementer har man indført et såkaldt BATOFF-kriterie. Iterativt sammenlignes og reformuleres definitionen med kriterierne indtil kriterierne er opfyldt. Det at vælge en systemdefinition er en vanskelig og vigtig proces, som bør foregå i fuld åbenhed og med forhandling mellem de involverede parter. Systemudviklerne kan komme med argumenter, men det er brugerne, der vælger. For at sikre at de ofte modsatte krav er relevante og rationelle bør argumentationen baseres på pålidelig information og dokumenterede erfaringer.

Et grundprincip i OOA, er at tage udgangspunkt i brugernes situation. For at dette skal blive reelt tilbyder metoden nogle værktøjer, der skal understøtte udviklere og brugeres fælles kommunikation. Værktøjerne er ikke specielt formaliserede i de områder, hvor der involveres brugere.

Her kan f.eks. nævnes rige billeder, der kan benyttes til at sætte fokus på de forandringer, der vil komme.

³² Mathiassen L., Munk-Madsen, A., Nielsen, P.A og Stage, J: *Objektorienteret analyse*

³³ Mathiassen L., Munk-Madsen, A., Nielsen, P.A og Stage, J: *Objektorienteret analyse*, s. 7

En anden grundlæggende egenskab ved OOA er, at specifikationen direkte afspejler den begrebsmæssige model. Herved undgås misforståelser under omformning fra et beskrivelseselement til et andet. En yderligere egenskab ved OOA er, modellen direkte kan anvendes til design og implementering i form af objektorienteret design, objektorienteret brugergrænseflade og objektorienteret programmering.

Metoden anbefaler at benytte eksperimenter til at skabe nye ideer. Metoden anbefaler kortfattet dokumentation med vægt på overblik frem for detaljer.

En afgørende aktivitet i OOA er at få lavet en velfungerende model af objektsystemet. Man skal her have dannet klasser af objekter med samme egenskaber og adfærdsmønstre. Man skal have beskrevet de enkelt objekter og deres egenskaber samt de hændelser som involverer objekterne. Denne modellering skal gerne være bæredygtig for både brugere og udviklere og gerne kunne udvides/ændres uden at man skal starte helt forfra. Metoden bygger på stor grad af genbrug.

En markant egenskab ved metoden er at den ikke er faseopdelt i traditionel forstand, idet analyse, design og implementering sker under hele systemudviklingen. At der tales om at være i analysefasen er udtryk for, at man er i et stadie, hvor der primært laves analysearbejde. Overgangen mellem stadierne bliver på den baggrund til et styringsværktøj for projektstyringen, f.eks. implementeret som referencelinier. Resultatet af analysesarbejdet er et analysedokument, men det er ikke udtryk for at analysen er slut: Det betyder blot at der på det tidspunkt er tilstrækkeligt grundlag til at træffe beslutninger om det videre udviklingsforløb. Skulle det vise sig nødvendigt kan man blive nødsaget til at vende tilbage. Modellen indeholder således også iterative elementer.

4. Sammenligning.

I dette kapitel vil vi sammenligne Struktureret Analyse med Objektorienteret Analyse ud fra de kriterier vi har opstillet i de foregående kapitler.

Først foretager vi en detaljeret sammenligning indenfor hver af de fem hovedområder, der er vores udgangspunkt. Derefter foretager vi en sammenfattende konklusion, for til sidst at opstille et forandringsbehov. Forandringsbehovet danner udgangspunkt for vores sidste kapitel, hvor vi vil opstille en handlingsplan.

4.1 Sammenlign SA og OOA.

Dette afsnit indeholder en detaljeret sammenligning inden for hvert af de opstillede hovedområder, svarende til vores taksonomi.

Flerfaglig disciplin.

Struktureret analyse bygger på en mekanistisk og rationel systemopfattelse, hvilket ikke udtrykkes direkte, men indirekte gennem hele metodens opbygning. Hermed bliver metoden også baseret på et statisk verdensbillede, hvor systemer ikke forandrer sig grundlæggende, men kun i den teknologiske implementering.

OOA bygger på et dialektisk syn, indirekte udtrykt gennem en række anbefalinger, såsom tolkningsmulighederne i rige billeder, dialogen og argumentationen i systemvalg. Metoden baserer sig på et verdensbillede, der er dynamisk, hvor både organisation, teknologi og omgivelser ændres.

Beslutningsstrukturen i struktureret analyse, bliver overordnet baseret på fasernes slutdokumenter. Metoden tager ikke højde for, at beslutninger ikke kun er rationelle, men at der kan indgå subjektive elementer.³⁴

Beslutningsstrukturen i OOA baserer sig eksplicit udtrykt på valg truffet i fuld åbenhed blandt deltagerne. Metoden anviser dog ikke hvordan overordnede beslutninger i brugerorganisationen træffes.

Struktureret Analyse er formalistisk og bygger på standarder, hvilket ses af de værktøjer, der anbefales såsom dataflow, E/R-diagram og strukturdiagrammer³⁵.

³⁴ Enderud H, Borum F: *Konflikter i organisationen-belyst ved systemarbejde*.

OOA er ikke på samme måde formalistisk, men slår til lyd for at skabe nye ideer. “Brug forbilleder, metaforer og eksperimenter til at skabe nye designideer.” Gamle velafprøvede ideer kan måske fremfindes ved benytte forbilleder, metaforer og eksperimenter.

I OOA tilbydes en teknik til beskrivelse af situationer og problemområder i form af rige billeder³⁶. De rige billeder kan skabe debat mellem parterne ligesom de kan skabe overblik. Værktøjet giver mulighed for flere fortolkninger og associationer. Dette giver mulighed for fokus på stabilitet eller forandring, ligesom forskellige enheder kan optræde: mennesker, fysiske objekter, steder, organisationer, roller og opgaver.

Struktureret Analyse forholder sig ikke til at systemanalysen påvirker organisationen, mens den gennemføres. Der gøres dog opmærksom på, at der er en række psykologiske problemer, der skal tages højde for.

I OOA integreres systemanalytikerens mere eller mindre i organisationen, under eksperimenter, diskussioner mm.. Man er opmærksom på psykologien i brugersamarbejdet bl. a. i forbindelse med udformning af en fælles begrebsmodel.

På baggrund af ovenstående fremgår det, at begge metoder inddrager flerfaglige discipliner, idet det dog mest udtrykkes implicit.

Med udgangspunkt i metodernes grundsyn, bliver OOA mest åben overfor inddragelse af andre discipliner, som følge af det dialektiske udgangspunkt. De mangler man finder i OOA vedrørende f.eks. projektstyring og kvalitetsstyring kan netop hentes fra andre discipliner.

³⁵ Yourdon, Edward: *Modern Structured Analysis*

³⁶ Mathiassen L., Munk-Madsen, A., Nielsen, P.A og Stage, J: *Objektorienteret analyse*

Brugerdeltagelse.

Følgende spørgsmål vil blive vurderet mht. analysemetodernes egnethed i forhold til brugerdeltagelse:

- lægges der op til brugerdeltagelse generelt
- produceres en for brugerne forståelig dokumentation
- i hvor høj grad er metoden iterativ
- tillades eksperimenter
- kan konfliktopfattelsen håndteres

I OOA anbefales følgende grundprincip: "Tag udgangspunkt i brugernes situation, men vær kritisk over for opgaveformuleringen³⁷." Desuden er der formuleret et såkaldt BATOFF-kriterie til vurdering af systemdefinitionen. I flere af disse kriterier indgår brugerforhold implicit eller eksplicit.

"Når vi skal sætte os ind i situationen for et udviklingsprojekt, skal vi forsøge at forstå brugernes situation på en righoldig og indlevet måde."

I OOA beskrives vigtigheden af brugerdeltagelse gennem fokusering på analysedokumentet, som en præcisering og detaljering af systemdefinitionen. Dokumentet indeholder hovedresultaterne af forhandlingerne med kunden og brugerne. Brugernes interesse består primært i at fastlægge funktionelle og brugsmæssige krav til edb-systemet. At man mener brugerdeltagelsen seriøst ses måske tydeligst i udsagnet: "Lad brugerne vælge edb-løsning ud fra forskellige alternative systemdefinitioner." Valget foregår som en åben og struktureret forhandling.

