

# **Vejledning til kravskabelon SL-07**

**- fra behov til løsning**

**Søren Lauesen**

Forlaget Samfundslitteratur

Søren Lauesen

*Vejledning til kravskabelon SL-07 - fra behov til løsning*

1. udgave 2007

© Forlaget Samfundslitteratur, 2007

Sats: Forfatteren

Omslag: Forfatteren og Samfundslitteratur Grafik

Tryk: Narayana Press

Forsideillustration: Rob Gonsalves: "The Sun Sets Sail".

Billedet er venligst stillet til rådighed af Saper Galleries,  
East Lansing, Michigan, USA.

ISBN: 978-87-593-1332-9

Forlaget Samfundslitteratur

Rosenørns Allé 9

1970 Frederiksberg C

Tlf. 38 15 38 80

Fax 35 35 78 22

forlagetsl@sl.cbs.dk

www.forlagetsl.dk

Alle rettigheder forbeholdes.

Kopiering fra denne bog må kun finde sted på institutioner, der har indgået aftale med COPY-DAN, og kun inden for de i aftalen nævnte rammer.

Undtaget herfra er korte uddrag til anmeldelse.

# Indhold

<b>1. Vejledningens og kravskabelonens formål.....5</b>	F. Systemets integration med eksterne systemer ....38
1.1. Farer ved at bruge skabelonen .....5	<i>F1. SKS</i> .....38
1.2. De største problemer med kravspecifikationer .....5	<i>F2. LabsysX</i> .....40
1.3. Det rigtige behovsniveau .....6	F10. Integration med nye eksterne systemer ....42
1.4. Præcise (verificerbare) krav .....7	G. Teknisk it-arkitektur .....44
1.5. Sporbarhed mellem behov og krav .....7	G1. Brug af eksisterende hardware og software .....44
1.6. Tidlig afdækning af store risici.....8	G2. Nyt hardware og software.....44
<b>2. Indsamling af krav .....8</b>	H. Sikkerhed.....46
2.1. Centraliser arbejdet.....8	H1. Adgangsret for brugere .....46
2.2. Inddrag interessenter og evt. leverandører.....9	H2. Sikkerhedsadministration .....48
<b>3. Vejledning til skabelonens dele .....10</b>	H3. Sikring mod tab af data.....48
Baggrund og vision .....12	H4. Sikring mod utilsigtet brugeradfærd .....50
A. Vejledning til leverandøren .....14	H5. Sikring mod trusler .....50
Praktiske forhold om formateringen.....14	I. Brugervenlighed og design .....52
B. Overordnede behov .....18	I1. Indlæring og effektivitet i daglig brug .....52
B1. Forretningsmæssige mål .....18	I2. Tilgængelighed og Look-and-Feel .....55
B2. Tidligt bevis for gennemførlighed (proof of concept).....20	J. Andre krav og leverancer.....56
B3. Tildelingskriterier .....20	J1. Andre standarder der skal følges.....56
C. Arbejdsopgaver systemet skal støtte.....22	J2. Uddannelse .....56
<i>C1. Indskriv patient inden ankomst</i> .....22	J3. Dokumentation.....58
Opgaver kontra delopgaver .....24	J4. Datakonvertering .....58
Kun få arbejdsopgaver.....26	J5. Installation .....58
Undgå at beskrive data som delopgaver .....27	K. Kundens leverancer .....60
Højniveau opgaver - oversigt, ikke krav.....28	L. Drift, support og vedligehold.....62
D. Data systemet skal anvende.....30	L1. Svartider.....62
<i>D1. Diagnoser</i> .....31	L2. Tilgængelighed (driftseffektivitet) .....66
D10. Data i eksisterende systemer og standarder .....32	L3. Datalagring.....68
E. Andre funktionelle krav .....34	L4. Support.....68
E1. Komplekse beregninger og regler .....34	L5. Vedligehold.....70
E2. Udskrifter og rapporter.....34	<b>4. Litteratur .....70</b>
E3. Udbygning af systemet.....36	

## Baggrund

Systemudviklere og it-konsulenter spørger ofte efter en eksemplarisk kravspecifikation som de kan tage udgangspunkt i til deres konkrete projekt. Dette skrift er vejledningen til en sådan kravspecifikation. Den er udformet som en skabelon udfyldt med et komplekst eksempel: *Krav til et elektronisk patientjournal system*. Jeg har dog enkelte steder digtet lidt for at kunne illustrere andre behov.

