

Handelshøjskole Syd, Kolding
cand. merc. aud.

TVÆRFAGLIGT SEMINAR:

**Klientens tilretteliggende (eller eget udviklede) EDB-systems
betydning for revisors arbejde.**

Opgaveløsere:

Mikael Juul Jensen
(studienr 41 2 90 025)

Peter Havskov Christensen
(studienr 41 2 92 024)

Vejleder:

Bent Warming-Rasmussen

INDHOLDSFORTEGNELSE

1	PROBLEMBAGGRUND	1
2	PROBLEMFÖRMULERING, SYSTEMATIK & AFGRÆNSNING	5
2.1	Problemformulering	5
2.2	Systematik	5
2.3	Afgrænsning	6
3	PROBLEMBEHANDLING	8
3.1	Hvilken virkning har EDB-udviklingen (specifikt vedr. udbredelsen af tilrettelige standardssystemer) for revisor ?	8
3.1.1	Påvisning af EDB-udviklingens betydning for revisor	8
3.1.2	Placering i revisionsprocessen (Generel for EDB-området)	14
3.1.3	Placering i revisionsprocessen (Specifik for tilretninger i standardssystemer)	19
3.1.3.1	Hvorfor er problemet med ændringer anderledes ved udbredelsen af tilrettelige standardssystemer på PC'ere ?	20
3.1.3.2	Hvorfor og hvornår revideres de administrative systemer for væsentlige ændringer ?	21
3.2	Tager kriteriesættet højde for EDB-udviklingen ?	23
3.2.1	Kriteriesættet vedr. ændringer/tilretninger	23
3.2.1.1	<Aggergren 1984>, EDB og den eksterne revisor	23
3.2.1.2	<Hougaard 1992>, Revision i PC-miljøer	26
3.2.1.3	<Klarskov 1994>, Dokumentation af systemer, metoder og procedurer	28
3.2.2	Cases	30
3.2.2.1	Case A: Manglende generelle kontroller	30
3.2.2.2	Case B: Konstater ændringer	31
3.2.2.3	Case C: Tilrettet faktureringsrutine	32
3.2.2.4	Case D: Tilføjet rutine til intern fakturering i profitcenter	33
3.2.2.5	Case E: Lagerliste anvendte forkerte data fra systemet	34
3.2.2.6	Case F: Udbetaling af for meget overtidstillæg	36
3.2.3	Vurdering af kriteriesættet	36
3.3	Hvorledes undersøger revisor hvilken gruppe systemet hører til og om systemet har fungeret uændret i hele perioden ?	38
3.3.1	Individuelle systemer og standardssystemer	38

3.3.2	Rammesystemer	43
3.3.3	Grupperingens betydning for revisionen	47
4	KONKLUSION	49
4.1	Problemstillingen	49
4.2	Hvad var de vigtigste hovedresultater	49
4.3	Hvad kunne vi have gjort bedre	51
4.4	Hvilke nye problemstillinger ser vi nu	51
5	LITTERATURLISTE	53
5.1	Bøger og publikationer	53
5.2	Tidsskrifter	55
5.3	Love m.v.	56
6	BILAG	58
6.1	Case A: Manglende generelle kontroller	58
6.2	Case B: Konstater ændringer	61
6.3	Case C: Tilrettet faktureringsrutine	63
6.4	Case D: Tilføjet rutine til intern fakturering i profitcenter	65
6.5	Case E: Tilrettet lagerliste anvendte forkerte data fra systemet	67
6.6	Case F: Udbetaling af for meget overtidstillæg	70
6.7	Kartoteksindhold	71
6.8	Efterbehandlingskode	78

1 PROBLEMBAGGRUND

De fleste virksomheder anvender i dag edb. En stor del af de oplysninger revisor baserer sit arbejde på stammer fra EDB-systemer. EDB-systemet er derfor en naturlig del af revisors arbejdsområde.

De første administrative EDB-systemer var baseret på store anlæg med separat udvikling og driftsafvikling. Efterhånden som udviklingen af hardware og software har muliggjort det, er mindre anlæg taget i anvendelse til administrative formål. Disse mindre anlæg består især af PC'ere (Personal Computer), der oprindeligt var tiltænkt en enkelt bruger. Ideen med PC'eren var, at brugeren skulle være sin egen operatør og selv varetage alle opgaver på maskinen. Udviklingen har bevirket, at PC'erne i større og større grad forbindes i netværk¹ og overtager de større maskiners områder².

PC'erens traditionelle styresystem (DOS) er udviklet til enkeltbrugere, og der er derfor ikke taget hensyn til funktionsadskillelse og adgangsbegrænsning. I netværksløsningerne flyttes vigtige data ofte til en server, hvor der er mulighed for adgangsbegrænsning.

De almindeligt anvendte administrative systemer på PC'ere har samme brugervenlighed og fleksibilitet (tilrettelighed), uanset om de anvendes på en enkeltstående PC'er eller i et netværk. Virksomheden står således nu med løsninger, hvor alle kan have mulighed for at tilpasse systemet efter individuelle ønsker.

Funktionsadskillelsen mellem udvikler og bruger som på de store systemer er nærmest selvfølgelig, da programmering kræver indgående kendskab til de traditionelle programmeringssprog, er således væk. Tilretning af de nuværende pc-baserede administrative systemer kan læres på et 2-dages kursus. Udover den manglende funktionsadskillelse giver dette også urutinerede "programmører".

Problemerne ved tilretteliggende standardsystemer er derfor specielle og får stigende betydning med øget udbredelse af disse systemer, specielt i mindre/mellemstore virksomheder.

Den beskrevne udvikling giver anledning til bl.a. følgende spørgsmål:

Hvilken virkning har EDB-udviklingen (specifikt vedr. udbredelsen af tilretteliggende standardsystemer) for revisor ?

¹ Ifølge analyseinstituttet IDC var 34% af alle kontor-pc'er i 1991 i Vesteuropa koblet til et LAN (Local Area Network), og det forventes at stige til 59% i 1996. Analysen er kort beskrevet i <Juhl 1993>.

² Denne udvikling kaldes ofte downsizing eller rightsizing <Friis 1994>.

Der er løbende udgivet en del skriftligt materiale om EDB og revision. Af de mere væsentlige kan nævnes (i tidsmæssig rækkefølge):

Yngre revisorer udgav i starten af 80'erne i deres artikelrække "faglige artikler" publikation nr 20, der indeholder 2 artikler: "EDB og den eksterne revisor" <Aggergren 1984> og "EDB, Grundlæggende begreber" <Tange 1984>. Disse artikler omhandler generelle forhold dels vedrørende revisor og EDB og dels fastlæggelse af nogle EDB-begreber.

FSR's revisionsvejledninger, der er udtryk for god revisionskik, omhandler i <Rev. vejl. 14> fra maj 1989 samt <Rev. vejl. 17> fra maj 1990 de krav, der stilles til revisor hvis virksomheden anvender EDB. De 2 vejledninger opdeler området i 3 underpunkter:

- a) Indledende vurdering af EDB-anvendelsen <Rev. vejl. 14>
- b) Gennemgang af generelle EDB-kontroller <Rev. vejl. 14>
- c) Gennemgang af EDB-baserede brugersystemer <Rev. vejl. 17>

I 1992 udarbejdede <Hougaard 1992> en kandidatafhandling der senere er udgivet som bog. Den omhandler primært revision i virksomheder der anvender PC'er.

FSR har i 1992 i samarbejde med IIA (Foreningen af Interne Revisorer i Danmark) og FIK (Foreningen for Informationskvalitet) udgivet publikationen "God EDB-skik"³ <Franck 1992>.

Publikationen definerer god EDB-skik som:

God EDB-skik er som norm de branchemæssige sædvaner og den praksis, der til enhver tid efterleves af kyndige og ansvarsbevidste fagfolk med henblik på at EDB-anvendelsen tilrettelægges og gennemføres på en klar, overskuelig og verificerbar måde. Endvidere at de sikkerhedsforanstaltninger, der skal iværksættes for at sikre en tilstrækkelig system-, data- og driftssikkerhed gennemføres, herunder at relevant lovgivning overholdes.

Revisors opgave i relation til god EDB-skik defineres nærmere i publikationens afsnit IV. Dette afsnit omhandler også revisors pligt til kontrol af overholdelse af relevant lovgivning.

³

Denne publikation er i 1994 udgivet på engelsk med titlen "Generally accepted IT Management practice". Der er tale om en direkte oversættelse af den danske publikation fra 1992.

Samarbejdet mellem FSR, IIA og FIK har i 1994 resulteret i udgivelsen af ringbindet "IT revision og sikkerhed" <IT rev. & sik.>, der indeholder flere artikler. På nuværende tidspunkt er følgende udkommet:

- 040 : Revision af organisatoriske forhold i et EDB-miljø <Torp 1994>.
- 070 : Dokumentation af systemer, metoder og procedurer <Klarskov 1994>.
- 080 : Fysisk sikkerhed <Saabye 1994>.
- 100 : Revision af databasesystemer <Søndergaard 1994>.
- 110 : Revision af datakommunikation <Groth 1994>.
- 140 : Revision af EDB-baserede brugersystemer <Skau 1994>.

Artiklerne har ikke status som revisionsvejledninger, men må antages at kunne anvendes som uddybende i forhold til disse og begrebet god EDB-skik.

På baggrund af den teknologiske udvikling er det relevant at spørge:

Tager kriteriesættet højde for EDB-udviklingen ?

Begrebet standardsystem er væsentligt for revisor da man traditionelt griber revision af applikationskontroller forskelligt an i standardsystemer og individuelle systemer. Dette er normalt begrundet i, at kontroller anses for bedre efterprøvede jo flere brugere der har anvendt systemerne.

<Tange 1984> definerer 3 typer af systemer:

- Individuelle systemer
- Standardsystemer
- Rammesystemer⁴

Definitionen af individuelle systemer og standardsystemer er åbenlyse. Rammesystemer, som er en mellemting mellem de 2 forannævnte, defineres således af <Tange 1984>:

Et rammesystem er et forsøg på at kombinere den store tilpasning til brugerens ønsker, som et individuelt system giver, med prisbilligheden i et standardsystem. Rammesystemet indeholder alle de karakteristiske delsystemer såsom indlæsning, kartoteksvedligeholdelse og udskrivninger, fastlagt på et "overordnet niveau" hvor

⁴ Disse betegnes i dag som tilretteliggende standardsystemer.

brugeren så selv kan fastlægge alle detaljerne (udseende af udskrifter, kartoteksindhold m.v.), og dermed give systemet et præg af et individuelt system.

Rammesystemer er i dag markedsstandard for PC-baserede⁵ administrative systemer.

Disse leveres normalt med et standardudseende⁶ for udskrifter og kartoteksindhold. Brugeren kan herefter vælge mellem at definere sit eget udseende, anvende standarden eller rette i standarden.

Ideen med de tilrettelige standardsystemer er bl.a., at brugeren skal opleve systemet som en helhed og ikke i den daglige brug spekulere over hvad der er standard og hvad der er specielt for den pågældende virksomhed. Dette opnås ved, at de tilrettede funktioner integreres fuldt ud i systemet. Der kræves derfor en detailviden om de enkelte tilrettelige standardsystemer for at afgøre om en given funktion i et system er designet af brugeren, er en standardfunktion eller er en tilrettet standardfunktion.

Da revisionen må gribes forskelligt an i de forskellige grupper af systemer er det relevant at stille følgende spørgsmål:

Hvorledes undersøger revisor hvilken gruppe systemet hører til og om systemet har fungeret uændret i hele perioden ?

⁵ Rammesystemer findes til andre maskintyper end PC'ere, men problemerne for revisor er de samme og PC-systemerne er de mest udbredte.

⁶ Programmer til økonomistyring kan meget let tilrettes efter behov såfremt man har anskaffet den del af systemet der muliggør dette. Dette skyldes, at programmer som Concorde XAL og Navigator er udviklet i et 4. generationsprogrammeringssprog (4 GL er det p.t. nærmeste computersprog kommer på almindeligt sprogbrug - i modsætning til 1., 2. og 3. generationssprog). Selve 4. GL sproget kan ikke ændres, men det kan de medleverede applikationer til økonomistyring.

2 PROBLEMFORMULERING, SYSTEMATIK & AFGRÆNSNING

2.1 Problemformulering

Udviklingen som beskrevet i afsnittet om problembaggrund har givet anledning til følgende spørgsmål:

- a) Hvilken virkning har EDB-udviklingen (specifikt vedr. udbredelsen af tilretteliggende standardsystemer) for revisor ?
- b) Tager kriteriesættet højde for EDB-udviklingen ?
- c) Hvorledes undersøger revisor hvilken gruppe systemet hører til og om systemet har fungeret uændret i hele perioden ?

Nærværende seminars formål er at forsøge at besvare disse 3 problemstillinger.

2.2 Systematik

Problemstillingerne som nævnt i afsnittet problemformulering behandles således ud fra nedenstående valg:

ad a En oplagt mulighed til besvarelse af denne problemstilling er empiriske undersøgelser. Dette kræver et forholdsvis detaljeret spørgeskema samt en rimelig stor mængde besvarelser. De spørgsmål der skulle være i et sådant spørgeskema er ikke kendt på nuværende tidspunkt. Desuden gør omfanget af dette seminar ikke denne metode mulig.

En anden mulighed er at tage udgangspunkt i relevant litteratur. Den til rådighed værende litteratur behandler ikke specifikt emnet tilretteliggende standardsystemer. Litteraturen baserer sig primært på den oprindelige opdeling i standardsystemer og individuelle systemer.

Ud fra ovennævnte muligheder har vi valgt, først at fastslå at EDB-udviklingen har betydning for revisor og dernæst at beskrive hvorledes EDB indplaceres i revisionsprocessen både generelt og specifikt. Herunder behandles hvilke problemer revisor kan støde på og hvilke problemer der er specielle for tilretteliggende standardsystemer (rammesystemer).

ad b Afprøvning af kriteriesættet kan foretages på flere måder. En empirisk undersøgelse vil her være mulig, men omfanget af seminaret bevirker, at metoden falder uden for de fastsatte rammer. I stedet for den ovennævnte statistisk korrekte metode er valgt en undersøgelsesmetode baseret på cases fra egne praktiske erfaringer. En sådan caseundersøgelse giver intet statistisk grundlag for konklusion. Problemstillingerne i casene er derfor ikke nødvendigvis udtømmende for test af kriteriesættet, men kan dog påvise nogle problemstillings relevans.

Systematikken for besvarelse af denne problemstilling bliver derfor:

- ☛ Beskrivelse og tolkning (via supplerende litteratur) af kriteriesættet specifikt vedr. ændringer/tilretninger.
- ☛ Ved hjælp af cases at afprøve kriteriesættet.

ad c Ved besvarelse af denne problemstilling kunne empiri anvendes. På baggrund af valgene i problemstillingerne a og b er det fundet mere relevant, at forsøge at anvende de i litteraturen anviste metoder til at konstatere, om et system er tilrettet så det har betydning for revisors arbejde. Forudsætningerne for metodernes anvendelse vil blive diskuteret.

Problemstilling c besvares ved:

- ☛ fremdragning af kriterier for om et system er tilretteligt
- ☛ fremdragning af mulige kriterier for hvornår et system er tilrettet (så det har betydning for revisor)

2.3 Afgrænsning

Begrebet EDB-revision anvendes i flere af vore kilder. Det anvendes i 2 forskellige betydninger:

- ☛ revisors arbejde i forbindelse med vurdering af kontrolmiljø og herunder generelle EDB-kontroller og applikationskontroller (revision af edb)
- ☛ EDB som hjælpeværktøj (revision ved hjælp af edb)

Seminaret omhandler "revision af EDB" og ikke "revision ved hjælp af EDB". Revision ved hjælp af EDB er mest egnet til substansrevision og har ikke nogen særstatus i revisionsprocessen i forhold til anden (manuel) substansrevision⁷.

⁷

Revision ved hjælp af EDB ændrer på mulige stikprøvestørrelser indenfor et rimeligt ressourcforbrug, men kan ellers ikke tilføre revisionen noget som manuel substansrevision med samme stikprøvestørrelse ikke kan.

Seminaret omhandler ikke de specielle problemer revisor kan stå overfor, hvis grundbilag ikke forefindes i klarskrift. Som eksempel kan nævnes EDI⁸. Det kan her tænkes, at EDB-værktøjer kan være nødvendige for revisionens gennemførelse. Området er relevant, men er også meget omfattende og ligger uden for dette seminars rammer. Området må endvidere betegnes som en delproblematik under EDB's betydning for revisor. Denne delproblematik er ikke speciel for tilretteliggende standardssystemer.

Seminaret vil ikke indeholde en generel teknisk beskrivelse af standardssystemer til økonomistyring, men beskrive de problemer revisor kan opleve ved tilretteliggende systemer. Dog vil krav til revisors viden om systemerne blive diskuteret.

Seminaret omhandler primært de systemer der af <Hougaard 1992> side 29-30 kategoriseres som økonomisystemer.

⁸

EDI = Electronic Data Interchange (Elektronisk dataudveksling).

3 PROBLEMBEHANDLING

3.1 Hvilken virkning har EDB-udviklingen (specifikt vedr. udbredelsen af tilretteliggende standardsystemer) for revisor ?

Helt overordnet er det af væsentlig betydning at få fastlagt om EDB-området overhovedet har betydning for revisors primære arbejdsområde (at bibringe regnskaber m.v. øget troværdighed gennem sin uafhængige påtegningsfunktion), fordi det er den forudsætning, der bygges på i den videre behandling.

3.1.1 Påvisning af EDB-udviklingens betydning for revisor

Det søges påvist via følgende områder, at EDB-udviklingen har betydning for revisor:

Fremtidsmæssigt perspektiv:

<Arens 1991> forudsiger, at visse af nutidens udviklingstendenser, herunder den teknologiske, gør at revisor må udvikle sig med tiden for at bevare sin rolle som en af fremskafferne af værdifulde beslutningsinformationer. Det forudses bl.a. at revisor vil blive mere "accounting-information-systems oriented". Dette vil blive en nødvendighed dels for at kunne leve op til virksomhedernes og den øvrige omverdens forventninger dels for at sikre revisor en fortsat vigtig rolle i fremskaffelsen af informationer. Han siger derfor bl.a. at:

As a result, all accountants must be knowledgeable about computer technology, and some must have highly specialized training.

Folketinget og revisorstanden selv har ligeledes givet edb-området en væsentlig placering i revisors arbejde (både direkte og indirekte) hvilket fremgår af bl.a. følgende lov- og god skik-regler:

Lovregler:

§ 6a stk 2 i <Lov om SR>

Revisor skal for at leve op til paragraf 6a stk 2 *udføre sit hverv med omhu, nøjagtighed og den hurtighed, som hvervets beskaffenhed tillader, samt i overensstemmelse med god revisorskik.*

Det betyder, at revisor ved hver opgave må vurdere, hvorledes denne løses mest effektiv under hensyntagen til de til enhver tid gældende love og god revisorskik-regler. For at kunne foretage denne vurdering kvalificeret er revisor "tvunget" til at følge med i den udvikling, der sker indenfor revisors arbejdsområder - herunder også den teknologiske udvikling.

§ 16 stk 3 i <Lov om SR>

Revisor, der i sin virksomhed skriftligt bevidner noget, hvorom den pågældende ingen kundskaber har, straffes med bøde eller hæfte, medmindre strengere straf er forskyldt efter den øvrige lovgivning.

Revisor må derfor have et kendskab til det genstandsfelt hvorom han afgiver sin påtegning. De administrative systemer er tit med til at "producere" og opbevare dette. Det er derfor nødvendigt for revisor at have et så stort et kendskab til de administrative systemer, at han kan overskue hvilken virkning, systemerne har på genstandsfeltet i det enkelte tilfælde.

<Bogf.bekendtg.>

Bekendtgørelsen gør, at revisor er nødt til at følge med i den løbende udvikling i virksomhedernes administrative systemer for at kunne påse om systemerne og virksomhedernes øvrige materialer samt procedurer omkring dette sikrer, at virksomheden opfylder bekendtgørelsens krav⁹ til registreringer, beskrivelser, grundbilag, åbningsbalance og årsregnskab samt opbevaring af regnskabsmateriale.

<A/S loven> §54 stk 3 samt <ApS loven> §36 stk 2

Begge paragraffer pålægger bestyrelse og direktion opgaver:

Bestyrelsen skal påse, at bogføringen og formueforvaltningen kontrolleres på en efter selskabets forhold tilfredsstillende måde.

Direktionen skal sørge for, at selskabets bogføring sker under iagttagelse af lovgivningens regler herom, og at formueforvaltningen foregår på betryggende måde.

⁹

I <CW 1994-13> som bygger på interview med Lars Ole Wiese m.fl. nævnes at *kun de færreste regnskabssystemer på markedet overholder bogføringsbekendtgørelsens krav om systembeskrivelse*. Artiklen og svar herpå i <CW 1994-16> omhandler hvor dybdegående systembeskrivelser skal være og hvem (leverandør/bruger) der bør udarbejde disse beskrivelser. Desuden behandles betydning af den "manglende" systembeskrivelse for revisors påtegning.

Disse bestemmelser har betydning for revisor, da han/hun har pligt til at oplyse i påtegningen, hvis der ved revisionen er fundet forhold, som kan medføre ansvar for bestyrelsesmedlemmer, direktører eller andre ledelsesorganer jf. <Påtegn.bekendtg.> §3 stk 4.

Revisor påtager sig desuden i henhold til god skik-reglerne denne opgave.

God skik-regler:

Af de aktuelle danske revisionsvejledninger (1 til 19) er det primært følgende der har speciel interesse ved vurderingen af om edb-udviklingen har betydning for revisor:

<Rev. vejl. 1, 14 og 17>

<Rev. vejl. 1> opdeler i afsnit 2 opgaverne mellem revisor og ledelse. Revisor påtager sig ifølge denne bl.a. ved en revision at gennemgå og vurdere, om de af ledelsen tilrettelagte registreringssystemer m.v. giver et pålideligt grundlag for udarbejdelsen af regnskabet¹⁰.

Revisor skal endvidere meddele ledelsen, hvilke væsentlige mangler der er forefundet vedrørende de regnskabsmæssige registreringssystemer m.v.¹¹.