I SA er der kun få konkrete anvisninger på, hvorledes samarbejdet med brugerne skal foregå. Dog lægges der ikke skjul på vigtigheden af samarbejdet³⁸, og rollerne fordeles groft sagt således, at brugerne skal begrænse sig til den logiske side af sagen, dvs. de bør f.eks. ikke beskæftige sig med valg af hardware og værktøjer. Det må således forudsættes, at metodens ophavsmænd mener, at brugerne er i stand til at forstå f.eks. dataflow- og ER-diagrammer.

Erfaringerne viser, at dette ofte ikke er tilfældet. Især giver ER-diagrammerne anledning til forståelsesmæssige problemer hos brugerne; men på det punkt har metoderne ikke meget at lade hinanden høre. Også OOA producerer vanskeligt tilgængelige diagrammer som f.eks. struktur- og adfærdsdiagrammer.

Med hensyn til brugervenlighed af dokumentationen har OOA dog to klare fordele i forhold til SA. For det første den kortfattede systemdefinition baseret på det såkaldte BATOFF-kriterium, som inden for en række foruddefinerede områder klart fortæller, hvad det kommende system drejer sig om. Selv om formalisme ikke nødvendigvis er noget at stræbe efter, er det en klar svaghed ved SA, at metoden ikke opererer med noget tilsvarende, idet man risikerer at skulle læse

³⁷ Mathiassen L., Munk-Madsen, A., Nielsen, P.A og Stage, J: *Objektorienteret analyse*

³⁸ DeMarco, Tom: *Structured analysis and system specification*, s. 14-15

hele dokumentationen igennem for blot at få en fornemmelse af helt essentielle forhold omkring analysen.

Den anden fordel er de rige billeder, som OOA opererer med. Dette værktøj giver brugerne mulighed for på fabulerende og ikke altid entydig vis at beskrive deres opfattelse af verden. På det punkt er SA meget formalistisk, hvilket klart er et handicap.

SA er pr. definition ikke iterativ. Man afslutter en fase, før man går i gang med den næste, og det bliver uhyre besværligt at håndtere fejl eller rettelser i senere faser. Qua sit rationelle ideal lægger metoden ikke op til det. Ved overgang til OOA lægges der implicit op til iterationer f.eks. ved opfordringer til gennemførelse af eksperimenter undervejs i forløbet, men det fremgår ikke eksplicit, hvor disse er placeret i forløbet

SA opererer ikke med eksperimenter, hvilket igen er en logisk følge af den snævre binding til det rationelle ideal. Hos OOA er eksperimenter derimod en integreret del af metoden, og den imødekommer således Parnas & Clements synspunkt om, at brugere ofte er uafklarede med hensyn til, hvad de vil have³⁹.

SA forudsætter, at den perfekte løsning eksisterer og at den kan findes ved korrekt anvendelse af metoden, igen det rationelle ideal. At dette skulle kunne lade sig gøre, forudsætter at man opererer i en harmonisk verden, dvs. at der kan opnås enighed om, at systemet er perfekt. Hermed er det indlysende, at SA ikke er særlig velegnet til en organisation, hvor konfliktopfattelsen hersker. Det er OOA muligvis heller ikke; men den har dog en række fordele i forhold til SA, idet den tillader iterationer, eksperimenter og rige billeder (med forskellige fortolkningsmuligheder), alt sammen værktøjer som kan hjælpe med til at opløse konflikter. Under SA vil uenighed på et vigtigt område i princippet sætte processen i stå. Under OOA vil ressourceforbruget til ændringer, der skyldes konflikter vedrørende krav, være mindre pga. det mindre omfang dokumentationen har i OOA og at dokumentationen kun omhandler det mest stabile i systemet.

³⁹ Parnas & Clements: *A rational design proces: how and why to fake it*, s. 346-47

Læreproces.

SA lægger ikke op til at se systemudvikling som en læreproces. Systemudvikling ses snarere som en konstruktionsproces, der kan standardiseres, rationaliseres og delvis automatiseres. Dette udelukker ikke en gammeldags opfattelse af læreprocessen, hvor eleven er et tomt kar, der skal fyldes, men er ikke overensstemmelse med et nutidigt syn på læreprocessen.

OOA lægger op til at der skal foregå en læreproces for såvel bruger som udvikler. Dette kommer til udtryk gennem brugen af eksperimenter, der dels giver erfaring til brugeren så behov kan afdækkes, dels giver mulighed opnåelse af systemforståelse for såvel brugere som systemudviklere. Dette svarer til brug af erfaringspædagogik.

Kompleksitet og usikkerhed.

Hovedbevægelsesretningen i analysen i SA er nedbrydning i delkomponenter. Denne opsplitning har til formål at mindske kompleksiteten af de enkelte problemer efter del og hersk princippet. Til gengæld opstår der et nyt problem med at holde styr på komponenterne og deres sammenhænge. SA tilbyder værktøjer til at holde styr på dette, såsom datadictionary, dataflowdiagrammer, E/R-diagrammer. Der findes edb-baserede udgaver af ligesom en totalløsning i form af CASE værktøjer (Computer Aided Software Engeneering).

Gennem sin analytiske metode, med indsamling af al relevant information søger SA, at bestemme og tage højde for usikkerheden. Der tages dog ikke højde for at der i praksis ofte er begrænsninger i tid og muligheder for indsamling og behandling af informationer. SA omtaler ikke eksperimenter, som en mulighed til afklaring af usikkerhed, men i praksis viser det sig at være til stor hjælp. Der er således opstået en afart af struktureret analyse med væsentlige elementer af prototyping, som beskrevet i Structured Rapid Prototyping⁴⁰, ligesom mange udviklere i praksis inddrager forskellige former for eksperimenter. Dette svarer helt til vores egne erfaringer.

OOA's reducerer kompleksitet gennem opbygning af objektstrukturer ved aggregering og associering samt gennem opbygning af klassestrukturer ved generalisering eller ved samling i klynger. Bevægelsesretningen ved dette er bottum-up. Metoden omtaler også dekomponering som modsætning til aggregering. Dette indebærer en top-down bevægelsesretning. Metoden anviser således begge retninger.

OOA erkender at der indgår en vis usikkerhed i analysen, da det ikke muligt eller hensigtsmæssigt at skaffe sig den komplette viden og at det ikke altid er muligt at definere kravene objektivt.

⁴⁰ Connell J.L., Shafer L.: *Structured Rapid Prototyping - An Evolutionary Approach to Software Development*

Problemet håndteres gennem brugen af eksperimenter og ved brug af mere uformelle beskrivelser, som rige billeder med fortolkningsmuligheder.

Metoden har anvisninger på mulige værktøjer, til beskrivelse af modellen, således at kompleksiteten kan rummes i et hierarki. Desuden kan man inddrage andre værktøjer, idet metoden ikke binder sig til bestemte værktøjer. CASE-værktøjer nævnes men anbefales ikke decideret, idet de betragtes som begrænsende for kreativiteten!

Dokumentation.

SA lægger i de forskellige udgaver vægt på en bestemt måde at udforme dokumentationen på, enten det nu drejer sig om funktionsorienteret, dataorienteret eller hændelsesorienteret analyse. Der er lagt op til en vis formalisering af dokumentationen, specielt i tilfælde, hvor dokumentation fremkommer som resultat af edb-baserede værktøjer. Her kan nævnes datadictionary og CASE-værktøjer. Dokumentation der afspejler procesforløbet forefindes ikke i metoden. Derfor bliver det vanskeligt, at videregive erfaringer for, hvordan projektarbejdet foregår i praksis.