Selve kravskabelonen kan hentes på

<http://www.itu.dk/people/slauesen/Papers/KravskabelonSL-07.doc>

Store dele af skabelonen og vejledningen er udarbejdet på opfordring fra VTU (Ministeriet for Videnskab, Teknologi og Udvikling) som led i arbejdet med Statens K02 kontrakt. Jeg vil gerne takke Sten Mogensen (VTU) for et konstruktivt samarbejde og Vibeke Søderhamn (IT-Universitetet) for en både indsigtfuld og minutøs kontrol af skrifterne.

Tidligere udgaver af skabelonen er blevet brugt med succes i 12 projekter af meget forskellig art, fx krav til Forsvarets CMS, krav til Novo's miljørapporteringssystem, krav til næste version af en leverandørs standardprodukt. Erfaringerne fra disse 12 projekter er indarbejdet i denne udgave.

Jeg er interesseret i løbende at forbedre kravskabelonen og hører derfor gerne om både problemer og fordele ved at anvende den.

Søren Lauesen  
IT-Universitetet i København, februar 2007  
slauesen@itu.dk  
<http://www.itu.dk/people/slauesen>

# 1. Vejledningens og kravskabelonens formål

Kravspecifikationer kan skrives på mange måder. Kravskabelon SL-07 viser en måde der er baseret på erfaringer fra offentlige anskaffelser, specielt når der er tale om standard/rammesystemer hvor store dele af systemet findes allerede. Men skabelonen har også vist sig hensigtsmæssig til andre slags anskaffelser og til systemudvikling.

Kravskabelon SL-07 bruger en elektronisk patientjournal (EPJ) som et gennemgående eksempel. Eksemplet er let simplificeret så det er mere forståeligt for personer uden for hospitalsverdenen. EPJ-området er meget komplekst, så eksemplet er ikke trivielt, men illustrerer hvordan man kan håndtere vanskelige situationer. Nogle kravtyper er dog ikke relevante i EPJ, så der har jeg inddraget andre behov.

Det er hensigten at de kursiverede dele af skabelonen udskiftes eller tilpasses til det enkelte projekt. Andre dele kan selvfølgelig også tilpasses efter behov. I nogle tilfælde er der vist alternative formuleringer, og i et konkret projekt skal man selvfølgelig kun vælge én af formuleringerne.

Kravspecifikationen er sædvanligvis et bilag til en kontrakt. Der er ingen faste regler for hvad der skal stå i kontrakten, hvad der skal stå i bilag, og hvor mange bilag der skal være. I mange kontrakter er fx svartider et separat bilag, mens det i kravskabelonen er kravene i afsnit L1.

## 1.1. Farer ved at bruge skabelonen

Skabelonen omfatter ikke alle krav der kan være brug for, men det er tilstræbt at vise typiske krav inden for hvert område. I det konkrete projekt må man selv tilføje krav til at dække projektets specielle behov.

Alligevel er skabelonen meget omfattende. Det betyder at man let kommer til at medtage en masse krav som der slet ikke er behov for i det konkrete projekt. Resultatet bliver i bedste fald at man kommer til at betale alt for meget for systemet, i værste fald at ingen vil give et tilbud. Fx viser skabelonen krav om at kunden selv skal kunne udbygge systemet. Det er et voldsomt krav som kun er relevant i særlige tilfælde.

Man kan også let komme til at stille alt for skrappe krav. Fx viser skabelonen svartider der er behov for ved systemer i intens daglig brug. Men ved fx web-sider med bred orientering til offentligheden, kan man udmærket nøjes med meget længere svartider.

## 1.2. De største problemer med kravspecifikationer

Erfaringer fra udbudsforretninger viser at en række problemer går igen fra sag til sag. Denne vejledning sigter især på at undgå følgende problemer.

- a. Kravene er på et forkert behovsniveau. De kan være for løsningsorienterede så der højst er en enkelt leverandør der kan opfylde dem; de kan være så forretningsorienterede at leverandøren ikke kan påtage sig ansvaret for dem; eller de kan gøre det svært at sammenligne leverandørernes tilbud.
- b. Kravene er for upræcise til at man kan verificere dem. Dvs. at man ikke kan afgøre om de er opfyldt.

- c. Kravene afspejler ikke de formål man har med systemet. Dvs. at selvom kunden får opfyldt kravene, bliver de egentlige behov og forretningsmæssige mål ikke nået.
- d. Kravene afspejler i for høj grad de eksisterende arbejdsgange så der ikke bliver plads til nytænkning eller kreativ udnyttelse af de systemer der kan tilbydes.
- e. De store risici viser sig ofte for sent. Typisk ser man at store dele af funktionaliteten leveres tidligt, og kunden tager dele af systemet i brug. De vanskelige ting udskydes til senere. Det viser sig til sidst at leverandøren ikke kan levere disse vanskelige ting, men på grund af det fremskredne tidspunkt bliver kunden nødt til at acceptere systemet alligevel.

