



APPENDIKS: DATABASEMODELLERING

I denne boken bruker vi entitet-relasjon-diagrammer (ER-diagrammer) og tabelldefinisjoner for å dokumentere en datamodell.

En *entitet* er en enhet du ønsker å lagre informasjon om i en database. Dette kan være en person, en vare, et dokument eller lignende. En samling like entiteter kalles en *entitetstype*. En samling personer som er kunder, vil det være naturlig å samle i entitetstypen Kunder, der entitetstypen skal beskrive hvilke egenskaper, også kalt *attributter*, vi er interessert i å lagre for hver entitet.

Tabeller er sentralt i relasjonsdatabaser. En tabell representerer en entitetstype, for eksempel Kunde, og *tabelloverskrifter* beskriver egenskapene, eller attributtene, ved entitetstypen. Hver ny rad som legges inn tabellen, representerer en entitet. I tabellen nedenfor ser vi at hver ny rad som legges inn, representerer en kunde.

Tabell A.6.1: Kundetabell.

Tabell: Kunde			
<u>Kundenummer</u>	Etternavn	Fornavn	Telefon
1234	Ås	Peder	55 33 55 44

Relasjonsdatabaser består av flere tabeller, og informasjonen ligger både i tabellene og i *koblinger* mellom tabellene. For eksempel vil vi ha en tabell med oversikt over alle produktene vi skal selge, og en tabell over kundene våre. Når vi kobler en kunde sammen med et produkt, representerer dette informasjon om at en transaksjon har blitt gjennomført - kunden har enten bestilt eller kjøpt varer. Vi trenger da en måte å *entydig identifisere* både en kundeføst og en vareføst på, og vi trenger en metode å *koble dem* sammen på.

Måten vi identifiserer en rad i en tabell på, er å bruke noe vi kaller *nøkkel* for å identifisere raden. For å sikre at rett vare blir koblet til rett kunde, må hver av radene ha en unik nøkkel. Denne typen nøkler kaller vi *primærnøkler*.



A2 Digital forretningsforståelse

Primærnøkkelen kan være en egenskap ved posten, for eksempel er et personnummer *unikt* for hver person. Det kan også være en *kombinasjon av egenskaper* som blir unik for raden, for eksempel en kombinasjon av et fakturanummer og et kundenummer, eller – som i tilfellet ovenfor med kundenummer – noe som er opprettet spesielt for å identifisere entiteten.

Vi merker primærnøkkelen i tabellen ved å understreke den:

Tabell A.6.2: Kundetabell med primærnøkkel.

Tabell: Kunde			
<u>Kundenummer</u>	Etternavn	Fornavn	Telefon
1234	Ås	Peder	55 33 55 44

På samme måten kan vi sette opp en tabell for varer, der varenummer er primærnøkkelen:

Tabell A.6.3: Varetabell.

Tabell: Varer			
<u>Varenummer</u>	Varenavn	Leverandør	Varetype
4321	Tie Guan Yin	Linda Lin	Oolong te
4421	Long Jing	Kitty Qing	Grønn te

For å registrere et kjøp i databasen må vi koble sammen et kundenummer med et varenummer slik at vi ser at kunde nummer 1234 har kjøpt varenummer 4321. En kunde kan kjøpe et eller flere produkter i ønsket antall. Før vi ser mer på denne koblingen, skal vi se litt mer på varetabelen.

En av egenskapene ved varetabelen heter Varetype. Vi må anta at formålet med å angi varetype er at det er flere varer som inngår i en varetype; hvis ikke blir varetypen en egenskap som er overflødig.

Når flere skal registrere data i en database, er det fare for at noen legger inn feil navn på en varetype. I databasen vil for eksempel Oolong te, oolong-te og oolongte bli oppfattet som ulike varetyper selv om de som registrerte varen, har ment å gi den samme varetypen til de registrerte varene. Dette kan løses ved å lage en egen tabell med oversikt over varetypene, koble den sammen med tabellen for varer og sørge for at varetype hentes fra varetypetabellen og inn til varetabelen.