OOA lægger vægt at dokumentationen er kortfattet og fyldestgørende. At den er kort er en fordel for både forfattere og læsere. Dette søges opnået gennem en minimumsstandard for analysedokumentet. Det kan dog ikke simplificeres til færdige opskrifter. "Analysen skal tilrettelægges med henblik på aktivt at inddrage mange andre i arbejdet. Det påvirker valget af aktiviteterne og dokumentationsformer." Teknikker, notationsformer og standarder kan udskiftes.

OOA har ingen eksplicit procesdokumentation, men man sige, at opfordringerne til situationsbestemt værktøjsanvendelse og gennemførelse af eksperimenter åbner for en uformel beskrivelse, der ikke udelukker procesbeskrivelse.

4.2 Sammenfatning.

Struktureret analyse tager sit udgangspunkt i det mekanistiske og rationelle syn og har sin primære fordel i en overvejende statisk organisation, hvor OOA bygger på et dialektisk syn og har sin fordel i en mere dynamisk organisation. Tilsvarende kan man fastslå, at SA bedre egner sig til faste forretningsgange, hvor der kan anvendes edb til at rationalisere i selve arbejdsopgaverne, hvor OOA specielt er velegnet til mere sagsorienterede arbejdsopgaver, hvor edb anvendes som støtteværktøj til arbejdsopgaven.

Da vores samfund som tidligere nævnt undergår hastige forandringer og de arbejdsopgaver, der ikke allerede er rationaliserede, netop er sagsorienterede, vil brugen af SA give anledning til større og større ulempe. Dette har igennem adskillige år stået klart også for fortalere for SA. Forfatterne til bogen "Struktureret analyse og integreret systemanalyse" stiller selv en række kritiske spørgsmål til systemanalyse, hvor de prøver at gøre en slags status i forhold til softwarekrisen⁴¹.

Softwarekrisen kommer til udtryk på den måde, at mindre ændringer og rettelser ikke kan indpasses i det traditionelle systemudviklingsarbejde.

"Systemets dele er godt indfiltret i hinanden, så brugeren skal have en virkelig god grund, hvis nytteværdien skal stå mål med omkostningerne."

Forfatterne har et afslappet syn på formalismen omkring systemudvikling og siger ligefrem at denne har gjort arbejdet mere kompliceret. Der tales om at "bøje" metoden efter tilfældet, således at metoden ikke bliver en eksakt anvisning på, hvad der skal gøres i hvilken rækkefølge. De hævder at "professionelt arbejde bygger på forståelse af formål og indhold af metoden som basis for at tilpasse retningslinierne efter forholdene".

SA kan med andre ord betragtes som et ideal, der ikke skal stræbes efter men opfattes som en støtte, hvor der i det konkrete systemarbejde arbejdes situationsbestemt. Overgangen til OOA kan hermed opfattes som en videre konsekvens af dette forhold.

Struktureret analyse er mere formalistisk end OOA. SA gør det nemmere at opstille standarder for arbejdsgangen og dermed også mere objektive mål for projektstyring og kvalitetsstyring. Faserne i SA giver nogle naturlige overordnede referencepunkter, der samtidig i dokumentationen giver beslutnings- og kontraktgrundlag for det resterende procesforløb. OOA ligger i større udstrækning disse opgaver ud til den enkelte bruger og systemudvikler, hvorved det samtidig bliver vanskeligere at sikre en overordnet koordination og planlægning. Netop på dette område bliver der behov for at supplere OOA, der ikke selv forholder sig til en løsning på disse problemer.

Problemet med mindre formel planlægning og koordination vil være særligt stort i en organisation der har tradition for meget formalistisk styring, idet der vil være tale om et kulturskifte. Samtidig vil en sådan organisation måske netop være i besiddelse af kompetencen til mindre formel styring og kun have problemer med kulturskiftet i en overgangsperiode. Det vil endvidere være oplagt at

⁴¹ Delskov, Lis & Lange, Threse : *Struktureret analyse*

holde fast i den del af det formalistiske system, der kan bibringe OOA noget positivt. F.eks. kan man godt udarbejde E/R-diagrammer eller dataflowdiagrammer, hvis dette i situationen bidrager til forståelse af systemet.

OOA giver brugeren bedre mulighed for at påvirke et nyt system end SA og må derfor betragtes som mere demokratisk. Samtidig gøres valg mere subjektive i OOA og dette kan øge politiseringen så interessekonflikter bliver mere synlige og dermed øges risikoen for konfrontation. Selv om dette kan virke som en ulempe ved brug af OOA er fordelene ved accepten af det mere subjektive og demokratiske så store, at ulemperne ved politiseringen opvejes. Ulemperne med den øgede risiko for konflikter må begrænses gennem forskellige former for procesfiltre ⁴². Der vil helt klart blive stillet større krav til systemudviklerens personlige kvalifikationer, når processen flyttes til et mindre formalistisk og mere politisk miljø, ligesom der vil blive stillet nye krav til hele organisationen, såvel brugerorganisationen som edb-organisationen.

Hverken SA eller OOA beskæftiger sig med hvorledes selve systemudviklingsprocessen dokumenteres, så man kan drage erfaring heraf.

Som det fremgår er en overgangen til OOA en naturlig konsekvens af udviklingen i vores samfund og problemerne med SA. OOA er dog ikke svaret på alle problemer ligesom SA også indeholder fordele, der med fordel kan medtages i en tilpasset OOA metode. Der vil stadig være behov for at forholde sig kritisk over for systemudviklingsmetoder også i fremtiden, og vi må således indstille os på at der til stadighed vil være behov for fornyelse i metoderne. Fornyelsen må udspringe af en brugbar praksis og ikke fra et rent teoretisk fundament, der som udgangspunkt tager afstand fra enhver nuværende praksis. Fornyelser vil desuden være kulturelt bestemte og må derfor hænge sammen med den generelle samfundsudvikling.

⁴² Enderud H, Borum F: *Konflikter i organisationen belyst ved systemarbejde*

4.3 Forandringsbehov.

Efter at have sammenlignet de to metoder og påpeget en række fordele og ulemper ved dem begge, vil vi nu udlede de forandringsbehov, som er nødvendige for at ændre organisationen fra at arbejde med SA til at arbejde med OOA.

Forandringsbehovene ved overgangen tager afsæt i følgende:

- Problemer og mangler i SA, som løses bedre i OOA, skal i fokus ved omstillingen.
- Hvad der er godt i SA må ikke mistes ved overgang til OOA.
- Mangler ved begge metoder bør også indgå i overvejelserne ved omstillingen.

Gennem sine forholdsvise faste forskrifter lægger SA op til et noget mere standardiseret forløb end OOA, hvorved inddragelse af andre discipliner lettere udelades end i OOA. Det samme kan man gøre gældende for læreprocessen. Behovet bliver således intet mindre end nytænke hele vores organisation ud fra et dialektisk syn.

Med hensyn til kompleksitet og usikkerhed kan man sige at SA er bedst til at eliminere kompleksiteten gennem sin analytiske tiltag, hvorimod OOA er bedre til at mindske usikkerheden om mål gennem brugersamarbejde. Vi har derfor behov for et øget brugersamarbejde, som opnås ved overgang til OOA. Analytiske tiltag, som man kender det fra SA kan fortsat anvendes som supplement.

Mindre ændringer og rettelser indebærer at store dele af den eksisterende dokumentation, skal revideres og dele af SA-forløbet skal gentages. Gennemføres denne aktivitet ikke mister dokumentationen sin værdi, idet dokumentationen skal afspejle det nye system helt ned i detaljerne. OOA afhjælper dokumentationsproblemet, da den ikke er så detaljeret og da der tillades fortolkninger, der således kan afspejle den ændrede systemopfattelse. Endvidere bygger OOA på, at der foretages analyse og design i hele systemudviklingsforløbet og at man derved ikke fastlåses i en tidlig fase. Dette burde mindske behovet for systemrevisioner.