Disse problemer uddybes i det følgende.

### 1.3. Det rigtige behovsniveau

Kravspecifikationen skal ikke beskrive systemet for detaljeret, for så risikerer man at højst en enkelt leverandør kan leve op til kravene. På den anden side må den ikke være så overordnet at leverandøren ikke kan tage ansvaret for kravene. Der skal være en balance mellem de to yderpunkter.

Et meget detaljeret krav angiver en bestemt systemmæssig løsning. Her er et eksempel:

**Krav 1 (for løsningsorienteret).** *Systemet skal kunne vise patientens diagnoser som en hierarkisk struktur. Ved at klikke på plus og minus skal man kunne se underordnede og overordnede diagnoser.* (Kommentar: Dette krav er tydeligt inspireret af et bestemt system som kunden har set - eller allerede har).

Et meget overordnet krav omfatter forretningsmæssige forhold der er kundens ansvarsområde. Resultatet kan blive at leverandøren ikke vil tage ansvaret for kravet. Her er et eksempel:

**Krav 2 (for forretningsorienteret).** *Systemet skal sikre at antallet af fejlmedicineringer reduceres fra de nuværende 10% til 2%.* (Kommentar: Dette krav kan leverandøren ikke opfylde alene. Det kræver i mindst lige så høj grad kundens indsats, fx uddannelse af personale og registrering af det nødvendige data).

Et krav kan udtrykke kundens behov på sådan en måde at leverandøren kan tage ansvar for det, og samtidig kan det opfyldes på mange måder. Her er et eksempel:

**Krav 3 (tilpas balance).** *Systemet skal støtte arbejdsopgaverne C1 til C7.* (Kommentar: Arbejdsopgaverne beskriver hvad bruger og system skal udføre tilsammen. Vejledningen anbefaler denne form, og kravskabelonen viser flere eksempler. Det populære begreb "use case" har mange betydninger. Nogle af dem svarer til "arbejdsopgave").

Et krav kan også være så vagt at man ikke kan sammenligne de tilbudte løsninger. Her er et eksempel:

**Krav 4 (svært at sammenligne svarene).** *Leverandøren bedes redegøre for sin integrationsstrategi.* (Hver leverandør skriver en "roman" om sin strategi, og kunden kan ikke objektivt sammenligne dem).

#### 1.4. Præcise (verificerbare) krav

Kravene skal være så præcise at de kan *verificeres*, dvs. at man kan afgøre om de er opfyldt. Præcisionen har intet med behovsniveauet at gøre. Krav 1 ovenfor kan fx verificeres når systemet leveres. Krav 2 kan verificeres når systemet har været i brug et stykke tid.

Krav 3 kan også verificeres, men systemet kan opfylde kravet mere eller mindre godt. Nogle systemer kan fx give en meget effektiv støtte til arbejdsopgaverne, andre en mindre effektiv, men dog tilstrækkelig støtte. Støttens effektivitet kan være afgørende for kundens valg af leverandør. Krav 3 kan så at sige opfyldes til en "karakter på 13-skalaen".

Krav 4 kan verificeres allerede på tilbudstidspunktet, idet man blot skal se efter om leverandøren har beskrevet en strategi.

Her er et krav der ikke kan verificeres sådan som det står:

**Krav 5 (ikke verificerbart).** *Systemet skal være let at bruge.* (Kommentar: Det er for uklart hvordan dette skal måles og hvornår det er godt nok).

#### 1.5. Sporbarhed mellem behov og krav

I praksis ser man mange krav der er præcise, men udtrykt på sådan en måde at man ikke kan afgøre hvor god løsningen er. Det skyldes at man ikke kan se det behov kravet skal dække. Her er et eksempel:

**Krav 6 (præcist krav med uklart behov).** *Systemet skal kunne vise en oversigt over patientens diagnoser.* (Kommentar: Man kan ikke se hvad denne oversigt skal bruges til, og derfor er det svært at afgøre om den tilbudte løsning er god nok. Hvis man fx kunne se i hvilke arbejdsopgaver oversigten skulle bruges, ville det være lettere at afgøre om den var god nok).

Behovet her er at støtte en eller flere arbejdsopgaver. Men der er også behov på det mere forretningsmæssige niveau. I et EPJ-system kunne det fx være at mindske antallet af fejlmedicineringer, at reducere længden af hospitalsophold, etc. Hvis sådanne forretningsmæssige målsætninger skal nås, er det ikke nok at anskaffe nye it-systemer. Nogle af de eksisterende arbejdsopgaver må nødvendigvis udføres på en anden måde. Der kan være tale om at nogle eksisterende arbejdsopgaver støttes bedre af it, at dele af arbejdet overtages af computeren, at arbejdsopgaver slås sammen eller deles på en anden måde, at nye arbejdsopgaver indføres, eller at dele af arbejdet udføres af andre, fx af borgerne eller af leverandører.