Dette gør, at revisor for at kunne løse opgaven må indsamle informationer om virksomhedens registreringssystemer m.v. Det fremgår af punkt 6.4 i <Rev. vejl. 1>, at revisor under informationsindsamlingen skal foretage en overordnet gennemgang af virksomhedens økonomisystemer, edb-anvendelse og væsentlige interne kontroller for at vurdere om systemer og procedurer kan forventes at identificere, registrere, behandle og rapportere regnskabsmæssige data rettidigt og pålideligt. Denne gennemgang skal gennemføres i overensstemmelse med <Rev. vejl. 14> afsnit 2.

<Ipsen 1993> skriver: *Det er vigtigt at slå fast, at muligheden for at forhindre såvel utilsigtede som til-sigtede fejl eller mangler, herunder besvigelser eller uregelmæssigheder først og fremmest beror på, om der ved tilrettelæggelsen af registreringssystemer og forretningsgange sikres en god intern kontrol.*

<Rev. vejl. 14> omhandler, foruden den overordnede gennemgang, hvilke områder revisor skal gennemgå hvis edb-anvendelsen findes betydende eller risikofyldt.

¹⁰ <Rev. vejl. 1> punkt 2.4

¹¹ <Rev. vejl. 1> punkt 2.5

<Rev. vejl. 17> omhandler hvilke områder revisor skal gennemgå omkring de edb-baserede brugersystemer. Vejledningen skal anvendes når revisor har overbevist sig om, at den del af de generelle edb-kontroller, der vedrører sikring af systemer og data, er tilstrækkelig til, at revisionen helt eller delvist kan bygge på de programmerede kontroller i virksomhedens brugersystemer¹².

Alle 3 vejledninger gør, at revisor er nødt til at følge med i edb-udviklingen for at leve op til god skik-reglerne.

God EDB-skik <Franck 1992>

Publikation beskriver særskilt hvad der er god EDB-skik. Vejledningen er primært relateret til virksomhedernes overordnede og daglige ledelser, hvorfor EDB-området belyses ud fra en ledelsesmæssig synsvinkel.

Dette til trods er det yderst væsentligt for revisor, at han/hun følger med i hvad der er god EDB-skik for at kunne bedømme, om der i den aktuelle situation er tale om en virksomhed, der arbejder med en god eller dårlig EDB-skik. Vurderingen af virksomhedens EDB-skik påvirker revisors vurdering af kvaliteten af de informationer der modtages fra de administrative systemer.

Flere og flere typer information opbevares i de administrative systemer og vurderingen af kvaliteten får derfor øget betydning.

Publikationen indeholder i afsnit IV desuden en oversigt over revisionens opgaver, som kort defineres som:

Revisionens funktion er, på vegne af virksomhedens ejere og uafhængigt af virksomhedens funktion, at vurdere omfanget af og efterprøve brugen af fastlagte forretningsgange og hensigtsmæssige interne kontroller og kontrolsystemer.

<IT Rev. & sik.>

Artiklerne omhandler flere emneområder. Disse har til formål dels at operationalisere nogle af hovedområderne i <Rev. vejl. 14 og 17> dels at tage nye generelle problemområder op og operationalisere disse.

¹²

<Rev. vejl. 17> punkt 1.3

De nye områder er medtaget i <Rev. vejl. 14 eller 17>, men kun som "biområder". Udviklingen siden udgivelsen af revisionsvejledningerne har gjort, at områderne har fået en væsentlig større rolle for revisors vurdering af de administrative systemer. Det har derfor været nødvendigt at uddybe og supplere revisionsvejledningerne.

Der er indtil videre udgivet 6 artikler som kan opdeles og sammenholdes med <Rev. vejl. 14 og 17> således:

Artikler i <IT rev. & sik.>	Sammenhæng til <Rev. vejl. 14 og 17>
040: Revision af organisatoriske forhold i et EDB-miljø <Torp 1994>	Operationalisering af <Rev. vejl. 14> punkt 3.4 vedrørende organisatoriske forhold.
070: Dokumentation af systemer, metoder og procedurer <Klarskov 1994>	Operationalisering af <Rev. vejl. 14> punkt 3.4 vedrørende dokumentation af systemer, metoder og procedurer.
080: Fysisk sikkerhed <Saabye 1994>	Operationalisering af <Rev. vejl. 14> punkt 3.4 vedrørende fysisk sikkerhed.
100: Revision af databasesystemer <Søndergaard 1994>	Udarbejdet fordi flere og flere systemer anvender relationsdatabase teknik, hvor data lagres et sted og alle brugersystemer trækker data herfra. Det medfører en 2 deling af revisionen af disse systemer. En af det/de udvalgte brugersystem(er) og en af databasen og dens konsistens. Denne artikel må betegnes som operationalisering af et "nyt" hovedproblemområde i forhold til <Rev. vejl. 14 og 17>.
110: Revision af datakommunikation <Groth 1994>	Udarbejdet fordi området datakommunikation er i rivende udvikling. Det er derfor vigtigt at få operationaliseret dette område, således at revisor kan afdække evt. risikofaktorer i kommunikationen. Denne artikel må betegnes som operationalisering af et "nyt" hovedproblemområde" i forhold til <Rev. vejl. 14 og 17>.
140: Revision af edb-baserede brugersystemer <Skau 1994>	Operationalisering af <Rev. vejl. 17>.

Kilde: Egen tilvirkning

Artiklerne har ikke status som revisionsvejledninger men må antages at være uddybende i forhold til disse.

Afrunding:

Alle de beskrevne områder påvirker i mere eller mindre grad revisors arbejdsområder og påviser at EDB-udviklingen har og vil få stigende indflydelse herpå.

Man kan kort opsummere dem i følgende hovedpåvirkningsgrupper:

- a) Revisors fremtidige eksistensberettigelse i den nuværende form som informationsfremskaffer og "tilfører" af troværdighedsværdi.

Revisor skal være opmærksom på at hans rolle ændrer sig. Ellers bliver værdien af revisors arbejde "udvasket" og dermed bliver revisor overflødig.

- b) Udviklingen bevirker, at flere og flere af de interne kontroller, som revisor kan bygge sin revision på, bliver programmerede.

Det bevirker, at såvel af hensyn til regnskabets udarbejdelse og troværdighed som af hensyn til revisors rapporteringspligter med hensyn til den interne kontrol, er det nødvendigt at gennemgå virksomhedens edb-anvendelse. <Tjagvad 1993>

- c) Virksomhedernes forventninger til revisors kundskaber.

Revisor må tilpasse sine metoder til teknologien for, at omverden skal bevare tiltroen til det udførte arbejde.

- d) Lovkrav.

Svært at vurdere om f.eks. <Bogf.bekendtg.> er opfyldt, hvis man ikke gennemgår virksomhedens edb-anvendelse.

e) Gældende god skik-regler.

Det væsentligste krav er her, set ud fra edb-anvendelsens betydning, punkt 2.3 i <Rev. vejl. 14>:

Revisor skal ud fra væsentlighed og risiko vurdere edb-anvendelsens betydning for virksomheden, herunder

- ☞ *om virksomheden kan risikere væsentlige økonomiske tab ved reduktion eller bortfald af virksomhedens edb-kapacitet;*
- ☞ *om ødelæggelse af elektronisk registrerede data kan medføre, at regnskaber ikke kan aflægges og/eller ikke kan revideres;*
- ☞ *om virksomhedens edb-anvendelse kan medføre åbenbare fejl, mangler eller usikkerheder i regnskaber.*

Efter at have konkluderet, at EDB-anvendelse kan have betydning for revisors arbejde, bliver det næste spørgsmål hvorledes placeres gennemgang af EDB i revisionsprocessen.

3.1.2 Placering i revisionsprocessen (Generel for EDB-området)

Revisor må afgøre, hvornår og hvordan delområdet revision af administrative EDB-systemer skal indarbejdes i revisionsprocessen for at kunne leve op til de under 3.1.1 nævnte forventninger og krav.

<Rev. vejl. 1> angiver grundlæggende principper for, hvordan revisionen skal gennemføres for, at revisor kan afgive en revisionspåtegning på et regnskab.

Overordnet for revisors arbejde er væsentlighed og risiko. Det defineres således i <Rev. vejl. 1>¹³, at hovedvægten lægges på de poster i regnskabet og de dele af registreringsystemerne inden for hvilke risikoen for væsentlige fejl er størst. Det fastlægges endvidere i vejledningen¹⁴, at væsentlighed skal vurderes såvel ud fra regnskabets helhed som dets enkelte dele. Revisor fastsætter væsentligheden ud

¹³ Punkt 4.1

¹⁴ Punkt 4.2

fra en vurdering af, hvilke forhold der forventes at påvirke regnskabslæserens bedømmelse af regnskabet.¹⁵

Man kan på baggrund heraf opstille følgende oversigt til brug for vurderingen af om det/de enkelte brugersystem(er) skal revideres:

Er systemet væsentligt og/eller risikofyldt set i forhold til regnskabet som helhed og/eller væsentlige poster heri ?

	Risikofyldt	Ikke risikofyldt
Væsentlig	Revidere systemet	Revidere systemet
Uvæsentlig	Hovedregel: Ikke revidere systemet. (Det må dog bero på en konkret vurdering)	Ikke revidere systemet

Kilde: Egen tilvirkning

Vurderingen af væsentlighed og risiko bygges bl.a. på systemstørrelse/kompleksitet samt hvilke regnskabsposter, der er afhængige af hvilke brugersystemer.

Vægtningen, af de enkelte systemer og applikationer på baggrund af om de er væsentlige og/eller risikofyldte for det grundlag revisor anvender til at afgive sin revisionspåtegning på, kan foretages på forskellige måder. Målet er, uanset hvilken man anvender, at få udvalgt de brugersystemer som har størst indvirkning på påtegningsgrundlaget og dermed giver størst revisionsmæssig værdi. <Skau 1994 side 6> giver vejledning om hvilke faktorer der under alle omstændigheder må indgå i en sådan "vægtningsmodel":

- ☛ Antals- og beløbsmæssig størrelse af de transaktioner der behandles af brugersystemet og som herigennem påvirker regnskabet. Dette kan udtrykkes gennem en væsentlighedsfaktor (VF). Afgørende for VF kan også eksempelvis være systemets anvendelse som grundlag for væsentlige

¹⁵

Jf <Rev. vejl. 1> punkt 4.2 anvendes væsentlighedskriteriet ikke ved revision af specifikke lovkrævede oplysninger i regnskabet. Risikoen skal ved planlægningen og udførelsen relateres til de i vejledningen punkt 4.4 nævnte 6 revisionsmål.

ledelsesmæssige beslutninger (beslutningsstøttesystemer) eller systemets betydning for virksomheden som going concern.

- ☛ Risikoen for at der opstår en fejl i forbindelse med brugersystemet. Dette kan udtrykkes som iboende risiko (**IR**), d.v.s. risikoen for at fejl/tab opstår.
- ☛ Risikoen for at opståede fejl ikke bliver opdaget af brugersystemets interne kontroller. Dette kan udtrykkes som intern kontrol risiko (**IKR**).
- ☛ Risikoen for at væsentlige fejl ikke afsløres under revisionen. Dette kan udtrykkes som opdagelsesrisikoen (**OR**).

Prioriteringen foretages herefter ud fra de enkelte brugersystemers risikoprofiler (**RP**) som udregnes ud fra de 4 faktorer:

$$RP = VF * IR * IKR * OR^{16}$$

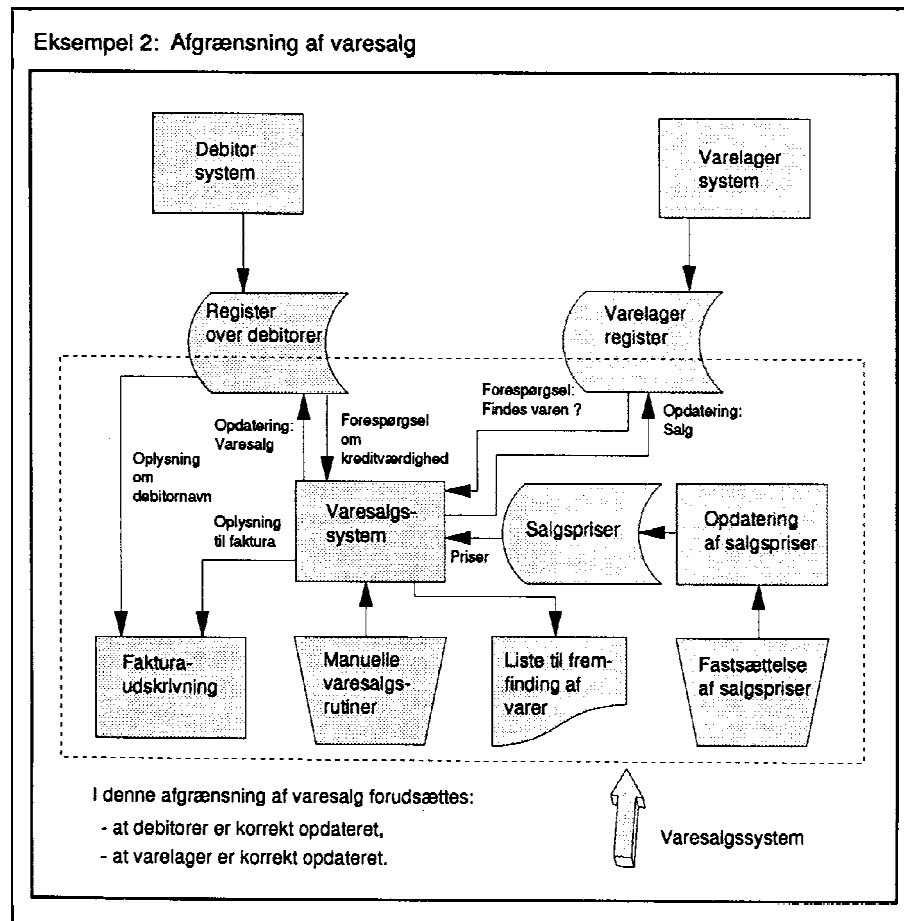
På baggrund af prioriteringen fastlægges revisionsstrategien og i den forbindelse sker planlægningen af hvilke brugersystemer/applikationer der skal revideres.

Det er vigtigt, i den sammenhæng at få foretaget en afgrænsning af det/de valgte brugersystem(er)/applikation(er) til de øvrige for, at klarlægge hvor input kommer fra, hvor der leveres output hen og hvilke programmerede og manuelle kontroller, der er tilknyttet det/de udvalgte brugersystem(er)/applikation(er).

Herved sikres det, at alle relevante områder revideres og at områder ikke revideres to gange.

¹⁶ Den svarer til modellen som er nævnt i <Rev. vejl. 1> punkt 4.3. Er bare suppleret med VF til direkte prioritering af de enkelte brugersystemer.

Arbejdsrapporten for afgrænsningen kan f.eks. se sådan ud:



Kilde: <Skau 1994> side 7

Revisionen af de valgte brugersystemer/applikationer gennemføres i overensstemmelse med <Rev. vejl. 14 og 17>. Det foreskrives i <Rev. vejl. 17>, at vejledningen finder anvendelse, hvor revisor har overbevist sig om, at visse dele af <Rev. vejl. 14> gennemgangen¹⁷ ikke har givet anledning til bemærkninger.

Det er dog vor opfattelse, at man som minimum uanset resultatet af <Rev. vejl. 14> gennemgangen må foretage vurdering af, om systemet lever op til lovgivningskrav, herunder specielt <Bogf.bekendtg.> krav.

¹⁷

Se punkt 1.3 i <Rev. vejl. 17>

Brugen af <Rev. vejl. 14 og 17> kan illustreres via følgende model:

Indledende vurdering:

<Rev.vejl.14> afsnit 2

Væsentlig / risikofyldt:¹⁸

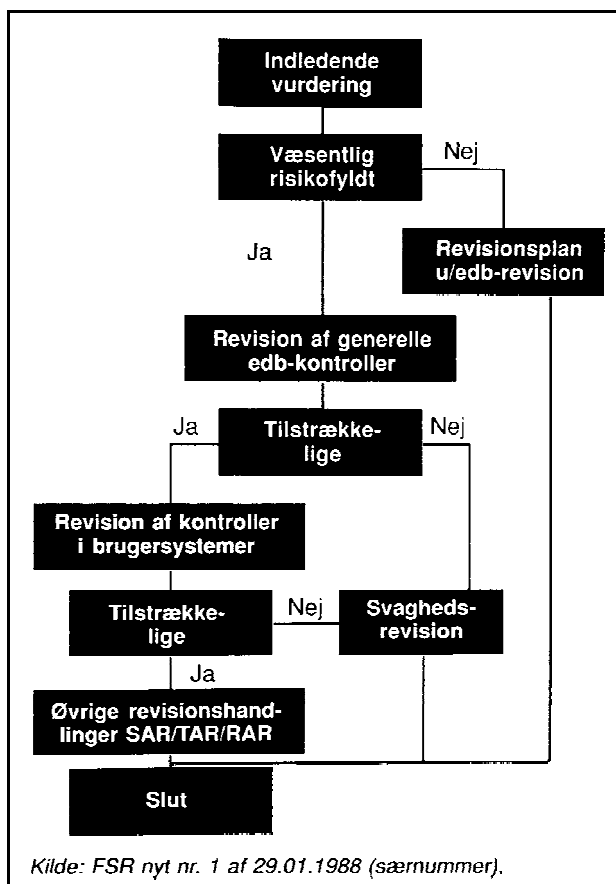
<Rev.vejl.14> pkt. 2.4+2.5

Rev. af gen. EDB-kontroller:

<Rev.vejl.14> afsnit 3

Tilstrækkelige:

Se næste model der er mere detaljeret og erstatter resten af denne model



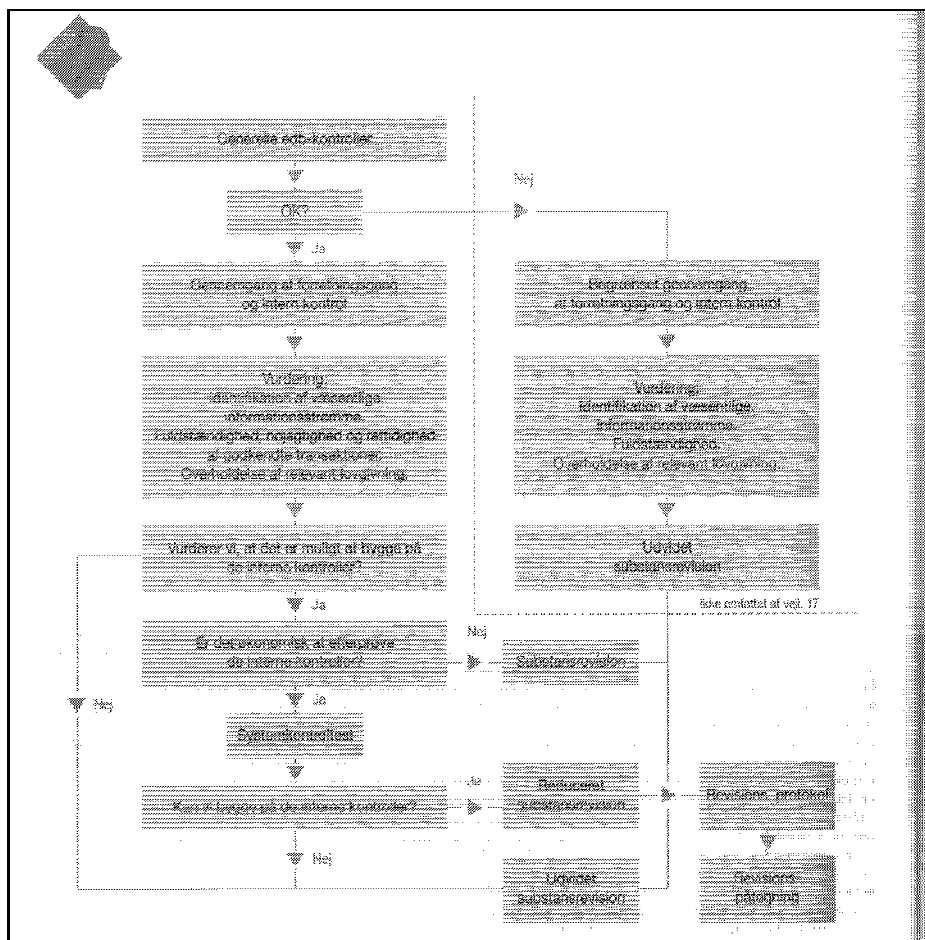
Rev.plan u. EDB-rev:
Ingen gennemgang af EDB

Kilde: <Schroeder 1993> side 31

¹⁸

Se vor model på side 15 der angiver, at der ikke nødvendigvis skal revideres i situationen risikofyldt men uvæsentlig.

Modellen på foregående side giver overblik over anvendelsen af <Rev. vejl. 14 og 17>. Den nederste halvdel af modellen er mere uddybet i følgende model:



Kilde: <Tjagvad 1993> side 14

3.1.3 Placering i revisionsprocessen (Specifik for tilretninger i standardssystemer)

Som det fremgår af problembaggrunden har edb-udviklingen ført til, at de mest udbredte administrative systemer på PC'ere i dag er tilretteliggende standard systemer.

Det er specielt udviklingen i programmeringssprog og disses anvendelse i de administrative systemer, kombineret med at brugerne er blevet mere edb-vante og avancerede, der gør, at revisor i dag kan stå med et meget brugerstyret systemmiljø. Hermed menes et system, hvor det er brugerne, der har foretaget tilretningerne af systemet.

Dette medfører både muligheder og risici for revisor.

Det er derfor vigtigt dels at få klarlagt hvilke problemer det medfører og dels hvordan vi placerer det revisionsmæssigt.

3.1.3.1 Hvorfor er problemet med ændringer anderledes ved udbredelsen af tilrettelige standardsystemer på PC'ere ?

Da systemerne er opbygget til at være fleksible med hensyn til individuelle brugerønsker (via tilretningsmoduler/designmoduler) vil revisor oftere møde ændrede administrative systemer. Det er derfor væsentligt for revisor, at få fastlagt om der er sket ændringer og i givet fald om de har betydning for revisor.

Der er specielt 2 væsentlige problemstillinger som kan forekomme:

- ☛ Alle brugerne kan som udgangspunkt lave tilretninger i systemerne uden at have et specielt kendskab til programmeringssprog m.v.