Et yderligere problem i SA er den manglende indflydelse på systemudviklingen for brugerorganisationen fra analysen er afsluttet til systemet leveres. Mangler og fejl opdages sent i forløbet. Brugerorganisationen har ikke undervejs i forløbet kunnet lære det nye system at kende. OOA kan afbøde disse problemer gennem øget brugerinddragelse og eksperimenter, der både giver muligheder for reel indflydelse og kan bidrage til indlæring af det nye system. Har man tidligere inddraget eksperimenter bliver overgangen mindre drastisk.

De vigtigste mangler man kan påpege i OOA, er måske ikke mangler men udeladelser, der gør metoden mere fleksibel og lægger op til højere grad af situationsbestemt ageren. Det der tænkes på er, at metoden ikke forholder sig konkret til projekt- og kvalitetsstyring. Ved overgang til OOA

metoden lægges der implicit op til iterationer f.eks. ved opfordringer til gennemførelse af revisioner undervejs i forløbet, men det fremgår ikke eksplicit, hvor disse er placeret i forløbet. Der bliver derfor ikke mindre behov for projekt- og kvalitetsstyring, ikke mindst til styring af disse iterationer. Man kan anvende de værktøjer man brugte under SA, idet der er behov for en tilpasning.

Af yderligere mangler eller måske snarere problemer med OOA, kan nævnes at den kortfattede dokumentation muliggør fortolkning. Dette er et problem, hvis dokumentationen skal fungere som kontraktgrundlag. Der bliver således behov for en anden form for kontraktgrundlag. BATOFF-kriteriet kan indgå i dette.

For at forbedre systemudviklerens måde at arbejde på, er der behov for løbende at dokumentere procesforløbet. Ingen af metoderne tager sig eksplicit heraf så der er behov for at dette indgår i omstillingsprocessen. En yderligere fordel herved kan blive en fastholdelse af erfaring i organisationen samt en mulighed for nye projektdeltagere til at komme ind i et konkret projekt.

Overgang til OOA vil som tidligere nævnt stille store krav til systemudviklerens personlige kvalifikationer og dermed blive behov for styrke og videreudvikle disse.

5. Omstilling fra SA til OOA

Systemudviklere, der er gode til struktureret systemudvikling, bliver gode til objektorienteret systemudvikling. Denne kendsgerning er dokumenteret i mange forsøg med omskoling af systemudviklere fra strukturerede til objektorienterede metoder⁴³. Man bruger ofte denne kendsgerning som et argument for, at systemudviklerens personlige egenskaber spiller en væsentlig større rolle for et givet projekts succes end den anvendte metode; men det er nok vigtigt, at man holder sig for øje, at erfaringen er høstet i relation til omskoling af systemudviklere, fordi erfaringen også kunne pege på nogle grundlæggende problemer i relation til den måde, man traditionelt tilrettelægger omskolingen på i form af teoretiske kurser i brug af metoder og værktøjer.

Et andet aspekt af problemer i relation til omskoling berøres af Stroustrup⁴⁴, som siger, at meget selvfølgelig afhænger af den enkelte systemudviklers evner; men ledelsens holdning spiller også en vigtig rolle. Ledelsen glemmer tit, at organisationen består af enkelte individer. Det er populært at sige, at alle organisationens individer er ens og udskiftelige. Det gælder imidlertid kun, hvis man ikke udnytter sine evner fuldt ud. Mindste fælles nævner kommer til at sætte standarden.

En anden trussel, som Stroustrup peger på, er at systemudviklerne selv søger væk til andre projekter. Det skyldes ofte, at systemudvikleren ikke får den fornødne frihed til den nødvendige personlige og faglige udvikling indenfor rammerne af et givet projekt. Denne trussel søger man traditionelt at imødegå ved formalisering af processen og standardisering af dokumentationen. Ifølge Stroustrup vil det være bedre at give folk optimale arbejdsvilkår og udfoldelsesmuligheder i projektorganisationen, for på den måde at få dem til at blive.

Stroustrups indfaldsvinkel kan bruges som argument for, at det er vigtigt - for ikke at sige altafgørende - for succes af en ændring i systemudviklingsmetoden, at man er opmærksom på behovet for en samtidig udvikling af organisationen - især på lederplan.

⁴³ Computer Language; March 1992

⁴⁴ Stroustrup, Bjarne: "The C++ Programming Language, 2. ed"; Addison-Wesley, 1991. Kapitel 11, Design and Development.

5.1 Omskoling af systemudviklere

Omskolingsplaner for systemudviklere er ikke det, der er skrevet mest om. Det er som om, der er større prestige forbundet med selve udviklingen af de nye metoder, end med implementeringen af de nye metoder. Det kan være vanskeligt at se logikken heri, fordi man skulle tro, at den, der har investeret mange ressourcer i udviklingen af en systemudviklingsmetode, også ville være interesseret i at sikre, at brugerne af metoden fik succes med den.

Da opstilling af en omskolingsplan er selve essensen af dette projekt, har vi selvfølgelig været ekstra opmærksomme på, hvad der er skrevet om emnet. I februar nummeret fra 1996 af Communications of the ACM fandt vi uventet en hjælp, som gør, at det følgende ikke udelukkende bygger på vore egne erfaringer; men også inddrager andres erfaringer.⁴⁵

Mange virksomheder tøver med at skifte til objektorienterede metoder på trods af de mange potentielle fordele, det stilles i udsigt at man vil opnå, hvis man skifter til disse metoder. Det kan skyldes, at man tidligere har oplevet, at et skifte til nye metoder faktisk ikke har medført de fordele, som man var blevet stillet i udsigt.

For den leder, der skal tage stilling til, om man skal skifte til en ny metode, står der meget på spil. Lederen er ansvarlig for, at softwareprojektet bliver gennemført, uden at de økonomiske rammer bliver sprængt. Hvis det er vanskeligt at gennemskue de mulige konsekvenser af at skifte til en ny metode, vil lederen være tilbøjelig til at satse på det kendte, hvor man bedre kan overskue konsekvenserne.

Det er en erfaring, at det i gennemsnit tager et år at skifte til en objektorienteret metode. Projektets størrelse spiller i denne forbindelse ingen rolle. Det drejer sig derimod om, at systemudviklerne skal have tid til at lære at arbejde med den nye metode. Man kan kun lære den nye metode ved, at man selv gennemlever processen, dvs. at man arbejder med metoden.

Ifølge Fayad m.fl. består omskolingsprocessen af tre faser:

- planlægning før projekt-start
- inddragelse af den objektorienterede metode
- projektledelse

Planlægning før projekt-start.

Det er en generel erfaring, at fornuftig forberedelse og planlægning øger chancen for at opnå succes. Inden man sætter omskolingsprocesser i gang, som resulterer i store forandringer, skal man

derfor planlægge, hvilke forandringer man ønsker. Ellers risikerer man, at det er tilfældigt, hvad der kommer ud af omskolingsprocessen.

En anden erfaring er, at indføring af en objektorienteret metode vil kræve ændringer i kulturen for hele projektorganisationen. Næsten alle trin i software udviklingsprocessen bliver påvirket af og skal tilpasses den nye måde at tænke på. Overgang til en objektorienteret metode vil påvirke alle, der deltager i projektet - ikke bare de involverede systemudviklere; men også de personer, systemudviklerne arbejder sammen med i projektet.

Når man skal ændre i en organisations kultur, skal man huske, at den vigtigste succesfaktor er, at ledelsen demonstrerer sin vilje til at arbejde for ændringen. Det er et grundprincip, at større ændringer i en organisation starter i toppen.⁴⁶

Ledelsen kan vise sin interesse for ændringerne på forskellige måder:

- Ledelsen skal engagere sig aktivt i at gennemføre ændringen. Det er ikke nok, at ledelsen bare starter processen op - man skal følge op med krav om, at ændringerne gennemføres eller med fornyet motivering, når det bliver svært, og systemudviklerne føler trang til at falde tilbage i de gamle vaner.
- Man skal sætte tid af til de fornødne kurser i de nye metoder og til opfølgende diskussioner af de ændringer, der skal gennemføres.
- Sørg for at de rette værktøjer er til rådighed. De rette værktøjer er dem, som systemudviklerne mener er nødvendige for at kunne løse opgaven og tilrettelægge arbejdet på en hensigtsmæssig måde. Ledelsen udsender forkerte og demotiverende signaler, hvis man ikke vil anskaffe de ønskede værktøjer i omskolingsfasen.
- Hold fast i metodens proces. Det er uundgåeligt, at der på et tidspunkt opstår et ønske fra en eller flere projektdeltagere om at gå i gang med at kode med det samme i stedet for at følge metodens anvisninger. Disse ønsker skal afvises med bestemthed i denne fase, hvor det handler om at lære at arbejde efter en ny metode.