Vi antar her at en vare kun kan tilhøre én varetype. Vi kan da beskrive koblingen mellom tabellene på denne måten: *En vare kan bare tilhøre én varetype, og til hver varetype kan det tilhøre mange varer.*

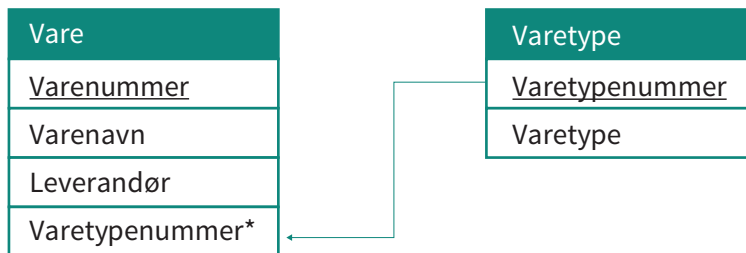


Nedenfor vises tabellen for varetyper. I denne tabellen er Varetypenummer primærnøkkelen.

Tabell A.6.4: Varetypetabell.

Tabell: Varetype	
<u>Varetypenummer</u>	Varetype
101	Hvit te
102	Grønn te
103	Oolongte
104	Svart te
105	Puerte

For å vise hvordan nøklene kobler sammen tabellene, setter vi opp en *relasjonsmodell*. Denne modellen viser entitetstypene og koblinger mellom entitetene.



Figur A.6.1: Varetabell koblet med varetypetabell.

I modellen ovenfor kobles varetabellen med varetypetabellen ved å hente primærnøkkelen inn som en egenskap i varetabellen. En primærnøkkel som brukes til å koble sammen to tabeller, kalles *fremmednøkkel* i tabellen den kobles sammen med.

Tabelldefinisjoner

I en tabelldefinisjon skriver en først navnet på tabellen (entitetstypen) og deretter egenskapene til entitetstypen. Primærnøkler understrekes, og fremmednøkler merkes med *. Tabelldefinisjonene på de tre tabellene i eksemplet ovenfor vil se slik ut:

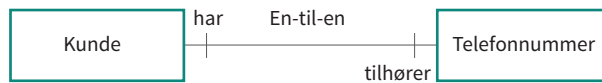
- Kunde (Kundenummer, Etternavn, Fornavn, Telefon)
- Vare (Varenummer, Varenavn, Leverandør, Varetypenummer*)
- Varetype (Varetypenummer, Varetype)



Koblinger – kardinalitet

I ER-diagrammer er det tre ting som er viktig: *entitetstyper, koblinger mellom entitetstypene og egenskaper ved koblingene*. I eksemplet ovenfor har vi sett på hvordan tabeller representerer entitetstyper, og vi koblet sammen tabellen for varer og tabellen for varetyper ved hjelp av nøkler. Vi beskrev koblingen mellom varer og varetyper med ord. Det går også an å beskrive koblingene med symboler. Det fins tre ulike typer koblinger som kan eksistere mellom tabeller, og disse beskrives med ulike symboler. Koblingene er beskrevet i teksten nedenfor.

- *En-til-en-kobling*: Dette betyr at det bare kan eksistere én tilsvarende verdi i de relaterte tabellene. For eksempel kan en kunde bare *ha* ett telefonnummer, og hvert telefonnummer kan bare *tilhøre* én kunde. En-til-en-relasjoner er unødvendige, fordi en da heller kan legge inn verdien (her telefonnummeret) som en egenskap i tabellen. I figuren nedenfor er det en strek nær begge entitetstypene; altså kan en kunde bare ha ett telefonnummer, og hvert telefonnummer kan bare tilhøre én kunde.

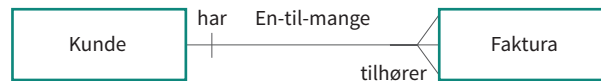


Figur A.6.2: En-til-en-kobling.