Det medfører, at man kan stå med en "blanding" af tilretninger, hvor "program-møren" kan være alt fra god til dårlig. Desuden vil systemudviklingen dermed ofte være tilfældig (mange udviklere) og risikoen for fejl bliver dermed større.

- ☛ Tilretningerne bliver som udgangspunkt integreret fuldt ud i systemerne med det samme. Det bevirker, at en bruger stort set ikke har mulighed for at se forskel på standarden og tilretninger foretaget af øvrige brugere.

Det medfører, at man kan miste kontrollen med systemudviklingen således at ændringer ikke bliver registreret, detailbeskrevet og brugertestet inden de tages i anvendelse.

Denne risiko er stor, fordi de fleste brugere kommer fra standardsystemer og dermed ikke har været vant til at foretage systemtilretninger. Dette kan medføre fejldata og -dispositioner i virksomheden, såfremt der ikke eksisterer effektive kontroller, der kompenserer for dette.

Desuden er den optimale funktionsadskillelse ikke tilstede når brugeren selv tilretter systemet, hvilket giver besvigelsesmuligheder m.v. Også dette skal afdækkes med kompenserende kontroller.

3.1.3.2 Hvorfor og hvornår revideres de administrative systemer for væsentlige ændringer ?

Ved udarbejdelsen af revisionsstrategien, som tidligere beskrevet, bliver omfanget af revisionsindsatsen fastlagt, så den ønskede sikkerhed for påtegningen opnås.

Gennemførelsen af den fastlagte revisionsindsats sker efter følgende revisionsmetoder:

- Systemrevision¹⁹
- Substansrevision, som kan opdeles i:
 - Regnskabsanalytisk revision²⁰
 - Detailrevision²¹

Revisor reviderer de administrative systemer som led i gennemførelsen af systemrevisionen. Det vurderes, hvor godt systemerne virker herunder specielt systemets indre sammenhænge samt brugen af de interne kontroller (både de programmerede og manuelle) for:

- at påse om den dybere substansrevision kan begrænses og
- at kunne vurdere om relevant lovgivning overholdes.

Når systemerne ændres kan både interne sammenhænge og kontroller blive ændret. Det bevirker, at man ikke kan udtale sig om systemet, før man har vurderet den enkelte ændrings betydning for systemernes sammenhænge og kontroller.

Da systemrevision ofte gennemføres i årets løb er det vigtigt, at revisor laver en revision af de resterende måneder i forbindelse med statusrevisionen, da der skrives under på, at systemet har fungeret i hele regnskabsåret.

Revisor behøver ikke at kontrollere alle de for påtegningsgrundlaget væsentlige administrative systemer hvert år, hvis det erfaringsmæssigt har vist sig, at systemerne er pålidelige²², såfremt der ikke er sket væsentlige ændringer indenfor det/de udvalgte system(er) siden sidste revisions gennemførelse.

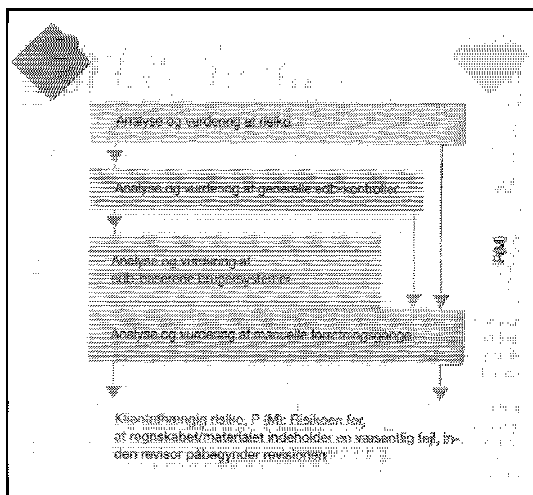
¹⁹ Er nærmere defineret og beskrevet i <Rev. vejl. 1> punkt 7.4 til 7.8.

²⁰ Defineret og beskrevet i <Rev. vejl. 1> punkt 7.10. Anvendelse i punkt 7.12.

²¹ Defineret og beskrevet i <Rev. vejl. 1> punkt 7.11. Anvendelse i punkt 7.12.

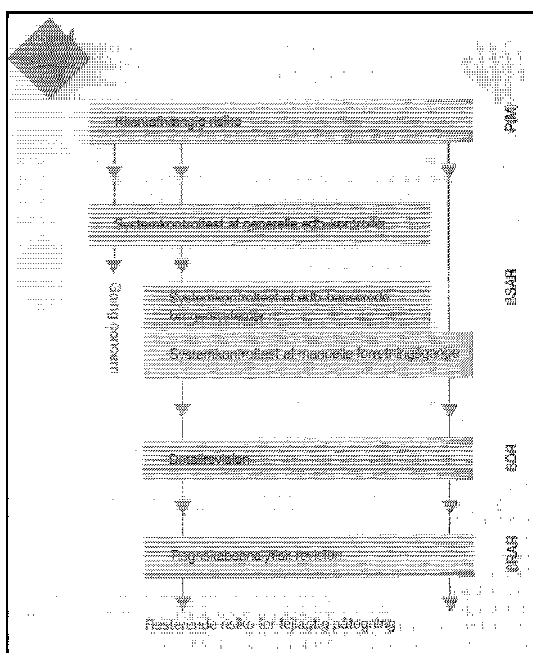
²² Jf. <Rev. vejl. 1> punkt 7.6.

Ovenstående proces kan vises via følgende modeller. Revisor starter med den samlede klientafhængige risiko som vist i nedenstående model:



Kilde: <Tjagvad 1993> side 15

Revisors opgave er billigst muligt, at komme fra den samlede klientafhængige risiko til den for revisor acceptable revisionsrisiko²³. Dette er illustreret i efterfølgende model:



Kilde: <Tjagvad 1993> side 16

Alt dette gør, at det er væsentligt for revisor, at kunne konstatere om et system er ændret. Det undersøges derfor, om revisor med udgangspunkt i kriteriesættet kan afgøre om et system er ændret.

23

Risikoen for at give fejlagtig påtegning.

3.2 Tager kriteriesættet højde for EDB-udviklingen ?

3.2.1 Kriteriesættet vedr. ændringer/tilretninger

Den del af det p.t. alment accepterede kriteriesæt blandt revisorer, som har noget med ændringer at gøre er <Rev. vejl. 14>, herunder specielt punkt 3.4 vedr. dokumentation af systemer, metoder og procedurer, punkt 3.6 om der er betryggende funktionsadskillelse mellem systemudvikling og drift samt punkt 3.9 om der er betryggende retningslinier for dokumentation af systemer og systemudvikling.

Hvis man skal bruge vejledningen som arbejdsredskab, vil man hurtigt komme i problemer, fordi den er alt for overordnet opbygget. Man må derfor søge efter skriftligt materiale, som sætter noget "kød" på vejledningen.

Man vil i denne proces bl.a. støde på følgende væsentlige arbejder, som hver på sin måde forsøger dette:

1. <Aggergren 1984>, Edb og den eksterne revisor
2. <Hougaard 1992>, Revision i PC-miljøer
3. <Klarskov 1994>, Dokumentation af systemer, metoder og procedurer

3.2.1.1 <Aggergren 1984>, EDB og den eksterne revisor

Udvalgs-kriterie → Grundlæggende

Det kan måske nok undre, at der er udvalgt en artikel, som er fra før vejledningens udgivelse. Den er udvalgt dels fordi den i sin behandling af analyse og vurdering af edb-området (afsnit 5 i artiklen) har en opbygning og fremgangsmåde som meget ligner den senere <Rev. vejl. 14> (der er ingen tvivl om at vejledningen er blevet stærkt påvirket af denne artikel) dels fordi artiklen er meget fremadrettet og har forudset en del af nutidens problemstillinger.

Specifik indhold vedrørende tilretninger:

Aggergren har i sin artikel behandlet tilretninger som en selvstændig del af punktet systemudvikling og -vedligeholdelse. Han foreslår, at det ved revisionen af brugersystemerne påses at virksomheden har følgende procedurer ved programændringer:

Når systemer er igangsat skal de løbende justeres og udbygges. Disse systemændringer bør principielt være underkastet samme kontrol som dem, der gennemføres for nye systemer.

Systemændringerne må således vurderes og prioriteres, ligesom ændringer skal detailbeskrives, programmeres og brugertestes, inden det ændrede program tages i anvendelse.

Problemet er så, om man kan anvende dette på de tilrettelige standard systemer. I 1984 var det primært programmører, der lavede de fleste systemændringer. Tilrettelige standard systemer var kun i opstartsfasen. Aggergren var dog klar over at edb-udviklingen ville føre til fremkomsten/udbredelsen af disse systemer og behandler dette i sit afsnit om overvejelser om fremtiden.

Behandlingen kan skematisk kort opstilles således:

<i>Automatisk registrering af systemændringer²⁴</i>	<i>Revision af systemgeneratorer²⁵</i>
<i>Der er tendenser til, at traditionel programmering vil blive fortrængt af "systemgeneratorer", hvor brugere selv gennem brug af særlige "høj-niveausprog" kan foretage ændringer i deres systemer uden hjælp af programmører. Kontrol og overvågning af systemændringer kan i disse systemer etableres ved tvangsmæssig dokumentation/udskrift af de foretagne systemændringer eller oprettelser. Samtidig bør disse systemer på ethvert tidspunkt kunne producere en dokumentation af det øjeblikkeligt kørende brugersystem således, at manuel vedligeholdelse af systemdokumentation vil være begrænset.</i>	<i>For at kunne undersøge systemerne og de indbyggede kontrolforanstaltninger, når klienterne anvender systemgeneratorer, må revisorer være fortrolig med systemgeneratorernes virkemåde. Revisor må tillige på lige fod med brugerne kunne drage nytte af udskrifter, der kan dokumentere brugen af systemet, således at han på egen hånd kan indsamle de nødvendige oplysninger. Der kan være tale om øjeblikbilleder af det bestående system eller udskrifter af alle systemændringer i en periode. (Der henvises til spalte 1)</i> <i>Revisor skal være i stand til at rådgive om såvel anskaffelse af som tilrettelæggelse af brugen af systemgeneratoren.</i>

Kilde: <Aggergren 1984> opsat i skema af os

²⁴ <Aggergren 1984> afsnit 9.2.4 (side 45).

²⁵ <Aggergren 1984> afsnit 9.3.2 (side 45).

Nutidens tilrettelige standard systemer har de samme egenskaber som de foran beskrevne systemgeneratorer, hvorfor forannævnte kan bruges direkte på disse.

Man kan kort opsummere hans krav til revision af tilretninger (systemændringer) i tilrettelige standard systemer således:

1. Påse at systemændringer → vurderes og prioriteres → detailbeskrives → programmeres → brugertestes inden de tages i anvendelse.
2. Påse at systemet "tvangsmæssigt" laver dokumentation af systemændringer.
3. Påse at systemet på ethvert tidspunkt kan lave en dokumentation af det øjeblikkeligt kørende system.
4. Krav til revisors kundskaber.
(Vil ikke blive behandlet særskilt, men ligger som en forudsætning for overhovedet at kunne foretage revision af ændringer).

Specifik indhold vedrørende pc-miljøer:

Vedrørende betydningen af at tilretningerne foretages i pc-miljøer siger han, at systemudvikling og -vedligeholdelse ofte her er stærkt afhængig af enkeltpersoner og at dokumentationen af selvudviklede systemer er mangelfulde.

Derfor foreslår han, at virksomheden benytter en programleverandør (softwarehus) til programmering, idet der i så fald enten benyttes standardsystemer eller kontraktudviklede, skræddersyede systemer.

Det bevirker, at oversætterprogrammet (for nærmere forklaring af dette henvises der til side 39 om compilere) kan fjernes fra maskinerne²⁶ og dermed formindskes risikoen for at der indlægges programmer, der ikke er ordentligt dokumenteret.

Ovennævnte er skrevet ud fra de i 1984 gældende forhold. Han beskriver ikke nærmere under sine overvejelser om fremtiden om det har nogen særskilt betydning, at tilretningen foretages i en systemgenerator som findes i et pc-miljø.

²⁶

<Aggergren 1984> side 25.

3.2.1.2 <Hougaard 1992>, Revision i PC-miljøer

Udvalgsriterie → Praktisk behandling af revision i pc-miljøer.

Bogen bygger på en hovedopgave lavet på cand. merc. aud studiet på Ålborg Universitetscenter. Den omhandler primært revision i virksomheder, der anvender pc'ere.

Bogen forsøger at operationalisere revisionen af edb-systemer i pc-miljøer.

Specifik indhold vedrørende tilretninger:

De behandler ikke tilretninger særligt dybt i deres verbale beskrivelse, men siger dog følgende²⁷:

Ved den mere uformelle systemudvikling, som kendetegner pc-miljøer, bør man altid foretage en vurdering af konsekvenserne af en uopdaget fejl i modeller/programmer. En sådan vurdering vil dog være svær at foretage, da man ikke kender fejls omfang på forhånd.

De beskriver herefter selve udviklingsfasen således:

- ☛ *Fastlæggelse af opgaven og planlægning.*
- ☛ *Udvikling/programmering, løbende test samt udarbejdelse af dokumentation.*
- ☛ *Indkøring samt sluttest.*

Det kan ved mindre opgaver være hensigtsmæssig, at punkterne udføres mere eller mindre under et, men det skal bemærkes, at der principielt bør stilles de samme krav til udviklingsfasen i et pc-miljø som i større edb-miljøer.

De har ved deres beskrivelse lagt op til en skelnen mellem egenudviklede systemer og købte systemer. De siger med hensyn til de købte systemer:

De købte systemer bør også påkalde sig speciel opmærksomhed, idet der i de seneste versioner er mulighed for, at brugeren selv kan foretage visse modifikationer. Det

²⁷

<Hougaard 1992> side 50.

skal dog bemærkes, at denne adgang kan og bør begrænses ved passwords i programmerne. Endvidere kan leverandøren foretage ændringer i de grundlæggende dele af programmet. Det er ved sådanne ændringer vigtigt, at der foreligger dokumentation for ændringerne til brug for evt. senere ændringer, samt naturligvis revisors gennemgang af systemet.

De siger endvidere at:

revisors gennemgang af systemudvikling og vedligeholdelse i pc-miljøer normalt ikke vil være et væsentligt punkt. Det skyldes, at der sjældent foregår systemudvikling af revisionsmæssige væsentlige programmer i pc-miljøer.

Andet steds i bogen siger de dog:

at idet de fleste standardprogrammer i dag kan tilrettes eller endog omprogrammeres, så får de karakter af specialprogrammer. F.eks. kan nævnes, at IBM's forhandlere af økonomiprogrammet Navigator 3.0 efter en uddannelse kan få certifikat til at rette i programkoden. Disse rettelser skal dog være fuldt dokumenteret og foreligge hos forhandleren. Det er således vigtigt, at revisor ved sådanne standardprogrampakker undersøger, om der er foretaget rettelser, og hvilken indflydelse disse får på revisionen. Omvendt kan det siges, at den totale program-sikkerhed er større i et tilrettet standardsystem end i en egentlig specialprogrammeret applikation, idet programudvikleren vil have brugt flere ressourcer på sikkerhedsbeskyttelse.

De giver i deres appendix A²⁸ forslag til revisionshandlinger ved edb-revision i pc-miljøer. De opstillede skemaer følger hovedpunkterne i <Rev. vejl. 14 og 17> og er ifølge forfatterne meget omfattende og detaljerede, hvilket har det formål, at de skal kunne tilpasses til såvel store som små pc-installationer. Spørgsmålene i skemaerne skal bl.a. bevirke, at alle væsentlige områder bliver vurderet og at revisor opnår et kvalificeret vurderingsgrundlag.

Skemaerne er især udarbejdet for at give generalistrevisoren en bedre indfaldsvinkel til edb-revision. Nedenstående skema viser deres krav til revisionen af systemudvikling og -vedligeholdelse i pc-miljøer:

Systemudvikling og -vedligeholdelse (ref. afsnit 4.3.3)			
	JA	NEJ	BILAG
Anvender virksomheden udelukkende standardsystemer?			
Hvis standardsystemer:			
- Er systemerne modificerede/tilrettede?			
Hvis specialsystemer:			
- Er systemerne programmeret af eksterne programmører?			
- Bruges der selvprogrammerede programmer?			
- Hvem udarbejder disse?			
- Findes der planlægning/test/dokumentation for disse?			
- Hvordan testes/godkendes nye systemer inden idriftsættelse?			
- Hvordan testes/godkendes systemændringer inden idriftsættelse?			
Anvendes regneark i virksomheden?			
Hvis ja, til hvad?			

Kilde: <Hougaard 1992> side 156

3.2.1.3 <Klarskov 1994>, Dokumentation af systemer, metoder og procedurer

Udvalgs-kriterie → Nyeste forsøg på at operationalisere området "dokumentation af systemer, metoder og procedurer" i <Rev. vejl. 14> punkt 3.4.

Den er udvalgt på grund af, at den er det nyeste forsøg fra de store faglige organer (FSR, FIK og IIA) på at operationalisere området dokumentation af systemer, metoder og procedurer.

Beskrivelsen tager sigte mod en større virksomhed med egenudvikling af edb-systemer og egen edb-drift. For virksomheder, der ikke foretager udvikling af edb-systemer, er kravene til dokumentation af procedurer og metoder ikke relevante. Der bør dog være etableret retningslinier for, i hvilket omfang købte systemer må tilpasses, og hvordan ændringer skal dokumenteres.

Det er derfor søgt, at pille de dele ud af beskrivelsen som direkte vedrører købte systemer og/eller mindre pc-miljøer.

Specifik indhold vedrørende tilretninger:

Virksomheder, der anvender købte færdigudviklede brugersystemer, vil ofte ikke have adgang til programmernes kildekode. Denne del af systemdokumentationen vil derfor ikke findes i virksomheden. Dette bør ikke give vanskeligheder for virksomheden, så længe leverandøren kan forestå vedligeholdelsen af systemet. Hvis systemet er udviklet specielt til virksomheden, bør det anbefales, at kildekoden deponeres, så virksomheden kan få adgang til koden i den situation, hvor leverandøren ikke kan varetage vedligeholdelsesopgaven.

Der bruges cirka 8 sider på beskrivelse af systemudviklingen, mens systemvedligeholdelse fylder 11 linier. Man må være opmærksom på, at tilretninger både kan være systemudvikling og systemvedligeholdelse, hvorfor det er vigtigt at se på begge dele.

<Klarskov 1994> opstiller følgende systemudviklingsmodel:

- *Planlægning og styring af projekt og kvalitet*
- *Analyse*
- *Design*
- *Programmering*
- *Test*
- *Accepttest*
- *Implementering*
- *Afslutningsreview*

Hvordan systemet skal revideres behandles ikke nærmere.

Vedrørende systemvedligeholdelse nævnes følgende:

Det er vigtigt, at den udarbejdede dokumentation føres ajour i forbindelse med fejlretning. Procedurene bør foreskrive, at der gennemføres kvalitetssikring på den ajourførte dokumentation umiddelbart i forbindelse med, at systemændringen sættes i produktion. Hvis der ikke er beskrevne procedurer for dette område, er der en stor risiko for, at den foreliggende systemdokumentation ikke svarer til realia.

For at undersøge om det beskrevne kriteriesæt tager højde for edb-udviklingen (herunder tilretteliggende standardsystemer), vil vi teste det med nogle cases fra praksis. Casene er søgt udvalgt, så de dækker forskellige problemstillinger. Antallet af cases gør, at der ikke opnås statistisk sikkerhed for konklusionerne, men det er vor opfattelse, at casene kan hjælpe til forståelse af hvorledes kriteriesættet kan anvendes.

3.2.2 Cases

Casene er anonymiserede og stammer fra de administrative systemer Navigator og Concorde XAL. De er gengivet i uforkortet form i bilagene. Her medtages kun et kort referat.

3.2.2.1 Case A: Manglende generelle kontroller

Referat:

I et firma hvor der anvendes automatisk datafangst for medarbejdertimer via strekcodeindlæsning, er den PC'er der anvendes som stempelur oprettet med maksimale rettigheder. Denne PC'er er den eneste der er fysisk tilgængelig for alle firmaets ansatte uden undtagelse. De maksimale rettigheder (superbrugerstatus i Navigator og supervisorstatus i Novell) bevirker, at alle medarbejdere har adgang til alle oplysninger og funktioner (herunder ændring i systemerne) i firmaet.

Problemstilling i relation til ændringer:

Alle brugeres mulighed for at ændre systemerne forhindrer effektiv kontrol med ændringer (de administrative medarbejdere loggede ind på Navigator med brugerID STREGKODE når de ønskede at ændre skærbilleder).

Test af kriteriesæt:

<Rev. vejl. 14> punkt 3.6 siger:

Revisor skal undersøge, om der er etableret betryggende funktionsadskillelse i og omkring edb-funktionerne, herunder mellem systemudvikling og drift. Revisor skal vurdere, om der er etableret fornødne adgangskontroller.

Ifølge <Rev. vejl 14> punkt 3.5 udføres revisors gennemgang hovedsageligt ved interviews, observation og efterprøvelse af indsamlet information.

<Rev. vejl. 14> dækker problemstillingen. Revisor må dog have et detaljeret kendskab til det anvendte system for at kunne efterprøve den indsamlede information.

Som nævnt på side 25 tager <Aggergren 1984> mere konkret højde for problemstillingen. Han anbefaler at oversætterprogram (compiler) fjernes fra maskinerne²⁹. Hvis dette råd skal følges i et tilretteligt standardssystem betyder det, at firmaet ikke skal købe de moduler der giver mulighed for at tilrette systemet. Tilretninger foretages så af softwareleverandør.

Herved mister brugeren en del fleksibilitet, men opnår til gengæld funktionsadskillelsen mellem udvikling og drift.

3.2.2.2 Case B: Konstater ændringer

Referat:

Firmaet oplyste at der ikke var foretaget ændringer i systemet i regnskabsåret. En efterprøvelse viste, at denne oplysning ikke var korrekt. Der var ændret i systemet, men firmaet opfattede det ikke som en ændring. Casen giver eksempler på hvorledes, revisor kan vælge at undersøge systemerne for ændringer.