Inden det første projekt med brug af en objektorienteret metode kan gå i gang, skal man have indhentet kundens tilladelse. I de fleste tilfælde vil kunden være imod, at man bruger en objektorienteret metode, og især hvis han får at vide, at det er vores første forsøg med en ny metode. Det skyldes, at kunder ofte associerer brug af ny metoder med høj risiko.

Inden projektet kan komme i gang, skal systemudviklernes modstand også bearbejdes. Det kan måske virke overraskende, at systemudviklere, som dog arbejder med indførelse af ny teknologi og metoder, er så afvisende overfor nye ideer, når det drejer sig om deres egen situation. Man kan blot konstatere, at forsvarsmekanismerne og modstanden mod forandringer i høj grad også er til stede

⁴⁵ Fayad, Mohamed E., Wei-Tek Tsai & Milton L. Fulghum: "Transition to Object-Oriented Software Development"; Communications of the ACM februar 1996, p. 108 - 121.

⁴⁶ Humphrey, W.: "Managing the Software Process"; Addison-Wesley, 1989.

her. Nøglen til systemudviklernes accept er indsigt og medindflydelse. Systemudviklerne deltager aktivt i tilpasningen af metoden til organisationen og får reel medindflydelse på beslutningerne. Herved opnår man, at systemudviklerne føler et medansvar for metoden.

Inddragelse af den objektorienterede metode

Den første aktivitet i denne fase består i at vælge én af de mange mulige objektorienterede metoder, der findes. Det må anbefales, at man vælger en objektorienteret metode, som dækker så mange som muligt af systemets livscyklus. Valg af en konkret metode ligger uden for rammerne af dette projekt, fordi vi allerede ved problemdefinitionen har lagt os fast på en konkret metode.

Hvis udviklerteamet er vant til at arbejde med CASE værktøjer, skal man sørge for at den valgte objektorienterede metode understøttes af et passende CASE værktøj, som også kan hjælpe systemudviklerne med at overholde de regler, som metoden måtte foreskrive. Vi er generelt ikke begejstrede for den idé, at en metodes regler skal følges slavisk; men netop i relation til det første projekt med en ny metode kan det være relevant, at man forsøger at overholde reglerne. I modsat fald er der fare for, at man slet ikke lærer reglerne at kende.

Fayad m.fl. har gjort den erfaring, at sammensætningen af projektgruppen skal være anderledes i et objektorienteret end i et struktureret projekt. Man har haft gode erfaringer med at supplere projektgruppen med nye nøglepersoner:

- Objekt gurun er den person, som er ekspert i den nye objektorienterede metode, og som kan svare på alle de spørgsmål, de andre systemudviklere stiller undervejs i projektet. Selvom systemudviklerne får formelle kurser i den nye metode, viser erfaringen, at der dukker mange spørgsmål op i relation til emner, som ikke nåede at blive berørt på kurset.
- Domæneanalytiker er ekspert med hensyn til det applikationsdomæne, der er genstand for analysen. En af domæneanalytikerens opgaver er at fremme genbruget af klasser, som er udviklet ved tidligere analyser af det samme eller beslægtede domæner. Domæneanalytiker er den person, som i kraft af sit overblik over applikationsdomænet er i stand til at få øje på potentielle muligheder for genbrug.
- Prototypeeksperten kaldes også super hackeren. Det er en person, som er i stand til at udvikle en prototype af et softwareprodukt hurtigt og fejlfrit uden at følge en specifik metode. Det er vigtigt, at der er tale om en person, som er i stand til at formidle resultatet af sit arbejde til de øvrige projektdeltagere, når prototypen senere skal konverteres til den aktuelle objektorienterede metode.

Det er nødvendigt, at projektdeltagerne får den fornødne træning i den nye metode. Det er erfaringen, at det er bedre at fordele træningen på et antal kurser af kortere varighed, som spredes ud over projektforløbet end at koncentrere det hele i starten af projektet. Når træningen spredes, giver det projektdeltagerne mulighed for at reflektere og øve sig imellem kurserne. Hvis man har mulighed for at tilknytte en konsulent til projektet, bliver mulighederne optimale, fordi der så

bliver mulighed for at give den enkelte systemudvikler individuel og situationsbestemt undervisning. En detaljeret uddannelsesplan bør indgå i den handlingsplan, der udvikles i forbindelse med omskolingsaktiviteterne.

Projektledelse

Anvendelse af objektorienterede metoder resulterer erfaringsmæssigt i optimering af udviklingsproces og genbrug med lavere udviklingsomkostninger som følge heraf. Til gengæld betaler man prisen i form af større forbrug af processor og memory ressourcer. Det er derfor en gylden regel, at de dele af softwaren, som er kritiske for projektets succes, udvikles så tidligt som muligt i projektforløbet, for at reducere risikoen for en senere fiasko.

Årsagen til det større forbrug af processor og memory ressourcer i forbindelse med objektorienteret implementering skal søges i den måde man implementerer objektorienterede faciliteter som polymorfi, arv og dynamisk binding. Det vil derfor også kunne betale sig at undgå at bruge disse muligheder mere, end løsningen kræver. Systemudviklere, der netop har stiftet bekendtskab med de nye objektorienterede muligheder, vil være fristet til at bruge dem mere, end løsningen strengt taget kræver. Reviews i det første projekt skal være med til at gøre systemudviklerne opmærksomme på dette problem.

Fayad m.fl. gør opmærksom på, at det er nødvendigt at dokumentere selve udviklingsprocessen. Det er i den forbindelse en stor hjælp, hvis metoden er baseret på den grundlæggende idé, at systemudviklingsprocessen også skal forstås som en læreproces. Som vi tidligere har beskrevet, indebærer det, at der sker en erfaringsopsamling i forbindelse med, at der udføres aktive eksperimenter. De opsamlede erfaringer er grundlaget for den senere refleksion, som kan resultere i forbedringer af selve metoden og/eller af systemudviklerens egne arbejdsmetoder. Det er vor egen erfaring, at man skal dokumentere udviklingsprocessen, mens den er frisk i erindringen. Efter blot nogle få dages forløb blegner erindringen om, hvordan man løste et konkret problem. En af de simpleste teknikker, man kan tage i brug til dette formål, er at lade systemudviklerne føre dagbog over de problemer og løsningsideer, som de arbejder med. Det er vores erfaring, at sådan noget skal indarbejdes som en fast vane i projektgruppen; men så er der næsten heller ingen grænse for hvilke fordele, man vil kunne høste.

Det er vigtigt, at metoden med tiden får lov til at udvikle sig, sådan som systemudviklerne ønsker det. Hvis man hele tiden udvikler og tilpasser metoden, kan man have held til at undgå den hjernelammelse, der efter Yourdons mening opstår hos nogle systemudviklere, som bliver så overvældet af en ny teknologi, at de efter nogen tid bruger al deres tid på at

- prøve at forstå den ny teknologi
- argumentere for dens fortræffelighed
- prøve at få den ny teknologi til at virke efter hensigten

I praksis har det vist sig vanskeligt at få eksisterende dokumentationsstandarder til at harmonere med den dokumentation, der er behov for at udvikle i et objektorienteret projekt. Man skal derfor være indstillet på at tilpasse eksisterende dokumentationsstandards, så de er kompatible med de nye objektorienterede metoder.

