



Håndbok i EDB-metode

EDB-metode 96: Minimetode for vedlikehold og utvikling av EDB-systemer i Statistisk sentralbyrå

Håndbok i EDB-metode

EDB-metode 96: Minimetode for vedlikehold og utvikling av EDB-systemer i Statistisk sentralbyrå

Forord

Denne håndboka er resultat av et samarbeid i Statistisk sentralbyrå (se metodegruppas forord neste side). En rekke personer i EDB-miljøet har bidratt. Metoden er behandlet og vedtatt i EDB-utvalget i januar 1997.

Vi vet at de fleste fagområder, fra statistikk og sosialøkonomi til Picassos maleri, har retningslinjer som fagfolkene følger når de arbeider. Også innen faget EDB er det slike retningslinjer, og en del av disse retningslinjene er nå samlet i denne håndboka under navnet «*metode*».

Noe av det viktigste Statistisk sentralbyrå venter å oppnå ved å utgi denne håndboka er at når et EDB-prosjekt er slutt foreligger det en god dokumentasjon av prosjektet. Metoden fokuserer sterkt på at all dokumentasjon skal lagres elektronisk og samlet på et *fastsatt sted* slik at vi lett kan finne fram til den. På lengre sikt vil den metodiske arbeidsmåten gi en mer ensartet dokumentasjon av statistikken i Statistisk sentralbyrå enn det vi til nå har sett.

Når vi bruker metoden i arbeidet for å løse en oppgave, bruker vi *standardiserte* teknikker. En fordel med at alle bruker de samme teknikkene er at vi da forstår hverandre raskere og mer nøyaktig. Med andre ord blir *kommunikasjonen* bedre - både EDB-medarbeidere seg i mellom på den ene side og mellom brukere og EDB-medarbeidere på den andre.

En metodisk arbeidsform hjelper oss også til å gjøre de riktige aktivitetene. Metoden gir oss blant annet en oppskrift for hvordan brukerne best kan samarbeide med EDB-medarbeiderne. Og når flere jobber sammen får vi automatisk en *kvalitetssikring* av prosjektet.

Til slutt noen ord om fremtiden. Det er nedsatt en gruppe som har redaksjonsansvar for metoden i Statistisk sentralbyrå. Denne gruppa vil ta stilling til alle endringsforslag som sendes gruppa, og de mest konstruktive bidrag vil inkluderes i metodehåndboka. Ikke minst regner vi med at fasen «Innledende arbeid ved fagseksjonen», som er spesielt utarbeidet for forholdene ved Statistisk sentralbyrå, vil forbedres en del i tiden frem til den neste versjonen av metodehåndboka kommer ut.

Statistisk sentralbyrå
Oslo/Kongsvinger, 21. februar 1997

Svein Longva

Forord fra metodegruppa

Bakgrunn

SSBs systemutviklingsmetode er en forenklet versjon av systemutviklingsmetoden SE-Express, levert av konsulentfirmaet ISI A/S. SE-Express er en videreføring og forenkling av den engelske systemutviklingsmetoden SSADM/System Engineering. «Rapport fra prosjektet Metodisk systemutvikling» 30. mars 1992 og «Skal SSB innføre systemutviklingsmetode?», 1. november 1993 forteller hvorfor og hvordan SSB tok utgangspunkt i SE-Express. Basert på erfaringene med «Rapport fra prosjektet Metodisk systemutvikling» ble det laget en «Minimetode» i 1994. Våren 1996 ble det etablert en arbeidsgruppe bestående av Erik Sjøberg@203@Oslo, Bjørn Roar Joneid@303@Oslo, Jan Erik Wålberg@403@Kvgr, Stein Tore Dale@701@Kvgr og Svein Jåvold@710@Oslo. Denne gruppa har *redaktøransvaret* for denne håndboka og for *metoden*. Men mange andre har bidratt til notatet: Fra EDB-siden Knut Vidar Hoholm, Hans Kristian Torvbråten, Liv Daasvatn, Håkon Berby, Jan Rønningen, Roy Lofthus, Sverre Nordseth, Solveig Lyby, Jon Folkedal, Per Olav Lande, Randi Wølner og Kristian Lønø. Fra brukersiden har Lars Sundell, Harald Dale-Olsen, Ove Vigen, Espen Sørensen, Anne Mundal og Jon Holmøy vært med på uttestingen og bidratt med synspunkter. Den fasen hvor brukerne forbereder EDB-arbeidet («Innledende arbeid ved fagseksjonen») kommer til å bli forbedret i neste versjon av metoden. Leiv Solheim og Rune Gløersen har ivaretatt synspunkter fra statistikkfaglig metode og prosjektstyring.

Andre håndbøker

Denne håndboka kan du selv skrive ut fra q:\dok\notater\handbok\metode96.doc. En del forhold som er nært knyttet opp mot systemutviklingsmetode er beskrevet i andre håndbøker. Det henvises derfor foreløpig til disse (*Håndbok i prosjektstyring*, *Håndbok for programmerere* med flere). Se referanselisten bak.

Veien videre

1. Elektronisk verktøy for dokumentasjon (DATA-DOK '97). Her får hele SSB blant annet en fellesløsning for definisjoner av variable. Systemet blir sannsynligvis tilgjengelig fra Word. Eventuelt også ONLINE hjelpesystemer/dokumentasjon.
2. Fasen «Innledende arbeid ved fagseksjonen» utarbeides mer detaljert.
3. EDB-Metode 96 utvides til å støtte objektorientert tankegang.
4. Opprydding: Vi vil lage en egen katalog for notater, håndbøker mm. på q:\dok\notater
5. Utgi «Håndbok for programmerere».
6. Legge metoden ut på Intranett.

Statistisk sentralbyrå
Oslo/Kongsvinger 10. februar 1997

Metodegruppa

Innholdsfortegnelse

1. Innledning: hvem, hva, hvor, hvorfor?	5
1.1 Leserveiledning	5
1.2 Hvordan er metoden oppbygd? (+ terminologi)	6
1.3 17 bud for EDB-systemer i SSB	8
1.4 Hvilke teknikker har vi til hjelp?	9
2. Innledende arbeid ved fagseksjonen	11
2.1 Statistikkfaglig forarbeid	12
2.2 Dataflytdiagram nivå 0 (DFD0)	13
2.3 Statistikkfaglig orientert grovspekifikasjon	15
3. EDB-initierings-fasen	17
3.1 EDB oppstart	18
4. Analyse-fasen	19
4.1 Datamodellering	20
4.2 Dataflytdiagram nivå 1 (DFD1)	23
4.3 Datadefinisjon	25
4.4 Detaljspesifikasjon	27
4.5 Prototyping	30
5. Gjennomførings-fasen	31
5.1 Programmering og testing	32
5.2 Fysisk dataflytdiagram (FDFD)	33
5.3 Dialogdesign	35
6. Avslutnings-fasen	37
6.1 Dokumentasjonsnotat	38
6.2 Sjekk dokumentasjon	41
6.3 Igangkjøring	42

7. Ordliste	43
8. Referanser	46
9. Vedlegg: Symbolbruk i diagrammene	47
10. Vedlegg: Dokumentasjonsverktøy	48
11. Vedlegg: Typisk dataflyt og datalagertyper i ssb	49
12. Vedlegg: Vedlikehold	50
13. Vedlegg: Noen erfaringer etter bruk av metoden	52

1. Innledning: hvem, hva, hvor, hvorfor?

1.1 Leserveiledning




Denne håndboka kan deles i tre deler: oversikt, detaljer og stoff til orientering. Stoffet er ordnet slik:

Kapittel 1 gir en *oversikt* over metoden. Merk spesielt kapittel 1.3 («12 bud for systemereren i SSB») som gir en kort tekstlig oppsummering av metoden. Dette blir utdypet i kapittel 1.4. Bli ikke skremt av kap. 1.4: Figuren er enkel, hvis du studerer den nøye. Men hvis du synes at den virker for komplisert, kan du hoppe over den. (Det viktigst står i kapittel 1.3.)

Kapitlene 2-6 utgjør den andre delen og er en mer *detaljert* gjennomgang av fasene og teknikkene i metoden. Stoffet er presentert i den rekkefølgen aktivitetene skal utføres i arbeidet med å lage EDB-systemet, og informasjonen i denne delen av håndboka er meget strengt strukturert: Hver teknikk starter med en tabell som gir noen sentrale fakta om denne teknikken:

Idé	Idéen som ligger bak å bruke teknikken i SSB.
Ambisjonsnivå	Hvilket ambisjonsnivå vi legger oss på i SSB (i denne utgaven av metoden).
Notat	Henvisning til eventuelt notat som er skrevet i tilknytning til denne teknikken.
Mal	Angivelse av hvor malen (eksempelen) finnes lagret på elektronisk form.
Ansvarlig	Hvem som er ansvarlig for resultatet av denne teknikken.
Utførende	Hvem som bruker teknikken og legger frem resultatet.

Deretter følger noen punkter om hvordan teknikken konkret brukes:

	Gir en beskrivelse av hvordan teknikken utføres
	Oppsummerer erfaringer som er gjort med denne teknikken i SSB.
	En mer teknisk anmerkning om denne teknikken.
EKSEMPEL	Et eksempel på bruk av teknikken anvendt i SSB. I tilknytning til denne er det listet opp hvilke symboler som er lovlige i teknikken.

Resten av håndboka består av utfyllende stoff. Ordlista gir en tekstlig beskrivelse av mye brukte metodiske uttrykk. Referanselista gir mer nøyaktige referanser til viktige notater som har tilknytning til metoden. Vedleggene består av oppsummerte fakta om metoden og om bruk av verktøy. De siste vedlegget viser prinsippet for et typisk prosjekt i Statistisk sentralbyrå og skisserer en metode for vedlikeholdsprosjekter.

Trenger alle å lese hele håndboka? Nei, det er ikke nødvendig! Her er et forslag til hvem som bør lese hva:

Ledere som ønsker informasjon om hva metoden kan tilby og hvordan den er oppbygd, bør spesielt studere kapittel 1.

Brukere fra fagseksjonene bør spesielt studere kapittel 2 og 6. Men kapittel 1 er selvsagt også nyttig som bakgrunn.

Kundestøtte som har ansvar for tilrettelegging av dokumentasjon bør studere organiseringen av Q-disken (kapittel 3) og de spesiellagede Word-kommandoene (kapittel 4.4).

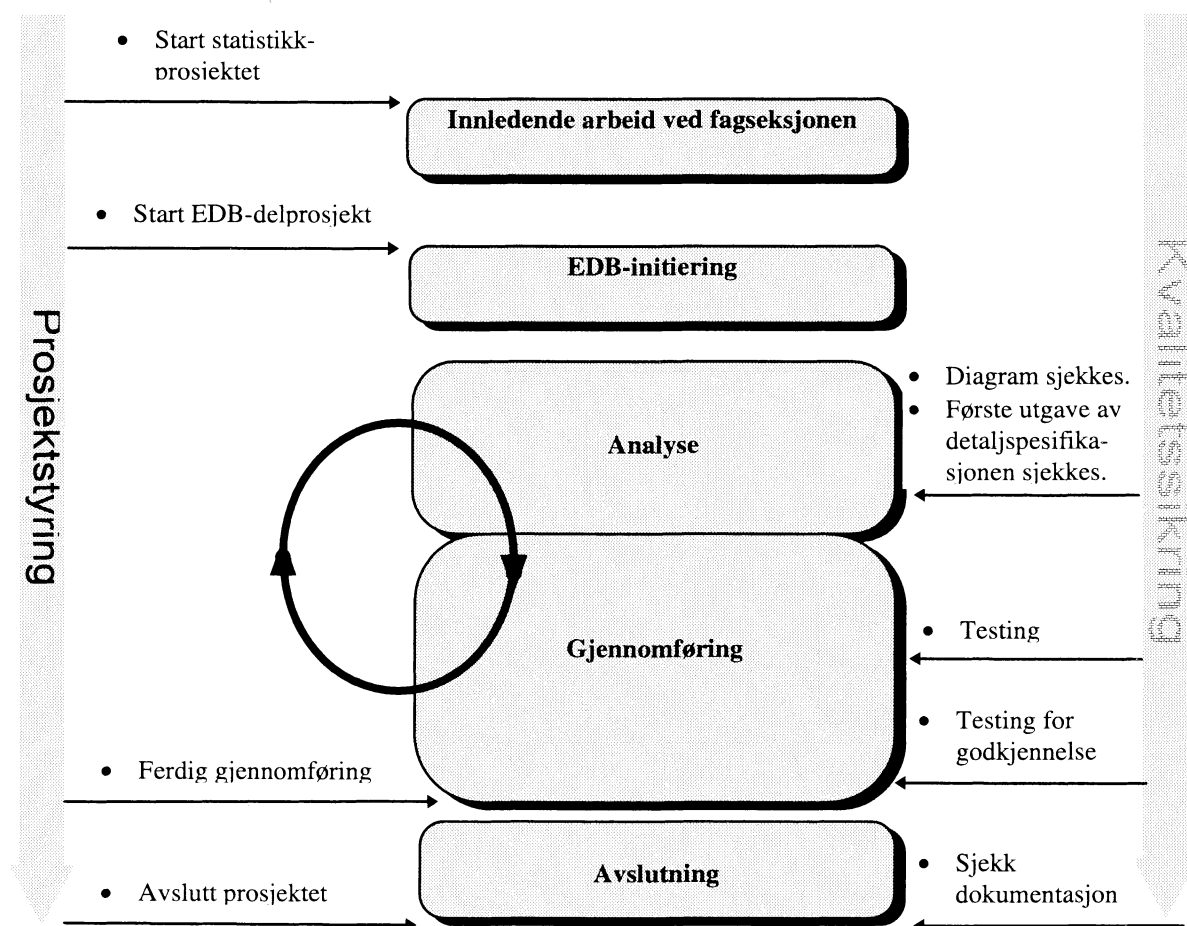
EDB-kontor-lederne som har ansvaret for å kvalitetssikre resultatet fra EDB-delen av prosjektet bør lese kapittel 1, 3 og 6.

Systemutviklerne (og spesielt den som er EDB-ansvarlig for prosjektet) bør lese hele håndboka nøye og siden bruke det som oppslagsverk.

1.2 Hvordan er metoden oppbygd? (+ terminologi)

«Metoden» beskriver hvordan et EDB-system bør utvikles. Dette gjør den ved å følge prosjektet fra *prosjektskriv*¹ til ferdig levert EDB-system. Hele tiden foreslår metoden hvordan arbeidsoppgaven kan gjøres for å få det best mulige resultat som utgangspunkt for neste trinn i arbeidet.

Prosjektarbeidet gjennomgår flere *faser*. Figuren nedenfor viser hvilke faser prosjektet gjennomgår og hvordan fasene følger etter hverandre i tid. På figuren er hver fase angitt med en boks med navnet på fasen. Ved venstre og høyre marg ser vi to piler som viser hvordan henholdsvis *prosjektstyring* og *kvalitetssikring* henger sammen med fasene i metoden. Fra prosjektstyringen i venstre marg skyter det ut piler som illustrerer hvordan ledelsen griper inn i prosjektet og styrer dette. Fra høyre marg illustrerer pilene tilsvarende hvilke tiltak som treffes for å sikre kvaliteten på delresultatene i prosjektet.



Figur: Oversikt over metoden.

¹ Metoden kan også brukes uten av prosjektskriv foreligger eller at arbeidet er organisert som et prosjekt.

Hver av fasene har sin bestemte oppgave:

- *Innledende arbeid ved fagseksjonen:* sikrer et godt forarbeid fra fagseksjonen
- *EDB-initiering* sikrer ryddig oppstart på EDB-arbeidet og skaper infrastruktur
- *Analyse* sikrer detaljert spesifikasjon av det systemet som skal lages
- *Gjennomføring* sikrer testet koding av EDB-systemet
- *Avslutning* sikrer at dokumentasjonen blir ferdig og avslutter prosjektet

Først gjøres et innledende arbeid ved fagseksjonen. Her legges grunnlaget for en ryddig utvikling av EDB-systemet. Dette arbeidet ender opp med en *statistikkfaglig grovspekifisering* av systemet som viser hvordan brukerne kunne ønske seg at det nye systemet skal være. Etter at fagseksjonen har foretatt dette innledende arbeidet, er det klart for å starte opp EDB-delen av prosjektet.

Arbeidet i *EDB-initieringsfasen* utføres i stor grad av prosjektlederen og den som er EDB-ansvarlig i prosjektet.

Analysefasen og *gjennomføringsfasen* henger nøye sammen. Når prosjektet er i analysefasen samarbeider brukerne og systemutviklerne om å skape den best mulige realiseringen av brukernes ønsker - når en tar omstendighetene i betraktning. Analysefasen ender opp med detaljerte spesifikasjoner for hvordan systemet skal lages. Når analysen anses som ferdig går prosjektet over i *gjennomføringsfasen* (koding og testing). Ofte vil det bli oppdaget mangler og uklarheter i detaljspesifikasjonen fra analysefasen, og det kan bli nødvendig å gå tilbake å gjøre endringer i spesifikasjonene. Denne iterasjonen mellom de to fasene «analyse» og «gjennomføring» symboliseres vha. en sirkel på figuren².

Avslutningsfasen er i hovedsak viet systematisering av dokumentasjonen i prosjektet - samt å få avsluttet prosjektet på en ryddig måte. Alt arbeid utover dette defineres som et nytt prosjekt - et vedlikeholdsprosjekt.

Hver av disse fasene består av en samling *teknikker*. Teknikkene brukes for å løse de utfordringene som oppstår i arbeidet med å utforme en EDB-løsning på brukernes ønskede system. Som et resultat av en teknikk får vi enten et tekstdokument eller et *diagram*. I neste avsnitt viser vi en oversikt over teknikkene, mens resten av håndboka kan leses som en manual som beskriver hver av teknikkene i detalj.