Problemstilling i relation til ændringer:

Hvorledes konstaterer revisor, om systemet har fungeret uændret i hele den periode der revideres? I hvilket omfang skal de indhentede oplysninger efterprøves?

Test af kriteriesæt:

<Rev. vejl. 14> punkt 3.4 gør, at revisors gennemgang skal omfatte en vurdering af systemudvikling og -vedligeholdelse.

Denne vurdering baseres jf. <Rev. vejl. 14> punkt 3.5 på interviews, observation og efterprøvelse.

De 2 punkter vedrører problemstillingen, men hvis revisor alene fastlægger sine revisionshandlinger ud fra disse, er det tvivlsomt om de handlinger, der effektivt fanger ændringerne vil blive gennemført. Ved denne problemstilling er det et alvorligt problem at <Rev. vejl. 14> er så generelt formuleret.

²⁹

<Torp 1994> side 14 anbefaler også, at compilere fjernes fra systemet (eller at der foretages adgangsbegrænsning til compiler).

<Hougaard 1992> side 51 anfører, at adgang til compiler kan og bør begrænses ved passwords i programmerne.

<Aggergren 1984> er mere konkret (jf. side 24 i dette seminar), idet han kræver tvangsmæssig dokumentation/udskrift af systemændringer. Metoden er korrekt, men hverken Concorde XAL eller Navigator muliggør umiddelbart anvendelsen af metoden.

<Hougaard 1992> side 77 anviser en metode til konstatering af om et program er ændret. Eksemplet baserer sig på Concorde Økonomi³⁰. Den beskrevne metode går ud på, at bruge DOS-kommandoen DIR for at se en oversigt over filer (herunder fildatoer) i et givet katalog.

I Hougaard's eksempel angives MULTI.COM som eksekverbar fil, men filerne MULTI.000, MULTI.001 o.s.v. indeholder også eksekverbar kode. Det er således ikke som anført kun filtyperne EXE og COM, der skal undersøges, hvis man vil undersøge om et program er blevet rettet eller opdateret. I nutidens mere databasebaserede systemer er metoden ikke anvendelig. (En beskrivelse af forudsætningerne for Hougaard's metode gives i vort afsnit 3.3.1).

Det må konstateres, at kriteriesættet ikke giver brugbar vejledning om hvordan ændringer i de tilrettelige standardssystemer konstateres.

De i casen beskrevne muligheder til konstatering af ændringer (beskrevet nærmere i afsnit 3.3.2) er bedre, selv om det ikke er muligt at se, om der er foretaget ændringer tidligere³¹.

Casen viser, at af de 3 muligheder i <Rev. vejl. 14> punkt 3.5 bør efterprøvelse ske når dette er muligt under iagttagelse af væsentlighed og risiko.

3.2.2.3 Case C: Tilrettet faktureringsrutine

Referat:

En efter firmaets opfattelse ubetydelig ændring, der kun skulle give det fakturaudseende man ønskede, satte en væsentlig programmeret kontrol ud af kraft. Konsekvensen heraf blev, at der ikke var kontrol med om udskrevne og bogførte fakturaer stemte overens. Det bevirkede, at der på grund af indtastningsfejl blev bogført fakturaer med beløb, der var forskellige fra beløbene på de udskrevne fakturaer.

Problemstilling i relation til ændringer:

³⁰ Concorde Økonomi er forgænger for Concorde XAL. Systemet er mere traditionelt opbygget end Concorde XAL og Navigator.

³¹ Metoderne giver et øjebliksbillede over hvilke dele, der er ændret i forhold til standarden.

Den ændrede rutine til fakturabogføring og -udskrivning var ikke tilstrækkelig testet. En sådan mindre rettelse testes ofte med et eller 2 sæt realistiske data. Der var tale om manglende kontrol. Systemet fungerede upåklageligt så længe brugeren indtastede data der stemte (hvor kontrollen ikke skulle bemærke noget).

Test af kriteriesæt:

<Rev. vejl. 14> punkterne 3.4 og 3.9 dækker så bredt, at denne problemstilling kan anses for omfattet.

Mere uddybende behandles problemstillingen af:

- ☛ <Aggergren 1984> jf. side 24 i dette seminar
- ☛ <Hougaard 1992> jf. side 26 i dette seminar³²
- ☛ <Klarskov 1994> jf. side 29 i dette seminar³³

<Rev. vejl. 17> punkt 2.4 dækker problemstillingen fra en anden indgangsvinkel. Såfremt ændringen i faktureringsrutinen var registreret som ændring i udseende af udskrift ville den næppe påkalde sig større opmærksomhed fra revisor³⁴. Først ved revisors gennemgang af brugersystemer jf. <Rev. vejl. 17> afsnit 2 vil den manglende programmerede kontrol blive opdaget. Dette vil dog kun ske hvis applikationen er fundet væsentlig og/eller risikofyldt.

Det må konstateres, at kriteriesættet behandler problemstillingen, men at revisors vurderinger af væsentlighed og risiko i forbindelse med ændringer er afgørende for om fejlen opdages (fejlen burde opdages i forbindelse med opfølgning på debitorer da det bogførte beløb fremgår af kontoudtogene).

3.2.2.4 Case D: Tilføjet rutine til intern fakturering i profitcenter

Referat:

³² <Hougaard 1992> erkender, at den mere uformelle systemudvikling kan være et specielt problem, men gengiver kravene til mere traditionel systemudvikling. Egentlig behandling af det specielle problem ses ikke.

³³ <Klarskov 1994> bruger som tidligere nævnt kun 11 linier på systemvedligeholdelse og cirka 8 sider på systemudvikling. Afsnittet om systemvedligeholdelse er drejet i retning af fejlrettelse og systemudviklingsafsnittet er mere traditionelt. Dette skyldes formentlig, at hans artikel tager sigte mod større virksomheder med egenudvikling fra bunden. Han nævner dog, at der bør være retningslinier for i hvilket omfang købte systemer må tilpasses og hvordan ændringer skal dokumenteres.

³⁴ En situation hvor de generelle edb-kontroller har fungeret og ændringen er identificeret.

I et firma med profitcentre blev internt salg genereret af en rutine i edb-systemet. Rutinen debiterede størsteparten af det interne salg i den sælgende afdeling således at eneste effekt af internt salg var opblæsning af omsætning og omkostninger i den sælgende afdeling. Afdelingslederen i den sælgende afdeling overbeviste direktionen om, at omkostningen var en fejlpostering og ikke hørte til hos ham. Ingen tog initiativ til at finde ud af, om der var en kreditpost et sted der skulle reduceres eller omkostningen manglede et andet sted.

Problemstilling i relation til ændringer:

En af edb-leverandøren tilføjet rutine blev ikke underkastet ordentlig aftenstning.

(Når konsekvensen af den manglende test blev som beskrevet, skyldes det, at faresignaler ikke blev håndteret fornuftigt)

Test af kriteriesæt:

Casen viser nødvendigheden af at teste tilføjede rutiner i systemerne. Kriteriesættet for test af tilføjede rutiner er det samme som for test af ændrede rutiner hvorfor der henvises til case C.

3.2.2.5 Case E: Lagerliste anvendte forkerte data fra systemet

Referat:

Firmaet var ikke tilfreds med de 2 lagerlister, der fandtes som standard i edb-systemet. Man definerede derfor sin egen liste med udgangspunkt i den ene af standardlisterne. Listen fik tilføjet et felt fra varekartoteket og dette felt blev ganget med et felt der allerede var på listen. Resultatet af denne udregning blev udskrevet og opsummeret. Firmaet antog, at det stemte med totalen på den anden standardliste. Der var dog væsentlige forskelle på de varenumre, hvor der var "svømmende varer". Det tilretteliggende standardsystem's håndtering af "svømmende varer" blev sat ud af kraft ved tilretningen.

Problemstilling i relation til ændringer:

Den person der foretog ændringen i listen forstod ikke alle sammenhænge i systemet. Der blev ikke skelnet mellem to næsten ens felter i varekartoteket:

- ☛ Antal på lager (fysisk tilstede)
- ☛ Antal firmaet ejer (fysisk tilstede korrigeret for svømmende varer)

Fejlen er typisk for ændringer foretaget af brugere, der ikke kender alle systemsammenhænge.

Man kan derfor spørge: Har det betydning om tilretningen er foretaget af edb-leverandør eller bruger?

Test af kriteriesæt:

<Rev. vejl. 14> giver ikke anledning til at skelne mellem tilretninger foretaget af edb-leverandør eller tilretninger foretaget af bruger³⁵.

Heller ikke <Aggergren 1984>, <Hougaard 1992> og <Klarskov 1994> lægger op til skelnen mellem om ændringen er foretaget af edb-leverandør eller brugeren.

Det må ud fra casene konstateres, at en sådan opdeling ikke ville være relevant, da der er fejl i både edb-leverandørernes og brugernes tilretninger.

Kriteriesættet må derfor betegnes som dækkende³⁶.

³⁵ Funktionsadskillelsen skal dog fortsat være opfyldt ved brugertilretning.

³⁶ Såfremt der gennemføres en empirisk analyse af edb-leverandørers og brugeres tilretninger, er det ikke sikkert, at denne konstatering holder.

3.2.2.6 Case F: Udbetaling af for meget overtidstillæg

Referat:

En rapport udviklet til opgørelse af normal- og overtimer pr. medarbejder pr. dag beregnede for mange timer til medarbejdere, der mødte efter normal arbejdstids begyndelse og gik hjem efter normal arbejdstids ophør. Der blev som konsekvens heraf udbetalt for meget i løn.

Problemstilling i relation til ændringer:

Den person der havde udviklet rapporten havde ikke taget højde for de mulige kombinationer af møde- og gåhjemtidspunkter.

Har den der programmerer tilstrækkelig viden til at overskue de betingelser, der skal være opfyldt for, at rapporten kommer til det ønskede resultat ?

Test af kriteriesæt:

Det må konstateres, at kriteriesættet ikke lægger op til egentlig vurdering af programmørens evner.

Revisor bør efter vor mening foretage en overordnet vurdering af de personer der tilretter firmaets systemer. En egentlig "kvalitetscertificering" af de pågældende personer kunne måske være løsningen.

3.2.3 Vurdering af kriteriesættet

For at give overblik over hvilke af problemstillingerne i casene der fanges af kriteriesættet, vises her en skematisk opstilling over problemstillingerne med henvisning til de relevante dele af kriteriesættet:

Case	Dækker kriteriesættet	Henvisning til litteratur ----- Kommentering af litteratur
A	Ja	<Rev. vejl. 14> punkt 3.6+3.5, <Aggergren 1984> side 25, <Hougaard 1992> side 51 og <Torp 1994> side 14 ----- <Rev. vejl. 14>: Klart dækkende. Både omfang og metode er anvist.
B	(Ja)	<Rev. vejl 14> punkt 3.4 og 3.5, <Aggergren 1984> og <Hougaard 1992> side 77 ----- <Rev. vejl. 14>: For overordnet formuleret. (Uddybning søges i anden litteratur) <Aggergren 1984> + <Hougaard 1992> anviser konkrete metoder. Disse er ikke anvendelige i de p.t. mest anvendte systemer.
C	(Ja)	<Rev. vejl. 14> punkt 3.4 og 3.9, <Aggergren 1984>, <Hougaard 1992> og <Klarskov 1994> ----- <Rev. vejl. 14>: For overordnet formuleret. (Uddybning søges i anden litteratur) <Aggergren 1984> anfører, at ændringer skal testes som nyudvikling. <Hougaard 1992> erkender, at der kan være specielle problemer ved den mere uformelle udvikling der kendetegner PC-miljøer. Der henvises dog til traditionel udviklingsmetodik.
D	(Ja)	<Rev. vejl. 14> punkt 3.4 og 3.9, <Aggergren 1984>, <Hougaard 1992> og <Klarskov 1994> ----- <Rev. vejl. 14>: For overordnet formuleret. (Uddybning søges i anden litteratur) <Aggergren 1984> anfører, at ændringer skal testes som nyudvikling. <Hougaard 1992> erkender, at der kan være specielle problemer ved den mere uformelle udvikling der kendetegner PC-miljøer. Der henvises dog til traditionel udviklingsmetodik.
E	Ja	<Rev. vejl. 14>, <Aggergren 1984>, <Hougaard 1992> og <Klarskov 1994> ----- <Rev. vejl. 14>: Området kun indirekte behandlet. (Uddybning søges i anden litteratur) <Aggergren 1984> + <Hougaard 1992> + <Klarskov 1994> behandler heller ikke området direkte.
F	Nej	----- Der er i litteraturen hverken direkte eller indirekte taget stilling til spørgsmålet.

(Ja) betyder, at litteraturen på grund af meget generelle formuleringer må siges at omfatte problemstillingen. De generelle formuleringer gør dog, at der ikke kan arbejdes ud fra denne.

Når kriteriesættet i case E anses for dækkende skyldes det, at vi ikke på grundlag af casene kan se nogen grund til at skelne mellem tilretninger foretaget af edb-leverandør eller bruger.

Generelt dækker kriteriesættet rimeligt, når man supplerer med den uddybende litteratur. Der er dog mangler. En væsentlig mangel set i forhold til tilretteliggende standardssystemer er, at der ikke gives konkrete anvisninger om hvorledes revisor undersøger om et system kan ændres. Dette kan revisor kun klarlægge ved at undersøge hvilken "systemtype" der er tale om.

3.3 Hvorledes undersøger revisor hvilken gruppe systemet hører til og om systemet har fungeret uændret i hele perioden ?

<Tange 1984> side 92 opdeler edb-systemer i 3 kategorier:

- ☛ Individuelle systemer³⁷
- ☛ Standardsystemer³⁸
- ☛ Rammesystemer³⁹

Det er afgørende for revisionsstrategien, at det indledningsvis fastslås hvilken kategori systemet hører til og specielt hvilke muligheder, der er for tilretning.

3.3.1 Individuelle systemer og standardsystemer

Typisk vil både individuelle systemer og standardsystemer på PC'ere være skrevet i et traditionelt⁴⁰ programmeringssprog.

De traditionelle programmeringssprog inddeles i 2 undergrupper:

- ☛ Compilersprog (oversættere)
- ☛ Interpretersprog (fortolkere)

Programmer udviklet i disse 2 grupper af sprog er væsentligt forskellige, når der henses til brugerens mulighed for at rette i programmerne.

³⁷ Eget udviklede eller specialudviklede systemer. Dette kendes i dag stort set ikke på PC'ere.

³⁸ Et eksempel herpå er PC-PLUS, hvor eneste tilretningsmulighed er placering af felterne på de udskrifter der typisk sendes til kunder.

³⁹ Systemer hvor alle basisfunktioner findes. Der kan tilrettes i basisfunktionerne og nye funktioner kan tilføjes. Eksempler på rammesystemer til PC'ere er Concorde XAL og Navigator.

⁴⁰ 3. generationssprog.

Compilersprog

Et program udviklet i et compilersprog består af en kildekode⁴¹ og en eksekverbar⁴² fil. Et eksempel på en kildekode⁴³ kan se sådan ud:

```
program Mængderabat;
var
  AntalEnheder, EnhPris, Total, RabatSats: real;
begin
  write('Indtast antal enheder :');
  readln(AntalEnheder);
  write('Indtast pris pr. enhed :');
  readln(EnhPris);
  Total := AntalEnheder * EnhPris;
  If (AntalEnheder > 10) and (Total > 100) then
    RabatSats := 5
  else
    RabatSats := 0;
  Writeln('Rabatprocenten udgør   : ',RabatSats:8:2);
  Writeln('Rabatten udgør       : ',RabatSats/100*Total:8:2);
end.
```

Denne kildekode, der er en ren tekstfil, kaldes i dette eksempel RABAT.PAS. Den danner grundlag for udfærdigelsen af programmet RABAT.EXE (den eksekverbare fil). Dette sker via en compiler⁴⁴ der læser RABAT.PAS og skriver RABAT.EXE⁴⁵. Når compileren har været anvendt, har programmøren to filer:

RABAT.EXE (eksekverbar fil)

RABAT.PAS (kildekode)

Det er nok at have RABAT.EXE for at kunne køre programmet og typisk leveres kildekoden ikke med til brugeren. Dette illustreres i <Hougaard 1992> side 77 hvor brugeren har MULTI.COM til rådighed, men ikke MULTI.PAS.

⁴¹ Den af programmøren skrevne tekstfil.

⁴² Den fil der anvendes ved afvikling af programmet.

⁴³ Eksemplet er skrevet i PASCAL.

⁴⁴ Oversætter til maskinkode

⁴⁵ Det er afhængigt af valg af compiler om filtypen bliver COM eller EXE. Begge typer er eksekverbare. EXE er en nyere type end COM.

Det er ikke muligt at lave meningsfyldte rettelser til en EXE eller COM fil, da strukturen i disse er alt for kompleks. Hvis brugeren vil have rettet i RABAT.EXE så er metoden:

- ✎ ret i kildeteksten RABAT.PAS (der så får en ny dato)
- ✎ dan med compileren en ny eksekverbar fil RABAT.EXE (der så får en ny dato)

Brugeren skal derfor have både kildeteksten og compileren til rådighed for at kunne rette i programmet.

Interpretersprog

Hvis programmet er skrevet i et interpretersprog⁴⁶ findes der kun én fil. I dette eksempel kaldes filen RABAT.CML⁴⁷:

```
0100 REM program til beregning af mængderabat
0110 DIM AntalEnheder, EnhPris, Total, RabatSats
0120 INPUT "Indtast antal enheder  ":" AntalEnheder
0130 INPUT "Indtast pris pr. enhed ":" EnhPris
0140 Total := AntalEnheder * EnhPris
0150 IF (AntalEnheder > 10) AND (Total > 100) THEN
0160   RabatSats := 5
0170 ELSE
0180   RabatSats := 0
0190 PRINT "Rabatprocenten udgør   : ",RabatSats
0200 PRINT "Rabatten udgør         : ",RabatSats/100*Total
```

Brugeren skal have RABAT.CML til sin rådighed for at kunne afvikle programmet.

Afviklingen sker via interpreteren (som brugeren også skal have). Interpreteren hedder i dette eksempel COMAL.EXE. Når brugeren kører programmet startes det således:

COMAL RABAT

Dette bevirker, at COMAL.EXE starter op og indlæser RABAT.CML. RABAT.CML fortolkes af COMAL.EXE under afviklingen. Det afgørende for revisor er, at brugeren pr. definition har mulighed for at rette i programmet, men dette kan ses på datoen på filen RABAT.CML.

⁴⁶ Fortolkersprog. Der dannes ikke en eksekverbar fil, men programmet oversættes til maskinkode linie for linie efterhånden som det afvikles.

⁴⁷ Eksemplet er skrevet i COMAL80 (Der findes i dag compilere der kan danne eksekverbare filer ud fra COMAL 80 programmer, men dette er ikke det normale).

Sammenligning compiler/interpreter

De væsentlige forskelle for revisor mellem compiler- og interpretersprog kan skitseres således⁴⁸:

	Compilersprog	Interpretersprog
Filtyper der opdateres ved ændringer (dette kan ses på fildatoer såfremt brugeren ikke har manipuleret med PC'ens indbyggede ur eller fildatoer)	PAS (ved pascalcompiler) COM (og 000, 001 m.v.) EXE (og OVR, OVL og DLL)	CML (ved comalinterpreter)
Nødvendige filer for afvikling	COM (og 000, 001 m.v.) EXE (og OVR, OVL og DLL)	CML (ved comalinterpreter) COMAL.EXE (ved comalinterpreter)
Nødvendige filer for ændring	PAS (ved pascalcompiler) PASCAL.EXE (compileren)	CML (ved comalinterpreter) COMAL.EXE (ved comalinterpreter)

Kilde: Egen tilvirkning

Den væsentligste forskel for revisor er, at brugeren af et program skrevet i et interpretersprog har alle de filer til rådighed, der er nødvendige for at ændre programmet. Funktionsadskillelse kan derfor ikke opnås.

Da der jf. skemaet ikke er sammenfald mellem de nødvendige filer til afvikling og ændring ved compilersprog er den ønskede funktionsadskillelse mulig⁴⁹.

Flertallet af standardsystemer til PC'ere⁵⁰ er skrevet i compilersprog. Det kan hænge sammen med, at programmørerne anser kildekoden for en forretningshemmelighed⁵¹.

En stor del af de individuelle systemer til PC'ere er skrevet i interpretersprog.

⁴⁸ Skemaet kan læses uden hensyntagen til det i parentes anførte. Det i parenteserne anførte er kun forudsætninger eller alternative filnavne.

⁴⁹ <Aggergren 1984> anbefaler, at compiler fjernes og <Torp 1994> anbefaler, at compiler fjernes eller der sker adgangsbegrænsning hertil. <Watne 1990> side 499 anbefaler, at kildekoden fjernes (det er mere sikkert da compiler kan erhverves nemmere end den konkrete kildekode).

⁵⁰ Eksempelvis Concorde Økonomi, PC-PLUS, DCC Admiral (DOS-udgaven), Kompass samt FK-2001.

⁵¹ <Klarskov 1994> side 8 spalte 1.

Metoder til konstatering af ændringer

<Hougaard 1992> side 77 anviser en metode til konstatering af, om et program er ændret. Eksemplet baserer sig på Concorde Økonomi⁵². Den beskrevne metode går ud på, at bruge kommandoen DIR for at se en oversigt over filerne i et givet katalog. Hvis fildato, klokkeslæt eller størrelse på eksekverbare filer er ændret i forhold til sidste revision, er programmet ændret.

```

DIR C:\CONCORDE >PRN2
-----
Volume in drive C is DISK C
Directory of C:\CONCORDE
.          <DIR>          1-10-91    8.00
..         <DIR>          1-10-91    8.00
MULTI     000     361472   20-05-88   17.05
MULTI     001     25344    20-05-88   17.05
MULTI     002     72448    20-05-88   17.05
MULTI     003     37376    20-05-88   17.05
MULTI     004     13056    20-05-88   17.09
MULTI     005     33536    20-05-88   17.10
MULTI     006    101120    20-05-88   17.13
MULTI     007     28672    20-05-88   17.13
MULTI     008     19712    20-05-88   17.14
MULTI     009     15616    20-05-88   17.15
MULTI     010     27904    20-05-88   17.16
MULTI     COM     39796    20-05-88   17.22
BRUGER    MCO      8576     16-05-92   13.17
SYSOPLYS  MDA     73600    16-05-92   13.17
-----
16 File(s)      2095104 bytes free
-----
Den indtastede kommando.
Disknavn.
Directorynavn.
Dato for oprettelse af directory.
-----
Eksekverbar fil
-----
Ledig diskplads
-----
Filnavn (max 8 kar.)
Filtype (max 3 kar.)
Filstørrelse
Dato seneste ændring
Klokkeslæt - -.