5.2 Omstilling af organisationen.

Projektet

Systemudviklingsarbejdet organiseres som oftest i projekter. Projekters organisation, indhold og styring er beskrevet righoldigt i litteraturen, f.eks. af Andreas Munk-Madsen i *Strategisk Projektledelse* - og han definerer projektet som "organisationens off-roader - fremragende i vildniset, men uøkonomisk på landevejen".

Hvis vi taler større systemudviklingsopgaver er projektet en velegnet organisationsform for både OOA- og SA-opgaver. Der er dog visse forskelle på metoderne som bør give anledning til overvejelser i den sammenhæng.

Set ud fra et organisatorisk synspunkt er den største forskel på OOA (f.eks. implementeret vha. Spiralmodellen) og Struktureret Analyse (implementeret vha. vandfaldsmodel) førstnævntes store fokusering på iteration og dermed usikkerhed med hensyn til tids- og ressourceforbrug.

Organisationen

Man kan skelne mellem to organisationstyper - den mekanistiske og den organiske⁴⁷.

Førstnævnte er kendetegnet ved klart definerede opgaver og ansvar, vertikal interaktion og loyalitet overfor opgaven; mens den organiske type baserer sig på løbende gensidig tilpasning, tværgående interaktion og engagement i organisationens opgaver.

Disse to typer kan således med god tilnærmelse betragtes som panderter til henholdsvis SA og OOA..

Hvis man således har med en overvejende mekanistisk organisation at gøre, kan der opstå problemer med at implementere OOA, idet den på afgørende punkter bryder med hævdivundne principper. Teoretisk set burde det ikke nødvendigvis give anledning til problemer at operere med såvel det organiske som det mekanistiske princip i den samme organisation; men det kan nemt give anledning til misundelse og murren i krogene, hvis en personalegruppe har væsentligt anderledes vilkår end de andre. Dette fænomen kaldes normkonflikter⁴⁸, og kan vanskeliggøre samarbejdet med brugerne, og dermed i sidste ende true projektets muligheder for succes.

Ovenstående er et eksempel på at edb-funktion og brugere er dele af den samme organisation. I det tilfælde bliver samarbejdsformen ofte således, at edb-afdelingen mere fungerer som konsulenter end leverandører⁴⁹, hvorimod der i langt de fleste tilfælde bliver tale om et egentligt leverandørforhold, hvis edb-folkene er eksterne. Traditionelle kunde leverandørforhold har vanskeligt ved at løse opgaver med stor usikkerhed, og de forløb som oftest finder sted i denne sammenhæng har flere lighedspunkter med SA end med OOA. Man bliver altså nødt til at bevæge sig væk fra disse traditionelle forhold, hvis OOA skal lykkes, fordi disse er karakteriseret ved en

⁴⁷ Burns & Stalker: *The management of innovation*, 1961

⁴⁸ J.A. Mortensen et al.: *Erbvervsøkonomi og produktionsorganisation*, s. 134, 1993

⁴⁹ Mathiassen, Lars, *Kontrol og kreativitet i Edb-arbejdet*, s. 3

klar ansvars- og arbejdsdeling baseret på statiske specifikationer. Der bliver altså tale om en ren konstruktionsproces, hvor lære- og forandringsprocesserne kun vanskeligt kan indgå.

For mindre virksomheder uden egen edb-organisation og med et begrænset edb-budget, kan man således ikke umiddelbart forvente, at den objektorienterede arbejdsform vil slå igennem. Her må man fortsat forvente, at virksomhedsspecifikke ændringer til et standardsystem (f.eks. Concorde, Navigator) vil blive gennemført efter vandfaldsmodellen.

Minzberg⁵⁰ har udviklet en meget anvendt organisatorisk grundmodel. Først definerer han fem arbejds- og styringsenheder, og derefter kombinerer han dem i fem forskellige organisationsformer (idealtyper): den basale form (f.eks. små "iværksættervirksomheder"), maskinbureaukratiet (f.eks. produktionsvirksomhed af standardvarer), adhokratiet (udvikling), fagbureaukratiet (f.eks. hospitalet, universiteter) og den divisionaliserede form (f.eks. datterselskaber af maskinbureaukratier). Hvilken form, der vælges, afgøres ved at vurdere en række interne såvel som eksterne forhold.

Traditionelt har edb-organisationer enten været organiseret som den basale form eller været adhokratier, idet disse former er specielt velegnede til at agere i en dynamisk omverden, og dermed gearede til at håndtere forskelligartede udviklingsprocesser.

Set i relation til overgang fra SA til OOA er der imidlertid en række forhold, som peger i retning af, at fagbureaukratiet er at foretrække⁵¹. Her tænker vi først og fremmest på den øgede kundeorientering og fokus på nødvendige bløde kvalifikationer, som trækker i retning et højere og bredere kompetanceniveau end det snævert edb-faglige. Hertil kommer, at arbejdsopgaverne er blevet stadig mere komplekse f.eks. i forbindelse med netværks- og client-server-løsninger.

Dernæst er opgaverne blevet mere ensartede, idet der i stigende grad anvendes kvalitetsstyringsystemer og standarder. Man kan i øvrigt sagtens argumentere for at fagbureaukratiet er mere velegnet end for eksempel adhokratiet ved anvendelse af SA - det har bare sjældent været tilfældet i praksis.

I den meget omtalte SEI Capability maturity Model⁵² som vurderer effektivitet og kvalitet i en edb-organisation opereres med fem niveauer, hvor trinene i stigende orden hedder: ad hoc, gentageligt, defineret, ledet og optimerende. Processen med at omdanne adhokratiet til et ægte fagbureaukrati eller det professionelle bureaukrati, som det også kaldes, kan altså betragtes som en tur op ad procesmodenhedsstigen.

Problemstillingen vedrørende valg af ændringsstrategi er selvsagt også relevant på dette sted. I forlængelse af betragtningerne i afsnit 2.3. om brugerdeltagelse, må det anses for fornuftigt at tage deltagelsesstrategien i anvendelse, idet den baserer sig på et aktivt menneskesyn og en konfliktopfattelse af organisationen, hvilket passer fint på de fleste edb-organisationer i dag. Strategien indebærer, at alle berørte personer skal involveres enten i en af tre følgende grupper: projektgruppe, brugergruppe, referencegruppe. Projektgruppen vil således efter modellen få til opgave at forstå det udførende arbejde i forbindelse med implementering af den nye metode.

⁵⁰ Minzberg, H.: *Structures in five*, 1983

⁵¹ Mathiassen, Lars, *Kontrol og kreativitet i Edb-arbejdet*, s. 18

⁵² Humphrey, Watts: *Managing the software process*; Addison-Wesley, 1989

Brugergruppen, som består af såvel menige brugere som deres chefer, står for problemformulering og fastlæggelse af mål, mens referencegruppen oplæres løbende og vurderer tvivlsspørgsmål. Uddannelse bliver et helt centralt element ved anvendelse af denne model⁵³, og det understreges, at det kommer til at foregå på brugernes præmisser med hensyn til såvel niveau som sprogbrug⁵⁴. I praksis oplever man desværre ofte, at edb-organisationer, som burde være velbevandrede ud i teorier om ændringsstrategier, selv vælger forkert, når de skal indføre ændringer i egen organisation.

Et væsentligt aspekt ved den objektorienterede metode, er genbrug af objekter, og for at tilsikre, at genbrug finder sted i optimalt omfang er det nødvendigt med visse organisatoriske tiltag.

Der kan anvises en række forskellige måder⁵⁵, som falder i to udgaver: discount modellerne ad-hoc og supply and demand, som lægger op til repositories med muligt genbrugelige objekter, som kunderne, dvs. systemudviklerne, selv må finde frem til. Expert services og product center modellerne kræver bemanding omkring de genbrugelige objekter, og disse personer kan eventuelt deltage i projektarbejdet. Sidstnævnte modeller er de dyreste; men også de mest effektive.

Undersøgelser har vist, at ad-hoc modellen indtil videre er den mest udbredte, hvilket utvivlsomt hænger sammen med omkostningerne. Man må således forvente, at ekspert og produkt modellerne vil vinde frem i de kommende år.