Dersom vi likevel velger å bruke to tabeller her, vil de kobles sammen med nøkler som vist ovenfor. For eksempel vil kunden ha et kundenummer eller en kunde-ID som primærnøkkel, og telefonnummeret vil være primærnøkkel i telefonnummertabellen. Telefonnummer kan da brukes som fremmednøkkel i kundetabellen. Når en primærnøkkel i en tabell blir fremmednøkkel i en annen tabell, kaller vi dette *nøkkelvandring*.

- *En-til-mange-kobling*: Dette betyr at mange entiteter i en tabell kan være koblet til en entitet i den sammenkoblede tabellen og omvendt. For eksempel kan en kunde være mottaker av *mange* fakturaer, men hver faktura kan bare *tilhøre* én kunde. Denne typen koblinger er den vanligste i relasjonsdatabaser. Nedenfor ser vi at en strek representerer én, og at en *kråkefot* (tre streker) representerer mange.

Her vil kundenummer være primærnøkkel i kundetabellen, og fakturanummer vil være primærnøkkel i fakturatabellen. Her vil primærnøkkel fra kundetabellen, som er én-siden i koblingen, gå inn som fremmednøkkel i fakturatabellen, som er



Figur A.6.3: En-til-mange-kobling.

på mange-siden av koblingen. Primærnøkkelen i kundetabellen vandrer til fakturatabellen. Slik vil hver faktura være koblet til én kunde.

- *Mange-til-mange-kobling:* Mange entiteter i en tabell kan høre til mange entiteter i den relaterte tabellen. Eksempel: Et produkt kan tilhøre mange fakturaer, og hver faktura kan inneholde mange produkter. Dette representeres med kråkefot på begge sidene av koblingen i modellen nedenfor.



Figur A.6.4: Mange-til-mange-kobling.

Mange-til-mange-koblinger kan ikke brukes i databaser. Dette løses ved å opprette en tredje tabell som kobler sammen de to opprinnelige tabellene. I *koblingstabellen* brukes en *kombinasjon* av to fremmednøkler som primærnøkler. Dette går an fordi kombinasjonen, i dette tilfellet mellom fakturanummer, som er primærnøkkel i fakturatabellen, og produktnummer, som er primærnøkkel i produkttabellen, vil være unik. Neste gang produktet selges til en kunde som får faktura, vil fakturanummeret være nytt, og kombinasjonen av disse to fremmednøklerne vil utgjøre en unik kombinasjon. Tabellen kan vi kalle Fakturalinje fordi hver post i tabellen vil representere en linje på fakturaen. Tabelldefinisjonene kan da se slik ut:

- Faktura (Fakturanummer, Kunde-ID*, dato)
- Produkt (Produktnummer, Produktnavn, Leverandør, Pris)
- Fakturalinje (*Produktnummer, *Fakturanummer, Antall)

Koblinger – medlemskap

I tillegg har vi også notasjon for medlemskap. Denne notasjonen kommer i tillegg til notasjonen for kardinalitet. En *strek indikerer at koblingen er obligatorisk*, mens en



runding indikerer at koblingen er ikke-obligatorisk. I figuren under er dette eksemplifisert med relasjonen mellom en person og en bil:



Figur A.6.5: En-til-mange-kobling med angivelser for kardinalitet og medlemskap.

Notasjonen *nærmest entitetene* representerer kardinalitet. I dette tilfellet betyr det at en person kan eie mange biler, mens hver bil bare kan ha én eier. Det neste settet med symboler sier noe om *medlemskapet*. Streken (strek nummer to) nærmest entiteten Person sier at det *ikke* er obligatorisk for en person å eie en bil (ikke-obligatorisk medlemskap), mens sirkelen ved kråkefot ved entiteten Bil sier at hver bil *må* ha én eier (obligatorisk eierskap). I dette tilfellet betyr notasjonen altså at det ikke er obligatorisk for en person å eie en bil, og at det er obligatorisk for en bil å ha én eier.