² Det må understrekes at forutsetningen for å kalle systemutvikling «metodisk» er at man får klarlagt kravene til systemet før systemet programmeres. Et kjennetegn ved metodisk systemutvikling er at man benytter diagrammer og skriftlige spesifikasjoner ved gjennomføringen av prosjektet. Men erfaringer fra SSB (og resten av verden) viser at man ikke alltid klarer å få i stand en perfekt detaljspesifikasjon av EDB-systemet som skal lages før man starter gjennomføringen. Derfor tillater SSBs metode *iterasjon* (at man kan gå tilbake til analysefasen for å gjøre endringer i spesifikasjonene til systemet, også etter at man har påbegynt gjennomføringen). Selv om vi må tilstrebe å få færrest mulig iterasjoner mellom analyse og gjennomføring må vi holde fast på at det viktigste er at prosjektet ender opp med en god løsning og god dokumentasjon.

1.3 17 bud for EDB-systemer i SSB

Innledende arbeid hos ledelsen

❶ **Prosjektskriv** for hele prosjektet. Prosjektskrivet inneholder noen basisopplysninger om prosjektet. Målet med prosjektet, hvor lang tid det forventes å ta og litt om behov for ressurser. Prosjektlederen har ansvaret for denne fasen, og fasen er beskrevet i detalj i «Håndbok i prosjektstyring».



Innledende arbeid ved fagseksjonen

❷ Fagseksjonen starter opp arbeidet med *statistikkfaglig forarbeid*. I denne teknikken gjør de seg kjent med dataene og velger metode. Arbeidet varierer i omfang og form etter type prosjekt.

Når dette er gjort kan brukerne sammen tegne en skisse over systemet i et *dataflytdiagram nivå 0* for å få oversikt over hvilke data som går inn og ut av systemet og hvem som eier disse dataene.

❸ **Statistikkfaglig orientert grovspesifikasjon** danner utgangspunkt for logisk-dataflytdiagrammet og eventuell datamodell i analysefasen. Her skal brukerne beskrive det systemet de ideelt ønsker seg, uten å bruke EDB-språk.



EDB-initierings-fasen

I denne fasen settes EDB-delen av prosjektet i gang på en ryddig måte.

❹ **EDB-prosjektoppstart:**

- Det opprettes en prosjektgruppe som består av minst én EDB-person (*systemutvikler*) og minst én fagperson (*bruker*).
- Hvis du bruker UNIX, må kundestøtte lage en stammekatalog på UNIX (Se notat for «Navnstandard på UNIX».)
- Sørg for at kundestøtte oppretter en katalog *q:\dok\<prosjektnavn>* hvor alle som jobber i prosjektet har skrivetillatelse.
- Prosjektnavn skal være det samme som stammekatalog på UNIX (hvis en slik benyttes i prosjektet).



Analyse-fasen

I denne fasen analyseres det systemet som skal lages (*kravspesifikasjon*). Analysen lages i samarbeid mellom brukeren og systemutvikleren:

- ❶ På grunnlag av brukerens «statistikkfaglig orientert grovspesifikasjon» tegner systemutvikleren og brukeren sammen opp en oversikt over systemet (*logisk dataflytdiagram*). For å sikre at diagrammet er så riktig som mulig, lønner det seg ofte å tenke litt mer detaljert enn det en tegner opp på diagrammet. Resultatet skal foreligge i et CASE-verktøy.
- ❷ Systemutvikleren (med hjelp av brukerne) lager nå en *datamodell* for å få oversikt over sammenhengen i dataene. Før en først til en god datamodell, er systemet godt forstått! Datamodellering skal alltid brukes i store prosjekter og i prosjekter hvor relasjonsdatabaser inngår. Resultatet skal foreligge i et CASE-verktøy.
- ❸ Brukerne skriver *detaljspesifikasjonen* i et tekstbehandlingsverktøy (Word). Hver prosess og datalager fra teknikken «logisk dataflytdiagram» beskrives med et avsnitt eller kapittel i detaljspesifikasjonen. Systemutvikleren diskuterer denne spesifikasjonen direkte med brukeren (ikke bare mottar dokumenter eller enda verre - får en del «uforpliktende» spesifikasjoner i telefonen).
- ❹ Kvalitetssikring av arbeidet kan gjøres ved at brukeren og systemutvikleren sammen går gjennom diagrammene og detaljspesifikasjonen og forsøker å finne flest mulig feil og mangler. Spesifikasjonen er ferdig når det ikke lenger synes å være åpenbare mangler i detaljspesifikasjonen eller diagrammene selvsagt
- ❺ Ikke minst i større systemer kan det være nyttig å lage en *prototype* av brukergrensesnittet i systemet. Systemutvikler og bruker jobber da sammen om å lage og knytte sammen skjermbildene. Arbeidet er ferdig når brukergrensesnittet fungerer tilfredsstillende.



Gjennomførings-fasen

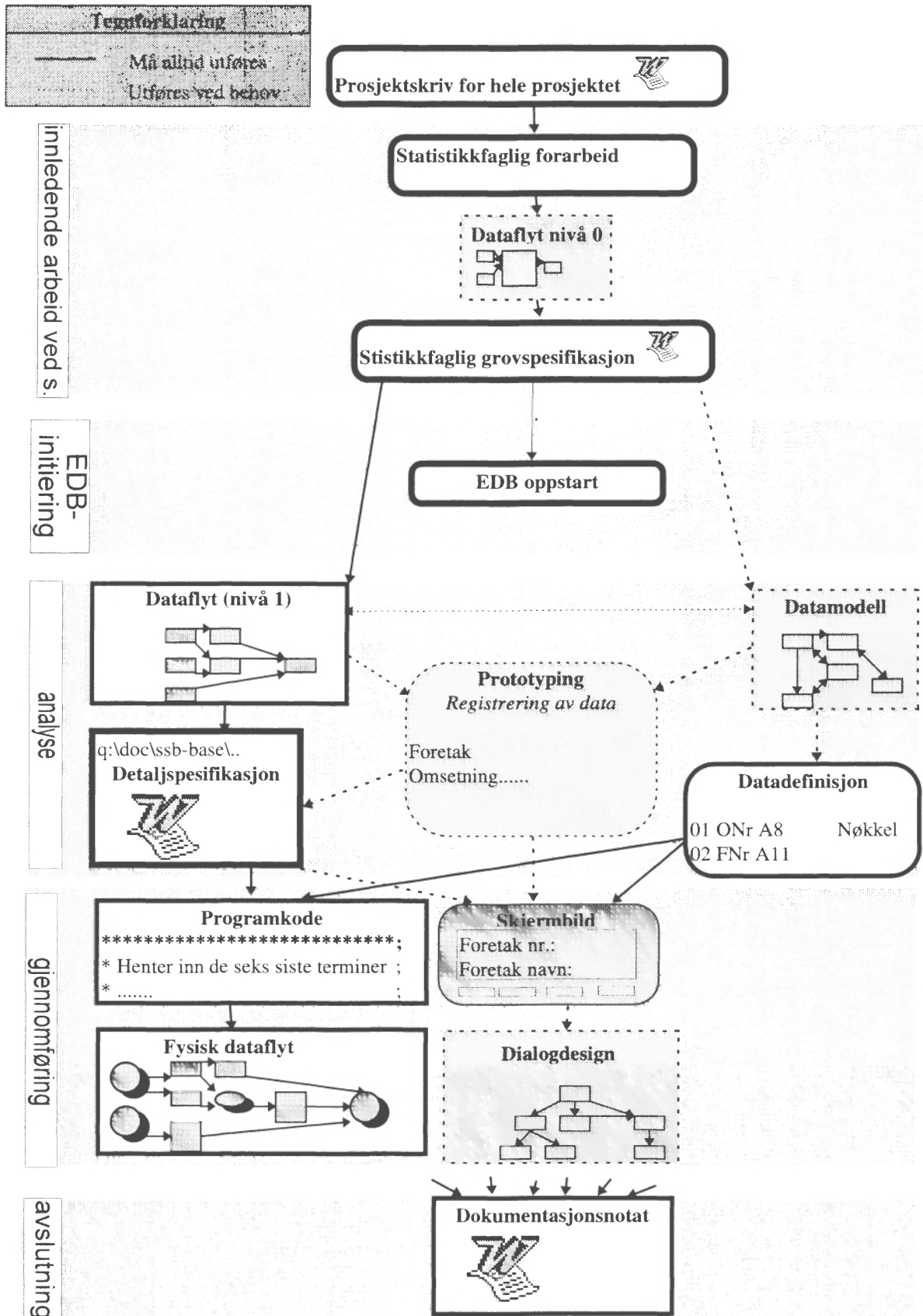
- ❶❶ Programmene kodes i det valgte verktøy: I starten av hvert program lages et «hode» som blant annet angir hva programmet gjør, hvem som har skrevet det og beskrivelse av inn- og ut-filer/tabeller.
- ❶❷ Programkoden testes.
- Dersom det på tross av kvalitetssikringen i analysefasen skulle oppdages feil eller mangler i detaljspesifikasjonen, er det selvsagt mulig å gå tilbake og gjøre deler av analysefasen om igjen («iterasjon»). Men husk at iterasjoner tar tid!
- ❶ Hvis det inngår mange skjermbilder i løsningen, tegner systemutvikleren en *dialogdesign* som viser sammenhengen mellom bildene og hvordan du kommer fra et bilde til det neste.
- ❶❶ Systemutviklerne tegner opp *fysisk dataflytdiagram* som viser sammenhengen mellom programmene.
- EDB-løsningen gjennomgår en *akseptansetest* på at systemet fungerer i samsvar med de spesifikasjonene brukerne gav.



Avslutnings-fasen

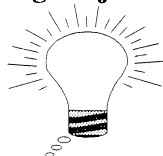
- ❶❶ Bruker og systemutvikler samarbeider om å skrive *dokumentasjonsnotatet*.
- ❶❶ Avdelingens EDB-leder påser at dokumentasjonen på katalogen *q:\dok\<prosjekt>* er i henhold til retningslinjene.
- ❶❶ Dokumentasjonsnotatet gis ut i serien *Interne dokumenter*.

1.4 Hvilke teknikker har vi til hjelp?



2. Innledende arbeid ved fagseksjonen

Innledende arbeid ved fagseksjonen



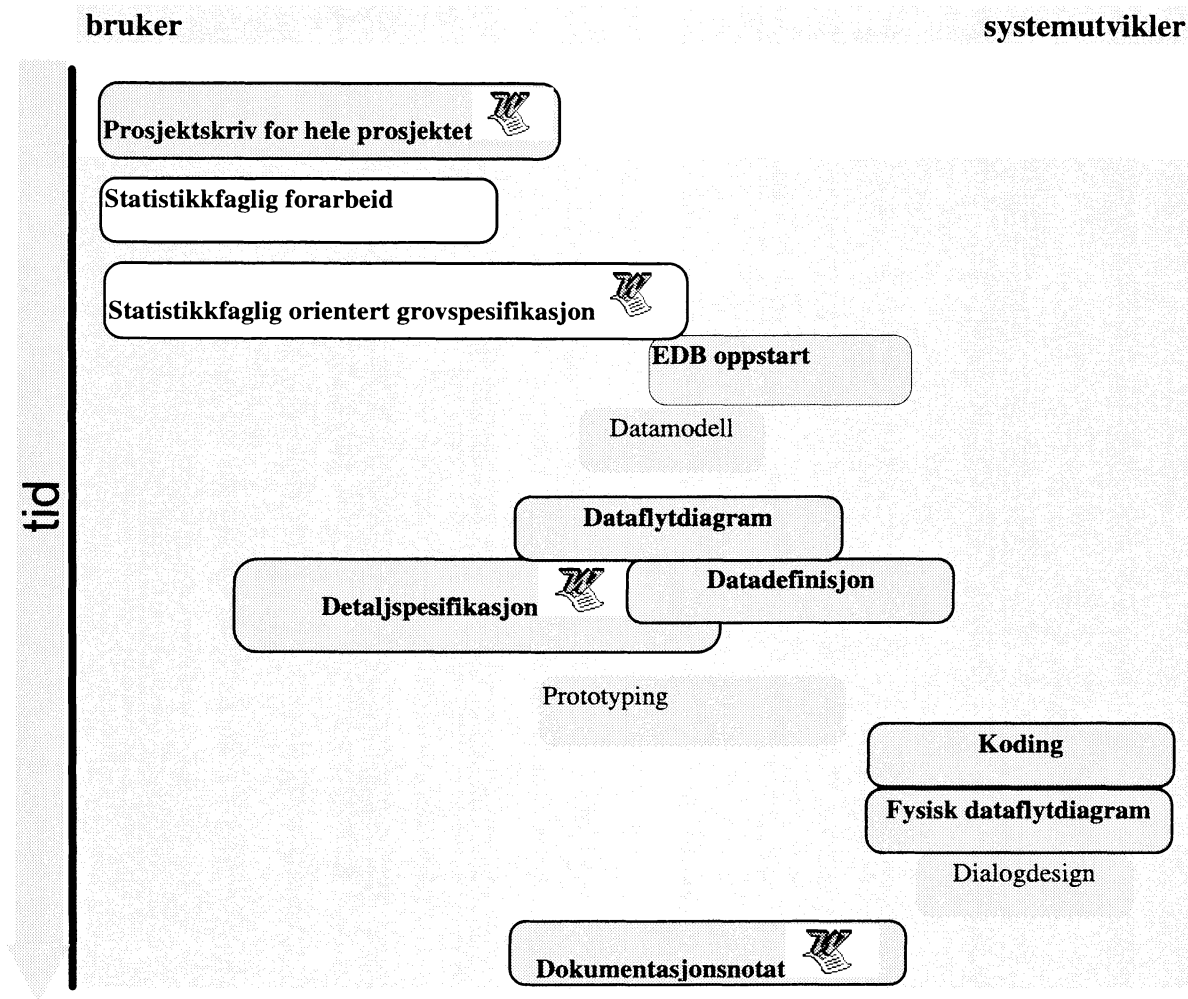
EDB-initiering

Analyse

Gjennomføring

Avslutning

Idé	Ryddig overgang fra statistikkseksjonens til systemutviklernes arbeid.
Mål	Skape et godt grunnlag for systemutviklernes videre arbeid.
Inn til fasen	1. Prosjektskriv for hele prosjektet.
Ut av fasen	1. Statistikkfaglig forarbeid 2. Statistikkfaglig orientert grovspeifikasjon



Figur 2.1: Kommunikasjonen mellom bruker og systemutvikler.

2.1 Statistikkfaglig forarbeid

Idé	Statistikkfaglig seksjon forbereder seg til å spesifisere det EDB-arbeidet trenger av info og deling av arbeidsoppgaver.
Ambisjonsnivå	Obligatorisk
Notat	-
Mal	-
Ansvarlig	Statistikkfaglig ansvarlig (bruker)
Utførende	Statistikkfaglig ansvarlig (bruker)



Beskrivelse:

Før det skrives notater eller tas kontakt med EDB-medarbeidere er det en del ting som må være klarlagt:

1. Forarbeid med data:
 - fastlegge hvilke data som skal brukes
 - få tilgang til dataene
 - hente dataene
2. Hvilke enheter består data av (f. eks. bedrift, person, bil, utdanning)
3. Kvaliteten på data
 - manglende data (fracfall)
 - forsinkelser på data
4. Statistikkfaglig metode velges (beregningsmetoder osv.)
5. For prosjekter større enn 1000 timeverk skrives eget prosjektskriv for EDB-delen av prosjektet.



Erfaringer:

1. Når datagrunnlaget er registre er ofte dataene ikke godt nok kjent av fagseksjonen. Dette kan ofte føre til at systemutviklerne kommer til å jobbe med for dårlige spesifikasjoner.
2. Denne teknikken vil bli videreutviklet i neste versjon av metoden. Særlig begrepet «statistikkfaglig metode» vil bli konkretisert. Teknikken gir imidlertid allerede i sin nåværende form viktige bidrag til å hjelpe med fagseksjonens forberedelser til EDB-arbeidet.
- 3.

2.2 Dataflytdiagram nivå 0 (DFD0)

Idé	Kartlegge omgivelsene til det informasjonssystemet vi nå skal lage.
Ambisjonsnivå	Frivillig. Angi kun leverandører og mottakere av data (ev. input/output-filer). Kan tegnes manuelt.
Notat	-
Mal	-
Ansvarlig	Bruker
Utførende	Bruker



Beskrivelse:

1. Prosjektleder arrangerer et møte med brukerne hvor det blir tegnet opp et *dataflytdiagram nivå 0* (DFD0).
2. Om opptegningen av diagrammet DFD0:
 - Viser avsendere og mottakere av data og hvilke data som flyter ut og inn av systemet.
 - NB: Det som skjer *inne* i systemet ignoreres!
 - Diagrammet tegnes som grunnlag for *Statistikkfaglig orienterte grovspefisikasjonen* fra det *Innledende arbeid ved fagseksjonen*.







Erfaringer:

1. Diagrammet hjelper med å få startet opp arbeidet med spesifikasjonene. Teknikken gir en god oversikt over hvilke datakilder som inngår i systemet, og er mer nyttig enn det kan se ut til ved første øyekast.
2. Etter at dataflytdiagram nivå 0 (DFD0) er ferdig kan det *kopieres* til et nytt diagram som utvides til også å vise innholdet i systemet (DFD1). Det er først i analysefasen at dataflytdiagrammet tegnes opp detaljert.