Vi har tidligere omtalt det nyttige i at anvende procesdokumentation som erfaringsmateriale i senere projekter. En måde at optimere genfindingen af procesdokumentation på ville være at organisere den under expert services modellen. Dermed vil man lynhurtigt kunne få svar på de to allervigtigste spørgsmål, når man står med et nyt problem: er der nogen som tidligere har oplevet dette problem og hvorledes angreb de det. Hvis organisationen opererer med en egentlig metodeafdeling vil ansvaret for procesdokumentation også kunne placeres her.

Projektlederen

Projektlederens foretrukne egenskaber må nødvendigvis være forskellige i et mekanistisk og i et organisk miljø. I det mekanistiske miljø er han først og fremmest beslutningstager, dernæst administrator. I det organiske miljø er evner på disse områder nødvendige; men på ingen måde tilstrækkelige for at kunne udfylde rollen som en god projektleder. Innovation, kreativitet, samarbejdsformidling bliver egenskaber og opgaver som projektlederen skal besidde og kunne løfte.

Projektledere rekrutteres ofte blandt dygtige systemudviklere, og dette viser sig tit at være en dårlig ide, fordi gode systemanalytikere eller programmører ikke nødvendigvis har de tilstrækkelige lederegenskaber til at blive gode projektledere, så det man reelt opnår, er at man mister gode systemfolk og får dårlige ledere. Med indførelsen af OOA er dette en endnu dårligere ide end under SA, idet de bløde kvalifikationer nævnt ovenfor udover lederegenskaberne er nødvendige for at kunne bibringe sit projekt succes.

⁵³ Nordby T., *Strategier for bedre systemer*

⁵⁴ Neergaard, Peter: *Planlægning af ændringer*, s. 87

⁵⁵ Goldberg & Rubin, *Succeeding with objects*, 1995

Man kan argumentere for, at projektlederens fornemste opgave i et OOA-projekt bliver at kontrollere kreativiteten⁵⁶. Det kan lyde paradoksalt, eftersom de fleste opfatter kreativitet som en meget spontan proces og dermed vanskeligt kontrollerbar. Efter Mathiassens mening kræver det såvel aktiv ledelse som selvdisciplin hos projektdeltagerene for at det kan lade sig gøre.

Hvis man forestiller sig OOA implementeret ved hjælp af spiralmodellen⁵⁷, kommer risikoledelse til at spille en betydelig rolle. Man kunne frygte, at det kunne påvirke arbejdet i negativ retning, at tage udgangspunkt i noget negativt; men fordelene er at man i en kreativ læreproces får indbygget kontrolmekanismer; men man kommer ikke udenom at projektlederen skal formå at skabe en projektkultur, hvor den megen fokus på risici ikke skaber modløshed og manglende motivation. I denne forbindelse vil tilvænning spille en rolle - dvs. det bliver nemmere, at abstrahere fra det negative, når man har arbejdet med modellen nogle gange.

Dokumentation af fremdrift og løbende beregning af færdiggørelsesgrad hører til blandt projektlederens vigtigste opgaver. Ud over måling af potentielt genbrug⁵⁸, har vi ikke i kilderne kunnet finde metoder eller anvisninger, som retter sig specifikt mod OOA, og må derfor henvise til de traditionelle metoder. Det falder uden for denne fremstillings rammer at gå i detaljer med disse.

Ledelsen

I den traditionelle projektmodel er styrekomiteen og som oftest dermed virksomhedens ledelse opdragsgiver og sætter de økonomiske og tidsmæssige rammer. I sin iver efter at udøve kontrol og sikre optimal udnyttelse af ressourcerne (læs: billige systemer), vil vandfaldsmodellen være mere attraktiv for ledelsen end den iterative model, fordi den er nemmere at styre, og set i et psykologisk perspektiv er der intet overraskende i, at top-down princippet tiltaler ledelsen mere end bottom-up. Man må forvente at skulle komme med særdeles gode argumenter for at den iterative model er at foretrække for at overbevise en mekanistisk orienteret ledelse.

Et eksempel på ledelsers fokusering på den økonomiske side af sagen er Information Engineering's succes i den sidste lille halve snes år. Metoden baserer sig på James Martins ideer⁵⁹ om, at de grundlæggende data- og processtrukturer i en virksomhed er uforanderlige og at det derfor er muligt at konstruere fremtidssikrede edb-systemer.

Martins ideer er siden deres fremkomst blevet afvist af stort set alle (Naur, som ikke tror på formelle specifikationer⁶⁰ og Brooks, som ikke mener at der findes nemme genveje til produktivetsforbedringer i hverken værktøjer eller metoder⁶¹); men alligevel kunne metoden i starten af 90'erne notere sig en betragtelig succes.

Årsagen hertil var ligetil. Virksomhedslederne var lette ofre for metodens løfter om fremtidssikrede edb-systemer og produktivetsforbedringer på op til 400%.

⁵⁶ Mathiassen, Lars, *Kontrol og kreativitet i Edb-arbejdet*, s. 1

⁵⁷ Boehm, Barry W., *A spiral model of software development and enhancement*, 1988

⁵⁸ Goldberg & Rubin, *Succeeding with objects*, 1995

⁵⁹ Martin & Finkelstein, *Information Engineering*, 1981

⁶⁰ Naur, P., *Intuition and formalization in program development*, 1990

⁶¹ Brooks F.P., jr, *No silver Bullet*

Inden for de sidste par år er metoden i kraft af brugernes negative erfaringer blevet modificeret, således at den er bedre i overensstemmelse med virkeligheden.

Man kan således ikke forvente, at ledere agerer rationelt, når der skal tages stilling til disse ting. Hvis man skal nå frem til, at en beslutningstager vælger OOA, skal han bibringes den fornødne indsigt, eller være villig til at acceptere rådgivning fra folk, som har den. Styrer han udelukkende efter kortsigtede økonomiske gevinster, hvilket lige nu synes at være tidens trend i ledelse, kan det blive meget vanskeligt at få ham overbevist. I den modsatte retning trækker det forhold, at et stigende antal større virksomheder og koncerner er begyndt at opfatte edb-investeringer som strategiske, hvilket betyder at disse ikke er underlagt samme stramme tilbagebetalingskriterier som andre investeringer.

6. Konklusion

Overgang fra SA til OOA er mere end overgang til en ny metode - der er tale om et paradigmeskift, som forudsætter en kulturændring i store dele af virksomheden. Det er nemlig ikke kun systemudviklerne, der påvirkes; men alle de personer, der på en eller anden måde arbejder sammen med systemudviklerne.

I dette kapitel vil vi fremdrage de konklusioner, som efter vor mening fremstår som de vigtigste resultater i vort projektarbejde

Ledelsens engagement

Den vigtigste konklusion er efter vor mening, at ledelsens direkte engagement i den nye metodes indførelse er af altafgørende betydning. Erfaringer peger på, at det tager systemudviklerne mindst et år at tilegne sig en ny metode. Især i begyndelsen af denne periode vil systemudviklerne være sårbare i forhold til de trusler, som projektet vil være udsat for.

Truslerne mod projektet kan komme udefra i form af pres fra kunder og brugere, som frygter at økonomiske rammer og tidsplaner ikke kan overholdes. Der vil være en markant tendens til, at enhver budget- eller ressourceoverskridelse bliver tilskrevet den nye metode. I en sådan situation er det altafgørende for successen, at lederen ikke mister overblikket og giver efter for presset. I stedet skal han lade systemudviklerne forstå, at han bakker dem op, og at han forventer, at den nye metode bliver indført, som planlagt.

Truslerne mod projektet kan også komme indefra i form af systemudviklernes usikkerhed og modstand mod den nye metode. I denne situation må lederen gå foran og vise systemudviklerne, at man mener det alvorligt, når man siger, at man vil indføre en ny metode. Det kan man bl.a. gøre ret effektivt ved at synliggøre i sin personalepolitik, at karrieremulighederne i virksomheden er knyttet til beherskelse af den nye metode.