Normalisering av databaser

For å unngå dobbeltlagring av data *normaliserer* man databasen. Denne prosessen gjør at tabellene blir i normalform. Normaliseringen skal sikre at samme data ikke er lagret flere plasser. Dersom telefonnummeret til en person er lagret flere plasser, og kun en av plassene blir oppdatert, kan en ikke vite hvilket nummer som er det riktige, og dataene har da mistet sin *integritet* eller troverdighet. Det fins fem grader av normalisering, men grad 4 og 5 er kun av teoretisk interesse, så vi vil bare gå gjennom de første tre.

Første normalform (1NF)

Data i databaser er lagret i tabeller, som vist nedenfor. Kravet til første normalform er at alle feltene i tabellen er *atomære*, det vil si at det kan bare være én verdi i hvert felt i tabellen. I eksemplet nedenfor er det to navn i feltet Navn, og dermed bryter det med kravet til første normalform. I tillegg er det både adresse, postnummer og poststed i tabellen; dette bryter også med første normalform.



Tabell A.6.5: Kundetabell, ikke normalisert.

Tabell: Kunder	
Navn	Adresse
Per Gullanger	Fjellet 12, 5195 Hjelmås

Ved å dele opp feltene får vi en tabell som tilfredsstill 1NF.

Tabell A.6.6: Kundetabell normalisert i forhold til én verdi i hvert felt.

Tabell: Kunder				
Etternavn	Fornavn	Adresse	Postnummer	Poststed
Gullanger	Per	Fjellet 12	5195	Hjelmås

Flere kunder kan ha samme navn og samme adresse. 1NF krever at hver rad i tabellen er unikt identifisert, og dette løses ved å gi hver rad et unikt nummer som identifiserer raden. Vi oppretter derfor Kundenummer som primærnøkkel i tabellen.

Tabell A.6.7: Kundetabell normalisert etter 1NF.

Tabell: Kunder					
<u>Kundenummer</u>	Etternavn	Fornavn	Adresse	Postnummer	Poststed
5164	Gullanger	Per	Fjellet 12	5195	Hjelmås

Andre normalform (2NF)

Normalformene bygger på hverandre, så det første kravet for 2NF er at kravene til 1NF er oppfylt. Dernest krever den at alle dataene i tabellen er direkte avhengig av primærnøkkelen i tabellen. Tabellen nedenfor tilfredsstill ikke kravene til 2NF. Det er fordi navn er avhengig av kundenummer, men ikke av firmakode. På samme måten er firma avhengig av firmakode, men ikke av kundenummer.

Tabell A.6.8: Kundetabell ikke normalisert etter 2NF.

Tabell: Kunder							
<u>Kunde- nummer</u>	<u>Firma- kode</u>	Firma	Etternavn	Fornavn	Adresse	Postnummer	Poststed
5164	81	Leknes skule	Gullanger	Per	Fjellet 12	5195	Hjelmås

Vi kan tenke oss at denne kunden bytter jobb, men likevel forblir kunde. En tabell som kobler sammen kunde med firma, gir informasjon om tilhørighet til firmaet. For å normalisere denne tabellen må man skille den i tre tabeller: en tabell som be-



A8 Digital forretningsforståelse

skriver kunde, en tabell som beskriver firma, og en tabell som kobler sammen firma og kunde.

Kunde

Tabell A.6.9: Kundetabell 2NF.

Tabell: Kunder					
<u>Kundenummer</u>	Etternavn	Fornavn	Adresse	Postnummer	Poststed
5164	Gullanger	Per	Fjellet 12	5195	Hjelmås

Firma

Tabell A.6.10: Firmatabell 2NF.

Tabell: Firma				
<u>Firmakode</u>	Firmanavn	Adresse	Postnummer	Poststed
81	Leknes skule	Skulevegen	5195	Hjelmås

Firmatilhørighet

Tabell A.6.11: Firmatilhørighetstabell 2NF.