I mange projekter ser man udmærkede forretningsmæssige målsætninger, men ingen har overvejet hvordan de skal opnås og hvordan it-systemet skal bidrage. Resultatet bliver ofte at gevinsterne udebliver. Afsnit B1 i skabelonen viser en simpel måde at forbinde de forretningsmæssige mål med kravene og sikre den nødvendige nytænkning af arbejdsopgaverne.

## 1.6. Tidlig afdækning af store risici

De vanskelige ting i projektet er ofte svartider når det fulde antal brugere nås, brugervenlighed og integration med andre systemer. Mangler på disse områder kan i praksis ikke udbedres sent i projektet.

Afsnit B2 i skabelonen stiller krav om at man tidligt i projektet afdækker disse risici, dvs. kontrollerer at disse ting realistisk kan nås. En sådan kontrol er dyr, og derfor er det ikke rimeligt at leverandøren skal have gjort det inden han giver tilbud. Til gengæld skal han, når han har fået kontrakten, udføre kontrollen hurtigt.

## 2. Indsamling af krav

Arbejdet med at skrive kravene kan virke overvældende, specielt i store organisationer. Det er derfor fristende at uddelegere skrivningen til de enkelte afdelinger og lade en central gruppe redigere det hele sammen. Dette må stærkt frarådes af flere grunde:

- Hver afdeling vil se på sine egne behov og har svært ved at se det hele fra den samlede organisations synspunkt. Kravene vil derfor afspejle de eksisterende arbejdsgange uden plads til nytænkning og forbedringer på tværs af afdelinger.
- Afdelingerne har sjældent ekspertise til at skrive krav, og kvaliteten af kravspecifikationen bliver ringe.
- Den centrale gruppe har ikke opnået den fornødne viden til at forstå afdelingens krav, og kan derfor ikke forbedre resultatet - bortset fra sproglige tilretninger. En gruppe udtrykte det således:

*Vi forstod ikke hvad de ville have, men redigerede det sammen og sendte det til leverandørerne. Vi regnede med at de forstod hvad det handlede om. Først langt senere gik det op for os at leverandørerne heller ikke forstod det. De lod bare som om de gjorde, og tænkte "det må vi finde ud af senere".*

### 2.1. Centraliser arbejdet

Det anbefales at have en lille arbejdsgruppe der udfører arbejdet:

- Indsaml behov, visioner og ønsker fra de forskellige interessenter (herunder afdelingerne, ekspertbrugere og ledere).
- Skriv det om til krav i overensstemmelse med denne vejledning.
- Gennemgå kravene med de enkelte afdelinger og foretag de nødvendige ændringer.
- Send kravene i udbud, normalt med inddragelse af juridisk ekspertise.
- Vurder de indkomne tilbud i samråd med de forskellige interessenter.

Arbejdsgruppen bør være på 2-4 medlemmer, sammensat så der er ekspertise om flest mulige arbejdsområder, inkl. it-funktionen. Gruppen kan suppleres med en konsulent med ekspertise i kravspecifikation.

Erfaringer med denne arbejdsform viser at det samlede arbejde kan reduceres til en femtedel. Samtidig går kvaliteten af kravspecifikationen drastisk op.



## 2.2. Inddrag interessenter og evt. leverandører

Selvom arbejdsgruppen har bred ekspertise, kan den ikke vide alt. Det er vigtigt at inddrage interessenterne undervejs. Det kan ske på mange måder, fx:

1. Interview af brugere, såvel ekspertbrugere som menige brugere. Spørg om eksisterende arbejdsopgaver, problemer i den måde de udføres på i dag, ønsker og visioner om fremtiden.
2. Få brugerne til at vise hvordan de udfører forskellige opgaver, specielt de sjældne, men vanskelige.
3. Indsaml relevante dokumenter, fx rapporter og blanketter der bruges i dag, print af skærbilleder, dokumentation af den eksisterende database og tekniske grænseflader til systemerne, statistikker og driftsrapporter.
4. Workshops hvor man sammen prøver at kortlægge de tværgående arbejdsgange og de ideelle arbejdsgange.
5. Forskellige former for brainstorm og fokusgrupper hvor man inspirerer hinanden til nye måder at gøre tingene på.
6. Besøg hos potentielle leverandører. Tit ved de hvordan andre udnytter deres produkt, og de kan henvise til kunder der bruger det. De kan også fortælle om muligheder kunden ikke selv har tænkt på, men som kan give anledning til nye måder at gøre tingene på.