```

<Hougaard 1992> gør opmærksom på:

at dato og klokkeslæt kan ændres med et utilityprogram, men at programstørrelsen kun kan manipuleres af en særdeles programmørkyndig person. Ligeledes skal revisor være opmærksom på, at filer kan være mærket HIDDEN (skjulte), og derfor ikke vil fremgå i oversigten. Disse kan dog ses med utilityprogram eller med kommandoen DIR /a:h i DOS 5.0.

<Hougaard 1992> anfører, at filen MULTI.COM er den eksekverbare fil i eksemplet. Dette er fejlagtigt, idet filerne MULTI.000, MULTI.001 o.s.v. også indeholder eksekverbar kode. MULTI.COM er groft sagt menuen i Concorde, og de enkelte delfunktioner ligger i de øvrige eksekverbare filer. Der kan således være foretaget programændringer uden at dato, klokkeslæt og størrelse for MULTI.COM er ændret.

Metoden er brugbar både for programmer skrevet i compiler- og interpretersprog, men stiller store krav til revisors edb-viden og kendskab til det konkrete program (skal kunne identificere alle filer der indeholder programkode).

⁵²

Concorde Økonomi er forgænger for Concorde XAL. Systemet er traditionelt opbygget og skrevet i et compilersprog.

Vi kan ikke foreslå alternativer til metoden og må derfor konstatere, at revisor må tilegne sig den nødvendige viden eller gøre brug af personer med denne viden⁵³.

3.3.2 Rammesystemer

Nutidens mest udbredte rammesystemer er Concorde XAL og Navigator. Disse systemer er ikke skrevet i compiler- eller interpretersprog, men i 4. generationsprog der kan beskrives som en slags blanding af disse. Systemerne består af 2 dele:

- ☛ Et 4. generationsprog
- ☛ Et program som er skrevet i 4. generationsproget

Systemerne indeholder en egen interpreter/compiler og opbevarer både den compilede udgave⁵⁴ og kildekoden⁵⁵ internt i sig.

I Navigator indeholder filen NAVI.EXE samt nogle DLL-filer⁵⁶ både compiler, versionskontrollsystem⁵⁷ samt funktioner til fejlsøgning i kildekode. Compiler og store dele af versionskontrollsystemet har lighedspunkter med compilersprog, dog med den væsentlige undtagelse at Navigators versionskontrollsystem gør, at kildekode og eksekverbar kode altid følges ad⁵⁸. Hermed bliver funktionsadskillelse ved fjernelse af kildekode umulig (interpreterkendetegn).

⁵³ Under hvilke betingelser revisor kan gøre brug af eksterne specialister fremgår af IFAC (ISA) vejledning nr. 18.

⁵⁴ Svarende til COM og EXE filer i compilersprog.

⁵⁵ Svarende til PAS filen ved compilersproget.

⁵⁶ Eftersom disse filer ikke ændres ved tilretninger er det mindre væsentligt at identificere disse.

⁵⁷ Versionskontrollsystemet sikrer, at kildekoden der hører til den compilede udgave af et objekt altid findes i databasen. Problemer med at finde kildekoden til den aktuelle version af programmet kendes således ikke. Versionskontrollsystemet har mange lighedspunkter med "Librarian package" beskrevet i <Watne 1990> side 214 i relation til compilersprog.

⁵⁸ Når sammenhæng kildekode og eksekverbar kode er sikret, har revisor ikke problemer med at undersøge om den version af programmet der anvendes er baseret på den kildekode revisor får udleveret til test. Problemstillingen er beskrevet af <Watne 1990> således: *The user can change the source code, recompile it, and use the modified objectcode for processing. The auditor, however, may be provided with an unmodified version for testing and be unaware of the changes in the production program.*

Alle tilretninger gemmes i én fil (normalt kaldet DATABASE) på harddisken. Denne fil indeholder også data og opdateres derfor både ved almindelig brug samt ved tilretning. Metoden med fildatoer er derfor ubrugelig til detektering af ændringer, da fildatoen opdateres hver gang brugeren eksempelvis bogfører, opretter/retter debitor eller retter i et objekt (tilretter systemet).

Der er nødvendigt at gå ind og analysere indholdet af databasefilen for at finde ud af, om der er foretaget ændringer i objekter og om disse ændringer har betydning for revisor. Ved denne gennemgang af databasen er det indbyggede versionskontrollsystem en stor hjælp.

Undersøgelsen foretages ved en gennemgang af objektoversigten⁵⁹ (funktion i versionskontrollsystemet) som i Navigator⁶⁰ kan se således ud:

Type	Nr	Navn	Dato	Tid	Ænd	VersionsListe
Kartotek	27	Vare	04-02-95	11.16	Ja	PCC5,BB
Efterbeh.kode	37	SalgsLinie	04-12-94	8.26	Ja	PCC7,AA
Vindue	30	Varekartotek	04-02-95	11.36	Ja	PCC1,BB
Rapport		Lager Beh.med pris	14-12-94	23.44	Ja	PCC1,CC
Printer		HP LaserJet 4	14-12-94	22.25	Ja	CC

Det fremgår af eksemplet, at der blandt andet er ændret i efterbehandlingskoden vedr. salgsliniekartoteket. Den seneste ændring i efterbehandlingskoden er foretaget 4. december 1994 klokken 8.26. Revisor skal være opmærksom på, at objektoversigten ikke indeholder oplysninger om tidligere ændringer. Det kan være et problem, hvis eksempelvis følgende er sket:

- ☛ standardrutinen anvendt indtil 20. november 1994 hvor ændring med fejl i rabatberegningen er sat i drift
- ☛ fejlen rettet den 4. december 1994

Det betyder, at der er fejl i rabatsatsen på de fakturaer der er udskrevet i perioden 20. november til 4. december.

⁵⁹ Oversigten er i dette tilfælde valgt således, at kun ændrede objekter vises (Ændret:Ja).

⁶⁰ Case B indeholder et tilsvarende eksempel fra Concorde XAL.

Det er ikke muligt via versionskontrollsystemet at konstatere, at der har været rettet tidligere end den 4. december. En manuel log omfattende alle ændringer er nødvendig for at opdage dette⁶¹. Versionskontrollsystemet har således kun 5 af de 6 kendetegn som <Watne 1992> side 214 nævner for "Librarian packages" idet "Maintains a program change log" ikke er med.

Objektoversigtens inddeling i typer giver revisor mulighed for at foretage en grov vurdering af ændringers betydning. Revisor skal kende systemet indgående for at kunne foretage denne vurdering ud fra typebetegnelserne.

I eksemplet kan dette illustreres ved:

Kartotek 27 Vare

Kartotek indeholder oplysninger om felters navne, typer, længder, tilladte tegn og relationer til andre kartoteker.

Ændringen i kartoteket Vare betyder ikke, at der er ændret i de allerede oprettede varer (dataene), men derimod at der er ændret i kartoteksdefinitionen (eksempelvis tilføjet nye felter eller rettet i definition af eksisterende felter). Hvilken ændring der er sket, kan revisor konstatere ved at sammenligne en udskrift af kartoteksdefinitionen fra sidste år med den nugældende definition⁶². Udskrift af kartoteksindhold for vare er medtaget som bilag 6.7.

Efterbeh.kode 37 Salgslinie

Efterbehandlingskode er instruktioner der udføres automatisk efter, at brugeren har indtastet data i et felt.

Ændringer konstateres ved at sammenligne efterbehandlingskode med sidste års efterbehandlingskode.

⁶¹ Lars Ole Wiese kræver i <CW 1994-13> og <CW 1994-16> blandt andet en log i PC-systemerne, der registrerer alle bevægelser. Det må tolkes således, at loggen skal indeholde både ajourføring-/ændring/sletning af data og programændringer. Den krævede funktionalitet er derfor mere vidtgående end den "Program change log" <Watne 1992> nævner som et kendetegn for "Librarian packages".

⁶² Det er muligt via menuer, at udskrive disse definitioner i Navigator såfremt man er i besiddelse af designmodulet. Denne funktion svarer til "File description generators" beskrevet i <Watne 1990> side 214.

En udskrift af den samlede efterbehandlingskode som den ser ud i dette eksempel er medtaget som bilag 6.8. Den del af koden som vedrører ændringen⁶³ i det aktuelle eksempel er fremhævet på dette skærbillede:

Demoregnskab	Dato 14-12-94	Tid 22.30.02	DB 29%	97 Kb
Linie Blok	Indsæt blok	Oversigt	Diverse	Generelt Hjælp
MængdeRabatKode				SalgsLinie
Lev.Dato				SalgsLinie
Tekst				SalgsLinie
Tekst 2				SalgsLinie
Enhed				SalgsLinie
Antal				SalgsLinie
<pre>IF (Antal > 10) AND (Antal*Apris > 100) THEN "Rabat%" := 5 ELSE "Rabat%" := 0; dbCALLFIELDPCODE("Rabat%");</pre>				

Vindue 30 Varekartotek

Vinduer indeholder oplysninger om hvad brugeren ser på skærbillederne. Eksempelvis opdateres vinduer blot et felt flyttes.

Rapport Lager Beh.med pris

Rapporter indeholder oplysninger om hvilke data, der skal sammentælles og skrives ud. En rapport kan godt være defineret således, at den opdaterer et kartotek samtidig med udskrivningen.

Printer HP LaserJet 4

Printere indeholder oplysninger om styrekoder således, at systemet kan benytte faciliteter som fed skrift, understregning m.m.

63

Efterbehandlingskoden er opdelt feltvis. For feltet antal er den nødvendige programmering (de 5 programlinier under den grå overskriftsline) væsentligt kortere end i eksemplerne under compiler- og interpretersprog. Dette skyldes, at i 4. generationssprog skal programmøren kun koncentrere sig om den kode, der har betydning for slutresultatet. Såvel indtastning som fremvisning klares af standardrutinerne.

Objekttyper - sammendrag

Revisor kan bruge sin viden om de forskellige typers indhold til at udvælge hvilke ændringer, der skal revideres. Dette kræver, at revisor er vidende om de karakteristiske træk ved de forskellige objekttyper. Af de 5 ændringer i ovenstående eksempel er kun 3 væsentlige at undersøge:

- ☛ Kartotek 27 Vare
- ☛ Efterbeh.kode 37 Salgslinie
- ☛ Rapport Lager Beh.med pris

Disse kan indeholde ændringer, som påvirker systemets sammenhænge og (programmerede) kontroller.

De 2 øvrige ændringer behøver ikke særskilt gennemgang, da de ikke betyder noget for revisionsgrundlaget.

Hovedtræk for rammesystemer

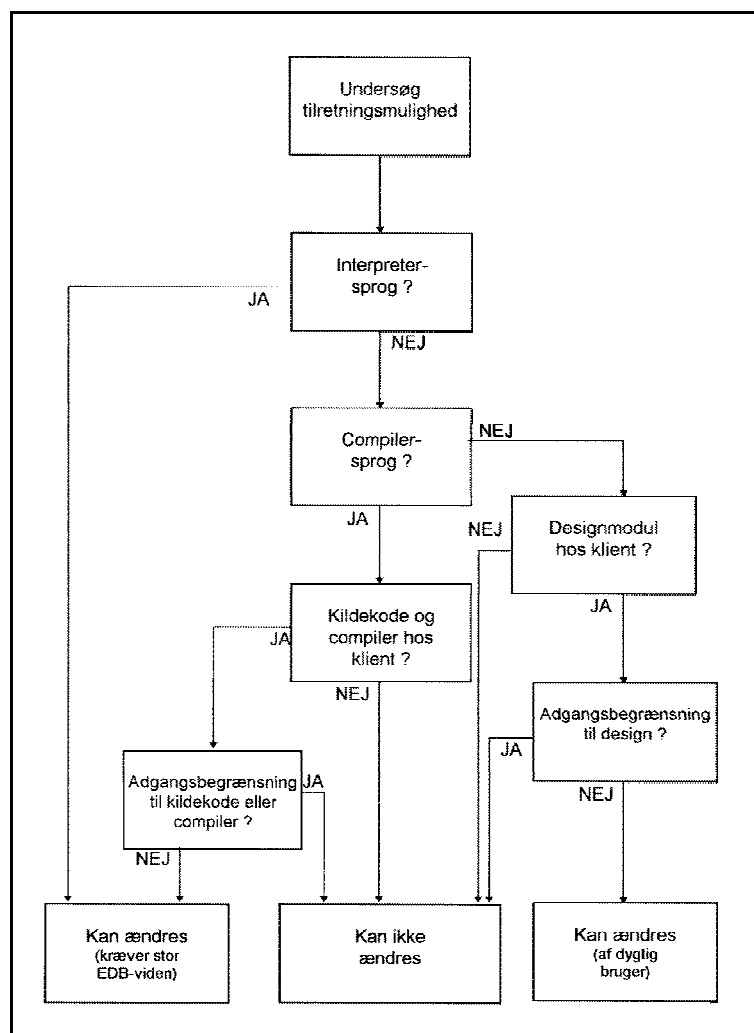
I rammesystemer er det ikke muligt at fjerne "kildekoden" eller "compileren" fra systemet, hvorfor man må adgangs begrænse brugen af compileren. Dette gøres nemmest ved ikke at anskaffe de moduler af systemet der giver designadgang. Hvis designmodul ikke er anskaffet er systemet revisionsmæssigt at betragte som et standardsystem, hvor brugeren hverken har compiler eller kildekode. Da rammesystemerne købes for deres fleksibilitet, er det ikke en særlig udbredt løsning.

I stedet bør brugen af designmoduler begrænses via passwords. Det er således passwordsystemet i rammesystemet der skal begrænse brugen af designmodul. Både ved tilretning og ved almindelig brug af rammesystemerne er det den samme eksekverbare fil (eks. NAVI.EXE) der afvikles. Dette umuliggør, at en kontrol i styresystemet kan anvendes til adgangs begrænsning af designmodul, da brugeren skal have læseret til NAVI.EXE og skriveret til databasen både ved almindelig brug og tilretning. Disse rammesystemer kan derfor ikke revideres med så skarp en afgrænsning mellem generelle EDB-kontroller <Rev. vejl. 14> og kontroller i brugersystemer <Rev. vejl. 17>, da det er brugersystemet der styrer adgang til designmodul (compiler).

3.3.3 Grupperingens betydning for revisionen

For at revisor kan arbejde sig gennem <Rev. vejl 14+17> må han/hun have kendskab til de i foregående afsnit beskrevne kategorier af programmer.

Revisor skal kunne vurdere i hvilket omfang det aktuelle system kan ændres og af hvem. Til brug for denne vurdering kan følgende model anvendes:



Kilde: Egen tilvirkning

I de tilfælde hvor der jævnfør modellen ikke eksisterer en effektiv adgangsbegrænsning kan brugernes manglende viden om systemerne bevirke, at en ikke begrænset adgang til compiler/interpreter/designmodul ikke får væsentlig indflydelse på revisionsstrategien.

Revisor bør dog, efter vor formening, altid undersøge fildatoer og -størrelser⁶⁴ eller objektoversigten⁶⁵, der så ikke skal indikere ændringer inden han/hun når til denne konklusion.

⁶⁴ Compiler- og interpreterbaserede systemer.

⁶⁵ Systemer baseret på 4. generationssprog som eksempelvis Concorde XAL og Navigator.

4 KONKLUSION

4.1 Problemstillingen

Formålet med seminaret var, at undersøge hvilken betydning udbredelsen af tilrettelagte standardsystemer har på revisors arbejdsområder. Undersøgelsen er opdelt i følgende 3 underområder:

- a) Hvilken virkning har EDB-udviklingen (specifikt vedr. udbredelsen af tilrettelagte standardsystemer) for revisor ?
- b) Tager kriteriesættet højde for EDB-udviklingen ?
- c) Hvorledes undersøger revisor hvilken gruppe systemet hører til og om systemet har fungeret uændret i hele perioden ?

De væsentligste resultater af behandlingen fremdrages kort herunder.

4.2 Hvad var de vigtigste hovedresultater

EDB-udviklingen bevirker, at revisors arbejdsgrundlag og omverdenens forventninger til revisor ændres således, at revisor skal have kendskab til edb for, at kunne leve op til de gældende lov- og god skik-regler samt bevare professionens eksistensberettigelse.

<Rev. vejl 1, 14 og 17> suppleret med <Skau 1994> giver et udmærket overblik over EDB-revisionens placering i den samlede revisionsproces samt en metode til udvælgelse af de brugersystemer, der har størst revisionsmæssig værdi. Der fokuseres ikke specielt på ændringer i systemerne.

Fra den eksisterende litteratur om EDB-revision har vi udvalgt 3 bidrag, vi finder væsentlige som supplement til revisionsvejledningerne og som behandler ændringer i en eller anden form.

Revisionsvejledningerne og de 3 bidrag betragtes som kriteriesættet for revision af ændringer i EDB-systemer. Kriteriesættets anvendelighed er testet på cases fra praksis. De væsentligste konklusioner heraf er, at kriteriesættet dækker rimeligt. Kriteriesættet efterlader dog visse huller:

- Der gives ikke tilstrækkelig konkrete anvisninger på hvorledes, revisor undersøger, om et system kan ændres / er ændret.

- ☛ Der lægges ikke op til en vurdering af "programmørers" evner.
- ☛ De meget brede formuleringer (specielt revisionsvejledningerne) gør det svært at arbejde ud fra kriteriesættet.

Set i forhold til tilrettelige standardsystemer er manglen på/utilstrækkeligheden af konkrete anvisninger til konstatering af ændringer i systemerne det mest alvorlige. Årsagen til manglen er formodentlig, at litteraturen er opbygget med tanke på de større og mere traditionelle EDB-miljøer, hvor tilretninger som hovedregel kræver personer med specialviden. <Hougaard 1992> behandler dog de specielle problemer, der knytter sig til PC-miljøer, men denne behandling bygger primært på (ikke tilrettelige) standardsystemer. De tilrettelige standardsystemer behandles meget overfladisk på trods af, at disse systemer havde betydelige markedsandele allerede i 1992.

Behovet for at kunne konstatere ændringer er stigende, da EDB-udviklingen bevirker, at flere og flere virksomheder anvender systemer med store tilretningsmuligheder. Dette har resulteret i, at mange virksomheder tilretter eller får tilrettet systemer således, at revisor meget oftere end tidligere vil skulle vurdere tilretninger og disses betydning for revisionen. Casene illustrerer, at selv mindre tilretninger kan have væsentlig betydning for systemernes samlede funktionalitet.

På den baggrund er det derfor vigtigt for revisor at kunne konstatere, om et system er ændret.

Der anvises i kriteriesættet kun en konkret metode til konstatering af ændringer. Metoden anvises af <Hougaard 1992> og har nogle meget væsentlige forudsætninger, som ikke er oplyst tilstrækkeligt.

I seminaret har vi derfor forsøgt at fremdrage disse forudsætninger og anviser alternative metoder til brug, hvor forudsætningerne ikke er opfyldt. Det er gjort ved en gruppering af systemerne i 3 typer. Disse systemtyper baseres på væsentligt forskellige programmeringsteknikker. Teknikkerne har afgørende betydning for mulighederne for at ændre i systemerne og mulighederne for at konstatere ændringerne.

Målet hermed er at give revisor et redskab til konstatering af, om et system er tilretteligt/tilrettet uanset systemtype.

Der lægges i behandlingen stor vægt på rammesystemer (tilrettelige standardsystemer), da disse er markedsstandard i dag.

Det må konstateres, at systemernes fleksibilitet giver revisor nye problemer i form af mange tilretninger, men revisor får samtidig nye muligheder for at håndtere problemstillingerne. Rammesystemerne

indeholder typisk både faciliteter, der kan oplyse hvilke dele, der er ændret og faciliteter, der kan udskrive dokumentation for systemets nuværende funktionsmåde. Revisor kan således sammenligne dokumentationen for de ændrede dele med den gamle dokumentation. Det kræver et ret indgående kendskab til det konkrete system at forstå betydningen af ændringen.

Det må derfor afsluttende konkluderes, at der ikke nogen nemme løsninger på problemet. Revisor må enten tilegne sig den nødvendige viden til at kunne foretage vurderingerne eller gøre brug af eksperter med denne viden. Det er vor opfattelse, at revisor som minimum skal have et grundlæggende kendskab til de forskellige systemer for overhovedet at kunne foretage en overordnet vurdering af systemet eller ekspertens konklusioner.

4.3 Hvad kunne vi have gjort bedre

Hovedresultaterne bygger for en stor dels vedkommende på cases. Der er derfor ikke statistisk belæg for konklusionerne. Det er dog vor opfattelse, at casene er rimeligt repræsentative for de systemer og problemer, som revisor møder.

Seminaret ville have været bedre dokumenteret, hvis det byggede på empiriske undersøgelser af hvorledes revisor i praksis vurderer virksomhedernes administrative systemer. Det har dog fra seminarets påbegyndelse været fravalgt dels af den årsag at spørgsmålene, der skulle stilles, ikke var kendte på det tidspunkt og dels af hensyn til seminarets omfang.

4.4 Hvilke nye problemstillinger ser vi nu

Efter at have arbejdet med seminar, har vi måttet konstatere nogle problemstillinger, som er nye eller væsentligere end først antaget.

Kravene til revisors viden er temmelig omfattende og derfor melder spørgsmålet sig, om arbejdet bør udføres af en generalistrevisor eller en mere specialiseret revisor. FSR's Informatikudvalg arbejder på etablering af en speciel uddannelse indenfor informatik. Det kan lede tanken hen på den amerikanske CISA-certificering, der også har vundet en vis udbredelse i Danmark. Hvorvidt udviklingen skal føre til en øget specialisering eller øgede krav til generalistrevisor kræver særskilt behandling.