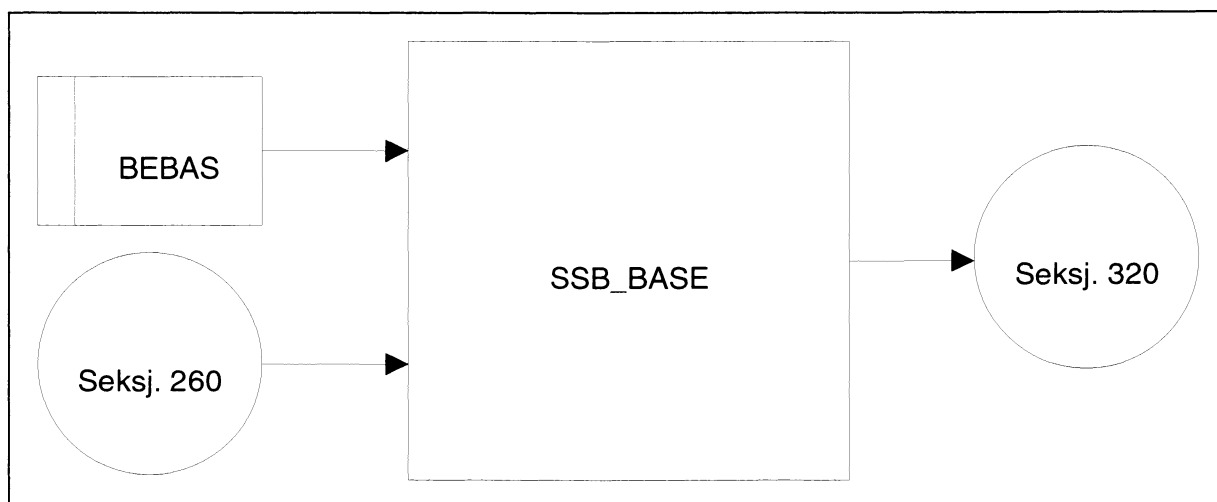
Teknisk anmerkning:

1. Vi bruker samme tegneteknikk i DFD0 og DFD1 (logisk dataflyt nivå 1 i analysefasen), men symbolbruken er noe forskjellig.
2. Det er lurt å bruke en tavle når diagrammet tegnes opp. Da blir alle feil tydelige og det øker den felles forståelsen.

Tillatte symboler i teknikken (SSADM) Data Flow:

Symbol				
Betydning	Dataflyt retning	Dataleverandør/ datamottaker	Datalager	EDB-systemet som skal lages

Vi kan vise datakilder/datamottakere (○) og/eller input/output-kilder (□).

Eksempel:

2.3 Statistikkfaglig orientert grovspesifikasjon

Idé	Statistikkfaglig seksjon tenker grundig gjennom systemet som skal lages. NB! Ta <i>ikke</i> hensyn til EDB-språk eller begrensninger i EDB-systemer.
Ambisjonsnivå	Obligatorisk. Word-notat med omfang avhengig av prosjekttype og størrelse.
Notat	-
Mal	q:\dok\metode\grovspec.doc
Ansvarlig	Prosjektleder (for hele prosjektet)
Utførende	Statistikkfaglig ansvarlig (bruker)



Beskrivelse:

1. En eller flere brukere (statistikkfaglig ansatte) skriver en grovspesifikasjon som et dokument. De som skriver må være bevisst på følgende:
 - Beskriv det du virkelig vil ha, ikke det som du tror er EDB-teknisk mulig. Hva som er mulig avgjøres i samarbeid med systemutviklerne i EDB-delen av prosjektet.
 - Ikke gå for mye i detaljer. Detaljene spesifiseres i *analysefasen* i samarbeid med systemutvikleren. Bruk dine egne ord. Ikke EDB-ord.
 - Framgangsmåten bør være at man har et «tenk-høyt»-møte (brainstorming). Resultatet fra dette organiseres i skrivet på neste side. Beskriv systemet som skal lages ut fra en statistikkfaglig synsvinkel. Dermed begrunner du også hvorfor dette er en fornuftig løsning. Denne spesifikasjonen danner grunnlaget for en felles forståelse for systemet og utgjør utgangspunktet for dataflytdiagrammet (og eventuell datamodell) som tegnes i analysefasen.
2. Mal for grovspesifikasjonen er gitt nedenfor.



Erfaringer:

1. Vær oppmerksom på at prosjekter kan være av ulike typer og at disse krever ulikt forarbeid:
 - *Konverteringsprosjekt* (overføring av et system til ny maskinplattform). Her er hovedsaken at du beskriver hvordan ting var før. Ofte kan konverteringsprosjekter regnes som utviklingsprosjekter fordi vi samtidig ønsker å forbedre eller utvide det eksisterende systemet.
 - *Utviklingsprosjekt* (utvikling av nye rutiner). Her kreves det mye forarbeid. Ofte kommer systemererne både for seint og for tidlig inn i dette arbeidet. Grunnen til dette er at statistikkfaglig ansatte spesifiserer for detaljert før de tar kontakt med systemutviklerne. Slik kan de komme til å binde prosjektet til ikke-optimale løsninger. På den annen side hender det ofte at systemutviklerne blir satt til å programmere ting som burde vært en del av «*statistikkfaglig forarbeid*» (se forrige teknikk).
 - *Vedlikeholdsprosjekt* (vedlikehold på et prosjekt som er avsluttet). Her vil ofte systemutvikleren kjenne systemet på forhånd, slik at forarbeidet ikke blir så omfattende.
 - *Betalte oppdrag*. Dette er ofte hastprosjekter hvor spesifikasjonene kommer via telefon. Husk at også her gjelder regelen «hastverk er lastverk». Brukeren bør derfor også her som et minimum levere en statistikkfaglig orientert grovspesifikasjon til systemereren.
2. Dette tvinger oppdragsgiver til å tenke igjennom hva vedkommende ønsker ut av systemet, og man unngår såkalte telefonoppdragsprosjekter.
3. Statistikkfaglig orientert grovspesifikasjon har mange likhetstrekk med det som ofte kalles en «kravspesifikasjon».

SVE/- 24.05.96 SIST ENDRET: 01.07.96 Q:\DOK\SSB-BASE\PROSJEKT\GROVSPEC.DOC v. 1.1

Avdeling for personstatistikk, Seksjon for persondata

Statistikkfaglig orientert grovspefikasjon for SSB-BASE

<Innholdsfortegnelse>

1. Innledning

- Referanse til prosjektskriv hvis det finnes³

2. Statistikkfaglig metode

- Presentasjon av statistikkfaglige problemer
- Begrunne valg av statistikkfaglig metode og kort beskrivelse av metoden
- Eventuelle usikkerheter tilknyttet løsningen
- Hva bør forbedres i forhold til tidligere metode

3. Oversikt og krav til systemet

- Mål: Hvilke hovedoppgaver systemet skal løse
- Basis (tidligere systemer, vedtak, standarder mm)
- Avgrensning
- Hva må forbedres i forhold til tidligere systemer
- Systemets forhold til omgivelsene
- Målgruppe

4. Rent statistikkfaglig beskrivelse av det nye systemet

- Kopi av eventuelt Dataflyt-diagram nivå 0
- Input (hvilke datalagre som hentes inn til systemet)
- Kontroll (inkludert beskrivelse av datakvalitet)
- Bearbeiding (utvalg, avledede variabler m.m.)
- Lagring (lagring på mikronivå, aggregering, lagring på makronivå)
- Analyse (algoritmer)
- Output (hvilke datalagre og rapporter/tabeller som skal leveres fra systemet)

5. Tanker om eventuelle framtidige utvidelser

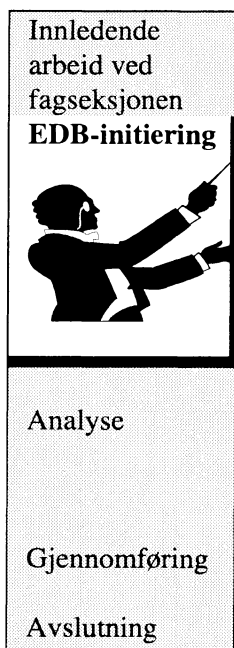
Vedlegg

- Ev. definisjoner
- Ev. skjema
- Referanser til relevante notater m.m.

³ Hvis prosjektskriv ikke finnes, skal innledningen inneholde følgende punkter:

- Bakgrunn for prosjektet
- Interessenter (oppdragsgiver, ev. styringsgruppe m.m.)
- Ansvarsforhold (prosjektleder, EDB-ansvarlig, eventuelt styringsgruppas og referansegruppas medlemmer)
- Presentasjon av statistikkfaglige problemer
- Referanser til annen dokumentasjon som kan være av interesse (hvor den finnes, hvem som står ansvarlig for den)
- Hvem som er berørt av endringene
- Systemets plassering i en større helhet

3. EDB-initierings-fasen



Idé	Ryddig start på EDB-delen av arbeidet i prosjektet.
Mål	Hente inn all den informasjon om prosjektet som allerede eksisterer. Slik skapes det ordnede forhold i EDB-delen av prosjektet når analysefasen starter.
Inn til fasen	<ol style="list-style-type: none"> 1. Statistikkfaglig forarbeid. 2. Dataflytdiagram nivå 0 (systemets omgivelser) [frivillig] 3. Statistikkfaglig orientert grovspeifisering.
Ut av fasen	-

Om dokumentasjonskatalogene:

I denne fasen opprettes «infrastrukturen» rundt EDB-prosjektet. Det lages blant annet dokumentasjonskataloger på Q-disken som i løpet av prosjektet fylles med resultatene fra de teknikkene som anvendes.

Dersom et prosjekt er delt opp i underprosjekter (med underkataloger), skal hvert delprosjekt utelukkende dokumenteres på sin tilhørende underkatalog. (Delprosjektene må ikke, helt eller delvis dokumenteres på hovedkatalogen.)

3.1 EDB oppstart

Idé	Starte EDB-delen av prosjektet.
Ambisjonsnivå	Oppfordring til at to eller flere systemutviklere jobber sammen om EDB-arbeidet i prosjektet.
Notat	- <i>Håndbok i prosjektstyring</i> - <i>Kort innføring i ny navnestandard for UNIX-filer</i> - <i>Datalagring på UNIX i Statistisk sentralbyrå</i>
Mal	-
Ansvarlig	EDB-leder, prosjektleder (ev. EDB-delprosjektleder)
Utførende	EDB-leder, systemutvikler (ev. EDB-delprosjektleder) og kundestøtte




Beskrivelse:

- EDB-leder har følgende funksjoner ved oppstarten av EDB-arbeidet i prosjektet:
 - Han mottar forespørselen fra fagseksjonen og velger ut prosjektdeltakere for EDB-delen av arbeidet i prosjektet.
 - Han velger ut en EDB-ansvarlig. Denne får funksjonen å lede EDB-arbeidet i prosjektet. Tittelen er EDB-ansvarlig, EDB-koordinator eller EDB-prosjektleder, alt etter størrelsen på prosjektet.
 - Han oppretter en prosjektgruppe som består av minst én *systemutvikler* (EDB) og minst én *bruker*. Hvis dette er samme person, bør det velges en kontaktperson ved den tilhørende EDB-gruppa som kan fungere som rådgiver for brukeren når han designer statistikksystemet og programmerer dette.
- EDB-ansvarlig har ansvaret for følgende:
 - Hvis det finnes et *prosjektskriv* eller et *EDB-delprosjekt-skriv*, tar EDB-arbeidet utgangspunkt i dette⁴.
 - Hvis prosjektet bruker UNIX, ber EDB-ansvarlig kundestøtte om å opprette en stammekatalog på UNIX. Samtidig spesifiserer EDB-ansvarlig medlemmer til lese- og skrivegruppene i dette prosjektet.
 - EDB-ansvarlig sørger for å be kundestøtte opprette en katalogstruktur for dokumentasjonen på PC. Han gir kundestøtte en liste over prosjektdeltakere.
- Kundestøtte er ansvarlig for:
 - å opprette en stammekatalog på UNIX lese- og skrivegrupper slik der er spesifisert av EDB-ansvarlig.
 - å opprette katalogstrukturen på PC: *Q:\DOK\ med underkataloger slik figuren i avsnitt 6.2 viser. Når et prosjekt er delt opp i delprosjekter, kan det være lurt å dele hovedprosjekt-katalogen opp i delprosjektkataloger. Vi får da en struktur som følger navnestandarden på UNIX. *q:\dok\⁵.**
 - at alle som jobber i prosjektet gis skrivetillatelse til den (de) aktuelle katalogen(e).
 - at prosjektnavn er det samme som stammekatalog på UNIX. (Dersom stammen på UNIX er delt inn i substammer, skal den samme stammeinndelingen brukes for prosjektets dokumentasjon på *Q:\DOK\.)*
- at listen over de som har skrivetillatelse på *Q:\DOK\ er identisk med tilsvarende liste på UNIX.*

⁴ Prosjektskrivet inneholder noen basisopplysninger om prosjektet: Målet med prosjektet, hva prosjektet skal dekke, hvor lang tid det forventes å ta, litt om behov for ressurser og bestemme prosjektets rammebetingelser. Merk at metoden kan brukes selv om det ikke foreligger noe prosjektskriv. Planlegging av tidsforbruk på personer må da gjøres i en *tidsplan* som utarbeides etter at *dataflytdiagram (nivå 1)* er skissert. Timeverkene fordeles da i prosjektet på personer og faser. Ferdigdatoen på hver av fasene kalles for en *milepel*. Det er milepelene som prosjektet følges opp etter. (Eventuell detaljplanlegging ned på enkeltoppgaver innen hver fase er utelukkende til bruk internt i prosjektet.)

⁵ Dette tilsvarer på UNIX: *<stamme>\<filklasse>\<type katalog>*

4. Analyse-fasen

Innledende arbeid ved fagseksjonen	Idé	Fastlegge brukernes ønsker og krav.
EDB-initiering	Mål	Lage spesifikasjonen til det systemet som skal lages. Brukerne er ansvarlig for den endelige kvaliteten på spesifikasjonen, men den lages i et nært samarbeid mellom bruker og systemutvikler.
Analyse	Inn til fasen	<ol style="list-style-type: none"> 1. Statistikkfaglig forarbeid. 2. Dataflytdiagram nivå 0 (systemets omgivelser) [frivillig] 3. Statistikkfaglig orientert grovspekifikasjon.
	Ut av fasen	<ol style="list-style-type: none"> 1. Datamodell (når relasjonsdatabaseløsning) 2. Logisk dataflytdiagram nivå 1 (DFD1) 3. Detaljspekifikasjoner 4. Ev. prototyping
Gjennomføring		
Avslutning		

Forslag til arbeidsform i analysefasen

1. Mål og tema for møtet er fastsatt på forhånd (f.eks. på forrige møte eller via mail).
2. Brukerne har tenkt gjennom problemstillingen før møtet og har et notatutkast med seg.
3. Notatet diskuteres i plenum. Bruk tavle og tegn diagram hvis mulig.
4. Etterarbeid fra møtet.
 - Brukerne endrer i spesifikasjonene
 - Systemutviklerne tegner elektronisk lagret diagram
5. Resultatene fra pkt. 4 er utgangspunkt for neste møte.

En brukers kommentar til fasen

1. Det brukes generelt for lite tid til planlegging i prosjektene, og det diskuteres løsninger i stedet for å klarlegge hvilke problemer prosjektet skal løse.
2. Brukerne er ikke involvert i prosjektet og blir påtvunget løsningene til systemereren.
3. Noen ganger er bruker og systemerer samme person. Da blir det ofte slik at han konsentrerer seg mest om den faglige oppgaven/resultater, ikke spesifisering og dokumentasjon.

4.1 Datamodellering

Idé	Oversikt over, og sammenheng mellom, dataene i systemet.
Ambisjonsnivå	Obligatorisk ved relasjonsdatabaser og store systemer. Ellers frivillig.
Notat	1. Skagestein (1991) 2. Clockey (1996) 3. Poland (1996)
Mal	-
Ansvarlig	Systemutvikler (ev. EDB-delprosjektleder)
Utførende	Systemutvikler (red) sammen med bruker



Beskrivelse:

1. Hvis løsningen er databaseorientert, innkaller EDB-ansvarlig alle prosjektdeltakerne til et møte hvor det tegnes en datamodell på tavla.
2. Datamodellen er en grafisk sammenheng mellom entiteter (objekter) som inngår i EDB-systemet. Hvis det skal ha noen hensikt å bruke datamodellen, må *minimum* følgende være med:
 - navn på entiteten
 - entydig nøkkel (identifikasjon) til entiteten
 - attributter (egenskaper) til entiteten



Erfaringer:

1. Teknikken er spesielt nyttig for å designe databasesystemer og særlig relasjonsdatabaser, men datamodellering kan være nyttig selv om ikke formålet er å få en relasjonsdatabase på tredje normalform⁶.
2. I SSB lagrer vi ofte data slik de kommer inn på skjemaene. Av og til ligger til og med hele skjemaet på samme record. Det kan diskuteres om dette krever datamodellering⁷. Eventuelt kan vi oppfatte det slik at vi i SSB modellerer forholdet mellom fysiske filer. På bakgrunn av dette er det naturlig at man i de alle fleste tilfeller konsentrerer seg om fysiske datamodeller.
3. Datamodellering har vist seg vanskelig å beherske på et tilfredsstillende høyt nivå. Men under veiledning av en (ev. innleid) ekspert kan prosessen utføres i gruppe (lagarbeid), og da gir den ofte en meget god læringseffekt.