Undgå at opstille alle disse oplysninger som krav. Det bliver let til en lang ønskeddel med krav på et alt for løsningsorienteret niveau og risiko for favorisering af enkelte leverandører. Spørg i stedet: Hvorfor er dette ønske interessant? Hvornår opstår ønsket? Hvilket behov dækker det? Hvilke arbejdsopgaver kan udnytte det? Resultatet bliver bredere behov der kan omsættes til krav.

### 3. Vejledning til skabelonens dele

Vejledningen i resten af hæftet kommenterer kravskabelonen afsnit for afsnit. De grå tekster er dele af skabelonen. Kapitelnummereringen i vejledningen følger kravskabelonen.

Bemærk at skabelonen nummererer kapitlerne med A, B, C i stedet for 1, 2, 3. Det er for at undgå konflikt med kontraktens bilagsnumre, som normalt er 1, 2, 3.

Skabelonens første egentlige side angiver systemets navn. Det er hensigtsmæssigt også at have et kort navn for systemet, idet der flere steder i kravene skal refereres til systemet ved navn, fx for at skelne det fra andre systemer som det skal samarbejde med.

Der angives også kundens navn, leverandørens navn og en kort beskrivelse af hvad leverancen omfatter. Fx er det vigtigt for læseren at have en fornemmelse af om der kun er tale om at levere et system eller også drift.

Forskellen mellem system og drift er afspejlet i kapitelstrukturen. Kapitel C til J beskriver hvad leverandøren skal levere til kunden på overtagelsesdagen. Kapitel K beskriver hvad kunden skal levere. Kapitel L beskriver hvad leverandøren skal levere efter overtagelsesdagen.

Hvis kravspecifikationen er et bilag til en kontrakt, vil systemets navn, kunden, etc. fremgå af kontrakten og man kan udelade det her. Kapitel K og L kan evt. være separate bilag til kontrakten og ikke kapitler i kravspecifikationen.

I nogle projekter vil man have en side med dokumentets versionshistorie, dvs. hvilke udgaver der har været og hvad der er rettet fra udgave til udgave. I skabelonen er det indskrænket til at hovedet på første side angiver hvem der sidst har rettet dokumentet og hvornår. MS-Word vil automatisk opdatere disse oplysninger når dokumentet printes eller åbnes igen.

Indholdsfortegnelsen er bekvem at have på forsiden, men kan blive så lang at den må placeres på en side for sig.

Sidst rettet af: Søren Lauesen  
07-03-2007, 22:27

## Kravspecifikation til *Elektronisk Patientjournal System* (i det følgende kaldet EPJ-systemet)

### Kunde

Region . . .

### Leverandør

. . .

### Leverancen omfatter

*Levering, drift og vedligehold af EPJ-system*

### Indhold

<b>Baggrund og vision</b> .....	3	F2. LabsysX .....	18
<b>A. Vejledning til leverandøren</b> .....	4	F10. Integration med nye eksterne systemer.....	19
A1. Funktionelle krav - tænkt eksempel .....	4	<b>G. Teknisk it-arkitektur</b> .....	<b>20</b>
A2. Kvalitetskrav - tænkt eksempel .....	5	G1. Brug af eksisterende hardware og software..	20
A3. Leverandørens svar.....	5	G2. Nyt hardware og software .....	20
A4. Praktiske forhold om formateringen .....	6	<b>H. Sikkerhed</b> .....	<b>21</b>
<b>B. Overordnede behov</b> .....	<b>7</b>	H1. Adgangsret for brugere .....	21
B1. Forretningsmæssige mål .....	7	H2. Sikkerhedsadministration .....	21
B2. Tidligt bevis for gennemførlighed (proof of concept).....	8	H3. Sikring mod tab af data .....	22
B3. Tildelingskriterier .....	8	H4. Sikring mod utilsigtet brugeradfærd.....	22
<b>C. Arbejdsopgaver systemet skal støtte</b> .....	<b>9</b>	H5. Sikring mod trusler.....	22
<b>Arbejdsområde 1: Patientadministration</b> .....	<b>9</b>	<b>I. Brugervenlighed og design</b> .....	<b>23</b>
C1. Indskriv patient inden ankomst .....	9	I1. Indlæring og effektivitet i daglig brug .....	23
C2. Indskriv . . . ..	9	I2. Tilgængelighed og Look-and-Feel.....	23
<b>Arbejdsområde 2: Patientbehandling</b> .....	<b>10</b>	<b>J. Andre krav og leverancer</b> .....	<b>24</b>
C10. Klinisk session .....	10	J1. Andre standarder der skal følges .....	24
C11. Ordiner medicin til patient .....	11	J2. Uddannelse .....	24
<b>D. Data systemet skal anvende</b> .....	<b>12</b>	J3. Dokumentation .....	25
D1. Diagnoser .....	13	J4. Datakonvertering .....	25
D2. Diagnosetyper .....	14	J5. Installation .....	25
D10. Data i eksisterende systemer og standarder	14	<b>K. Kundens leverancer</b> .....	<b>26</b>
<b>E. Andre funktionelle krav</b> .....	<b>15</b>	<b>L. Drift, support og vedligehold</b> .....	<b>27</b>
E1. Komplekse beregninger og regler .....	15	L1. Svartider .....	27
E2. Udskrifter og rapporter.....	15	L2. Tilgængelighed .....	28
E3. Udbygning af systemet.....	16	L3. Datalagring .....	28
<b>F. Systemets integration med eksterne systemer</b>	<b>17</b>	L4. Support .....	29
F1. SKS .....	17	L5. Vedligehold .....	30