Tabell: Firmatilhørighet			
<u>Kundenummer*</u>	<u>Firmakode*</u>	Ansatt fra	Ansatt til
5164	81	12.07.1985	

Her kan vi legge merke til at primærnøkklene i kundetabellen og firmatabellen inngår i tabellen for firmatilhørighet. Når vi bruker primærnøkler til å koble sammen tabeller på denne måten, sier vi at primærnøkkelen i parti går inn som en *fremmednøkkel* i tabellen firmatilhørighet. En annen ting å merke seg her at de to fremmednøkklene i firmatilhørighetstabellen til sammen utgjør primærnøkkelen til tabellen. Her vil kombinasjonen av nøklene være unik.

Tredje normalform (3NF)

Den tredje normalformen krever, i tillegg til at 2NF er tilfredsstillt, at alle attributter som ikke er nøkler, er innbyrdes uavhengige av hverandre. I tabellen nedenfor er poststed avhengig av postnummer, og ikke bare av primærnøkkelen i tabellen.



Tabell A.6.12: Kundetabell, ikke normalisert etter 3NF.

Tabell: Kunder					
<u>Kundenummer</u>	Etternavn	Fornavn	Adresse	Postnummer	Poststed
5164	Gullanger	Per	Fjellet 12	5195	Hjelmås

Vi kan derfor ta poststed ut av tabellen og lage en egen tabell med bare postnummer og poststed, som vist nedenfor. Formålet er å lage databasen mer effektiv, med lite overflødige data, eller på fagspråket: å unngå dataredundans. Et eksempel med tanke på tabellen ovenfor er å ha en egen tabell med postnummer og poststed, se nedenfor:

Tabell A.6.13: Postnummertabell 3NF.

Postnr.	Poststed
5915	Hjelmås
5916	Isdalstø
5917	Rosslund
5918	Frekhaug
...	...

Å lage en kobling mellom entiteten Kunde og entiteten Poststed vil være effektivt av følgende grunner:


- Man slipper å skrive inn det samme poststedet flere ganger dersom en har flere kunder som bor på samme poststed. Poststedet vil i stedet komme opp automatisk når man skriver inn postnummer.
- Ettersom postnummeret er koblet direkte til poststedet, unngår man at feil kombinasjon av postnummer/poststed blir skrevet inn manuelt i databasen. Man unngår også at poststed skrives feil i en eller flere rader.
- Poststed lagres bare én plass, databasen vil få mindre tekst og blir derfor enklere å håndtere.

Eksempel: Normalisering av fakturadatabase

Mange-til-mange-relasjoner lar seg ikke implementere i databaser. Ved å normalisere databasen får man effektiv lagring av data, man fjerner uheldige avhengigheter mellom dataene, og som et resultat av normaliseringsprosessen blir flere tabeller opprettet. De opprettede tabellene løser opp mange-til-mange-koblingene slik at vi står igjen med tabeller med en-til-mange-koblinger i stedet.



A10 Digital forretningsforståelse

 TEDRAGEN				
Tedragen AS Bøkkersmauet 1 5032 BERGEN		Org. Nr.: 912 684 989 Bank: 3633 38 18234	Dato Faktura Betalingsfrist 14 dager	15-Jun-2017 123717
Børs Cafe Strandkaien 12 5000 Bergen		Org. Nr.:	Dykkar referanse: Knut Nesbø	
	Prod.nr.	Pris	Antall	Total
Da Hong Pao - Stor raud kappe - Keisarleg Oolong-te - 50 g	303	200	7	1,400.00 kr
Tekanne i borosilikatglas 3 dl	1003	300	15	4,500.00 kr
Bai Hao Yinzhen - Sølvnål - Keisarleg kvit te - 50 g	101	180	22	3,960.00 kr
Dian Hong - Gyldne knoppar - Keisarleg svart te - 50 g	401	150	20	3,000.00 kr
Sum				12,860.00 kr
15 % mva på te				1,254.00 kr
25 % mva på brukarutstyr og frakt				1,125.00 kr
Totalt mva				2,379.00 kr
Sum			64	15,239.00 kr

Figur A.6.6: Faktura.¹

I forklaringen ovenfor har vi sett at det er en mange-til-mange-relasjon mellom faktura og varer, og vi har sett på hvordan tabellen fakturalinje kobler sammen fakturatabellen og varetabellen. En faktura består av et fakturahode og varelinjer, se figur nedenfor.