⁶ Ved normalisering av data til 3. normalform har vi følgende retningslinjer:

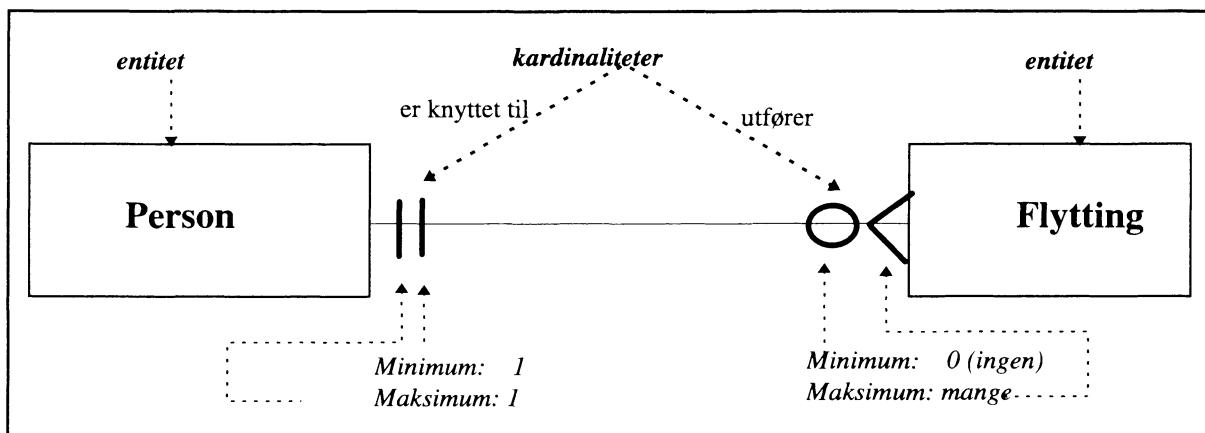
- 1. normalform: Fjerne repeterende grupper.
- 2. normalform: Fjerne felter som bare er avhengige av deler av nøkkelen.
- 3. normalform: Fjerne avhengighet mellom ikke-nøkkel-felter.

⁷ Vi kan kanskje si at det allerede har foregått en viss datamodellering ved design av skjemaet ved at de dataene (variablene) som hører sammen ofte er plassert nær hverandre på skjemaet.

Tillatte symboler i teknikken «Entity Relationship Datamodell» i

Designer/2000	Symbolbruk	ABC Flowcharter
Fysisk datatabell i databasen	Entitet	
Illustrerer relasjoner / sammenhenger mellom databasetabeller	Relasjonslinje	
Illustrerer relasjoner / sammenhenger mellom databasetabeller	Relasjonslinje med kardinaliteter / mengdeforhold NULL eller EN	
Illustrerer relasjoner / sammenhenger mellom databasetabeller	Relasjonslinje med kardinaliteter / mengdeforhold EN og BARE EN	
Illustrerer relasjoner / sammenhenger mellom databasetabeller	Relasjonslinje med kardinaliteter / mengdeforhold NULL eller MANGE	
Illustrerer relasjoner / sammenhenger mellom databasetabeller	Relasjonslinje med kardinaliteter / mengdeforhold EN eller MANGE	
Illustrerer relasjoner / sammenhenger mellom databasetabeller	Relasjonslinje med kardinaliteter / mengdeforhold FLERE ENN EN	

Teoretisk anmerkning: Anvendelse av relasjoner i datamodellering.

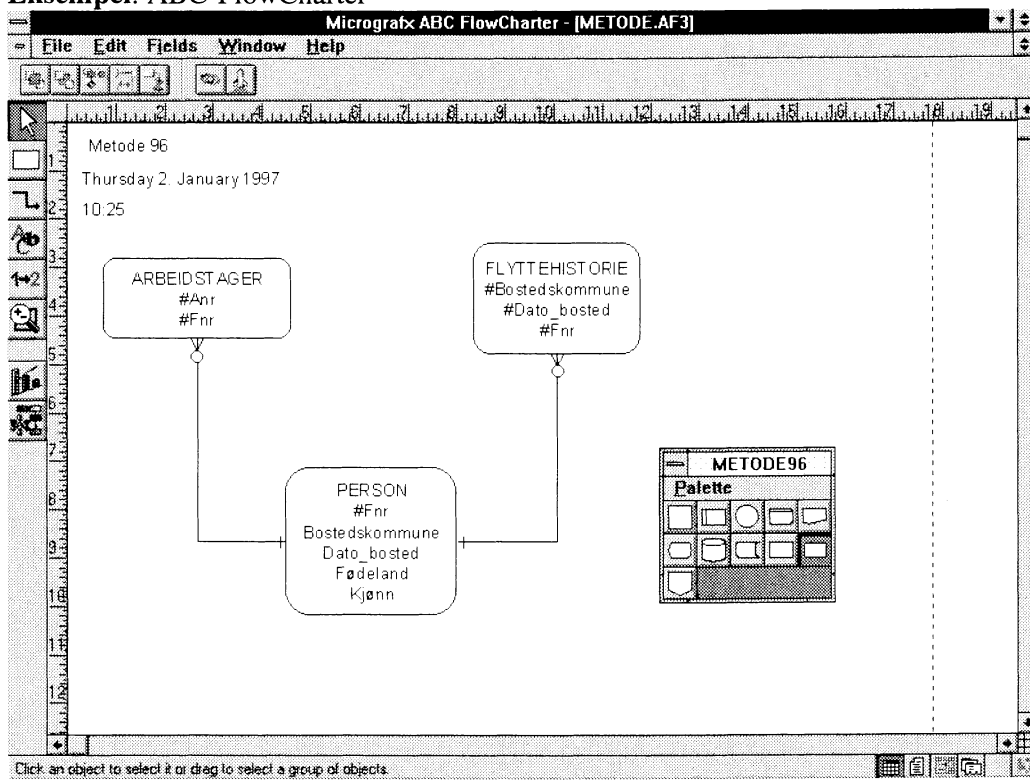


Figur: Elementer i datamodellen.

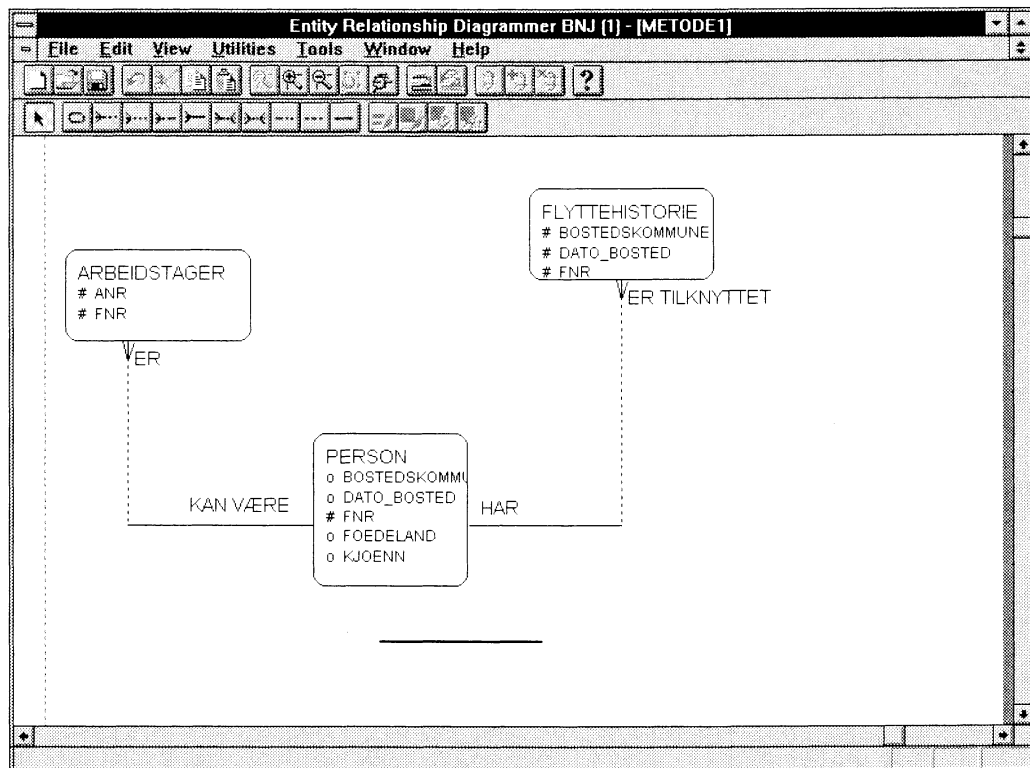
Figuren over viser et enkelt eksempel på hvordan datamodellering ved hjelp av ER-metoden kan se ut. Legg merke til betydningen av minimums- og maksimums-kardinalitet.

- Fra venstre mot høyre leser vi at *en person kan utføre flytting ingen, én eller flere ganger.*
- Fra høyre mot venstre leser vi at *en flytting er knyttet til minimum én person og maksimum én person.*

Eksempel: ABC-FlowCharter



Eksempel: Designer 2000 fra ORACLE



Legg merke til at Designer 2000 benytter andre notasjoner for minimum- og maksimumskardinaliteter.

4.2 Dataflytdiagram nivå 1 (DFD1)

Idé	Oversikt
Ambisjonsnivå	Obligatorisk. Oversikten bør kunne tegnes opp på én side i små prosjekter. I større prosjekter brukes også et nivå 2 (DFD2).
Notat	-
Ansvarlig	Systemutvikler (ev. EDB-delprosjektleder)
Utførende	Systemutvikler (red.) sammen med bruker



Beskrivelse:




1. Dataflytdiagram nivå 1 (også kalt *logisk dataflytdiagram*) bygger på den statistikkfaglig orienterte grovspesifikasjonen fra innledende fase.
2. EDB-ansvarlig samler alle prosjektdeltakerne for å tegne *dataflytdiagram nivå 1*. Dataflytdiagrammet gir prosjektdeltakerne oversikt over *prosesser (moduler)*, *datalagre (logiske filer)*, *dataflyt* mellom dem, *datakilder* og *datamottakere*. *Prosesser* er aktiviteter som mottar, transformerer og/eller sender ut data. Husk å få med alle datalagre!
3. En prosess kan igjen deles opp i flere *delprosesser* hvis det er behov for det. Dette blir da nye dataflytdiagram på nivå 2 (DFD2). La data flyte i én retning i diagrammet (fra venstre mot høyre eller ovenfra og nedover). Diagrammet får da en tidsakse slik eksemplene på neste side viser.
4. Husk at en prosess alltid kommer mellom to datalagre (og omvendt).



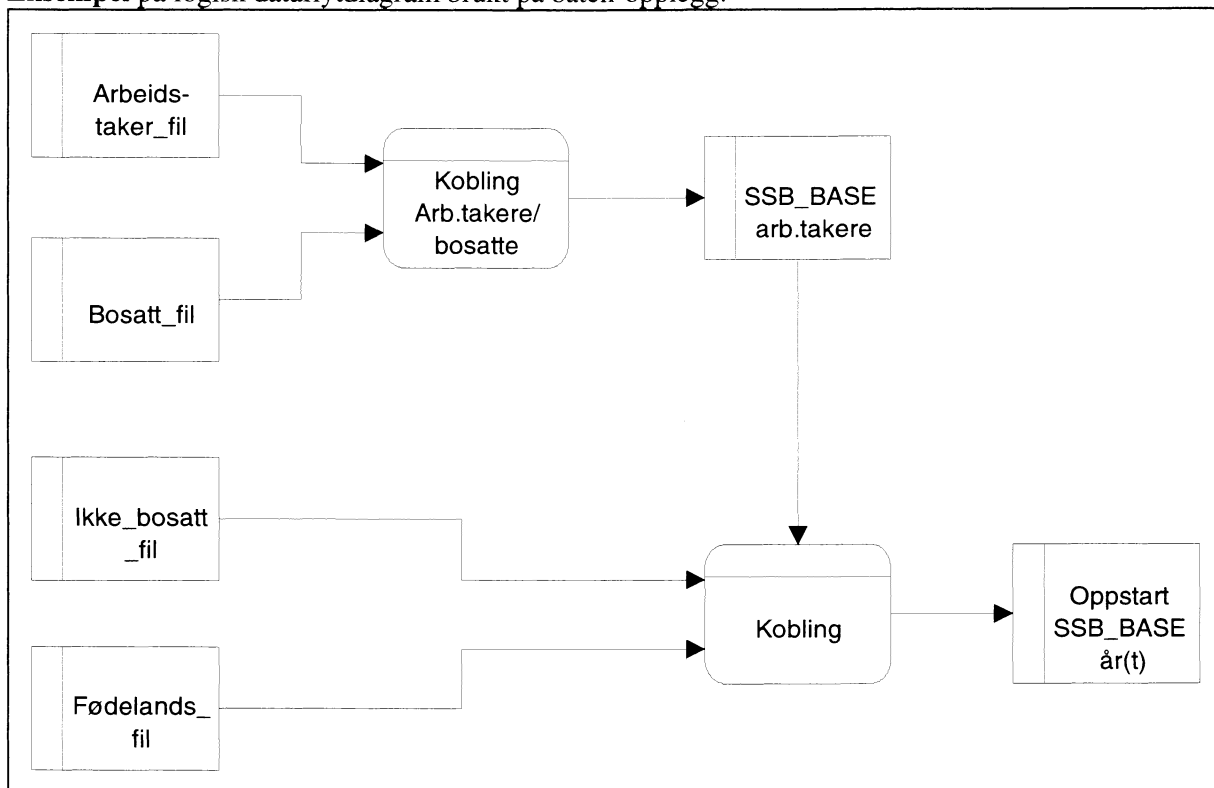
Erfaringer:

1. Dataflytdiagram nivå 1 (og eventuelt nivå 2) blir ofte blandet sammen med det fysiske dataflytdiagrammet i gjennomføringsfasen. Forskjellen er at logisk dataflyt tegnes *før* systemet er programmert for å finne ut hvordan det *bør* være. Fysisk dataflyt tegnes *etter* at systemet er programmert og forteller hvordan systemet virkelig ble. Fysisk dataflyt er derfor mye mer detaljert enn logisk dataflyt.
2. *Logisk dataflytdiagram* kunne også kalles for *oversiktsdiagram*. Det er mange grunner til at diagrammet har vist seg nyttig å bruke også i Statistisk sentralbyrå:
 - Å lage diagrammet gir en sjekk (kvalitetssikring) av spesifikasjoner fra brukeren at spesifikasjonene er riktige og at bruker og systemutvikler har forstått hverandre.
 - Det er en visuell teknikk og kan forstås av de fleste uten mye trening.
 - Det gir den nødvendige grove oversikten for å kunne velge løsning.
 - Det lager en modell som alle kan forstå av det systemet som skal lages.
 - Alle parter kan bruke diagrammet til å følge med hvor langt vi til enhver tid er kommet.
 - Det er en oversiktlig dokumentasjon av systemet til bruk for dem som ikke kjenner systemet fra før.
3. Det er viktig å være forberedt på mange endringer i diagrammet før det endelige utkastet i analysefasen er klart. Avsett tilstrekkelig med tid til dette arbeidet! Den tiden får du godt betalt for senere.
4. For å unngå altfor mange endringer i «den endelige» modellen er det som regel nødvendig å bryte opp noen av prosessene i dataflytdiagram nivå 1 slik at vi får diagrammer på nivå 2 (DFD2). På denne måten får vi en kontroll på at vi har tenkt langt nok, og vi kan være relativt sikre på at DFD1 beskriver en god systemløsning. Diagrammet bør derfor (i alle fall mer uformelt på papir) jobbes med ned på nivå 2 for å rette opp detaljer som kan påvirke diagrammet på nivå 1.
5. Det bør foreligge dataflytdiagram nivå 1 også for relasjonsdatabaseløsninger. Disse bør beskrive flyten av data inn til databasen (lasterutiner) og uttrekksrutiner til filer og rapporter. Det er imidlertid også mulig å beskrive prosessene internt i databasesystemet ved hjelp av dataflytdiagrammer. (Se eksempler på neste side.)

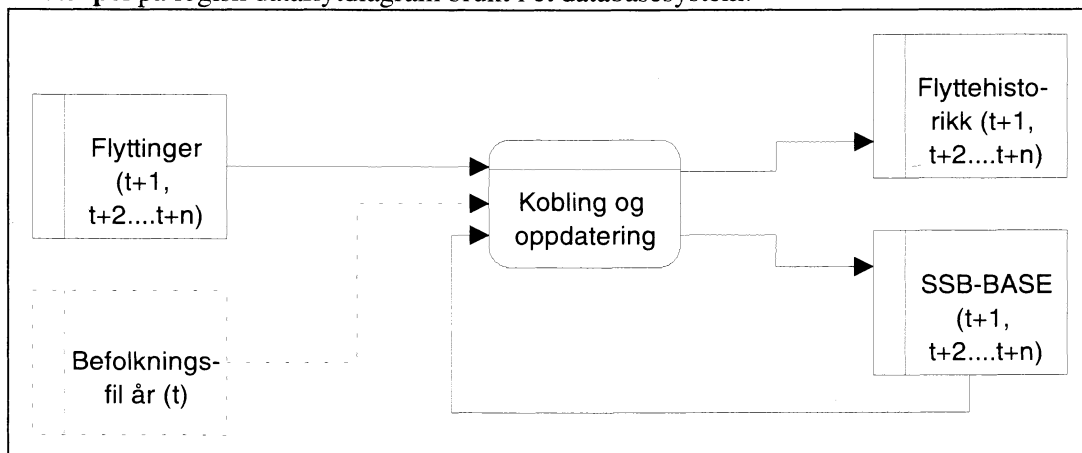
Tillatte symboler i teknikken (SSADM) Dataflow:

Symbol			
Betydning	Dataflytretning	Prosess	Datalager

Eksempel på logisk dataflytdiagram brukt på batch-opplegg:



Eksempel på logisk dataflytdiagram brukt i et databasesystem:



4.3 Datadefinisjon

Idé	Data defineres slik at alle i SSB har ON-LINE tilgang til dem.
Ambisjonsnivå	Obligatorisk. Alle filer som ikke er midlertidige dokumenteres.
Notat	<i>Datalagring på Unix i Statistisk sentralbyrå Kort innføring i ny navnestandard for Unix-filer Brukerveiledning for datadokumentasjonssystemet!</i>
Mal	-
Ansvarlig	Bruker
Utførende	Systemutvikler (red.) ev. med assistanse av bruker.