## **Baggrund og vision**

Dette afsnit giver læseren et hurtigt overblik over hvad systemet drejer sig om. Det er en god idé kort at forklare de vigtigste forretningsmæssige mål (uddybes i afsnit B1), hvad kunden har nu og hvad han forventer.

Kontekstdiagrammer for den nuværende og kommende situation er en god illustration. Læg vægt på at markere det system der skal leveres som én kasse med dobbelt væg. I forbløffende mange kravspecifikationer er det uklart for såvel kunde som leverandør hvor meget der skal leveres og hvem der skal integrere. I eksemplet skal leverandøren levere sit eget system integreret med et andet nyt system.

Ofte ser man at kunden skriver en lang redegørelse for sin it-strategi, den historiske udvikling, osv. Det kan være udmærket som orientering for leverandøren hvis det kan holdes på et par sider og er relevant for det han skal levere. Men ofte afspejler redegørelsen mere kundens interne overvejelser og politiske udsagn, som ikke er relevante for leverandøren.

Pas også på at der ikke kommer til at stå krav i dette afsnit.

## Baggrund og vision

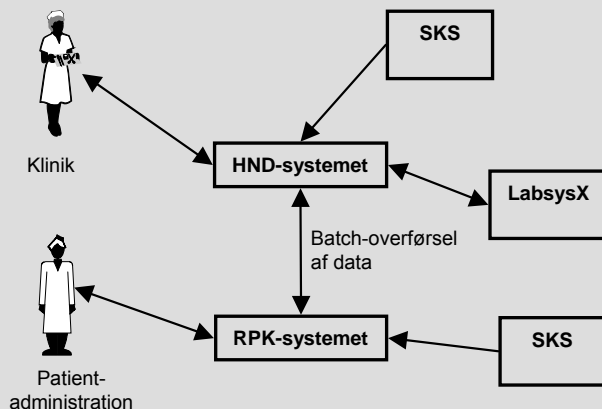
Kunden har i øjeblikket flere EPJ-systemer af ældre dato, som han ønsker at erstatte med ét for at opnå:

1. Mere effektiv støtte til det kliniske arbejde
2. Bedre mulighed for integration med fremtidige systemer
3. Lavere driftsomkostninger

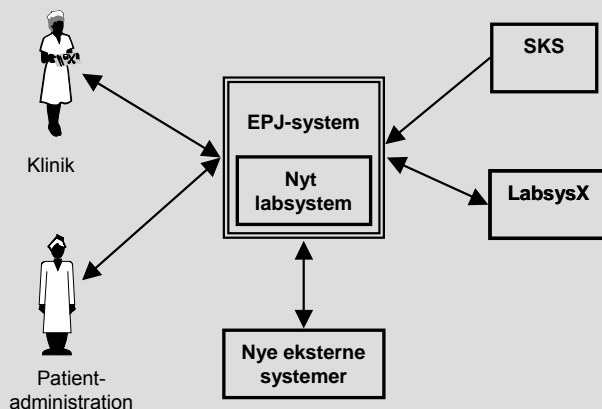
Kunden forventer at leverandøren allerede har et standardsystem der kan opfylde mange af kravene. Kunden er til gengæld villig til i rimelig udstrækning at ændre sine arbejdsgange så de passer til standardsystemet, så længe de overordnede mål nås.

Den nuværende og fremtidige situation kan anskueliggøres med disse kontekstdiagrammer. I dag er der fx dårlig integration mellem HND og RPK systemet, og SKS opdateringer skal foretages uafhængigt i de to systemer.