Fakturahodet inneholder info om kunde, dato, fakturanummer og selvsagt selger. I tillegg inneholder fakturaen en linje for hver vare som er lagt til fakturaen. Relasjonen mellom fakturalinjer, produkt og faktura blir slik:

¹ Bokens forfatter er medeier i firmaet Tedragen.



- En fakturalinje vil inneholde et produktnummer, et produktnavn og antall. Hver fakturalinje kan kun være med på én faktura, men hver faktura kan ha mange fakturalinjer.
- Et produkt kan være med på mange fakturalinjer (samme produkttype), men hver fakturalinje kan bare inneholde ett produkt.

Mange-til-mange-relasjonen mellom faktura og produkt er nå løst opp eller normalisert, og vi har fått entiteten Fakturalinje:

- Fakturalinje (*Produktnummer, *Fakturanummer, Antall)

Merk her at det er to fremmednøkler som til sammen utgjør primærnøkkel for tabellen. Dette går an fordi *kombinasjonen* produktnummer og fakturanummer er unik ettersom hvert produkt kun kan inngå i én varelinje på én faktura.

For hele eksemplet har vi nå følgende tabelldefinisjoner:

- Kunde (Kundenummer, Etternavn, Fornavn, E-post, Telefon, Adresse, Postnummer*)
- Poststed (Postnummer, Poststed)
- Faktura (Fakturanummer, Kundenummer*, Dato)
- Produkt (Produktnummer, Produktnavn, Leverandør, Pris)
- Fakturalinje (*Produktnummer, *Fakturanummer, Antall)

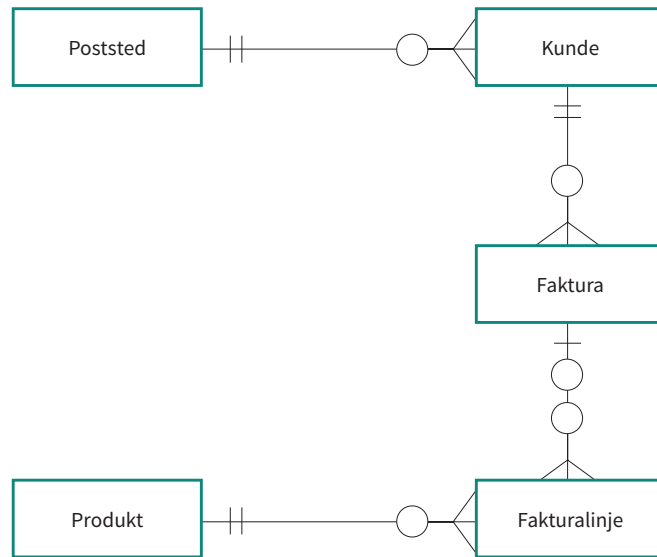
1NF er oppfylt så sant det bare blir registrert atomiske verdier som en e-postadresse eller et telefonnummer i hver rad i tabellen. 2NF er oppfylt ved at koblingstabellen Fakturalinje kobler Faktura og Produkt. 3NF er tilfredsstillt ved at vi har laget en egen tabell for Postnummer og Poststed.

Dette gir oss følgende fem entitetstyper (se figur under).