Beskrivelse:

- Følgende regler gjelder for lagring av data:
 - Alle data lagres på UNIX.
 - Alle data langtidslagres på sekvensielle filer, med fast recordlengde.
 - Alle langtidslagrede data og produksjonsdata dokumenteres i elektronisk verktøy.
 - Andre data kan dokumenteres i elektroniske verktøy hvis ønskelig.
- Inntil nye DATA-DOK er ferdig, lagres detaljert filbeskrivelse for sekvensielle filer på `q:\dok\<prosjekt>data`.
- Følgende aspekter ved produksjonsdata skal defineres (dokumenteres):
 - Navn på datalagrene i prosjektet (fil/databasetabell/datasett) med tilhørende beskrivelse.
 - Input-datalagre til systemet kan være definert tidligere.
 - Langtidslagrede filer som lages i prosjektet («produksjonsdata»).
 - Eventuelt andre filer som er spesielt nyttige.
 - Definisjon av entydige nøkler («Identer») fil tabellen/filen.
 - Variablene (på datalageret) med tilhørende beskrivelse.
 - Definisjonsmengde: Kodene (til variablene) med tilhørende beskrivelse.
- Metaformat: For flate filer har SSB ennå ikke definert noe fast format på dokumentasjonen. Vi foreslår derfor at denne standarden følger DATA-DOK '97 som defineres slik:



Erfaringer:

- Dokumentasjonen av data i SSB må økes for å unngå feilinformasjon.
- Unngå å kaste bort tiden på grunn av udokumenterte data.
- Midlertidige filer kan med fordel dokumenteres sammen med produksjonsdata, men dette er ikke obligatorisk.

Teknisk anmerkning

- I DATA-DOK '97 kan en hente fram opplysninger direkte inn i Word slik at informasjonen lett kan inkluderes i tekstdokumenter (dokumentasjonsnotatet osv.). Mens vi venter på et godt verktøy til å definere data (DATA-DOK⁸), er vi henvist til å definere datalagrene i andre steder. Datadefinisjoner som lages i ORACLE og SAS (kanskje også FAME⁹) kan kopieres over i DATA-DOK '97. Ingen data definisjon er derfor bortkastet.
- Hva så med dokumentasjon av sekvensielle datalagre (filer)? Under arbeidet med DATA-DOK er det utarbeidet et forslag til beskrivelse av dataene («metaformat») som vi kan bruke inntil DATA-

⁸ Dagens versjon av systemet støtter blant annet ikke dokumentasjon av variablenes koder.

⁹ Det er uklart om FAME-«databaser» skal beskrives i datadokumentasjons-systemet. Brukere vil neppe dokumentere serie for serie manuelt i datadokumentasjons-systemet. En variant er at det blir laget et program som genererer ønsket beskrivelse og dokumentasjon av FAME-baser og at dette oppdateres i datadokumentasjons-systemet automatisk.

DOK '97 kommer. Disse formatene vil kunne leses automatisk inn i DATA-DOK '97 når dette systemet er klart (sannsynligvis høsten '97). Beskrivelser av dette formatet følger på neste side, og det lagres på underkatalogen q:\dok\<<prosjekt>\data som er opprettet spesielt for dette formålet. Sekvensielle datalagre (filer) skal dokumenteres slik:

Datalager-definisjonen

1-8	9-64	65-72	73-75	76-78	79-178	179	180-189	190-290
stamme	Underkataloger (inntil 7 ledd) ¹⁰	Filnavn	Type	Seksjon	Beskr/def	Kr ¹¹	Ko ¹²	R ¹³
STRYG DFOB	data/bosatte/v01	g91m12	Sek	320	0	2332	...
SLONN	handel/data/detalj	g96	Sek	420			

Variabeldefinisjonen

1	2-33	34-41	42-61	62-65	66-70	71-72	73-74	75-94	95-194
Nøkkel	Navn	Kortnavn	Datatype	Startposisjon	Lengde	Desimaler	Katalog-type	Verdiområde	Beskrivelse/definisjon
1	Fødselsnummer	FNr	Numerisk	1	11			1-9999999999
	Bedriftstype	BType	Kategorisk	12	1			1-9

Kodelister for kataloger (inkl. kategoriske variable)

1-32	33-47	48-79	80-179
Variabel	Kategori	Navn	Beskrivelse/definisjon
Bedriftstype	001	Stor bedrift	Bedrift med mer enn 1000 ansatte
Bedriftstype	002	Middels bedrift	Bedrift med mellom 50 og 1000 ansatte
Bedriftstype	003

¹⁰ Hvorav 3 av leddene inneholder hhv. «DATA», versjon og filklasse + max 4 subklasser.

¹¹ Krypteringsserie.

¹² Konesjonsnummer hos datatilsynet.

¹³ Referanse til Word-dokument.

4.4 Detaljpesifikasjon

Idé	Hver datalager/fil og hver prosess/program får et eget kapittel i detaljpesifikasjonen.
Ambisjonsnivå	Obligatorisk. Detaljpesifikasjonen lages som kapitler på ett eller flere tekstbehandlingsdokumenter.
Notat	-
Mal	q:\dok\metode\ detaljsp.doc
Ansvarlig	Prosjektleder
Utførende	Bruker (red.) sammen med systemutvikler



Beskrivelse:

1. Brukeren skriver *detaljpesifikasjonene* i Word som ett eller flere dokumenter.
2. Detaljpesifikasjonene baserer seg på det logiske dataflytdiagrammet og eventuelt en datamodell ved at dette/disse fungerer som en «grafisk innholdsfortegnelse» for detaljpesifikasjonen.
 - For hver prosess i logisk-dataflytdiagrammet lages det et avsnitt eller kapittel i detaljpesifikasjonen. Summen av disse kapitlene utgjør detaljpesifikasjonen.
 - Når datadefinisjonsverktøyet for SSB er ferdig til drift, dokumenteres data direkte i datadefinisjonsverktøyet. Datadefinisjonen blir da en del av detaljpesifikasjonen. Inntil dette skjer, kan datadefinisjonen skje på følgende måte: Dersom det er laget datamodell, dokumenteres hver av entitetene i datamodellen i et verktøy (CASE eller fjerdegenerasjonsverktøy). Dersom det bare er laget logisk-dataflytdiagram, tas det utgangspunkt i dette diagrammet og hver av prosessene dokumenteres. Kodelister må dokumenteres på *q:\dok\<prosjekt>data* (eventuelt på egne databasetabeller).
3. Brukeren følger malen i eksemplet to sider lengre fram.






Erfaringer:


1. I små prosjekter kan en ta utgangspunkt i logisk dataflytdiagram på nivå 1 (DFD1). For større prosjekter er det naturlig å ta utgangspunkt i DFD2.
2. Fagseksjonene har tradisjonelt problemer med å lage en tilstrekkelig nøyaktig spesifisering med det resultat at spesifikasjonene endrer seg fortløpende i prosjektets gang. Noen kontrollspørsmål til spesifikasjonene:
 - Har du husket alle spesialtilfellene?
 - Har du tatt ting for gitt? (husk at systemutvikleren ikke kjenner området like godt som deg)
 - Har du brukt ord eller uttrykk, som har en spesiell statistikkfaglig betydning, som ikke vil bli forstått?
3. Det er stor forskjell på hvor mye systemutviklerne på forhånd kjenner til av systemet som skal lages. Men dette må ikke føre til at det nye systemet blir mindre nøyaktig spesifisert fordi systemutvikleren kjenner det godt på forhånd. Dette vil nemlig lett kunne føre til manglende nytenkning samt dårlig dokumentasjon.




Teknisk anmerkning:

Word hjelper brukeren til å få en enkelt håndterbar detaljpesifikasjon.


1. Klikk på  og deretter på . Den viktigste knappen der er  som gir informasjon om dokumentet. Innholdet kommer alltid på første linje i dokumentet og består av følgende informasjon:

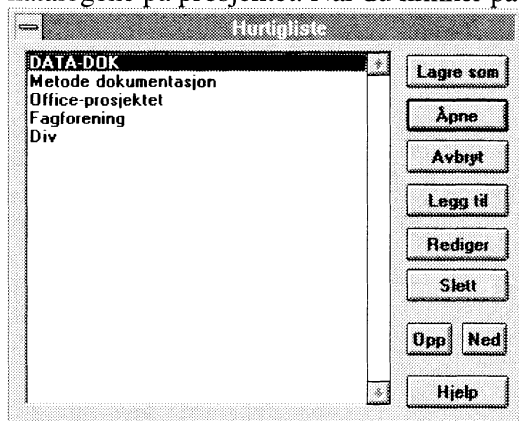
- SVE: Initial på den som har skrevet dokumentet
- 24.05.96: Dato dokumentet er skrevet
- SIST ENDRET: 21.08.96: Dato sist endret
- Q:\DOK\METODE\METODE96.DOC: Fullstendig navn på dokumentet. Hvis dokumentet flyttes, kan du trykke på knappen en gang til (uansett hvor i dokumentet du står) og navnet vil bli oppdatert riktig.
- V1.0: Versjon av dokumentet. Du må selv endre versjonsnummeret når du har endret så mye på dokumentet at du vil kalle det en ny versjon.
- For dette dokumentet blir det for eksempel slik:
SVE/- 24.05.96 SIST ENDRET: 21.08.96 Q:\DOK\METODE\METODE96.DOC V1.0

2. Bruk Word's automatiske overskriftsdesign  (også kalt «stiler») sammen med følgende knapper for å få full nytte av systemet¹⁴:

-  gir innholdsfortegnelse
-  gir overskriftsnummerering
-  gir sidenummerering

3. For raskt å finne fram i katalogene lønner det seg å bruke hurtiglista i Word. Hurtiglista finner du



på knapperaden i Word under ikonet . Ved hjelp av denne lista kan du lett finne fram i de ulike katalogene på prosjektet. Når du klikker på en av linjene i lista, kommer du rett inn i riktig katalog:



4. For å sammenligne ulike versjoner av dokumentet gjør du følgende:

Sammenligne to versjoner av et dokument

1. Åpne den redigerte versjonen av dokumentet.
2. Velg **Korrektur** på **Verktøy**-menyen.
3. Velg Sammenlign versjoner.
4. Skriv inn eller velg navnet på det opprinnelige dokumentet, i boksen "Opprinnelig filnavn", og velg OK.
Det redigerte dokumentet vises med korrekturmerking for tekst som er satt inn, slettet og korrigeret. Alternativene for visning av korrekturmerking er angitt under Korrektur i dialogboksen **Alternativer**. Hvis du vil ha flere opplysninger, se [kategorien Korrektur](#).

¹⁴ Disse to knappene er også nyttige:  gir bytting mellom små og store bokstaver i en tekst og  gir formel-editor

Eksempel:

SVE/- 24.05.96 SIST ENDRET: 01.07.96 Q:\DOK\SSB-BASE\ANALYSE\INIT.DOC v 1.0

Avdeling for personstatistikk, Seksjon for persondata

Detaljspesifikasjon for Befolkningsbasen

<Innholdsfortegnelse>

Innledning

- Muligheter og begrensninger i systemet

For hver prosess på logisk dataflytdiagram:

Detaljert beskrivelse av prosess YYYYYY

- Matematiske formler og algoritmer
- Kontroller av variabler og skjemafelter
- Koblinger av datalagre (husk å spesifisere koblingsnøkler)
- Annet som gjøres med dataene
- Eventuelle fremtidige utvidelser (bearbeidet versjon av grovspesifikasjonen)

For hvert datalager som ikke er dokumentert/definert andre steder:

Detaljert beskrivelse av datalager XXXXXX

- Navn på datalagrene i prosjektet (fil/databasetabell/datasett) med tilhørende beskrivelse
 - Input-datalagre til systemet kan være definert tidligere
 - Langtidslagrede filer som lages i prosjektet («produksjonsdata»)
 - Eventuelt andre filer som er spesielt nyttige
- Variablene (på datalageret) med tilhørende beskrivelse
- Kodene (til variablene) med tilhørende beskrivelse
- Definisjon av entydige nøkler («Identert»)

Vedlegg (ev. kopi fra grovspesifikasjon)

- Ev. definisjoner
- Skjema
- Tabeller
- Filbeskrivelser

4.5 Prototyping

Idé	Spesifikasjoner angitt ved «prøving og feiling»
Ambisjonsnivå	Frivillig
Notat	1) Eventuell kommende «styleguide» ¹⁵ 2) Kapittel ang. grafiske brukergrensesnitt i <i>Håndbok for programmerere</i>
Mal	-
Ansvarlig	Systemutvikler (ev. EDB-delprosjektleder)
Utførende	Systemutvikler (red.) sammen med bruker



Beskrivelse:

1. Skjermbilder kan lages ved hjelp av prototyping. Systemutvikler(e) og brukerne lager da skjermbildene sammen slik at de er i overensstemmelse med *styleguiden*. Styleguiden er et spesielt nyttig hjelpemiddel for å få et ensartet brukergrensesnitt slik at brukeren skal kjenne seg igjen fra program til program og systemløsning til systemløsning. Styleguiden er et dokument som beskriver retningslinjer for design av:
 - skjermbildet (vinduer, menylinje, knapperad, statuslinje, kommandoknapper, fargebruk)
 - feedback (visualisering av ventetid, feilmelding, andre meldinger)
 - navnestandarder (variable, programmer)
 - standardrutiner som kan kalles opp (Se *Håndbok i programmerere*)

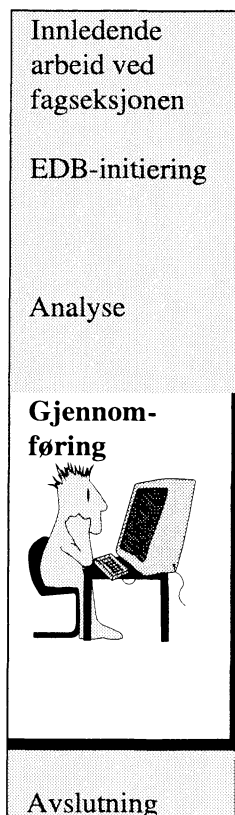


Erfaringer:

1. Prototyping virker som kvalitetssikring på den felles forståelse for EDB-systemet. Den gir god kommunikasjon, rask respons og er gøy å jobbe med.
2. En gruppe brukere bør delta i prototypingen slik at flere interesser blir tilgodesett.
3. Prototyper kan også brukes til andre formål enn skjermbilder når:
 - Systemene er meget store og derfor kompliserte.
 - Den totale funksjonen ikke kan bli implementert innen fastsatt tid.
 - Behovene ikke kan bli forstått uten operativ erfaring.
 - Videre design ikke kan komme igang uten at et forberedende system er bygd og kjørt.
 - ON-LINE-systemer med skjermbilder. I de fleste verktøy med grafiske brukergrensesnitt kan teknikken brukes ved finpussing av brukergrensesnittet i gjennomføringsfasen.
- Systemet kan da spesifiseres bit for bit. Hver bit programmeres og testes ut før neste bit påbegynnes.

¹⁵ En *styleguide* er et notat som angir hvordan skjermbildene skal utformes.

5. Gjennomførings-fasen



Idé	Programmering og testing. ¹⁶
Mål:	Ende opp med en ferdig programkode som er i samsvar med spesifikasjonene fra brukerne og retningslinjene i SSB.
Inn til fasen	<ol style="list-style-type: none"> 1. Logisk dataflytdiagram 2. Detaljspesifikasjoner 3. Datamodell (når databaseløsning) 4. Eventuell prototyping
Ut fra fasen	<ol style="list-style-type: none"> 1. Kjørbare programmer 2. Dokumentasjon innen det enkelte program 3. Fysisk dataflyt med sammenhengen mellom programmene 4. Dialogsystem (når skjermbilder)

Merk at detaljerte retningslinjer for hvordan programmeringen skal utføres er beskrevet i *Håndbok i programmerere*.

¹⁶ Denne fasen er i systemutviklingsmetoder fra '70 og '80-tallet oppdelt i én eller flere av fasene «logisk design», «fysisk design» og «konstruksjon». I vår metode er dette forenklet for å unngå for mye detaljregulering. Dette gir også økt fleksibilitet mht. verktøy (operativsystem, programmeringsspråk, databaser osv.).