Hvis lederen ikke på forhånd har forpligtet sig til at stå last og brast med den nye metode, vil det være for let for alle parter at give efter for presset, når det opstår.

Uddannelse af projektdeltagere

Den næstvigtigste konklusion er efter vor mening, at rigtig uddannelse og træning i de nye metoder er af største betydning for successen. Uddannelse og træning kan tilrettelægges på mange forskellige måder; men den mest hensigtsmæssige er efter vor mening et forløb, der spredes ud over hele projektets levetid, sådan at systemudviklerne får tid til at øve sig og drage deres erfaringer ind imellem de enkelte delkurser.

I den ideelle model er der fast tilknyttet en konsulent til projektet. Det giver mulighed for at give projektets systemudviklere målrettet hjælp og situationstilpasset (differentieret) undervisning.

Vi har påpeget, at man ikke kan skifte fra en struktureret til en objektorienteret metode, uden at det får konsekvenser for den herskende kultur i alle de dele af virksomheden, hvor man direkte eller indirekte kommer i berøring med den nye metode. Vi mener, at det er vigtigt, at man understøtter de kulturelle forandringer med relevante kurser, bl.a. for at modvirke medarbejdernes angst for det nye og ukendte.

Vi har flere gange fremhævet, at vi mener, at systemudvikling forudsætter en mangfoldighed af kvalifikationer hos systemudvikleren. Efter vor mening er dette forhold så vigtigt, at vi har medtaget det som et af de bedømmelseskriterier, der indgår i vor taksonomi. Overgang til en ny systemudviklingsmetode er et perfekt tidspunkt for afvikling af kurser, som kan styrke systemudviklernes almene kvalifikationer, fordi man må forvente at systemudviklerne i overgangsfasen generelt er åbne og modtagelige for nye tanker.

Projektorganisering

Den tredje konklusion, vi vil pege på, er den, at projektorganisationen efter vor mening stadig er den eneste reelle mulighed for at organisere en objektorienteret systemudvikling.

Det står os dog klart, at organisationen må igennem en tilpasning, for at være korrekt trimmet til de nye krav, der stilles i forbindelse med anvendelse af en objektorienteret metode. Vi har allerede påpeget nødvendigheden af, at man styrker systemudviklerens almene kvalifikationsniveau i flere forskellige retninger.

Objektorienterede projekter siges at være vanskelige at styre. Det kan skyldes, at de anvendte metoder giver systemudviklerne nogle nye frihedsgrader. Det betyder dog ikke, at edb-udviklingsprojekter hermed skal blive store ta'-selv-borde for de involverede parter. Tværtimod stiller de opnåede frihedsgrader store krav med hensyn til udøvelse af selvdisciplin hos den enkelte, såvel projektdeltager som mellem- og projektleder, samt indførelse af passende kontrolmekanismer, som på et tidligt tidspunkt kan tilsikre, at projekterne ikke løber af sporet⁶².

Det afgørende er imidlertid, at ledelsen ikke kun går efter kortsigtede økonomiske mål; men er villig til at acceptere, at gevinsterne kan komme på et lidt længere sigt end det indeværende budgetår. For at kunne fastholde ledelsens tillid er det derfor vigtigt, at man er i besiddelse af værktøjer, som kan måle projekternes fremdrift og færdiggørelsesgrad, og som kan danne rammen for en dokumentation, som kan godtgøre, at man ikke blot har spildt tiden med ligegyldige eksperimenter.

Procesbeskrivelser

Den fjerde konklusion, vi vil pege på, er, at det er vigtigt, at man som systemudvikler lærer at opsamle procesdokumentation. Vi har fra starten af denne opgave peget på, at opfattelsen af systemudvikling som en læreproces af os blev opfattet som et af de fem vigtigste kriterier, vi ville pege på, når vi skulle opstille en taksonomi til vurdering af systemudviklingsmetoder.

Hvis sammenhængen mellem systemudviklingen og læreprocessen skal virke i praksis er det dog vigtigt, at systemudvikleren er i stand til at opsamle og huske sine egne erfaringer. Disse erfaringer er jo råmaterialet, når systemudvikleren skal reflektere over sin måde at arbejde på i særdeleshed og metodens anvendelighed i almindelighed.

Gennem refleksionen finder systemudvikleren gradvist frem til de forbedringer af egen arbejdsmåde og den anvendte metode, som er så livsvigtige, hvis metoden skal kunne anvendes i længere tid. Metoder er ikke statiske - de skal hele tiden trimmes og tilpasses de skiftende forhold de anvendes i og de systemudviklere, der bruger dem. Efter vor opfattelse må det være sådan, at metoden tilpasses de mennesker der arbejder med dem - ikke omvendt.

⁶² Mathiassen, Lars: *Kontrol og kreativitet i edb-arbejdet*

7. Litteraturliste

- Andreas Munk-Madsen: Strategisk projektledelse.
Blum, Bruce I: A Taxonomy of Software Development Methods. Comm. of the ACM Nov. 1994 p. 82.
Boehm, Barry W., A spiral model of software development and enhancement, 1988
Bohm, C & G. Jacopini: Flow Diagrams, Turing Machines and Languages with Only Two Formation Rules; Comm.
Booch, Grady: Object Oriented Design with Applications.
Brooks F.P., jr, No silver Bullet
Burns & Stalker: The management of innovation, 1961
Chidamber, S. R. & Kermerer, C. F.: Towards a Metrics Suite for Object Oriented Design. SIGPLAN Notices, Oct.
Communications of the ACM februar 1996, p. 108 - 121.
Computer Language; March 1992
Connell J.L., Shafer L.: Structured Rapid Prototyping - An Evolutionary Approach to Software Development
Dahlbom, Bo og Mathiassen, Lars: Computer in context
Delskov, Lis & Lange, Threse :Struktureret analyse
DeMarco, Tom: Structured Analysis and System Specification; Prentice-Hall, 1978.
Development Methods. ACM SIGSOFT Software Engineering Notes Apr. 1993
Dijkstra, E.W.: Cooperating Sequential Processes i F. Genuys (ed.): Programming Languages; Academic Press,
Enderud H, Borum F: Konflikter i organisationen belyst ved systemarbejde
Enderud, Harald: Beslutninger i organisationer - i adfærdsteoretisk perspektiv.
Fayad, Mohamed E., Wei-Tek Tsai & Milton L. Fulghum: Transition to Object-Oriented Software Development;
Goldberg & Rubin, Succeeding with objects, 1995
Humphrey, W.: Managing the Software Process; Addison-Wesley, 1989.
J.A. Mortensen et al.: Erhvervsøkonomi og produktionsorganisation, 1993
Kolb, D.A.: Experiential learning: Experience as the source of learning and development
Kowal, James A.: Analyzing Systems; Prentice-Hall 1988.
Leavitt, H.J.: Applied organizational change in industri: Structural, technological and humanistic approaches
Martin & Finkelstein, Information Engineering, 1981
Mathiassen L., Munk-Madsen, A., Nielsen, P.A og Stage, J: Objektorienteret analyse
Mathiassen, Lars, Kontrol og kreativitet i edb-arbejdet
McMenamin, Stephen M. & John F. Palmer: Essential Systems Analysis; Yourdon Press, 1984.
Minzberg, H.: Structures in five, 1983
Monarchi & Puhr: A Research Typology for Object Oriented Analysis and Design. Comm. of the ACM Sept. 1992
Mumford & Ward: edb, system og menneske.
N.E. Andersen et al: Professionel Systemudvikling.
Naur, P., Intuition and formalization in program development, 1990
Neergaard, Peter: Planlægning af ændringer.
Nordby T., Strategier for bedre systemer
Parnas & Clements: A rational design proces: how and why to fake it.
Shable, Robert C. & Cohen, Samuel S.: The Object-Oriented Brewery: A Comparison of Two Object-Oriented
Stroustrup, Bjarne: The C++ Programming Language.
Yourdon, E. & L. L. Constantine: Structured Design: Fundamentals of..., Prentice-Hall 1978.
Yourdon, Edward: Modern Structured Analysis