Med udbredelsen af de tilretteliggende standardssystemer opnås stor fleksibilitet, men der afgives samtidig hermed en del af den sikkerhed, som ligger i de ikke tilretteliggende systemer og den adskillelse af udvikling og drift det giver. Problemet må forventes at få stigende betydning, da markedet ikke vil give afkald

på fleksibiliteten. Revisor må derfor overveje, hvorledes han/hun kan kompensere dette evt. via brug af EDB som værktøj i revisionsprocessen.

Seminaret anviser mere konkrete metoder til konstatering af ændringer end fundet i litteraturen. Det giver anledning til spørgsmålet om, hvorvidt mere konkrete (instruksagtige) vejledninger er løsningen. Risikoen er, at revisor blot følger disse og ikke foretager selvstændige kritiske vurderinger, selv om EDB-udviklingen går hurtigere end instrukserne opdateres. Instrukserne kan blive betragtet som udtømmende, hvorved den professionelle vurdering kan gå tabt.

5 LITTERATURLISTE

5.1 Bøger og publikationer

- <IT rev. & sik.> IT revision og sikkerhed
Flere forfattere
FSR's Forlag
(indeholder artiklerne 040, 070, 080, 100, 110 og 140)
- <Torp 1994> Artikel 040: Revision af organisatoriske forhold i et edb-miljø
Jørgen Torp, statsautoriseret revisor
FSR's forlag 1994
ISBN 87-7747-101-6
- <Klarskov 1994> Artikel 070: Dokumentation af systemer, metoder og procedurer
Søren Klarskov, CMA, CISA
FSR's forlag 1994
ISBN 87-7747-109-1
- <Saabye 1994> Artikel 080: Fysisk sikkerhed
Søren Saabye
FSR's forlag 1994
ISBN 87-7747-105-9
- <Søndergaard 1994> Artikel 100: Revision af databasesystemer
Hanne Søndergaard, cand. scient, CISA og Carsten Tjagvad, statsautoriseret revisor
FSR's forlag 1994
ISBN 87-7747-103-2
- <Groth 1994> Artikel 110: Revision af datakommunikation
Thomas Groth, cand. scient. og Jørn Johannsen, cand. merc.
FSR's forlag 1994
ISBN 87-7747-107-5

- <Skau 1994> Artikel 140: Revision af edb-baserede brugersystemer
Esben Skau, cand. merc., CISA og Henning Bendixen, HD, CISA
FSR's forlag 1994
ISBN 87-7747-111-3
- <Hougaard 1992> Revision i PC-miljøer
Henrik Hougaard & Hans Schroeder Nielsen
FSR's Forlag 1992
ISBN 87-7747-041-9
- <Watne 1990> Auditing EDP Systems
Donald A. Watne & Peter B. B. Turney
Prentice-Hall Inc. 1990
ISBN 0-13-052986-9
- <Arens 1991> Revisorprofessionen i det 21. årh. (side 13-22)
Alvin A. Arens, Ph.D., CPA
Foreningen af Yngre Revisorer 1991
ISBN 87-8710-112-2
- <Aggergren 1984> Faglige artikler 20 (side 3-47)
EDB og den eksterne revisor
Lars Aggergren, statsautoriseret revisor
Yngre revisorer 1984
(intet ISBN nr)
- <Tange 1984> Faglige artikler 20 (side 49-101)
EDB, Grundlæggende begreber
Carl Tange, uddannelsesleder
Yngre revisorer 1984
(intet ISBN nr)

- <Franck 1992> God EDB-Skik⁶⁶
Et ansvar for ledelse og organisation
Jan Franck, Lars Aggergren, Børge Jørgensen, Niels Malta Rousing, Vagn
Lausten og Carsten Stenstrøm
FSR's Forlag 1992
ISBN 87-7747-054-0

5.2 Tidsskrifter

- <Friis 1994> Revision & Regnskabsvæsen nr 4 1994 (side 64-65)
Månedens EDB-begreb: "downsizing/rightsizing"
Henrik Friis, civilingeniør og Henrik Juhl, civilingeniør
FSR 1994
ISSN 0034-6918
- <Ipsen 1993> Revision & Regnskabsvæsen nr 12 1993 (side 9-20)
Ny vejledning om "Grundlæggende principper for revision af regnskaber".
Lis Ipsen, statsautoriseret revisor
FSR 1993
ISSN 0034-6918
- <Juhl 1993> Revision & Regnskabsvæsen nr 8 1993 (side 42-43)
Månedens EDB-begreb: "LAN, MAN eller WAN"
Henrik Juhl, civilingeniør og Linda Pedersen, akademiingeniør
FSR 1993
ISSN 0034-6918
- <Tjagvad 1993> Revision & Regnskabsvæsen nr 7 1993 (side 8-16)
Revision af edb-baserede brugersystemer
Carsten Tjagvad, statsautoriseret revisor og Peder Toft, statsautoriseret
revisor
FSR 1993
ISSN 0034-6918

⁶⁶

Denne publikation er kommenteret/udfyldt af Lars Aggergren i Revision & Regnskabsvæsen nr 4 1992 side 34-35.

- <Aggergren 1994> INSPI nr 5 1994 (side 12-14)
Status NU for edb-revision
Lars Aggergren, statsautoriseret revisor
Foreningen af Yngre revisorer 1994
- <Schroeder 1993> Inspi nr 3 1993 (side 31-34)
Revision hos klienter, der anvender EDB, del 1
Inspi nr 5 1993 (side 11-13)
Revision hos klienter, der anvender EDB, del 2
Hans Schroeder, cand. merc. aud.
Foreningen af Yngre revisorer 1993
- <CW 1994-13> ComputerWorld nr. 13 1994
De fleste økonomisystemer på markedet er ulovlige (side 1)
Snyd skattevæsenet så vandet driver uden at blive opdaget (side 27)
K. Krabbe Hermansen, journalist
(artiklerne er baseret på interview med lektor Lars Ole Wiese og skatterevisor Henning Lassen)
IDG Danmark A/S
ISS 0107-5217
- <CW 1994-16> ComputerWorld nr. 16 1994
Virksomhederne har ikke de nødvendige informationer (side 16)
Lars Ole Wiese, lektor
(svar på indlæg fra Merete Lydolph Larsen (PC&C) der besvarer artikel i <CW 1994-13>)
IDG Danmark A/S
ISS 0107-5217

5.3 Love m.v.

- <Lov om SR> Bekendtgørelse af lov om statsautoriserede revisor
Industri- og Samordningsministeriets lovbekendtgørelse nr. 762 af 22. august 1994
Industri- og Samordningsministeriet 1994

- <Bogf.bekendtg.>** Bekendtgørelse om erhvervsdrivende virksomheders bogføring, årsregnskaber og opbevaring af regnskabsmateriale⁶⁷
Erhvervs- og Selskabsstyrelsens bekendtgørelse nr. 598 af 21. august 1990.
Erhvervs- og Selskabsstyrelsen 1990.
- <A/S loven>** Bekendtgørelse af lov om aktieselskaber
Industri- og Samordningsministeriets lovbekendtgørelse nr. 572 af 24. juni 1994.
Industri- og Samordningsministeriet 1994.
- <ApS loven>** Bekendtgørelse af lov om anpartsselskaber
Industri- og Samordningsministeriets lovbekendtgørelse nr. 573 af 24. juni 1994.
Industri- og Samordningsministeriet 1994.
- <Påtegn.bekendtg.>** Bekendtgørelse om revisors påtegning på regnskaber
Erhvervs- og Selskabsstyrelsens bekendtgørelse nr. 121 af 23. februar 1989.
Erhvervs- og Selskabsstyrelsen 1989.
- <Rev. vejl. 1>** Revisionsvejledning nr. 1
Vejledning om grundlæggende principper for revision af regnskaber
Revisionsteknisk udvalg (FSR) 1993
ISBN 87-7747-082-6
- <Rev. vejl. 14>** Revisionsvejledning nr. 14
Vejledning om revision i virksomheder som anvender EDB
Revisionsteknisk udvalg (FSR) 1989
(intet ISBN nr.)
- <Rev. vejl. 17>** Revisionsvejledning nr. 17
Vejledning om revision af edb-baserede brugersystemer
Revisionsteknisk udvalg (FSR) 1990
(intet ISBN nr.)

⁶⁷

Kommenteret af Finn Elkjær Nielsen i Revisionsorientering A 1.9 (oktober 1991).

6 BILAG

6.1 Case A: Manglende generelle kontroller

I et firma hvor der anvendes automatisk datafangst for medarbejdertimer via strekkodeindlæsning, er den PC'er der anvendes som stempelur oprettet med maksimale rettigheder. Denne PC'er er den eneste der er fysisk tilgængelig for alle firmaets ansatte uden undtagelse. De maksimale rettigheder (superbrugerstatus i Navigator og supervisorstatus i Novell) bevirker, at alle medarbejdere har adgang til alle oplysninger og funktioner (herunder ændring i systemerne) i firmaet.

For at undgå at beskæftige en administrativ medarbejder cirka 15 timer om ugen har firma A fået tilsluttet en PC'er på værkstedet til erstatning for det traditionelle stempelur. Når værkføreren modtager en ordre oprettes denne i systemet med oplysning om sagsnummer, kundenummer samt kort beskrivelse af det arbejde, der skal udføres. Efter værkføreren har oprettet sagen, udskrives et sagskort med de ovennævnte oplysninger i klarskrift samt sagsnummer som strekkode.

Dette kort udleveres til medarbejderen. Alle medarbejdere i firmaet har et navneskilt med strekkode på. Når så medarbejderen skal stemple ind på en sag, går vedkommende hen til strekkodelæseren (elektronisk stempelur) og indlæser først sin egen strekkode og derefter strekkoden på sagskortet. Disse tidsregistreringer ligger i en sagskladde, der gennemgås af værkføreren ved dagens slutning og derefter bogføres⁶⁸.

Udover sagskort kan medarbejderne efter at have indlæst deres egen strekkode også vælge at indlæse en standardkode fra strekkodeskema på væggen (lyspen kan føres hen over strekkoden) i stedet for at indlæse strekkode fra et sagskort. Disse standardkoder omfatter blandt andet Fri (stemple ud), Oprydning/renholdelse værksted, Tomgang (spildtid) og Vedligeholdelse ejendom. Disse koder på væggen vil udføre en af følgende funktioner:

- ☛ stemple ud fra tidligere sag uden at stemple ind på en ny (Fri)
- ☛ stemple ind på "internt arbejde" hvorved lønnen fordeles til foruddefineret finanskonto (Oprydning/renholdelse, Tomgang og Vedligeholdelse ejendom)

⁶⁸

Denne databehandlingsmetode kaldes online inddatering/batch opdatering. I det konkrete tilfælde bevirker batchopdateringen, at eksempelvis faktura til kunder først kan udskrives dagen efter at arbejdet er afsluttet, da de sidste timer ellers ikke kommer med.

Da firmaets arbejdsopgaver bevirker, at en medarbejder på en typisk arbejdsdag er beskæftiget på 1 til 8 sager, er den PC'er der fungerer som stempelur naturligvis placeret på værkstedet på et lettilgængeligt sted.

Firmaets administrative løsning er leveret af en af de større edb-forhandlere. Leverandøren har foretaget opsætning af passwords ud fra mundtlig beskrivelse fra regnskabschefen. Leverandøren har (af nemhedshensyn) ikke oprettet en brugerprofil for streghkodeindlæsningen, men i stedet anvendt en superbrugerprofil.

Oprettelse af brugerrettigheder i firmaets system sker således:

- ☛ Først tages stilling til om brugeren skal have superbrugerrettigheder. Hvis dette er tilfældet, skal der ikke gøres andet end at svare ja hertil. En superbruger kan slette andre brugere, inklusive andre superbrugere, og har pr. definition adgang til alle delsystemer og funktioner i systemet.
- ☛ Hvis der svares nej til superbruger kan brugeren som udgangspunkt intet i systemet. Der skal så tildeles rettigheder (af en superbruger) til hvert eneste kartotek, funktion og skærbillede. Disse rettigheder tildeles ofte til en standardprofil f.eks. STD_ADM. De medarbejdere der arbejder i administrationen oprettes så med samme rettigheder som STD_ADM.

I firmaet ser skærbilledet for brugeroprettelsen således ud:

[F1]=Hjælp [Alt]=Pull-down-menu Dato 20-11-94 Tid 13.28.39 DB 38% 137 Kb
 Linie Adgang(database) Generelt Hjælp

BrugerID	Adgangs- Kode	Super	Rettigheder ligesom
ANDERS	~FRTEBMBF	Ja	
BENT	~AIQDXKZC		STD_ADM
CARL	~GMTCESIB		STD_MEK
LOTTE	~TRRDXYLC		STD_ADM
PER	~WVOAHUBF		STD_MEK
PIA	~ZQMWYSCR		STD_ADM
STD_ADM	~TDMVVSXC		
STD_MEK	~VMUFKCDP		
STREGKODE		Ja	

S/N 45-N3E0565

Christensen
 rvej 6
 lding

BrugerID: ANDERS Indsæt

På grund af arbejdet med tildeling af netop de rettigheder som brugeren må have, vælger nogle firmaer at oprette alle brugere med Super=Ja, da det er det nemmeste ved oprettelsen. I det konkrete tilfælde havde man (leverandøren) givet fulde rettigheder til brugerne med brugerID STREGKODE.

Den skærm der anvendes som stempelur er logget på nettet døgnet rundt med brugerID'en STREGKODE. Virkningen af dette er, at hele det administrative system er åbent for alle brugere af denne skærm.

PC'eren er endvidere logget på netværket som supervisor⁶⁹.

Da firmaets system er meget brugervenligt kan man med hoptaster⁷⁰ nemt gå fra stregkodeindlæsning til finans og eksempelvis bogføre eller udskrive købsrekvisitioner, salgsfakturaer eller salgskreditnotaer. For firmaet betyder det, at alle medarbejdere kan få adgang til alt ikke blot i det administrative system, men også tekstbehandling (ansættelseskontrakter, forhandleraftaler, bestyrelsesreferater m.m.) samt regneark (budgetter m.m.).

Forholdet er opdaget ved gennemgang af de generelle EDB-kontroller herunder passwordsystemer. Revisionsmetoden var "efterprøvelse" ved forevisning af skærbillede svarende til foregående figur.

Efter at have konstateret forholdet i forbindelse med gennemgangen af generelle EDB-kontroller rapporteredes dette til firmaet via revisionsprotokollen.

⁶⁹ Den højeste rettighed i Novell Netware. Det betyder blandt andet, at man kan oprette, ændre og slette brugere på netværket og har adgang til alt på netværket og kan give denne adgang til andre.

⁷⁰ Taster der gør det muligt, at springe rundt i systemet uden at skulle ud i hovedmenuen. Eksempelvis kan man fra en kundegruppe hoppe til en kundegruppeoversigt. I denne kundegruppeoversigt kan man stille sig på kontonummeret for samlekonto i finans og trykke en gang mere på hoptasten. Det vil bevirke direkte adgang til finans.

6.2 Case B: Konstater ændringer

Firmaet oplyste at der ikke var foretaget ændringer i systemet i regnskabsåret. En efterprøvelse viste, at denne oplysning ikke var korrekt. Der var ændret i systemet, men firmaet opfattede det ikke som en ændring. Casen giver eksempler på hvorledes, revisor kan vælge at undersøge systemerne for ændringer.

Når revisor skal undersøge, om systemet har fungeret uændret i den periode der revideres⁷¹, kan det være teknisk kompliceret at undersøge dette med andre metoder end forespørgsel⁷². Forespørgsel er ikke altid tilstrækkeligt, hvilket denne case demonstrerer. Firmaet svarede nej til ændringer i systemet da vi forespurgte. Da vi efterprøvede svaret, fandt vi dog ud af noget ganske andet.

Der var i det konkrete tilfælde ikke tale om forsøg på vildledning, men der var ikke overensstemmelse mellem vor og firmaets opfattelse af hvad der var ændringer.

Det er i mange systemer muligt at få en oversigt over, hvad der er tilrettet. Dette eksempel er fra Navigator.

Firma B		Dato 17-11-94 Tid 7.43.17 DB 38% 129 Kb			Generelt Hjælp	
Linie	Opdater	Indlæs	Udlæs	Diverse		
Type	Nr	Navn	Dato	Tid	Ænd	VersionsListe
Kartotek	25	KreditorPost	24-11-92	14.48		PCC4.03
Kartotek	27	Vare	05-01-93	11.31		PCC5
Kartotek	28	VareSalgsPris	17-09-90	2.54		PCC1
Kartotek	29	VareSalgsMængdeRabat	17-09-90	2.54		PCC1
Kartotek	30	VareTekst	17-09-90	2.54		PCC1
Kartotek	32	VarePost	11-06-94	23.29	Ja	PCC6,BB
Kartotek	36	SalgsHoved	16-05-93	15.38		PCC6
Kartotek	37	SalgsLinie	17-11-94	0.16	Ja	PCC6,AA
Kartotek	38	KøbsHoved	16-05-93	15.39		PCC6
Kartotek	39	KøbsLinie	16-05-93	14.41		PCC6
Kartotek	40	VareSalgsPrisForslag	17-09-90	2.54		PCC1
Kartotek	42	Afrunding	18-09-90	2.44		PCC1
Kartotek	43	KøbsBemærkning	18-09-90	2.44		PCC1
Kartotek	44	SalgsBemærkning	18-09-90	2.44		PCC1
Kartotek	45	FinansJournal	18-09-90	2.44		PCC1
Kartotek	46	VareJournal	18-09-90	2.44		PCC1
Kartotek	48	LagerRegul.Buffer	14-05-93	13.54		PCC6

VersionsListe: PCC6 Indsæt

I eksemplet står der ja ved ændret udfør 2 kartoteker (tabeller). Den praktiske betydning af JA i kartotek 37 (salgslinie) er beskrevet i case C.

Det er ikke muligt at se om den tilretning, der er foretaget 17. november 1994 klokken 0.16 er den eneste eller der har været andre før. Det er kun den seneste ændring der vises. Dette er i modsætning til en traditionel log, hvor alle ændringer vises.

⁷¹ <Rev. vejl. 1> punkt 7.4.

⁷² <Rev. vejl. 14> punkt 3.5.

Alle objekter (linier) hvor datoen er nyere end sidste gennemgang af systemet bør vurderes for væsentlighed og risiko.

Et tilsvarende eksempel fra Concorde XAL ser således ud:

CONCORDE			Form	1/2	1/4021	Usr	Time/Sag	Løn
	Finans	D	AdgangskontrolFile			Sys		
Dagligt	CONCORDE		AdgangskontrolFile			Sys		
Kartotek			AdgangskontrolRights			Sys		
Udskrifter			AdgangskontrolUsers			Sys		
Periodisk			AdgangskontrolUtil			Sys		
Tilpasning			Afdeling			Sys		
Licens:			AftaleKalender			Sys		
			Applikationskoder			Sys		
			ArtMedarbejder			Sys		
			Ascii-Tabel			Sys		
			Batch			Sys		
			Betaling			Sys		
			BrugerOpsætning			Sys		
			BrugerStatus			Sys		
			Bærer			Sys		
			Calculator			Sys		
			CalculatorNew			Sys		
			ConcordeØkonomi2Xal			Sys		
			DatabaseRelationer			Sys		

I eksemplet ses at AdgangskontrolFile findes både som Sys (System = Standardversion) og Usr (User = brugertilpasset version).

Alle nye Usr i forhold til sidste år bør vurderes for væsentlighed og risiko.

6.3 Case C: Tilrettet faktureringsrutine

En efter firmaets opfattelse ubetydelig ændring, der kun skulle give det fakturaudseende man ønskede, satte en væsentlig programmeret kontrol ud af kraft. Konsekvensen heraf blev, at der ikke var kontrol med om udskrevne og bogførte fakturaer stemte overens. Det bevirkede, at der på grund af indtastningsfejl blev bogført fakturaer med beløb, der var forskellige fra beløbene på de udskrevne fakturaer.

I et ordreproducerende firma med et dyrt kapacitetsapparat anvendes følgende system ved registrering af omsætningen. Der fastsættes standardsalgspriser på:

- timer (arbejds løn),
- materialer (indkøbspris + procenttillæg)
- maskintillæg (maskintimeløn)

Når en ordre skal faktureres, opgøres forbruget til standardsalgspriser for hver af ovennævnte kategorier. Den faktiske salgspris er nogle gange fast aftalt og andre gange efter regning. Det opgjorte forbrug til standardsalgspris konteres automatisk ved fakturering på 3 separate salgskonti og differencen mellem det opgjorte beløb og den faktiske salgspris konteres på konto for over/underdækning.

Firmaet har i sin råbalance opdelt nettoomsætningen i de 4 poster:

- timer (standardpriser)
- materialer (standardpriser)
- maskintillæg (standardpriser)
- over/underdækning

Af nemhedshensyn er systemet tilrettet så arbejdsgangen ved fakturering er følgende:

- Indtast sagsnummer (maskinen finder selv kundenummer hertil)
- Indtast fakturadato og tekst til faktura.
- Indtast salgspris.

Maskinen udregner selv de realiserede standardsalgspriser og fordeler den indtastede salgspris på de 4 salgskonti. Da man naturligvis ikke vil vise kunderne denne opdeling af salget (specielt ved

overdækning), har man fået tilrettet faktureringsrutinen således, at oplysningerne kun fremtræder på skærbillederne. Et eksempel på et sådant skærbillede er:

Firma C		Dato 17-11-94 Tid 1.19.25 DB 38% 119 Kb						
Linie	Vindue	Funktion	Generelt Hjælp					
Salgsfaktura								
Nummer.....: 1008		SagsNr.....: M0346	Salgspris: 22.000,00					
		KundeNr....: 75504515	Bogf.Gr...: JYLLAND					
		Navn.....: Jens Jensen						
		Adresse...: Maskinvej 1						
Bogf.Dato....: 10-01-94		Adresse 2..:						
		PostNr+By..: 6000 Kolding						
A	Nummer	Tekst	Ant	Enh	Apris	Rabat %	NettoBeløb	Bogf Udsk
K	1010	Løn	1		12.364,30		12.364,30	Bogf
K	1012	Materialer	1		4.623,43		4.623,43	Bogf
K	1014	Maskintillæg	1		7.213,11		7.213,11	Bogf
K	1016	Over-/underdækning	1		-2.200,84		-2.200,84	Bogf
		Renovering af maskine med ny styreenhed og motorer.						Udsk
K	9999	Pris iflg. tilbud	1		20.000,00		20.000,00	Udsk
K	9999	Pulverlakering	1		2.000,00		2.000,00	Udsk
Rabat%:								Indsæt

Den centrale i tilretningen er det sidste felt på linierne, hvor der enten står Bogf eller Udsk. Her vælges om linien skal bogføres eller udskrives. Det er således ikke muligt, at indtaste en linie der både skal bogføres og udskrives. Hvis man sælger noget f.eks. 2 timers arbejde til standardsalgsspris skal der indtastes 2 næsten ens linier. Eneste forskel er sidste felt på linien hvor den ene er Bogf og den anden er Udsk.