5.1 Programmering og testing

Idé	Programmer med tanke på at andre skal lese programmet!
Ambisjonsnivå	Obligatorisk. Følge SSBs standarder for programmering.
Notat	1. <i>Håndbok i programmerere</i> 2. Div. håndbøker om SAS, UNIX osv.
Mal	q:\dok\metode\heading.txt (PC) \$FELLES\maler\heading.txt (UNIX)
Ansvarlig	Systemutvikler
Utførende	Systemutvikler + bruker og systemutvikler på testing



Beskrivelse:

- Systemutvikleren programmerer etter retningslinjer i «*Håndbok i programmerere*».
- Det er 3 typer tester. Hver av testene har som funksjon å finne feil i systemet:
 - *Modultesting*: Testing av enkeltprogrammer
 - *Systemtest*: Test av sammenhengen mellom programmene
 - *Akseptansetesting*: Test på om systemet som helhet kan godtas av brukerne. For konverteringsprosjekter innebærer dette som et minimum at resultater produsert av det nye systemet sammenlignes med resultatene fra det gamle. For nyutviklede systemer sjekkes det at resultatene er i samsvar med forventningene til brukerne.
- Når en systemutvikler driver programutvikling både på PC/Windows og på UNIX skal programmene ikke kopieres mellom plattformene. For å sikre tilgjengelighet til programmer på PC brukes «q:\dok\

Heading i begynnelsen av programmene:

```

/*****
Prosjekt .....: SSB-BASE
Programnavn .....: arb_Bosatt
Skrevet av .....: Bjørn Joneid
Dato .....: 24.06.96
Versjon .....: 1
Programmets funksjon .: Kobler filene og tester lovlige koder.
Anmerkning.....: Programmet forutsetter at data er korrekte.
Programmet kaller ....: -
Filer inn.....: $SSB-BASE\data\arbtak.dat
                  $SSB-BASE\data\bosatt.dat
Filer ut.....: SSB-BASE m. arb.takere
Endret når .....: 26.11.96
Endret av .....: Erik Søberg
Grunn til endring ....: Tester nå på lovlige koder.
*****/

```

5.2 Fysisk dataflytdiagram (DFD)

Idé	Oversikt over informasjonssystemet vi <u>har</u> laget.
Ambisjonsnivå	Obligatorisk. Viser sammenhenger mellom programmer og filer.
Notat	-
Mal	-
Ansvarlig	Systemutvikler
Utførende	Systemutvikler



Beskrivelse:

- Etter at programmene er kodet og testet må de dokumenteres. Systemutvikleren som har kodet programmene tegner da *fysiske dataflytdiagrammer* som viser sammenhengen mellom programmene.
- Fysisk dataflytdiagram inneholder følgende bestanddeler: Grafiske symboler for filer og programmer og flyten mellom dem.
- På filer/databasetabeller skal følgende spesifiseres:
 - Navn (inkl. SAS-libname/ORACLE-brukernavn og stammenavn på UNIX)
 - Eventuelt omtrentlig antall records/rader
- På programmer skal følgende spesifiseres:
 - Programnavn (ikke katalognavn)
 - Beskrivelse av hva programmet gjør
 - Annet er frivillig, men unngå rot!
- Diagrammet skal ha en dokumentasjonsheading som inneholder:
 - Diagramnavn (automatisk)
 - Versjonsnummer
 - Ansvarlig for programmene i diagrammet
 - Ansvarlig for spesifikasjonene til programmene
 - Hvilke kataloger programmene er lagret på (stammenavn på UNIX). Hvis det er mange «programmer» (funksjoner, subrutiner, prosedyrer) lagret på samme programfil, oppgis filnavnet



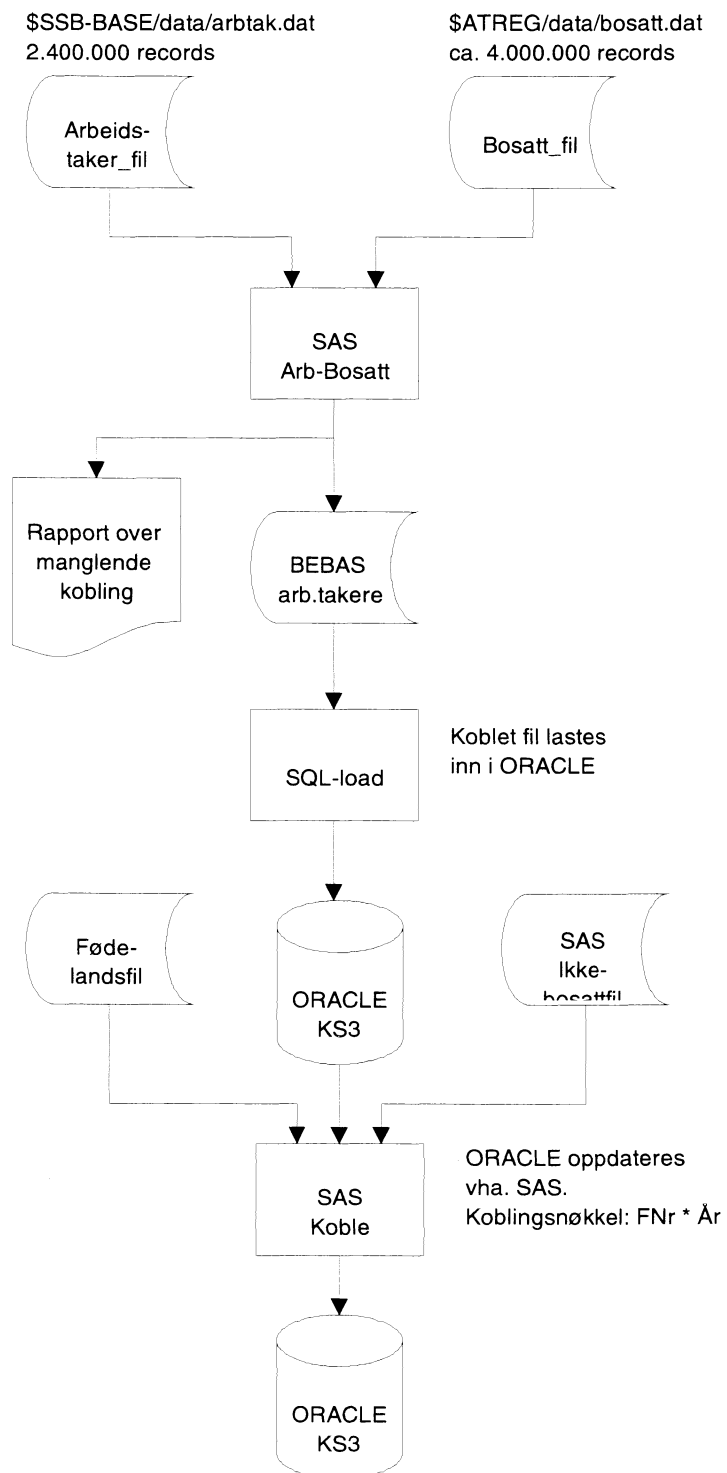
Erfaringer:

- Fysisk dataflytdiagram* er en teknikk som tradisjonelt har stått sterkt i SSB og som mange kjenner igjen som såkalte «*flowchart*», «*flytkart*», «*flytskjema*», «*programflyt*» eller «*systemkart*».
- Diagrammet ligner på *logisk dataflytdiagram (DFD)*, men brukes for dokumentasjonens skyld og ikke som et verktøy for å *designe* de enkelte *prosesser* og *transaksjoner* som inngår i systemet.
- Hvis det er spesielt viktige skjermbilder i tilknytning til prosesser i EDB-løsningen, kan disse tas med i *fysisk dataflytdiagram* i tillegg til prosessene.
- Det anbefales ikke å bruke *fysisk dataflytdiagram* til å kommentere strukturen i et program. Det enkelte programmet kommenteres utelukkende internt i programmet (se «*koding*»). Programmer som er kompliserte må ha desto flere kommentarlinjer!

Tillatte symboler i teknikken «FlowChart»:

Symbol							
Betydning	Dokument (papirtabell)	Skjerm bilde	Database/dataset oracle, sas, fame	Sekvensiell fil	Program	Dataflyt-retning	Peker til forts.

Eksempel:



5.3 Dialogdesign

Idé	Vise hvordan skjermbildene henger sammen (menysystem).
Ambisjonsnivå	Interaktive systemer. Frivillig på mindre prosjekter.
Notat	-
Mal	-
Ansvarlig	Systemutvikler
Utførende	Systemutvikler



Beskrivelse:

- Dialogsystem viser hvordan dialogen mellom programsystemet og brukeren fungerer. Systemutvikleren som har kodet dialogen tegner opp et dialogsystem (dialogdesign):
 - Det er mange skjermbilder som henger sammen.
 - Det er valg til mange skjermbilder som utgang fra skjermbilde/meny.
- Det som tegnes opp er skjermbildene og de mulige veiene mellom skjermbildene.



Erfaringer:

- Resultatet av denne teknikken kalles ofte «menysystem».

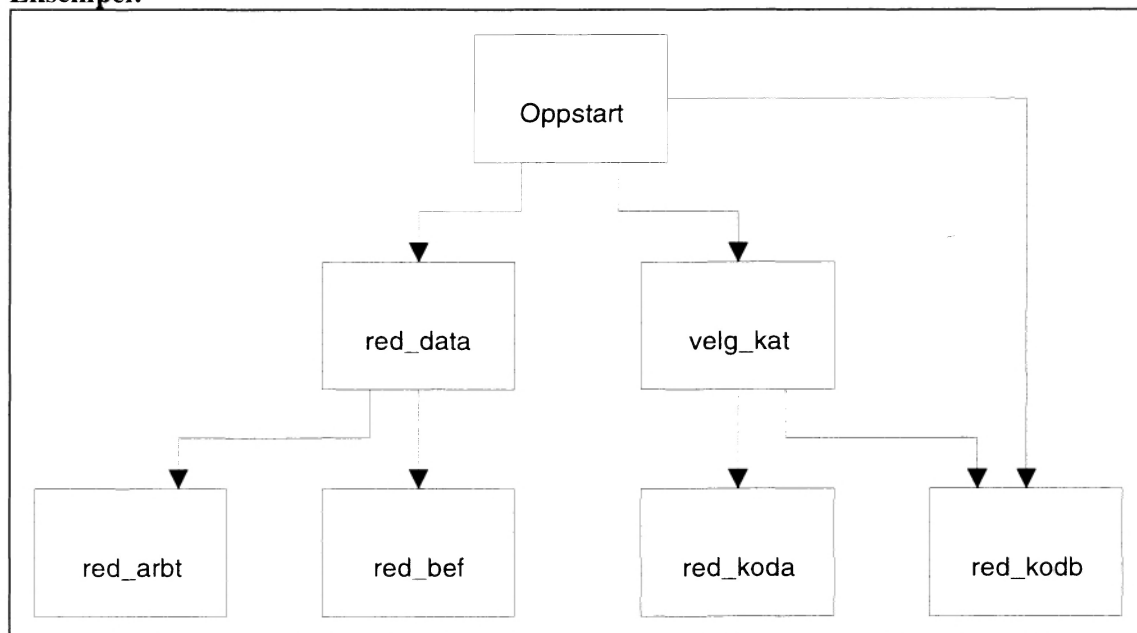
Teknisk anmerkning:

- Siden vi ikke har noe spesialisert verktøy for denne teknikken, benytter vi isteden samme verktøy som f.eks. *fysisk dataflyt*-teknikk


Tillatte symboler i teknikken «Dialogdesign»:

Symbol			
Betydning	Skjermbilde	Valg	Peker til forts.

Eksempel:



6. Avslutnings-fasen

Innledende arbeid ved fagseksjonen	Idé	I denne fasen ryddes det opp i den dokumentasjonen som allerede eksisterer - slik vi ønsker prosjektet skal framstilles for ettertiden. Den gjenstående dokumentasjonen lages i denne fasen.
Initiering	Mål	Lagre informasjon om systemet slik at det er lett å finne fram til og så lett å forstå at den blir brukt i praksis.
Analyse	Inn til fasen	<ol style="list-style-type: none"> 1. Dataflytdiagram nivå 0 2. Logisk dataflytdiagram 3. Samlede detaljspesifikasjoner 4. Eventuell datamodell 5. Eventuell prototype 6. Eventuelt dialogsystem 7. Dokumentasjon innen det enkelte program 8. Fysisk dataflytdiagram med sammenhengen mellom programmene
Gjennomføring	Ut fra fasen	<ol style="list-style-type: none"> 1. Dokumentasjonsnotat 2. Informasjon om prosjektet som helhet
Avslutning 		

6.1 Dokumentasjonsnotat

Idé	Samle all dokumentasjon av systemet på ett sted. Skape brukerens bilde av systemet og oversikt over systemet.
Ambisjonsnivå	Obligatorisk 1) Lagre den som Word-dokument på <i>q:\dok\...</i> 2) Systemdokumentasjon og brukerdokumentasjon
Notat	-
Mal	q:\dok\metode\doknot.doc
Ansvarlig	Prosjektleder
Utførende	Systemutvikler (red.) og bruker (red.)



Beskrivelse:

1. Bruker og systemutvikler skriver sammen en endelig dokumentasjon av systemer etter malen gitt nedenfor. Innholdet kan gjerne hentes fra tidligere utarbeidede dokumenter og diagrammer.



Erfaringer:

1. Innhold og struktur vil kunne variere fra prosjekt til prosjekt. I hvert fall i mindre prosjekter er det unødvendig å skille mellom brukerdokumentasjon og systemdokumentasjon (de slås sammen til et «dokumentasjonsnotat»).

Eksempel:

SVE/- 22.08.96 SIST ENDRET: 22.08.96 Q:\DOK\SSB-BASE\DOKNOT.DOC v 1.0

Dokumentasjonsnotat for SSB-BASE

- Fullstendig filnavn på dokumentet
- Dato for utarbeidelsen av dokumentet
- Forfatter(ne)s navn
- Navn på prosjektet og seksjonen(e)

<Innholdsfortegnelse>

1. **Innledning** (ev. delvis kopi av prosjektskriv)
 - Bakgrunn for prosjektet
 - Leseveiledning (hvem som skal lese hva osv.)
 - Interessenter (oppdragsgiver, ev. styringsgruppe mm)
 - Ansvarsforhold:
 - kontaktperson ved drift, forslag, endringer og systemfeil
 - prosjektleder
 - EDB-ansvarlig
 - eventuelt styringsgruppas og referansegruppas medlemmer
 - Referanser til annen dokumentasjon som kan være av interesse:
 - hvor den finnes
 - hvem som står ansvarlig for den

*When in trouble,
when in doubt;
run in circles,
scream and shout!*



2. Oversikt over systemet (eventuelt delvis kopi fra prosjektskrivet)

- Kort beskrivelse av systemet (ev. kopi fra detaljspesifikasjon)
 - hensikten med å lage det nye systemet
 - hvilke hovedoppgaver systemet skal løse
 - hvem som er berørt av endringene
 - oppsummering av muligheter og begrensninger i systemet
- Overordnede krav til systemet
 - krav til omgivelsene (endringer av manuelle rutiner, grensesnitt, hjelpesystemer)
 - krav til opplæring i bruk av systemet (kunnskapsnivå hos brukerne, kurs)
- Omgivelsene systemet skal fungere i
 - beskrivelse av standardprogramvare, programmeringsspråk som er benyttet, databasesystem, programutviklingshjelpemidler, standardarder, programbibliotek og henvisninger til sikkerhetskopier
 - beskrivelse av kommunikasjonsutstyr, datamaskiner osv. som har betydning for systemet

3. Statistikkfaglig forarbeid (ev. delvis kopi fra statistikkfaglig grovspesifikasjon)

- Systemets plassering i en større helhet.
- Hvilke statistikkfaglige problemer EDB-systemet skal løse
- Begrunne valg av statistikkfaglig metode og kort beskrivelse av metoden
- Eventuelle usikkerheter tilknyttet løsningen
- Hva forbedres i forhold til tidligere system

4. Bruk (brukerdokumentasjon)

- Forkunnskaper som forventes av brukeren
- Krav til opplæring i bruk av systemet
- Systemets restriksjoner
- Veiledning for normal bruk av systemet
 - hvordan sette systemet igang
 - hvis batch-kjøring: tekstlig beskrivelse av kjøringene
 - beskrivelse av menysystemet (inklusive ev. kopi av skjermbilder og liste over lovlige verdier på skjermfeltene)
 - beskrivelse av hjelpesystemet
- Spesialsituasjoner ved bruk av systemet
 - hvordan stoppe systemet og komme igang igjen?
 - hvordan håndtere feilsituasjoner?
 - hvordan få evt. annen hjelp/assistanse fra systemet?
 - beskrivelse av hvordan korrigerer blir utført, reserverutiner ved driftsstans, sikkerhetskopiering av database. Oppgi hvem som skal kontaktes ved uforståelige feil.
- Liste over datafiler som systemet krever
- Detaljert beskrivelse for hvert enkelt program¹⁷
 - navn
 - formål
 - eventuell kommentar og spesiell anmerkning
 - inndata
 - referanser til involverte skjermbilder
 - rapporter (med eksempler)
 - muligheter for utskrift fra skjermbilder

¹⁷ Dette punktet kan i mange tilfeller legges inn som en oversikt i tabellform.