**Figur 1: Eksisterende system**



**Figur 2: Vision om nyt system**



## A. Vejledning til leverandøren

Første kapitel af skabelonen starter med at forklare hvordan kravene er opstillet og hvordan leverandørens svar skal struktureres. Specielt forklares hvordan tabellerne bruges, hvad der er krav og hvad der ikke er. Princippet er at venstre søjle beskriver kundens behov (de egentlige krav), mens midterkolonnen beskriver en mulig løsning eller en tilbudt løsning. Om nødvendigt henvises til bilag.

Det er erfaringen at mange leverandører er usikre på hvordan de skal bruge tabellerne i deres svar. Derfor er forklaringen ret lang med et detaljeret eksempel af generel natur. Det er tanken at eksemplet kan bruges ordret i de fleste udbudsforretninger.

Eksemplet viser hvordan kunden i udbudsmaterialet har brugt varianter og problemer, og hvordan han på nogle punkter har skrevet eksempler på løsninger. I svaret har leverandøren dels anført løsninger der er en del af standardsystemet, dels løsninger der leveres specielt eller er en del af en kommende version.

Leverandøren har også mulighed for at tilbyde alternative løsningspakker og opdeling i delleverancer.

Nogle krav er kvalitetskrav hvor leverandøren skal tilbyde et antal, en svartid eller lignende. Det kan være svært for leverandøren at give et tal hvis han ikke ved hvad kunden forventer. Skal han tilbyde et dyrt system med en kort svartid eller et billigt med en længere svartid? Her kan kunden angive hvad han forventer, fx 20 sekunder. Det skal fortolkes sådan at en kortere svartid ikke giver leverandøren fordele, mens en længere vil stille ham dårligere. Hvis kunden ikke kan leve med en længere svartid, må han i stedet skrive svartiden som et krav i venstre side.

Princippet om at kunden skriver sine forventninger, giver leverandøren mere frihed, men princippet skal ikke misbruges. Kundens forventninger bør være realistiske og nødvendige. Fx er det urealistisk og unødvendigt at en kompliceret rapport der sjældent bruges, kan beregnes og vises på 1 sekund. Det kan få en seriøs leverandør til at gøre sig store anstrengelser for at tilbyde et dyrt system tæt på forventningen, mens et langt billigere var nok.

### Praktiske forhold om formateringen

Skabelonen giver også råd om formateringen. Kravspecifikationen er skrevet i MS-Word. Dokumentet er struktureret med standard overskrifter på niveau 1, 2, undertiden 3, samt en Overskrift uden nummer. De vil automatisk generere indholdsfortegnelsen. Der er brugt tvungent sideskift for nogle overskrifter for at øge overskueligheden. Det ændres via liniens Format → Afsnit → Sideskift.

Tabellerne har tykke rammer med 1½ punkt og tynde med ½ punkt. Venstre søjle har hængende indrykning på 0,75 cm. Inden for en tabelcelle skal man tabulere med Ctrl+Tab, idet Tab alene rykker markøren til næste celle.

## A. Vejledning til leverandøren

Dette kapitel forklarer kravenes form og hvordan leverandøren beskriver sin løsning.

Kravene er opdelt i kapitler efter deres art, fx kapitel C om arbejdsopgaver systemet skal støtte. Inden for hvert kapitel er kravene opstillet i tabeller, fx en tabel med krav der vedrører en bestemt arbejdsopgave. Tabellerne har tre søjler. Venstre søjle beskriver kundens behov, midterste søjle et eksempel på en løsning eller den tilbudte løsning. Højre søjle beskriver hvornår løsningen leveres og om det er en standardløsning.

### A1. Funktionelle krav - tænkt eksempel

Her er et tænkt eksempel uden forbindelse til nærværende leverance. Eksemplet skal blot illustrere de funktionelle kravs form. Det overordnede krav er at systemet skal støtte en række arbejdsopgaver, herunder C5, og helst fjerne de problemer man oplever i dag.

#### C5. Henvendelse om en klage

...

**Brugere:** Hotline-medarbejdere, ofte i kortvarige ansættelser.

**Miljø:** Kontorlandskab hvor man sidder tæt, og hvor der er en del støj.

**Hypighed:** Totalt ca. 500 henvendelser pr. dag, og op til 100 henvendelser pr. bruger.

...