Denne tilretning var foretaget af edb-leverandøren og gik i sin enkelhed ud på, at fakturaudskriftsrutinen så bort fra linier mærket Bogf og bogføringsrutinen så bort fra linier mærket Udsk. Desværre var der ikke indbygget nogen form for kontrol der sikrede, at summen af linier Bogf var lig summen af linier Udsk.

Tilretning bevirkede derfor, at der ikke nødvendigvis var overensstemmelse mellem bogførte og udskrevne fakturaer.

Denne manglende kontrol betød, at firmaet i løbet af 12 måneder (cirka 2500 fakturaer) havde udskrevet 8 fakturaer, hvor der ikke var overensstemmelse mellem det bogførte og det udskrevne.

6.4 Case D: Tilføjet rutine til intern fakturering i profitcenter

I et firma med profitcentre blev internt salg genereret af en rutine i edb-systemet. Rutinen debiterede størsteparten af det interne salg i den sælgende afdeling således at eneste effekt af internt salg var opblæsning af omsætning og omkostninger i den sælgende afdeling. Afdelingslederen i den sælgende afdeling overbeviste direktionen, om at omkostningen var en fejlpostering og ikke hørte til hos ham. Ingen tog initiativ til at finde ud af, om der var en kreditpost et sted der skulle reduceres eller omkostningen manglede et andet sted.

I et firma der styrer efter profitcentertankegang, var der tilføjet en rutine der automatisk skulle postere de interne "faktureringer" mellem profitcentrene. Rutinen identificerer posterings, der ikke er noteret på eksterne ordrer og finder så ud af hvad timer (og evt. materialer) er brugt til. På baggrund af disse oplysninger bliver der genereret 2 linier i kasserapporten, der krediterer beløbet på en intern salgskonto i den afdeling der "ejer" ressourcerne og debiterer beløbet på en konto i den afdeling, der har nydt godt af arbejdet.

I det pågældende tilfælde blev timer der ikke var noteret på eksterne ordrer ikke identificeret korrekt. Timer der var anvendt på eksterne ordrer, der tilhørte andre afdelinger, blev korrekt krediteret på internt salg i medarbejderens afdeling, men fejlagtigt debiteret på konto for uproduktiv tid i samme afdeling⁷³. Dette bevirkede store fordele for de afdelinger der købte timer fra andre afdelinger og store ulemper for de afdelinger der solgte timer internt i huset.

Da omfanget af internt salg var ganske betydeligt (over 100.000 kr om måneden), burde en sådan fejl straks opdages. Afdelingslederen i den sælgende afdeling konstaterede, at omkostninger ved uproduktiv tid var helt ude af proportioner og bad administrationen rette det. Da administrationen ikke kunne finde ud af, hvad der var galt, blev beløbet ikke omposteret. Afdelingslederen fik direktionen overbevist om at tallet var fuldstændig forkert og blev derfor målt uden hensyntagen til denne konto.

I den afdeling der skulle have båret størsteparten af omkostningerne, var lederen forholdsvis ny og afdelingen havde ikke noget grundlag for at læse de afdelingsopdelte balancer. Direktionen var tilfreds med afdelingens økonomiske resultater og man fortsatte som hidtil.

73

Der "betales" således IKKE for brug af medarbejdere fra andre afdelinger.

I praksis betød det at firmaet kørte i et år, hvor man så bort fra en debetpost på et par millioner i balancen uden at spekulere over, om den tilsvarende kreditpost var i resultatopgørelsen og skulle omposteres eller om debetposten burde stå et andet sted i resultatopgørelsen.

6.5 Case E: Tilrettet lagerliste anvendte forkerte data fra systemet

Firmaet var ikke tilfreds med de 2 lagerlister, der fandtes som standard i edb-systemet. Man definerede derfor sin egen liste med udgangspunkt i den ene af standardlisterne. Listen fik tilføjet et felt fra varekartoteket og dette felt blev ganget med et felt der allerede var på listen. Resultatet af denne udregning blev udskrevet og opsummeret. Firmaet antog, at det stemte med totalen på den anden standardliste. Der var dog væsentlige forskelle på de varenumre, hvor der var "svømmende varer". Det tilrettelige standardsystem's håndtering af "svømmende varer" blev sat ud af kraft ved tilretningen.

Firmaet ønskede en lagerliste med kostpris pr. stk., antal og samlet værdi til brug for status. Firmaets administrative system indeholder en liste kaldet LAGERBEHOLDNING som vist her:

Firma E		LAGERBEHOLDNING							31. december 1993	
Stk	Nummer	Navn	Enhed	Beholdning	Reserveret	Bestilt leverandør	Restordre leverandør	Disponible	Minimum beholdning	Genbestil
	B-60	Bordplade 60 cm	Stk	50				50	20	
	B-70	Bordplade 70 cm	Stk	50				50	10	

Systemet indeholdt også en liste kaldet LAGERVÆRDI med følgende oplysninger:

- ☛ værdi af primolager
- ☛ tilgang til kostpris
- ☛ afgang til kostpriser
- ☛ værdi af ultimolager

Listen, der er vist her, indeholder ikke oplysninger om stykantal.

Firma E		LAGERVÆRDI					31. december 1993	
Stk	Nummer	Navn	Enhed	Lagerværdi d. 01-01-93	Køb og tilgang	Forbrug og afgang	Lagerværdi d. 31-12-93	Finansbogf. lagerværdi
	B-60	Bordplade 60 cm	Stk	()	(50.000,00)	()	(50.000,00)	0,00
	B-70	Bordplade 70 cm	Stk	()	(55.000,00)	()	(55.000,00)	55.000,00
		Ialt.....		()	(55.000,00)	()	(55.000,00)	105.000,00
				()	(105.000,00)	()	(105.000,00)	

Tallene i parentes er udregnet på grundlag af antalsmæssige bevægelser på lageret (f.eks leverancer), mens de øvrige tal er baseret på beløbsmæssige bevægelser på lageret (f.eks fakturaer). Tallene i parentes udskrives kun, hvis de afviger fra de beløbsmæssige bevægelser.

Firmaet brød sig ikke om en værdiliste, hvor man ikke kunne efterregne værdien (antal*kostpris) og definerede derfor en ny rapport baseret på LAGERBEHOLDNING (kaldet LAGERBEHOLDNING med

pris), men med tilføjelse af kostpris og kostværdi (antal*kostpris). Kolonnen kostværdi blev talt sammen og lagerreguleringen blev bogført i kassekladden ud fra resultatet.

Firma E		LAGERBEHOLDNING med pris								31. december 1993		
Stk	Nummer	Navn	Enhed	Beholdning	Reserveret	Bestilt leverandør	Restordre leverandør	Disponible	Minimum beholdning	Genbest	Kostpris	Kostværdi
	B-60	Bordplade 60 cm	Stk	50				50	20	Ja	1.000,00	50.000,00
	B-70	Bordplade 70 cm	Stk	50				50	10	Ja	1.100,00	55.000,00
		TOTAL										105.000,00

Tilretningen tog ikke højde for systemets kompleksitet omkring styring af svømmende varer. De oprindelige lister var baseret på forskellige antalsoplysninger, idet systemet arbejdede både med begrebet antal og begrebet værdiantal. Begrebet antal dækker over det fysisk tilstedeværende lager (d.v.s. modtagne varer som firmaet endnu ikke har modtaget købsfakturaer på er medtaget, og afsendte varer som der er lavet salgssølgessedel (men ikke salgsfaktura) på er fratrukket).

Begrebet værdiantal omfatter derimod kun transaktioner, hvor købs- og salgssølgessedel er bogført.

Forskellen på begreberne kan illustreres således:

Firmaet har regnskabsår den 31. december og modtager 50 stk af vare B-60 den 27. december. Faktura fremkommer først den 4. januar. I dette tilfælde er antal=50 og værdiantal=0 den 31. december. Det er således kolonnen værdiantal (og ikke antal) der skal ganges med kostpris for at finde lagertal til regnskabet. Forskellen kan også illustreres ved dette skærmbillede:

Firma E		Dato 14-12-94 Tid 23.57.14 DB 29% 159 Kb							
Linie Udligning		Generelt Hjælp							
Vareposter - VareNr: B-60									
Dato	A Nr	Tekst	Antal	Værdi- Antal	Beløb	Å	LøbeNr		
27-12-93	K		50			J	8		
04-01-94	K	1005		50	50.000,00		9		
			50	50					

Dato: 04-01-94 Indsæt

Denne fejl bevirkede systematisk ignorering af svømmende varer ved statusopgørelse og kan alt efter mængden af svømmende varer bevirke over- eller undervurdering af lager og dermed resultat.

Fejlen er opdaget ved, at vi vidste at lagerlisten ikke var standard med det udseende den havde. Vi valgte derfor, at kontrollere dens virkemåde.

6.6 Case F: Udbetaling af for meget overtidstillæg

En rapport udviklet til opgørelse af normal- og overtimer pr. medarbejder pr. dag beregnede for mange timer til medarbejdere, der mødte efter normal arbejdstids begyndelse og gik hjem efter normal arbejdstids ophør. Der blev som konsekvens heraf udbetalt for meget i løn.

I et firma der anvendte timetal fra et time/sag-system som løngrundlag, var den rapport der opdelte dagens timer i normaltimer og overtimer ikke i overensstemmelse med de faktiske timer, der var registreret i systemet. Rapporten var udviklet fra bunden af firmaet.

Rapporten gav overtidstillæg for tidligt, hvis medarbejderen var mødt efter normal arbejdstids begyndelse den pågældende dag.

Såfremt medarbejderen var gået hjem efter normal arbejdstids ophør blev antal normaltimer sat til 7,4 og antal overtimer sat til antal timer efter normal arbejdstids ophør. Det medførte, at medarbejdere der mødte senere end normal arbejdstids begyndelse og gik hjem efter normal arbejdstids ophør fik løn fra normal arbejdstids begyndelse til det faktiske tidspunkt for arbejdets ophør.

Ingen af medarbejderne havde gjort opmærksom på problemet (rutinen havde heller aldrig snydt dem).

Fejlårsagerne var manglende erfaring hos "programmør", manglende test af rapporten og manglende kontrol af uddata.

6.7 Kartoteksindhold

Udskrift af kartoteksindhold for kartoteket Vare:

Kartotek 27 Vare
PCC5,BB

Rettet d. 04-02-95 11.16.22

Felter

Nr	A	Navn	Type	Læn	K	Kal	Attributter
1	+	Nummer	Kode	20			Ikke blank
2	+	Nummer 2	Kode	20			
3	+	Navn	Tekst	30			
4	+	SøgeNavn	Kode	30			Søgefelt
5	+	Navn 2	Tekst	30			
6	+	StykListe	JaNej				
7	+	Klasse	Kode	10			
8	+	Enhed	Tekst	10			
9	+	PrisFaktor	Heltal				
11	+	Bogf.Gr	Kode	10			Kartoteksrelation=VareBogf.Gr
12	+	Placering	Kode	10			
13	+	SalgsMængdeRabatKode	Kode	20			Specialrelation
14	+	VareDeb.RabatGr	Heltal				
15	+	Beregn Fak.Rabat	JaNej				Initialværdi=Ja
16	+	StatistikGr	Heltal				
17	+	ProvisionsGr	Heltal				
18	+	SalgsPris	Decimaltal				Positiv,Decimaler=2,5
19	+	AvanceBeregning	Valgfelt				Valgsæt=Av=Salg-Kost,Salg=Kost+Av, ,
20	+	Avance%	Decimaltal				Max.værdi=99,99999
21	+	LagerAfgang	Valgfelt				Valgsæt=FIFO,LIFO,ÅbenPost
22	+	KostPris	Decimaltal				Positiv,Decimaler=2,5
23	+	KostBeregning	Valgfelt				Valgsæt=Fast,Sidste,Gn.snit
24	+	KøbsPris(fast)	Decimaltal				Positiv,Decimaler=2,5
25	+	KøbsPris(sidste)	Decimaltal				Positiv,Decimaler=2,5
26	+	KøbsPris(gns.)	Decimaltal				Positiv,Decimaler=2,5
27	+	NettoNetto	Decimaltal				Positiv,Decimaler=2,5
28	+	Omkostnings%	Decimaltal				Positiv
29	+	I.MomsKode	Kode	10			Kartoteksrelation=Moms
30	+	U.MomsKode	Kode	10			Kartoteksrelation=Moms
31	+	LeverandørKode	Kode	20			Specialrelation
32	+	Lev.VareNr	Tekst	20			
33	+	LeveringsTid	Tekst	20			Datoformel
34	+	Min.Beholdning	Decimaltal				
35	+	Max.Beholdning	Decimaltal				
36	+	OrdreKvantum	Decimaltal				
37	+	Alt.VareNr	Kode	20			Kartoteksrelation=Vare
38	+	Vejl.Pris	Decimaltal				Positiv,Decimaler=2,5
39	+	AfgiftsDel	Decimaltal				Procent
40	+	AfgiftsKode	Kode	10			
41	+	BruttoVægt	Decimaltal				Positiv
42	+	NettoVægt	Decimaltal				Positiv
43	+	Antal pr. Kolli	Decimaltal				Positiv
44	+	Rumfang	Decimaltal				Positiv
45	+	Holdbarhed	Kode	10			
46	+	FragtForm	Kode	10			
47	+	ToldPosition	Kode	20			Kartoteksrelation=BrugsTarifNummer
48	+	ToldFaktor	Decimaltal				
49	+	KøbsLand	Kode	10			Kartoteksrelation=Land
50	+	Budget(antal)	Decimaltal				
51	+	Budget(beløb)	Decimaltal				Beløbsdecimaler
52	+	Budget(DB)	Decimaltal				Beløbsdecimaler
53	+	Bemærkning	JaNej			K [■]	Ingen indtastning
54	+	Spærret	JaNej				
62	+	Rettet d.	Dato				Ingen indtastning
64	+	DatoAfgrænsning	Dato			K	
65	+	AfdelingsAfgrænsning	Kode	10		K	Kartoteksrelation=Afdeling
66	+	ProjektAfgrænsning	Kode	10		K	Kartoteksrelation=Projekt
67	+	LokationsAfgrænsning	Kode	10		K	Kartoteksrelation=Lokation
68	+	Beholdning	Decimaltal			K [■]	Ingen indtastning

69 + VærdiBeholdning	Decimaltal	K [■]	Ingen indtastning
70 + Bevægelse	Decimaltal	K [■]	Ingen indtastning
71 + Køb(antal)	Decimaltal	K [■]	Ingen indtastning
72 + Salg(antal)	Decimaltal	K [■]	Ingen indtastning
73 + Tilgang(antal)	Decimaltal	K [■]	Ingen indtastning
74 + Afgang(antal)	Decimaltal	K [■]	Ingen indtastning
77 + Køb(DKK)	Decimaltal	K [■]	Ingen indtastning, Beløbsdecimaler
78 + Salg(DKK)	Decimaltal	K [■]	Ingen indtastning, Beløbsdecimaler
79 + Tilgang(DKK)	Decimaltal	K [■]	Ingen indtastning, Beløbsdecimaler
80 + Afgang(DKK)	Decimaltal	K [■]	Ingen indtastning, Beløbsdecimaler
83 + Forbrug(DKK)	Decimaltal	K [■]	Ingen indtastning, Beløbsdecimaler
84 + Forv.Tilgang(antal)	Decimaltal	K [■]	Ingen indtastning
85 + Forv.Afgang(antal)	Decimaltal	K [■]	Ingen indtastning
87 + SalgsPris Inkl.Moms	JaNej		
88 + Tilhører VareGr.Nr	Kode	20	Kartoteksrelation=VareGruppe

Kalkulationsbeskrivelse for feltet Bemærkning

Kartotek.....: KartoteksBemærkning
 Afgrænsning.....: Kartotek=3,Nummer=Nummer

Afgrænsning af KartoteksBemærkning	Konstant/ felt på...Vare	Kun til-dato
Kartotek	= Konstant	3
Nummer	= Felt	Nummer
		[]
		[]

Kalkulationsbeskrivelse for feltet Beholdning

Kartotek.....: VarePost
 Afgrænsning.....: VareNr=Nummer,AfdelingsKode=AfdelingsAfg
 Sumfelt.....: Antal
 Vend fortegn.....: Nej

Afgrænsning af VarePost	Konstant/ felt på...Vare	Kun til-dato
VareNr	= Felt	Nummer
AfdelingsKode	= Felt	AfdelingsAfgrænsning
ProjektKode	= Felt	ProjektAfgrænsning
LokationsKode	= Felt	LokationsAfgrænsning
		[]
		[]
		[]
		[]

Kalkulationsbeskrivelse for feltet VærdiBeholdning

Kartotek.....: VarePost
 Afgrænsning.....: VareNr=Nummer,AfdelingsKode=AfdelingsAfg
 Sumfelt.....: VærdiAntal
 Vend fortegn.....: Nej

Afgrænsning af VarePost	Konstant/ felt på...Vare	Kun til-dato
VareNr	= Felt Nummer	[]
AfdelingsKode	= Felt AfdelingsAfgrænsning	[]
ProjektKode	= Felt ProjektAfgrænsning	[]
LokationsKode	= Felt LokationsAfgrænsning	[]

Kalkulationsbeskrivelse for feltet Bevægelse

Kartotek.....: VarePost
 Afgrænsning.....: VareNr=Nummer,AfdelingsKode=AfdelingsAfg
 Sumfelt.....: Antal
 Vend fortegn.....: Nej

Afgrænsning af VarePost	Konstant/ felt på...Vare	Kun til-dato
VareNr	= Felt Nummer	[]
AfdelingsKode	= Felt AfdelingsAfgrænsning	[]
ProjektKode	= Felt ProjektAfgrænsning	[]
LokationsKode	= Felt LokationsAfgrænsning	[]
Dato	= Felt DatoAfgrænsning	[]

Kalkulationsbeskrivelse for feltet Køb(antal)

Kartotek.....: VarePost
 Afgrænsning.....: Art=0,VareNr=Nummer,AfdelingsKode=Afdeli
 Sumfelt.....: VærdiAntal
 Vend fortegn.....: Nej

Afgrænsning af VarePost	Konstant/ felt på...Vare	Kun til-dato
Art	= Konstant 0	[]
VareNr	= Felt Nummer	[]
AfdelingsKode	= Felt AfdelingsAfgrænsning	[]
ProjektKode	= Felt ProjektAfgrænsning	[]
LokationsKode	= Felt LokationsAfgrænsning	[]
Dato	= Felt DatoAfgrænsning	[]

Kalkulationsbeskrivelse for feltet Salg(antal)

Kartotek.....: VarePost
 Afgrænsning.....: Art=1,VareNr=Nummer,AfdelingsKode=Afdeli
 Sumfelt.....: VærdiAntal
 Vend fortegn.....: Ja

Afgrænsning af VarePost	Konstant/ felt på...Vare	Kun til-dato
Art	= Konstant 1	[]
VareNr	= Felt Nummer	[]
AfdelingsKode	= Felt AfdelingsAfgrænsning	[]
ProjektKode	= Felt ProjektAfgrænsning	[]
LokationsKode	= Felt LokationsAfgrænsning	[]
Dato	= Felt DatoAfgrænsning	[]

Kalkulationsbeskrivelse for feltet Tilgang(antal)

Kartotek.....: VarePost
 Afgrænsning.....: Art=2,VareNr=Nummer,AfdelingsKode=Afdeli
 Sumfelt.....: VærdiAntal
 Vend fortegn.....: Nej

Afgrænsning af VarePost	Konstant/ felt på...Vare	Kun til-dato
Art	= Konstant 2	[]
VareNr	= Felt Nummer	[]
AfdelingsKode	= Felt AfdelingsAfgrænsning	[]
ProjektKode	= Felt ProjektAfgrænsning	[]
LokationsKode	= Felt LokationsAfgrænsning	[]
Dato	= Felt DatoAfgrænsning	[]

Kalkulationsbeskrivelse for feltet Afgang(antal)

Kartotek.....: VarePost
 Afgrænsning.....: Art=3,VareNr=Nummer,AfdelingsKode=Afdeli
 Sumfelt.....: VærdiAntal
 Vend fortegn.....: Ja

Afgrænsning af VarePost	Konstant/ felt på...Vare	Kun til-dato
Art	= Konstant 3	[]
VareNr	= Felt Nummer	[]
AfdelingsKode	= Felt AfdelingsAfgrænsning	[]
ProjektKode	= Felt ProjektAfgrænsning	[]
LokationsKode	= Felt LokationsAfgrænsning	[]
Dato	= Felt DatoAfgrænsning	[]

Kalkulationsbeskrivelse for feltet Køb(DKK)

Kartotek.....: VarePost
 Afgrænsning.....: Art=0,VareNr=Nummer,AfdelingsKode=Afdeli
 Sumfelt.....: Beløb
 Vend fortegn.....: Nej

Afgrænsning af VarePost	Konstant/ felt på...Vare	Kun til-dato
Art	= Konstant 0	[]
VareNr	= Felt Nummer	[]
AfdelingsKode	= Felt AfdelingsAfgrænsning	[]
ProjektKode	= Felt ProjektAfgrænsning	[]
LokationsKode	= Felt LokationsAfgrænsning	[]
Dato	= Felt DatoAfgrænsning	[]

Kalkulationsbeskrivelse for feltet Salg(DKK)

Kartotek.....: VarePost
 Afgrænsning.....: Art=1,VareNr=Nummer,AfdelingsKode=Afdeli
 Sumfelt.....: Beløb
 Vend fortegn.....: Ja