De mange koblingene mellom tabellene gjør at det kreves prosessorkapasitet for å hente frem data i søk. Dette kan føre til treghet i systemet og vil kunne hindre registrering av transaksjoner, som er systemets hovedoppgave. Modellering av relasjonsdatabaser blir ofte for komplekst og vanskelig tilgjengelig for forretningsbrukere. En annen måte å modellere data på er *dimensjonal modellering*, som brukes i multidimensjonale databaser. Mer om dette i kapittel 18 om behandling, analyse og presentasjon av digital forretningsinformasjon.



A12 Digital forretningsforståelse



Figur A.6.7: ER-diagram.



Eksempel på bruk og effekter av informasjonssystemer: Old@home

Vimarlund et al. (2008) undersøkte i artikkelen «Organizational effects of information and communication technology (ICT) in elderly homecare: a case study» effektene av innføringen av IKT i hjemmehjelpen i en kommune i Sverige. Vi ser her på konklusjonene i artikkelen, satt inn i det teoretiske rammeverket som er presentert i denne boken.

I denne kommunen tok man i bruk mobil teknologi for å kunne registrere opplysninger (data) om de pleietrengende, med andre ord kunder, direkte i hjemmet til kunden. Dataene som ble registrert, ble tilgjengelige for alle de involverte profesjonelle: leger, helsesøstre, helsesøsterassistenter og hjemmehjelpen. De samme dataene ble brukt av ansatte på ulike nivåer og med ulike funksjoner i organisasjonen. Hovedfordelene med systemet var at registreringen av data skjedde hos den pleietrengende, da de ennå var i friskt minne til den som hadde besøkt den pleietrengende, og at dataene ble delt i et felles system for alle de involverte, i motsetning til at de tidligere ble registrert i flere ulike systemer.

Et viktig moment i utviklingen, og senere for implementeringen av systemer, var at det var en høy grad av brukermedvirkning. Det gjorde at systemet ble bedre tilpasset brukernes behov, i tillegg til at det gjorde implementeringen lettere. Motstand mot endring, og gjerne spesielt endring der IT er involvert, er et kjent fenomen. At det i dette tilfellet har vært en suksess, er litt av en seier.

Verdikjedeanalyse for hjemmehjelp

Ofte omfatter en virksomhet flere ulike verdikonfigurasjoner. I tillegg er teoriene bak verdikonfigurasjonene utarbeidet med kommersielle foretak som utgangspunkt, så det kan derfor være vanskelig å plassere et helseforetak.

Når det er sagt, vil de verdikonfigurasjonene som passer best her, være verdikjede og verdiverksted. For de involverte legene vil verdiverksted være nærliggende: Man får inn et sykdomstilfelle, man vurderer ulike behandlingsalternativer, man velger og gjennomfører behandlingen, og til slutt evaluerer man effekten av behandlingen. For hjemmehjelpen vil en verdikjede være nærliggende: Man gjennomfører et sett med oppgaver som kan være ganske standardiserte, men også individuelt tilpassete, hos den pleietrengende. Effektivitet i et verdiverksted er å kunne gi den rette diagnosen og den rette behandlingen med minst mulig bruk av ressurser. Effektivitet i en tradisjonell verdikjede er å kunne gjennomføre oppgavene ved å bruke minst mulig tid.



A14 Digital forretningsforståelse

Men for hjemmehjelpen er det å kunne gi best mulig stell det viktigste, ikke raskest mulig stell – best mulig stell, men likevel effektivt.

Systemperspektivet

I teorien om systemperspektivet omtales fem elementer i et informasjonssystem. Ved å sette denne casen inn i teorien får vi følgende beskrivelser av elementene:

- Programvareressurser: muliggjør registrering og tilgang til data på flere ulike plattformer.
- Maskinvareressurser: mobile enheter som støtter lagring og tilgang til informasjon uavhengig av lokalitet, spesielt på POC (point of care).
- Kommunikasjonsressurser: bedriftens vanlige datanettverk samt mobile datanettverk.
- Dataressurser: enhetlig strukturerte og definerte data, tilgjengelige på tvers av ulike systemer.
- Menneskelige ressurser: kompetent IT-personell for å drifte systemet, og IT-kompetent helsepersonell for å kunne bruke systemet.