- meldinger, feilmeldinger og behandling av feil



5. Datalagre (datalagre som ikke er dokumentert i DATA-DOK)

- Hvor er filen/tabellen lagret?
- Hva inneholder filen/tabellen?
- Hva brukes filen/tabellen til?
- Utlisting av filens/tabellens variabler og variablene koder. [kan ev. henviser til DATA-DOK]

6. EDB (systemdokumentasjon/oversikt)

- Oversikt over systemet (diagrammer som limes inn fra tegneverktøyet):
 - datamodell [dataoversikt]
 - dataflytdiagram nivå 0 (DFD0) [omgivelser]
 - logisk dataflytdiagram (DFD) [oversikt]
 - fysisk dataflytdiagram (hvis det blir svært mange, bør de bli vedlegg)
 - dialogdesign [menysystem mm.]
- EDB-systemets oppbygning (moduler)
- Beskrive hvert program for eksempel i form av en kopi av programmenes headinger
- Hvor er programmene lagret
- Beskrivelse av data (konverteringer, overføringer, drift og vedlikehold av data)

Vedlegg

- Liste over feilmeldinger (som er programmert i SSB) systemet gir og deres sannsynlige årsak
- Eventuelle logisk-dataflytdiagrammer på nivå 2 (DFD2) og høyere nivå
- Programlistinger
- Tabeller

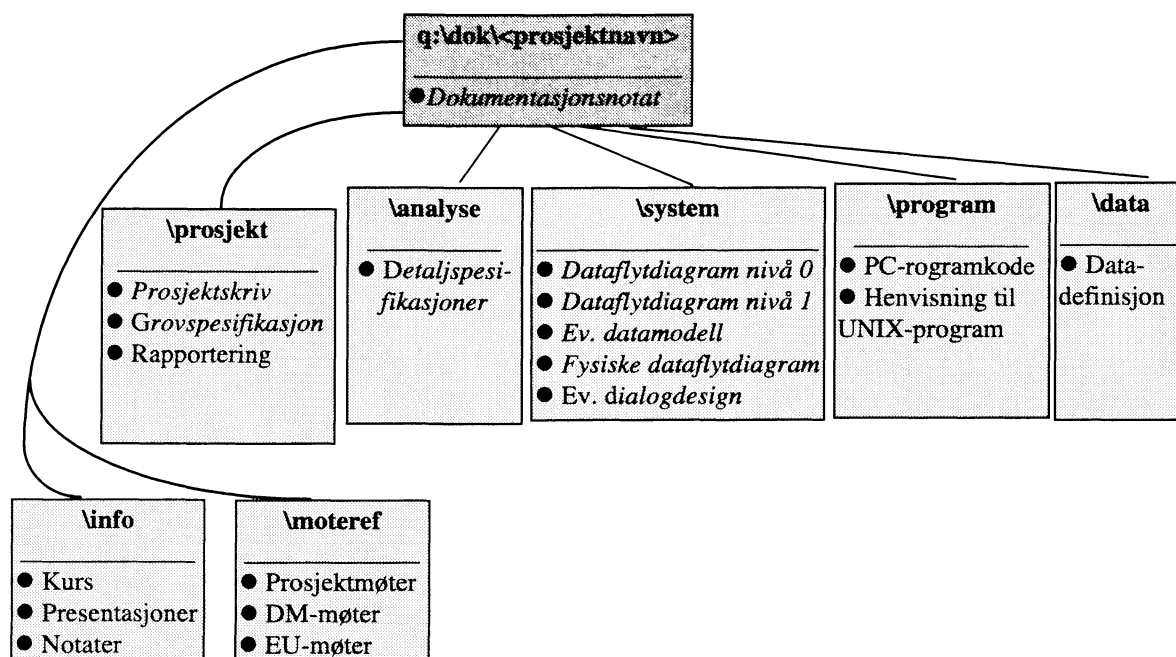
6.2 Sjekk dokumentasjon

Idé	Sjekk at dokumentasjonen er i henhold til reglene i SSB.
Ambisjonsnivå	Obligatorisk
Notat	-
Mal	-
Ansvarlig	EDB-kontorleder
Utførende	EDB-kontorleder



Beskrivelse:

1. Det er EDB-kontor-lederens oppgave å kvalitetssikre at dokumentasjonen foreligger. Her må det understrekes at kvalitetssikringen *ikke* er kontroll av faglig innhold i dokumentene, men å se til at all dokumentasjon faktisk foreligger, og at den ligger på riktig plass på Q-disken.
2. Underveis i prosjektet er alle viktige produkter lagret på PC under Q-diskens katalog «DOK» med prosjektkatalog bestemt ved navnet etter prosjektet: Q:\DOK\



Figur: Oversikt over hva som skal ligge hvor på Q-disken



Erfaringer:

1. Det kan være lurt å lage enda et nivå av kataloger. Eksempler på dette er:
 - Når et prosjekt er delt opp i delprosjekter, kan det være lurt å dele hovedprosjektkatalogen opp i delprosjektkataloger. Vi får da en struktur som følger navnestandarden på UNIX.
 - Noen kataloger blir så fulle av filer at det blir vanskelig å holde oversikten. Det lages da underkataloger. Underkatalogene kan da organiseres slik som punktene på figuren ovenfor viser: q:\dok\
2. Umiddelbart under katalogen q:\dok\

6.3 Igangkjøring

Idé	EDB-systemet leveres til brukerne og dokumentasjonen i SSB.
Ambisjonsnivå	Obligatorisk. Brukerne har testet og godtatt at systemet er i samsvar med spesifikasjonene. Dokumentasjonen godkjent av EDB-leder.
Notat	-
Ansvarlig	Prosjektleder
Utførende	Systemutvikler sammen med bruker og EDB-leder



Beskrivelse:

1. Brukerne og systemutviklerne tester det endelige systemet sammen. Hvis systemet fungerer som forventet, overleveres systemet til brukerne sammen med dokumentasjonen.
2. Bestem hvem som er ansvarlig for endringer i systemet.
3. Send ut dokumentasjonsnotatet i serien *Interne dokumenter*.
4. En tur på byen når alt er OK.
5. Alt som gjøres etter dette i prosjektet er vedlikehold (det vil si at utviklingsprosjektet eller konverteringsprosjektet er avsluttet).



Erfaringer:

1. Dokumentasjonen er en del av produktet som er laget. Dårlig dokumentasjon gir lite brukervennlig system.

7. Ordliste

- ✍ angir at dette er en teknikk i metoden
- ◆ angir at dette er en fase i metoden
- ☺ angir at dette er en deltaker i arbeidet med å lage EDB-systemet
- ⇒ angir annet som angår metode

- ⇒ **aktiviteter**: De arbeidsoperasjoner som en prosess kan deles inn i. Se innledningskapitlet!
- ⇒ **analyse**: Statistisk begrep i forbindelse med analyse av data. Men også brukt som EDB-begrep i forbindelse med å identifisere krav til EDB-systemer og designe disse.
- ◆ **analyse(fasen)**: Fase nr. 2 i metoden. Systemet spesifiseres og planlegges.
- ⇒ **attributt**: Egenskap ved en entitet. Se teknikken *datamodellering*!
- ◆ **avslutning(sfasen)**: Prosjektet avsluttes og det sjekkes at EDB-koden og dokumentasjonen er i orden.
- ⇒ **bearbeiding**: Ofte kalt «massering av data», «transformering av data» eller «tilrettelegging av data». Mellom input og lagring av data i SSB (nevnes i *statistikkfaglig orientert grovspeifisering*).
- ⇒ **beslutningspunkt**: Ved avslutningen av en fase, før en ny fase påbegynnes, skal det besluttes om neste fase skal gjennomføres. Sammenlign fig. i innledningskapitlet!
- ✍ **blackbox-testing**: Testing av programmer uten å kjenne koden (dvs. sende inn data og undersøke om output-data er riktige).
- ☺ **bruker**: Den som skal bruke EDB-systemet som lages. Se særlig fasen *Innledende arbeid ved fagseksjonen*.
- ⇒ **CASE-verktøy** (= *Computer Aided Software Engeneering*): Verktøy for å automatisere deler av *systemutviklingsprosessen*. Vi har flere typer CASE-verktøy:
 - ⇒ *upper CASE*: administrasjon og dokumentasjon
 - ⇒ *lower CASE*: kodegenerering
 - ⇒ *I-CASE*: *upper-CASE* + *lower-CASE*
- ✍ **CBA** (= *Cost benefit analysis*): Se *nytte-kostnadsanalyse*!
- ✍ **CRA** (= *critical requirement analysis*) Se *kritisk kravanalyse*!
- ✍ **dataflyt**: Hvordan data beveger seg fra datalager til datalager via prosesser. Dataflyt er av to typer: *logisk dataflyt* og *fysisk dataflyt*. Logisk dataflyt er igjen av to typer: *dataflytdiagram nivå 0* og *dataflytdiagram nivå 1...n*.
 - ✍ **dataflytdiagram nivå 0** (DFD0): Diagram som viser hvile kilder og mottakere av data som systemet har. Brukes i fasen *innledende arbeid ved fagseksjonen*.
 - ✍ **logisk dataflytdiagram nivå 1** (DFD1): Diagram som viser en grov oversikt over EDB-systemet som skal lages med dets omgivelser. Inneholder *prosesser* og *datalagre* og sammenhengen (*dataflyten*) mellom dem. Brukes i fasen *analyse*.
 - ✍ **logisk dataflytdiagram nivå 2** (DFD2): 1. nedbrytning av logisk dataflytdiagram (DFD1). Delprosesser med tilhørende dataflyt. Brukes i fasen *analyse*. (Ikke obligatorisk i vår metode.)
 - ✍ **fysisk dataflyt** (DFDF) Se «*fysisk dataflytdiagram*»!
- ⇒ **datalager**: Logisk fil. Brukes i logiske dataflytdiagrammer.
 - ⇒ **manuelt lager** (mappe i kontorskuff, listing osv.)
 - ⇒ **datamaskinbasert lager** (sekvensiell fil, databasetabell, SAS-datasett osv.)
 - ⇒ **temporært lager** (midlertidig lagring)
- ✍ **datamodell(ering)** (DM): Sammenhengene i data: entiteter, attributter og nøkler. Teknikken som brukes i analysefasen.
- ⇒ **delprosess**: Del av en hovedprosess i DFD1.

- ◆ **design(fasen):** I vår metode er det en del av *gjennomføringsfasen*. Dette er egne faser i *Systems Engeneering*:
 - ⇒ - logisk design
 - ⇒ - fysisk design
- ⇒ **Designer 2000:** CASE-verktøy i *Developer 2000 (ORACLE)*.
- ✍ **detaljspesifikasjoner:** Detaljerte spesifikasjoner fra bruker til systemutvikler om hvordan systemet skal fungere. Teknikk i analysefasen.
- ✍ **DFD0 = dataflytdiagram nivå 0.**
- ✍ **DFD1 = logisk dataflytdiagram nivå 1.**
- ✍ **DFD2 = logisk dataflytdiagram nivå 2.**
- ⇒ **diagram:** Resultat av tegneteknikk.
- ✍ **dialogdesign (= dialogsystem):** Teknikk for å tegne opp sammenhengen mellom skjermbildene. Teknikk i analysefasen.
- ✍ **DM = datamodellering.**
- ✍ **dokumentasjon:** Tilrettelegging og presentasjon av informasjon om det som skal gjøres eller allerede er gjort.
- ✍ **ELH (= entity life history).** Se «entitetenes livssyklus»!
- ⇒ **entitet:** Et dataelement/objekt (f.eks. person, foretak, bolig).
- ✍ **entitetenes livssyklus (ELH):** Historien til en entitet: oppretting, endring, sletting.
- ✍ **estimering:** Planlegging av ressurser: prosjektdeltakernes tidsforbruk, maskinressurser osv.
- ⇒ **event = hendelse.**
- ◆ **fase:** Groveste oppdeling av systemutviklingsarbeidet (se innledningskapitlet!). I vår metode er det 4 faser (prosjektinitiering, analyse, gjennomføring og avslutning).
 - ⇒ Hver fase er oppdelt i *aktiviteter*.
 - ⇒ Hver aktivitet er oppdelt i konkrete (arbeids)oppgaver.
- ✍ **Flowchart = fysisk dataflytdiagram.**
- ✍ **fysisk dataflytdiagram:** Oversikt over systemet som er laget. Skjematisk sammenheng mellom programmer og sekvensielle filer/databasetabeller. Se kapitlet for *gjennomføringsfasen*!
- ⇒ **funksjon:** Begrep i *logisk-dataflyt*-teknikken i *System Engeneering*. Ikke benyttet i vår metode!
- ✍ **funksjonsanalyse:** Ligner vår *logisk-dataflyt*-analyse.
- ⇒ **fysisk design:** Begrep ikke benyttet i vår metode. Se *design(fasen)*!
- ◆ **gjennomføring(sfasen):** Koding, testing og dokumentasjon av dette.
- ⇒ **hendelse (= event):** Begrep som kommer mer inn med objektorientering. En hendelse setter i gang («trigger») en prosess.
- ⇒ **IE = Information Engeneering.**
- ◆ **avslutning:** Iverksetting av systemet som lages (installering m.m.) Se fasen *avslutning*.
- ⇒ **Information Engeneering:** *Systemutviklingsmetode*, konkurrent til *Systems Engeneering*.
- ⇒ **Inkrementell metode:** *Modell* for *systemutviklingsmetode* som baseres på at systemet ikke kan spesifiseres i detalj før kodingen starter. Motsatt av «*fossefallsmodellen*». Ved *inkrementell* utvikling utvikles systemet bit for bit, for så til slutt å ende opp med et komplett system (som å bygge med legoklosser). Man velger gjerne ut den viktigste delen av det nye systemet, og lager denne først. Deretter bygger man på med stadig nye deler, slik at man til slutt får et komplett system. Inkrementell systemutvikling Det er mange fordeler med denne formen for systemutvikling, men det er alltid en fare for at en ny modul kan utløse feil/svakheter i systemet, noe som vil føre til forsinkelser i prosjektet.
- ✍ **inspeksjon:** (sml. *Fagans* metode for gjennomgang av hvorvidt dokumenter tilfredsstillende kvalitetskrav).
- ⇒ **installasjon:** Se *avslutningsfasen*.
- ⇒ **iterasjon:** Gjentatte gjennomganger av en fase med gradvis bedre resultat (forhåpentligvis).
- ✍ **JAD (= Joint Application Development).** Se «*lagarbeid*».
- ✍ **koding:** Programmering. Se også «*faser*».

- ⇒ **kravspesifikasjon:** (Godkjente) spesifikasjoner fra brukerne av systemet.
- ✍ **kritisk kravanalyse (CRA).** Teknikk i analysefasen i *System Engeneering*. Brukes ikke i vår metode.
- ⇒ **kvalitetssikring (= QA).** Kvalitetssikring er ivaretatt på mange måter i vår metode. Følgende teknikker:
 - ✍ **Analysefasen:** Tegne sammen diagrammer for å sjekke at alle har samme *modellen* av systemet som skal lages.
 - ✍ **Gjennomføringsfasen:** *Testing* av programkoden m.m.
 - ✍ **Avslutningsfasen.** «Sjekk dokumentasjon» og «Igangkjøring».
 - ✍ I den grad arbeidet utføres i et prosjekt med flere deltakere kontrolleres resultatene automatisk og løpende.
- ✍ **lagarbeid:** Arbeidsform hvor alle prosjektdeltakerne jobber sammen. Systemutviklerne lager diagrammer og brukerne skriver tekst. Se fotnote i *analysefasen*.
- ✍ **logisk design:** Fase i *System Engeneering*. Benyttes ikke i vår metode. Se *design!*
menysystem: Skjema over alle skjermbildene og deres tilhørende programmer. Se *dialogdesign*.
- ⇒ **metode:** Begrepet betegner formalismen bak en metodisk arbeidsform. Brukes her ensbetydende med *systemutviklingsmetode* (se kapittel 1).
- ☺ **metodeansvarlig:** Gruppe med spesielt ansvar for vedlikehold og opplæring i metoden.
- ⇒ **milepel:** Dato når en *fase* eller viktig *aktivitet* i prosjektet er avsluttet.
- ⇒ **modul:** Samling av programmer som hører naturlig sammen.
- ✍ **nytte-kostnadsanalyse (= Cost Benefit Analyse (CBA)).** Teknikk for på forhånd å beregne kostnader og gevinster ved å utføre et prosjekt sammenlignet med ikke å utføre prosjektet.
- ⇒ **objektorientering:** Programmering basert på å modellere *objekter* som har egenskaper (*attributter*) som utfører handlinger (prosedyrer) + arving fra «foreldre»/(parent)-objekter.
- ⇒ **oppgave:** Den minste arbeidsoperasjon i metoden. Se innledningskapitlet og *fase!*
- ⇒ **produkter:** Resultat av teknikk (*diagram, spesifikasjon, program* m.m.)
- ✍ **programspesifikasjoner:** Se *detaljspesifikasjoner!*
- ⇒ **prosess:** Enhet i en grov oppdeling av EDB-systemets bearbeiding av data. Komponent i *logisk-dataflyt*-diagrammet. Se innledningskapitlet og *analysefasen*.
- ⇒ **prosjekt:** Arbeidsform hvor personer fra ulike faggrupper jobber sammen for å nå et mål innen en tidsfrist. Engangsoppgave med et klart mål og klare ressurs- og tidsrammer. Se innledningskapitlet!
- ☺ **prosjekteier:** Den som har satt i gang prosjektet og skal bruke resultatet.
- ◆ **prosjektinitiering:** Se under «faser»!
- ☺ **prosjektgruppe:** Summen av de personene som deltar i prosjektet.
- ☺ **prosjektleder:** Den som leder prosjektet. (Dersom prosjektgruppa består av bare én person, er denne personen prosjektleder.) Se innledningskapitlet!
- ✍ **prosjektskriv:** SSBs tradisjonelle *prosjektskriv*. Se kapittel for *prosjektinitiering!*
- ⇒ **prosjektstyring (PS):** Rutiner for å planlegge og følge opp prosjektet. Dette ligger på siden av *systemutviklingsmetoden*, men de er nyttige å bruke sammen (synergieffekt).
- ✍ **prototyping:** Se kapittel for *analysefasen*.
- ⇒ **QA = Quality assurance.** Se *kvalitetssikring*.
- ⇒ **RAD = Rapid application development:** Utvikling vha. ferdigskrevne standardmoduler og kodegenerering (se kodefase).
- ☺ **referansegruppe:** Gruppe av spesielt kvalifiserte personer som kan delta i vurdering av diverse oppgaver i prosjektet.
- ⇒ **SE-Express:** Engelsk *systemutviklingsmetode* levert av *ISI A/S*. Utgangspunktet for SSBs *systemutviklingsmetode*.
- ⇒ **SE-ONLINE:** Hjelpesfunksjon til *SE-Express*. Finnes på PC.
- ⇒ **SSADM: Systemutviklingsmetode** som ble utarbeidet som standard for forvaltningen i Storbritannia. Forløper for *System Engeneering*.
- ⇒ **styleguide:** Retningslinjer for hvordan du skal lage skjermbilder og *dialoger* i grafiske brukergrensesnitt.

- ☺ **styringsgruppe:** Prosjektets øverste beslutningsmyndighet. Prosjektlederen rapporterer til denne.
- ⇒ **SU = systemutvikling.**
- ⇒ **systemering:** Arbeidet med å analysere EDB-delen av et system. Tilsvarende *analysefasen* og deler av *prosjektinitieringsfasen*.
- ⇒ **System Architect:** CASE-verktøy brukt i SSB 1992-1996, levert av *Popkins Software* (USA).
- ⇒ **System Engineering (SE):** Kommersialisering av *SSADM*. *SE-Express* er en forenkling av SE. Vår metode er igjen en forenkling av *SE-Express*.
- ☺ **systemutvikler:** EDB medarbeider som utformer løsningen for EDB-systemet.
- ✍ **systemkart = fysisk dataflytdiagram.**
- ⇒ **systemutvikling (SU):** Innbefatter hele arbeidet med utforming av et informasjonssystem, både systemering og koding (dvs. alle fasene i *systemutviklingsmetoden*).
- ⇒ **systemutviklingsmetode (SU-metode):** *Metoden* en bruker når en vil arbeide med *systemutvikling*. Se definisjon i innledningskapitlet!
- ✍ **teknikk:** Standardisert måte å løse et problem på. *Teknikkene* er presentert i kapitlene fra *prosjektinitiering* til *avslutning*.
- ☺ **teknisk prosjektleder:** EDB-faglig person som leder den EDB-faglige delen av prosjektet.
- ✍ **testing:** Teknikker for å finne feil.
 - ✍ *modultest:* programtest
 - ✍ *systemtest:* test av sammenhenger mellom programmer
 - ✍ *akseptansetesting:* test av systemet som helhet
- ⇒ **testplan:** Plan for å teste ferdige produkter (*program, spesifikasjon* m.m.).
- ⇒ **tidsplan:** Oversikt over hva de enkelte prosjektdeltakerne skal gjøre i de ulike *fasene* i prosjektet.
- ⇒ **transaksjon:** Begrep i *logisk-dataflytdiagrammet* i *System Engineering*. ikke benyttet i vår metode.
- ⇒ **vaske data:** Begrep fra *datawarehouse*. Konsistenssjekking m.m. av inputdata.
- ✍ **white-box-testing:** Testing om programmet er riktig ved å lese programkode.
- ⇒ **3. normalform:** Se *datamodellering* i kapitlet om *analysefasen*.