Afgrænsning af VarePost	Konstant/ felt på...Vare	Kun til-dato
Art	= Konstant 1	[]
VareNr	= Felt Nummer	[]
AfdelingsKode	= Felt AfdelingsAfgrænsning	[]
ProjektKode	= Felt ProjektAfgrænsning	[]
LokationsKode	= Felt LokationsAfgrænsning	[]
Dato	= Felt DatoAfgrænsning	[]

Kalkulationsbeskrivelse for feltet Tilgang(DKK)

Kartotek.....: VarePost
 Afgrænsning.....: Art=2,VareNr=Nummer,AfdelingsKode=Afdeli
 Sumfelt.....: Beløb
 Vend fortegn.....: Nej

Afgrænsning af VarePost	Konstant/ felt på...Vare	Kun til-dato
Art	= Konstant 2	[]
VareNr	= Felt Nummer	[]
AfdelingsKode	= Felt AfdelingsAfgrænsning	[]
ProjektKode	= Felt ProjektAfgrænsning	[]
LokationsKode	= Felt LokationsAfgrænsning	[]
Dato	= Felt DatoAfgrænsning	[]

Kalkulationsbeskrivelse for feltet Afgang(DKK)

Kartotek.....: VarePost
 Afgrænsning.....: Art=3,VareNr=Nummer,AfdelingsKode=Afdeli
 Sumfelt.....: Beløb
 Vend fortegn.....: Ja

Afgrænsning af VarePost	Konstant/ felt på...Vare	Kun til-dato
Art	= Konstant 3	[]
VareNr	= Felt Nummer	[]
AfdelingsKode	= Felt AfdelingsAfgrænsning	[]
ProjektKode	= Felt ProjektAfgrænsning	[]
LokationsKode	= Felt LokationsAfgrænsning	[]
Dato	= Felt DatoAfgrænsning	[]

Kalkulationsbeskrivelse for feltet Forbrug(DKK)

Kartotek.....: VarePost
 Afgrænsning.....: Art=1,VareNr=Nummer,AfdelingsKode=Afdeli
 Sumfelt.....: LagerVærdi(v.antal)
 Vend fortegn.....: Ja

Afgrænsning af VarePost	Konstant/ felt på...Vare	Kun til-dato
Art	= Konstant 1	[]
VareNr	= Felt Nummer	[]
AfdelingsKode	= Felt AfdelingsAfgrænsning	[]
ProjektKode	= Felt ProjektAfgrænsning	[]
LokationsKode	= Felt LokationsAfgrænsning	[]
Dato	= Felt DatoAfgrænsning	[]

Kalkulationsbeskrivelse for feltet Forv.Tilgang(antal)

Kartotek.....: KøbsLinie
 Afgrænsning.....: BilagsArt=1,Art=2,Nummer=Nummer,Afdeling
 Sumfelt.....: RestOrdre(antal)
 Vend fortegn.....: Nej

Afgrænsning af KøbsLinie	Konstant/ felt på...Vare	Kun til-dato
BilagsArt	= Konstant 1	[]
Art	= Konstant 2	[]
Nummer	= Felt Nummer	[]
AfdelingsKode	= Felt AfdelingsAfgrænsning	[]
ProjektKode	= Felt ProjektAfgrænsning	[]
LokationsKode	= Felt LokationsAfgrænsning	[]
Lev.Dato	= Felt DatoAfgrænsning	[]

Kalkulationsbeskrivelse for feltet Forv.Afgang(antal)

Kartotek.....: SalgsLinie
 Afgrænsning.....: BilagsArt=1,Art=2,Nummer=Nummer,Afdeling
 Sumfelt.....: RestOrdre(antal)
 Vend fortegn.....: Nej

Afgrænsning af SalgsLinie	Konstant/ felt på....Vare	Kun til-dato
BilagsArt	= Konstant	1 []
Art	= Konstant	2 []
Nummer	= Felt	Nummer []
AfdelingsKode	= Felt	AfdelingsAfgrænsning []
ProjektKode	= Felt	ProjektAfgrænsning []
LokationsKode	= Felt	LokationsAfgrænsning []
Lev.Dato	= Felt	DatoAfgrænsning []

Nøgler

A Nøgle	Grupper	Sumfelter
-		
+ Nummer		
+ SøgeNavn		
+ Bogf.Gr		
+ Placering		
+ LeverandørKode		
+ Tilhører VareGr.Nr		

Oplysninger

Feltoverf.

6.8 Efterbehandlingskode

Udskrift af efterbehandlingskode for kartoteket SalgsLinie:

Efterbeh.kode 37 PCC7,AA	Rettet d. 04-12-94 8.26.31
-----------------------------	----------------------------

Interne var.

B	Navn	Type	Kartotek	Læn	Dim1	Dim2	Dim3
	Hoved	Record	SalgsHoved				
	RegnskabsData	Record	RegnskabsData				
	FinansKonto	Record	FinansKonto				
	Vare	Record	Vare				
	VareTekst	Record	VareTekst				
	VareBogfGr	Record	VareBogf.Gr				
	DebBogfGr	Record	DebitorBogf.Gr				
	DebVareRabat	Record	Deb.VareRabat				
	MængdeRabat	Record	VareSalgsMængde				
	Moms	Record	Moms				
	VareSalgsPris	Record	VareSalgsPris				
	VP	Record	VarePost				
	Ressource	Record	Ressource				
	ResBogfGr	Record	Res.Bogf.Gr				
	ResSalgsPris	Record	Res.SalgsPris				
	ResKostPris	Record	Res.KostPris				
	ArbejdsType	Record	ArbejdsType				
	xArt	Heltal					
	xNummer	Kode		20			
	PrisIValuta	JaNej					
	PrisInklMoms	JaNej					
	SagsPost	Record	SagsPost				
	VareGruppe	Record	VareGruppe				

Beskrivelse

OPRET	SalgsLinie
-------	------------

```
dbLOCKFILE(SalgsLinie);
Hoved.Nummer := '';
HentHoved;
```

SLET	SalgsLinie
------	------------

```
dbTESTFIELD("Lev.EjFak.(antal)",0);
```

SPECIALRELATION	SalgsLinie
-----------------	------------

```
IF (FeltNr = dbFIELDNO(UdalignLøbeNr)) OR (FeltNr = dbFIELDNO(SerieNr)) THEN
  IF Art=2 THEN BEGIN {VareLinie}
    dbSELECTKEY(VP.VareNr,VP.Åben);
    dbSETRANGE(VP.VareNr,Nummer);
    dbSETRANGE(VP.Åben,TRUE);
    IF callWINDOW(63,VP) THEN BEGIN
      IF FeltNr = dbFIELDNO(UdalignLøbeNr) THEN
        UdalignLøbeNr := VP.LøbeNr;
      LokationsKode := VP.LokationsKode;
      SerieNr := VP.SerieNr;
    END;
  END;

IF (FeltNr = dbFIELDNO(SagsPostLøbeNr)) OR (FeltNr = dbFIELDNO(SagsUdaligningsID)) THEN BEGIN
  dbSELECTKEY(SagsPost.SagsNr,SagsPost.Åben,SagsPost.Positiv,SagsPost.Dato);
  dbSETRANGE(SagsPost.SagsNr,SagsNr);
  dbSETRANGE(SagsPost.Åben,TRUE);
```

```

IF NettoBeløb <> 0 THEN
  IF BilagsArt = 3 THEN {Kreditnota}
    dbSETRANGE(SagsPost.Positiv,NettoBeløb < 0)
  ELSE
    dbSETRANGE(SagsPost.Positiv,NettoBeløb > 0);
  IF callWINDOW(92,SagsPost) AND (FeltNr = dbFIELDNO(SagsPostLøbeNr)) THEN
    SagsPostLøbeNr := SagsPost.LøbeNr;
END;

```

Art	SalgsLinie
-----	------------

```

dbTESTFIELD("Lev.EjFak.(antal)",0);
xArt := Art;
dbINITREC(SalgsLinie);
Art := xArt;

```

Nummer	SalgsLinie
--------	------------

```

dbTESTFIELD("Lev.EjFak.(antal)",0);
IF Nummer = '' THEN EXIT;

```

```

xArt := Art;
xNummer := Nummer;
dbINITREC(SalgsLinie);
Art := xArt;
Nummer := xNummer;

```

```

HentHoved;
dbTRANSFERFIELDS(Hoved,SalgsLinie);

```

```

dbTESTFIELD(Art);

```

```

CASE Art OF

```

```

  1 : BEGIN {Konto}
    dbGETREC(FinansKonto,Nummer);
    dbTESTFIELD(FinansKonto.Spærret,FALSE);
    dbTESTFIELD(FinansKonto.KontoArt,0);
    dbCALCFIELDS(FinansKonto.Bemærkning);
    IF FinansKonto.Bemærkning THEN
      MESSAGE('Der er bemærkning på konto %1',FinansKonto.Nummer);
    dbTRANSFERFIELDS(FinansKonto,SalgsLinie);
  END;

```

```

  2 : BEGIN {Vare}
    dbGETREC(Vare,Nummer);
    dbTESTFIELD(Vare.Spærret,FALSE);
    dbTESTFIELD(Vare."Bogf.Gr");
    dbCALCFIELDS(Vare.Bemærkning);
    IF Vare.Bemærkning THEN
      MESSAGE('Der er bemærkning på vare %1',Vare.Nummer);
    dbTRANSFERFIELDS(Vare,SalgsLinie);

```

```

  IF Hoved.SprogKode<>' ' THEN
    IF dbGETREC(VareTekst,Vare.Nummer,Hoved.SprogKode) THEN
      dbTRANSFERFIELDS(VareTekst,SalgsLinie);

```

```

  IF MomsKode = '' THEN BEGIN
    dbGETREC(RegnskabsData);
    IF RegnskabsData.VareSalgsBogføring = 1 THEN BEGIN
      dbGETREC(DebBogfGr,Hoved."Bogf.Gr");
      dbTESTFIELD(DebBogfGr.VareSalgsKonto);
      dbGETREC(FinansKonto,DebBogfGr.VareSalgsKonto);
    END ELSE BEGIN
      dbGETREC(VareBogfGr,"Bogf.Gr");
      dbTESTFIELD(VareBogfGr.VareSalgsKonto);
      dbGETREC(FinansKonto,VareBogfGr.VareSalgsKonto);
    END;
    MomsKode := FinansKonto.MomsKode;
  END;

```

```

  IF dbGETREC(DebVareRabat,Hoved."Deb.VareRabatGr",Vare."VareDeb.RabatGr") THEN
    "Deb.VareRabat%" := DebVareRabat."Rabat%";

```

```

  END;

```

```

  3 : BEGIN {Ressource}
    dbGETREC(Ressource,Nummer);
    dbTESTFIELD(Ressource.Spærret,FALSE);
    dbTESTFIELD(Ressource."Bogf.Gr");
    dbCALCFIELDS(Ressource.Bemærkning);
    IF Ressource.Bemærkning THEN
      MESSAGE('Der er bemærkning på ressource %1',Ressource.Nummer);
    dbTRANSFERFIELDS(Ressource,SalgsLinie);

```

```

HentResKostPris;

IF MomsKode = '' THEN BEGIN
  dbGETREC(RegnskabsData);
  IF RegnskabsData.VareSalgsBogføring = 1 THEN BEGIN
    dbGETREC(DebBogfGr,Hoved."Bogf.Gr");
    dbTESTFIELD(DebBogfGr.VareSalgsKonto);
    dbGETREC(FinansKonto,DebBogfGr.VareSalgsKonto);
  END ELSE BEGIN
    dbGETREC(ResBogfGr,"Bogf.Gr");
    dbTESTFIELD(ResBogfGr."Res.SalgsKonto");
    dbGETREC(FinansKonto,ResBogfGr."Res.SalgsKonto");
  END;
  MomsKode := FinansKonto.MomsKode;
END;
END;
4 : BEGIN {Varegruppe}
  dbGETREC(VareGruppe,Nummer);
  dbTESTFIELD(VareGruppe.Spærret,FALSE);
  dbTESTFIELD(VareGruppe."Bogf.Gr");
  dbCALCFIELDS(VareGruppe.Bemærkning);
  IF VareGruppe.Bemærkning THEN
    MESSAGE('Der er bemærkning på varegruppe %1',VareGruppe.Nummer);
  dbTRANSFERFIELDS(VareGruppe,SalgsLinie);

  CASE VareGruppe.SalgsAvanceValg OF
    0: "Avance%" := VareGruppe."Avance%(fast)";
    1: "Avance%" := VareGruppe."Avance%(sidste)";
    2: "Avance%" := VareGruppe."Avance%(gns.)";
  END;

  IF MomsKode = '' THEN BEGIN
    dbGETREC(RegnskabsData);
    IF RegnskabsData.VareSalgsBogføring = 1 THEN BEGIN
      dbGETREC(DebBogfGr,Hoved."Bogf.Gr");
      dbTESTFIELD(DebBogfGr.VareSalgsKonto);
      dbGETREC(FinansKonto,DebBogfGr.VareSalgsKonto);
    END ELSE BEGIN
      dbGETREC(VareBogfGr,"Bogf.Gr");
      dbTESTFIELD(VareBogfGr.VareSalgsKonto);
      dbGETREC(FinansKonto,VareBogfGr.VareSalgsKonto);
    END;
    MomsKode := FinansKonto.MomsKode;
  END;

  IF dbGETREC(DebVareRabat,Hoved."Deb.VareRabatGr",VareGruppe."VareDeb.RabatGr") THEN
    "Deb.VareRabat%" := DebVareRabat."Rabat%";
END;

END;

dbCALLFIELDPCODE(MomsKode);

IF (Art = 2) OR (Art = 3) THEN {VareLinie,ResLinie}
  HentApris;
dbCALLFIELDPCODE(Antal,xSalgsLinie.Antal);

```

Antal	SalgsLinie
-------	------------

```

IF (Antal > 10) AND (Antal*Apris > 100) THEN
  "Rabat%" := 5
ELSE
  "Rabat%" := 0;
dbCALLFIELDPCODE("Rabat%");

```

tilføjet kode

Faktureres	SalgsLinie
------------	------------

```

IF ((Faktureres*Antal) < 0) OR (ABS(Faktureres) > ABS("Leveret(antal)+"Leveres-"Faktureret(antal)")) THEN
  ERROR('Der kan max. faktureres %1',"Leveret(antal)+"Leveres-"Faktureret(antal)");

```

Leveres	SalgsLinie
---------	------------

```

IF ((Leveres*Antal) < 0) OR (ABS(Leveres) > ABS("RestOrdre(antal)")) THEN
  ERROR('Der kan max. leveres %1',"RestOrdre(antal)");

```

```
Faktureres := "Leveret(antal)+"Leveres-"Faktureret(antal)";
```

Apris	SalgsLinie
-------	------------

```
IF (Antal > 10) AND (Antal*Apris > 100) THEN
  "Rabat%" := 5
ELSE
  "Rabat%" := 0;
```

tilføjet kode

```
dbCALLFIELDPCODE("Rabat%");
```

MomsKode	SalgsLinie
----------	------------

```
IF MomsKode <> '' THEN BEGIN
  dbGETREC(Moms,MomsKode);
  dbTESTFIELD(Moms.Art,1);
  "Moms%" := Moms."Moms%";
  HentHoved;
  IF NOT Hoved.MomsPligtig THEN
    "Moms%" := 0;
END ELSE
  "Moms%" := 0;
BeregnNettoBrutto();
```

MængdeRabat%	SalgsLinie
--------------	------------

```
dbGETREC(RegnskabsData);
IF RegnskabsData.LinieRabatBeregning = 0 THEN {K/V+Mængde}
  "Rabat%" := "Deb.VareRabat%"+MængdeRabat%
ELSE
  "Rabat%" := ROUND(100-(100*(1-"Deb.VareRabat%"/100)*(1-"MængdeRabat%"/100)));
IF "Rabat%" > 100 THEN
  "Rabat%" := 100;
dbCALLFIELDPCODE("Rabat%");
```

Rabat%	SalgsLinie
--------	------------

```
RabatBeløb := ROUND(Antal*Apris*"Rabat%"/100);
BeregnNettoBrutto();
```

RabatBeløb	SalgsLinie
------------	------------

```
dbTESTFIELD(Antal);
BeregnNettoBrutto();
```

NettoBeløb	SalgsLinie
------------	------------

```
dbTESTFIELD(Art);
dbTESTFIELD(Antal);
BruttoBeløb := ROUND(NettoBeløb*(1+"Moms%"/100),0.00001);
HentHoved();
IF Hoved.BruttoPriser THEN
  RabatBeløb := ROUND(Antal*Apris)-BruttoBeløb
ELSE
  RabatBeløb := ROUND(Antal*Apris)-NettoBeløb;
"RestOrdre(beløb)" := ROUND(BruttoBeløb * "RestOrdre(antal)"/Antal);
"Lev.EjFak.(beløb)" := ROUND(BruttoBeløb * "Lev.EjFak.(antal)"/Antal);
```

BruttoBeløb	SalgsLinie
-------------	------------

```
dbTESTFIELD(Art);
dbTESTFIELD(Antal);
NettoBeløb := ROUND(BruttoBeløb/(1+"Moms%"/100),0.00001);
HentHoved();
IF Hoved.BruttoPriser THEN
  RabatBeløb := ROUND(Antal*Apris)-BruttoBeløb
ELSE
  RabatBeløb := ROUND(Antal*Apris)-NettoBeløb;
"RestOrdre(beløb)" := ROUND(BruttoBeløb * "RestOrdre(antal)"/Antal);
"Lev.EjFak.(beløb)" := ROUND(BruttoBeløb * "Lev.EjFak.(antal)"/Antal);
```

PrisGruppeNr	SalgsLinie
--------------	------------

```
IF Art = 2 THEN BEGIN {VareLinie}
```

```
HentApris;
dbCALLFIELDPCODE(Antal);
END;
```

ArbejdsTypeKode	SalgsLinie
-----------------	------------

```
IF Art = 3 THEN BEGIN
HentApris;
dbCALLFIELDPCODE(Apris);
HentResKostPris;
IF dbGETREC(ArbejdsType,ArbejdsTypeKode) THEN
  Enhed := ArbejdsType.EnhedsKode;
END;
```

Deb.VareRabat%	SalgsLinie
----------------	------------

```
dbCALLFIELDPCODE("MængdeRabat%");
```

HentHoved()	SalgsLinie
-------------	------------

```
dbTESTFIELD(BilagsNr);
IF Hoved.Nummer <> BilagsNr THEN
  dbGETREC(Hoved,BilagsArt,BilagsNr);
```

HentApris()	SalgsLinie
-------------	------------

```
HentHoved;
PrisIValuta := FALSE;
PrisInklMoms := FALSE;
```

```
CASE Art OF
  2 : BEGIN
    dbINITREC(VareSalgsPris);
    VareSalgsPris.VareNr      := Nummer;
    VareSalgsPris.PrisGruppeNr := PrisGruppeNr;
    VareSalgsPris.ValutaKode  := Hoved.ValutaKode;
    IF (BilagsArt = 0) OR (BilagsArt = 1) THEN {Tilbud,Ordre}
      VareSalgsPris.Dato      := Hoved.OrdreDato
    ELSE
      VareSalgsPris.Dato      := Hoved."Bogf.Dato";
    callFUNCTION(52,VareSalgsPris); {Find VareSalgsPris}
    Apris := VareSalgsPris.SalgsPris;
    "Beregn M.Rabat" := VareSalgsPris."Beregn M.Rabat";
    "Beregn Fak.Rabat" := VareSalgsPris."Beregn F.Rabat";
    IF NOT VareSalgsPris."Beregn D.Rabat" THEN
      "Deb.VareRabat%" := 0;
    PrisIValuta := VareSalgsPris.ValutaKode <> '';
    PrisInklMoms := VareSalgsPris."Inkl.Moms";
  END;
  3 : BEGIN
    dbINITREC(ResSalgsPris);
    ResSalgsPris.SagsNr := SagsNr;
    ResSalgsPris.Kode   := Nummer;
    ResSalgsPris.ArbejdsTypeKode := ArbejdsTypeKode;
    callFUNCTION(221,ResSalgsPris);
    Apris := ResSalgsPris.SalgsPris;
  END;
END;
```

```
IF (Hoved.ValutaKode <> '') AND (NOT PrisIValuta) THEN BEGIN
  dbTESTFIELD(Hoved.ValutaKurs);
  Apris := Apris / (Hoved.ValutaKurs / 100);
END;
```

```
IF Hoved.BruttoPriser <> PrisInklMoms THEN
  IF Hoved.MomsPligtig THEN
    IF Hoved.BruttoPriser THEN
      Apris := Apris * (1 + "Moms%" / 100)
    ELSE
      Apris := Apris / (1 + "Moms%" / 100)
  ELSE
    IF MomsKode <> '' THEN BEGIN
      dbGETREC(Moms,MomsKode);
      IF Hoved.BruttoPriser THEN
        Apris := Apris * (1 + Moms."Moms%" / 100)
      ELSE
        Apris := Apris / (1 + Moms."Moms%" / 100);
    END;
```

```
Apris := ROUND(Apris,0.00001);
```

```
HentResKostPris()
```

```
SalgsLinie
```

```
dbINITREC(ResKostPris);  
ResKostPris.Kode := Nummer;  
ResKostPris.ArbejdsTypeKode := ArbejdsTypeKode;  
callFUNCTION(220,ResKostPris);  
KostPris := ResKostPris.KostPris;
```

```
BeregnNettoBrutto()
```

```
SalgsLinie
```

```
dbTESTFIELD(Art);  
HentHoved;  
NettoBeløb := ROUND(Antal*Apris)-RabatBeløb;  
BruttoBeløb := NettoBeløb;  
IF Hoved.BruttoPriser THEN  
  NettoBeløb := ROUND(BruttoBeløb/(1+"Moms%"/100),0.00001)  
ELSE  
  BruttoBeløb := ROUND(NettoBeløb*(1+"Moms%"/100),0.00001);  
IF Antal <> 0 THEN BEGIN  
  "RestOrdre(beløb)" := ROUND(BruttoBeløb * "RestOrdre(antal)"/Antal);  
  "Lev.EjFak.(beløb)" := ROUND(BruttoBeløb * "Lev.EjFak.(antal)"/Antal);  
END ELSE BEGIN  
  "RestOrdre(beløb)" := 0;  
  "Lev.EjFak.(beløb)" := 0;  
END;
```