Gevinster

Her er utgangspunktet de ulike gevinsttypene; rasjonaliseringsgevinster, styringsgevinster, organisasjonsgevinster og markedsgevinster.

Systemet førte til en del tilfeller av *effektivisering* av arbeidet til alle de involverte. Eksempler på dette er:

- På grunn av de felles dataene brukte man mindre tid på å finne info om den pleietrengende.
- Som følge av dette sparte man tid og fikk dermed mindre press på ansatte og andre ressurser.
- I og med at dataene kun ble lagret i ett system, fikk man mindre dobbeltlagring av informasjon. Tidligere ble dataene gjerne ført både for hånd og i system, eller i flere systemer.
- Det ble kortere responstid mot kunder.

Disse gevinstene er helt klart *rasjonaliseringsgevinster*: effektivisering av prosesser på operativt nivå.

Noen av gevinstene hadde positive effekter på både organisasjonsstrukturen og organisasjonskulturen:



- Bedre forhold mellom de involverte aktørene. I helsevesenet har lege tradisjonelt sett vært et høystatusyrke, mens hjemmehjelp har vært et lavstatusyrke. Systemet har lagt til rette for bedre samarbeid mellom yrkesgruppene, og denne muligheten ble utnyttet.
- Bedre definert arbeidsdeling mellom aktørene. Ved å få bedre innblikk i hverandres arbeidsoppgaver kunne man lettere identifisere dobbeltarbeid og eliminere dette.
- Bedre kompetanse på bruk av IS/IT. Gjennom utviklingsprosessen fikk hele den involverte organisasjonen et kompetanseløft innenfor IT-kompetanse.
- Positiv holdning til å dele og å aksessere informasjon. Det opplevde skillet mellom yrkesgruppene var tidligere en sperre for å dele informasjon.
- Transparens: å kunne se hva andre gjør i den samme prosessen.
- Reduserer geografiske barrierer og organisasjonskulturbarrierer.
- Forståelse for endringer i arbeidsoppgaver og organisasjonsstruktur som innføring av systemet innfører.

Alle disse punktene hører inn under *organisasjonsgevinster*.

På taktisk nivå drives det gjerne med overordnet planlegging og fordeling av ressurser. Gevinster for dette nivået inkluderer:

- Bedre data rundt hjemmebesøk, noe som gjør det lettere å drive med ressursplanlegging.
- Lettere å identifisere og bytte ut ineffektive deler av prosessen.
- At dataene blir registrert under hjemmebesøkene, og ikke på slutten av en travel arbeidsdag, gjør at dataene som registreres, vil være mer korrekte. Dette vil gi bedre datagrunnlag for beslutninger.

Disse gevinstene representerer *styringsgevinster*.

Til slutt: Er det relevant å finne markedsgevinster i en næring som ikke er konkurranseutsatt? Ja, det er det. Det er ikke umulig at næringen blir konkurranseutsatt, men uansett er det viktig å finne ut om implementeringen av et IT-system fører til et bedre produkt for kundene. Dette er noe som gjør bedriften konkurransedyktig, og et godt levert produkt vil kunne gi fordeler overfor de delegerende myndigheter. Så på et strategisk nivå kan vi se følgende:

- Det nye systemet har gjort det lettere å utarbeide en kundetilpasset omsorgsprosess. En tidligere standardisert prosess kan nå tilpasses – uten at ressursbruken øker.



A16 Digital forretningsforståelse

- At informasjonen fra pågående og tidligere hjemmebesøk nå er tilgjengelig i sanntid for alle yrkesgruppene, fører til kortere responstid på tilfeller som måtte oppstå.
- Koordineringen av data og arbeidsoppgaver, villigheten til å dele informasjon og den hevede IT-kompetansen har ført til en mer fleksibel og tilpasningsdyktig organisasjon.

Disse gevinstene er altså *markedsgevinster*.

Kilde: Vimarlund et al. (2008)