8. Referanser

Litteratur fra Statistisk sentralbyrå:

Cecilie Alnæs (1996): *Brukerveiledning for Datadokumentasjons-systemet, Brukerdokumentasjon*, SSB

Avd. 303 (1996): *HB-96: Håndbok for programmerere*, SSB (ennå upublisert pr. 10/2/97).

Håkon Berby (1994): *Navnestandard på UNIX*, SSB

Rune Gløersen (1996): *Håndbok i prosjektstyring*, SSB

Knut Vidar Hoholm og Svein Jåvold (1992): *Rapport fra prosjektet Metodisk systemutvikling*, 30. mars. SSB

Svein Jåvold og Knut Vidar Hoholm (1993): *Skal SSB innføre systemutviklingsmetode?*, 1 november, SSB

Svein Jåvold (1995): *Minimetode*, SSB

Stein Turtumøygard og Håkon Berby (1995): *Kort innføring i ny navnestandard for UNIX-filer*, SSB

Noen eksempler på annen litteratur om metode:

Bræk, R et al (1985): *Håndbok i systemarbeid*, Tapir

J. D. Clockey (1996): *Delphi Guidelines & Notes*, Sunsite











Poland (1996): URL: <http://sunsite.icm.edu.pl/~robert/delphi/ftp/does/gideln.zip>

G. Skagestein (1991): *Data i fokus*, Universitetsforlaget.

9. Vedlegg: Symbolbruk i diagrammene

Diagrammer som skal benyttes i vår minimetode er DFD0, DFD1 fysisk dataflytdiagram. Det brukes færrest mulig symboler i diagrammene for å gjøre dem letteste. Det er minst like viktig at hvert enkelt symbol betyr det samme i hvert diagram, noe som igjen resulterer i nettopp færre symboler. Det må imidlertid understrekes at datamodeller må ha en egen standard for symbolbruk, og at et eventuelt nytt databaseutviklingsverktøy vil påvirke symbolbruk for datamodeller.

Tabell: Symboler som er vedtatt for bruk i de forskjellige diagrammene (ikke datamodell).

Symbol	Symbolnavn	Anmerkning	Tilhørende teknikk
	Skjerm bilde(r)		Fysisk dataflyt
	Database/datasett (ORACLE, SAS, FAME, ...)	Formaterte data	Fysisk dataflyt
	Sekvensielle filer (flate filer)	Uformaterte data	Fysisk dataflyt
	Dokument (papirtabeller)		Fysisk dataflyt
	Program	Benyttes også i DFD0 for å avgrense systemet	Fysisk dataflyt, dialogsystem, DFD0
	Dataflytretning	Fra venstre mot høyre eller fra topp til bunn	Fysisk dataflyt/ DFD0, DFD1, dialogsystem
	Prosess	Samling av programmer	DFD1
	Datalager		(DFD0), DFD1
	Dataleverandør/datamottaker		DFD0
	Fortsettelsesmerke		Diverse

10. Vedlegg: Dokumentasjonsverktøy

I denne håndboka er det lagt vekt på å beskrive systemutviklingsprosessen mest mulig uten verktøy. Her følger likevel et forslag til CASE-verktøy:

Tabell: Hva dokumenteres hvor?

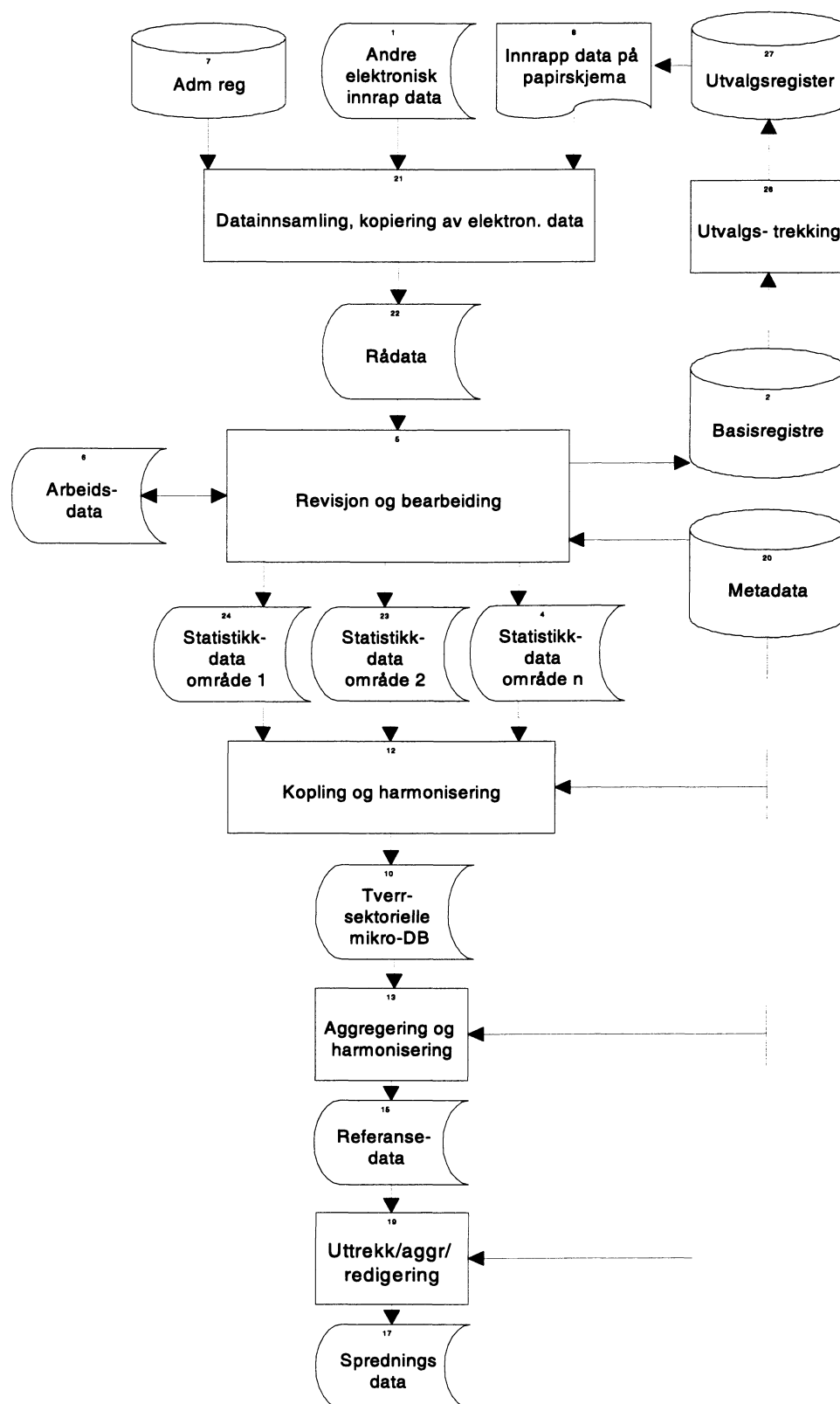
Programmer	DATA-DOK	CASE uavh. av prog. verktøyet	CASE integrert i prog.verktøyet	bruker-dok
Kommentarer til program	Filer/db-tabell	dataflyt nivå 0	Ev. datamodell	Ref. til annen dok
	Variabel	logisk dataflyt	Ev. prototype	Feilmeldinger
	Lovlige verdier	fysisk dataflyt ev. dialogsystem		Skjerm-bilder Lovlige koder

Tabell: Hvilke verktøy kan brukes til hva?

Type data	Datamodell	Dataflyt	Fysisk dataflyt	Dialogsystem	Definisjoner
Sekvensielle filer	ABC	ABC	ABC Flowcharter	-	DATA-DOK
	Flowcharter/ Developer 2000	Flowcharter			
SAS-datasett	ABC	ABC	ABC Flowcharter	ABC Flowcharter	DATA-DOK SAS
	Flowcharter/ Developer 2000	Flowcharter			
FAME	-	-	ABC Flowcharter	ABC Flowcharter	FAME
ORACLE	Developer 2000	ABC Flowcharter / Developer 2000	ABC Flowcharter	ABC Flowcharter/ Developer 2000	DATA-DOK Developer 2000

Word tillates også brukt som dokumentasjonsverktøy dersom de tilrådte symbolene benyttes.

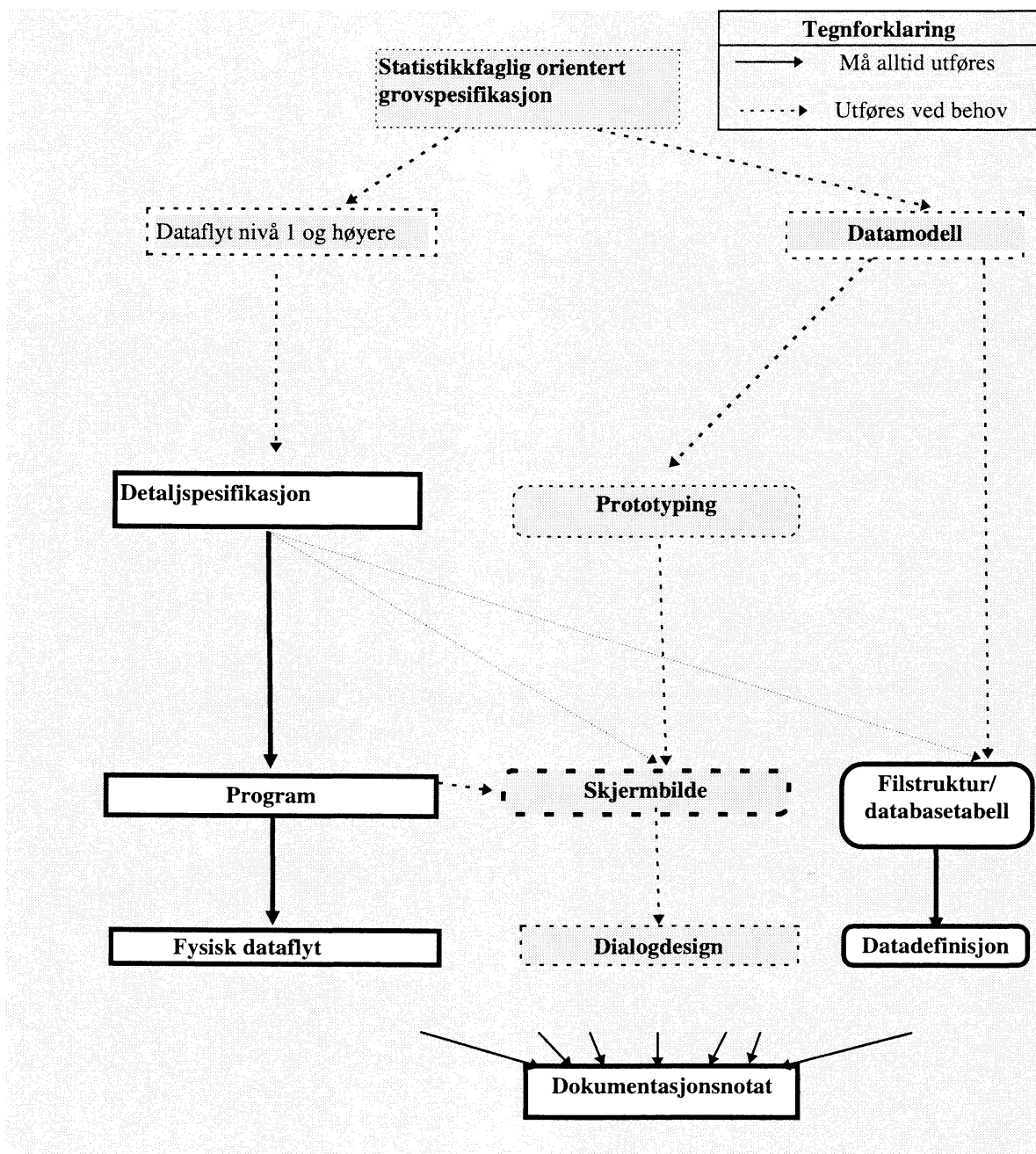
11. Vedlegg: Typisk dataflyt og datalagertyper i SSB



Figur: Typisk dataflyt og datalagertyper i SSB.

12. Vedlegg: Vedlikehold

Idé	Alt som gjøres etter at utviklingsprosjektet er ferdig er vedlikehold. Vedlikehold er et nytt prosjekt!
Ambisjonsnivå	Ved endringer i EDB-rutiner skal all dokumentasjon som berøres av endringene bli oppdatert. Dersom det mangler dokumentasjon må slik utarbeides.
Notat	-



Figur: Skisse av en vedlikeholdsmetode for SSB

**Beskrivelse:**

1. *Prosjektskriv*: Ikke påkrevet.
2. *Logisk dataflyt* og *datamodell*: Oppdateres hvis endringene er så omfattende at de gjenspeiles i diagrammene.
3. *Detaljspesifikasjoner*: Endringer i detaljspesifikasjoner gjøres i Word. For at det skal framgå klart hva som er siste endring, benyttes versjonshåndtering. Siden SSB pr. idag ikke har noe automatisert system for versjonshåndtering, må dette gjøres manuelt ved at alle notater og diagrammer påføres versjonsnummer. Et minimum av versjonshåndtering er at det tas vare på to versjoner: Gammel versjon: `q:\dok\<prosjekt>\dokument.Vxx` og gyldig (ny) versjon `q:\dok\<prosjekt>\dokument.doc`.
4. *Prototyping*: Dersom vedlikeholdet er så omfattende at prototyping må gjøres på nytt, bør det vurderes om dette ikke i stedet burde gjennomføres som nytt prosjekt.
5. *Fysisk dataflyt*: Oppdateres.
6. *Dialogsystem*: Utføres ved behov.
7. *Dokumentasjonsnotat*: Oppdateres. Dersom programmene inngår i tidligere versjon av dokumentasjonsnotatet, må de nye (eventuelt oppdaterte) programmene legges inn der.

**Erfaringer:**

1. Dersom det viser seg at et helt menysystem må lages, kan dette være et tegn på at prosjektet er et utviklingsprosjekt og ikke et vedlikeholdsprosjekt.

Teknisk anmerkning:

Endringer i detaljspesifikasjonene kan sammenlignes automatisk ved hjelp av Word tekstbehandler. Se «teknisk anmerkning under «detaljspesifikasjon».

13. Vedlegg: Noen erfaringer etter bruk av metoden

Jeg har faktisk begynt å bruke metodeopplegget dere har laget, dokumentasjonen og alt. Det TAR mye tid å dokumentere, men det er faktisk litt moro også, og faktisk temmelig nyttig iblant...

Ove Viggen

Jeg ser, når jeg leser håndboka, at den vil være et nyttig verktøy også for slike som meg, som skal følge opp prosjektene. Og jeg merker meg at EDB-kontorleder er ansvarlig for at det blir utarbeidet dokumentasjon. Jeg tror det vil virke positivt, også for den jobben jeg er satt til å utføre.

Rune Gløersen

Å bruke en felles metode for systemutvikling og dokumentasjon er en forutsetning for å organisere våre data hensiktsmessig for produksjon, gjenbruk og analyse på tvers av SSBs mangfoldige statistikk- og analyseområder.

Håkon Berby

Intet er mer oppklarende enn en god figur. Dataflytdiagrammer er helt nødvendige i kommunikasjon i prosjektet.

Leiv Solheim

De sist utgitte publikasjonene i serien Statistisk sentralbyrås håndbøker

- | | | | |
|----|---|----|---|
| 45 | Håndbok i datasikkerhet og fysisk sikring. 1994. 53s. | 54 | Internkontroll. 1995. 25s. |
| 46 | Telefonkatalog. 1996. | 55 | Nordisk statistikk på CD-ROM: Veiledning. 20s. |
| 47 | EØS-avtalen. Det statistiske samarbeid og konsekvenser for Statistisk sentralbyrås statistikkproduksjon. 1994. 55s. | 56 | PC-Axis versjon 2.2: Brukerhåndbok. 69s. |
| 48 | Håndbok i tilsettingssaker. 1994. 32s. | 57 | Produktregister versjon 4.0: Brukerveiledning. 49s. |
| 49 | Oppgaveplikt og tvangsmulkt. 1995. 55s. | 58 | Håndbok i prosjektstyring. 20s. |
| 50 | Emneinndeling 1995. 1995. 43s. | 59 | Personalreglement for Statistisk sentralbyrå. 22s. |
| 51 | Intervju: EDB-arbeidsbok. 1995. | 60 | Produktnummerkatalog pr. 28.02.1996. 55s. |
| 52 | Intervju: EDB-oppslagsbok. 1995. | 61 | Innkjøpshåndbok. 1996. |
| 53 | Intervju: Opplæring og administrasjon. 1995. | 62 | Timeplan versjon 3.0: Brukerveiledning. 16s. |



Statistisk sentralbyrå
Statistics Norway