Delopgaver og varianter:	Eksempel på løsning:	Kode:
1. Modtag henvendelsen pr. brev, telefon eller e-mail. Kan være en ny klage eller oplysninger om en eksisterende klagesag.		
2. Find sagen frem . . .	Data overføres automatisk fra e-mail til søgebilledet.	
2a. Opret en ny sag.		
2p. Problem: Sagen kan være svær at finde, fx fordi klageren ikke kender sagsnummeret eller ikke kan huske sit cpr-nummer.	Systemet viser mulige match på klagerens navn eller dele deraf.	
...		

Tabellen beskriver de enkelte delopgaver, mv. der indgår i arbejdsopgave C5. Der er således et krav om at støtte hver af delopgaverne i en eller anden grad. Oplysningerne om *Brugere*, *Miljø* og *Hypighed* er ikke i tabellen, og er derfor ikke krav, men **forudsætninger** for kravene. I eksemplet vil det sige at arbejdsopgaven skal støttes for den slags brugere, omgivelser, mv. der er anført. (Egentlige krav om fx svartider fremgår af kapitel L).

De enkelte krav nummereres fortløbende. Varianter af det enkelte krav anføres med bogstaverne a, b, osv. Problemer der vedrører det enkelte krav anføres med bogstaverne p, q, osv. Henvvisninger til et krav, variant eller problem ser fx sådan ud:

Se problem C5-2p.

**Behov.** Venstre søjle i tabellen angiver kundens behov, fx en delopgave systemet skal støtte eller noget data systemet skal opbevare.

**Løsning.** Midterste søjle angiver hvad støtte systemet giver. I udbudsmaterialet kan søjlen indeholde kundens forestilling om en løsning. Dette er *ikke et krav* eller et ønske, men blot kundens forestilling om nogle muligheder til orientering for leverandøren. I mange tilfælde vil feltet slet ikke være udfyldt. I svaret udfylder leverandøren søjlen med den løsning han tilbyder (se afsnit A3).

**Kode.** Højre søjle udfyldes af leverandøren med en talkode der angiver om løsningen er en del af standardsystemet, en tilføjelse, etc. (se afsnit A3). Undertiden har det ikke mening at angive en kode. I så fald har kunden skrevet N/A i feltet.

## A2. Kvalitetskrav - tænkt eksempel

Nogle krav er kvalitetskrav, hvor kunden ikke ønsker en funktionalitet men et antal, en tidsgrænse eller lignende. Her er et tænkt eksempel:

### G1. Brug af eksisterende hardware

Kunden har i øjeblikket følgende it-udstyr der forventes brugt til drift af systemet:

1. 20 servere af typen . . .
2. . . .

Belægningen af udstyret til andre opgaver i spidsbelastningsperioder forventes i fremtiden at være:

3. Servere . . .
4. . . .

Krav til kundens hardware og software:	Eksempel på løsning:	Kode:
1. Det tilbudte system skal i starten køre på det eksisterende udstyr uden udbygning, og overholde svartidskravene i L1.	Systemet kan under disse omstændigheder trække ___ brugere. (Kunden forventer 20 brugere).	
2. . . .		

Kunden beskriver atter sit behov i venstre side, men midterste søjle viser nu hvordan han ønsker løsningen målt eller svaret struktureret. Desuden kan han anføre hvad han forventer, dvs. hvornår det er godt nok. Kunden kan dog leve med mindre end forventningerne, men vil i så fald vurdere løsningen lavere på dette punkt.

Indledningen til tabellen angiver de forudsætninger leverandøren skal regne med.

## A3. Leverandørens svar

I tilbudet udfylder leverandøren midterste og højre søjle så de viser den tilbudte løsning. Han har mulighed for at angive alternative løsninger og for at angive i hvilken delleverance eller version han vil levere løsningen. Her er et muligt svar på det første eksempel ovenfor:

### C10. Modtagelse af en klage

...

Delopgaver og varianter:	Tilbudt løsning:	Kode:
1. Modtag henvendelsen pr. brev, telefon eller e-mail. Kan være en ny klage eller oplysninger om en eksisterende klagesag.	(Systemet fanger ikke email, etc. Brugeren må selv starte registreringen).	5
2. Find sagen frem . . .	Se søgebillede 12 i bilag x. <del>Data overføres automatisk fra e-mail til søgebilledet.</del>	1
	Der leveres semi-automatisk overførsel fra e-mail i version 18. Se beskrivelsen i bilag x, side y.	4.18
2a. Opret en ny sag.	Se skærmbillede 13 i bilag x.	1
2p. Problem: Sagen kan være svær at finde, fx fordi klageren ikke kender sagsnummeret eller ikke kan huske sit cpr-nummer.	Systemet viser mulige match på klagerens navn eller dele deraf. Systemet har fonetisk søgning (se søgebillede 12).	2.1 (alt.A) 2.3 (alt.B)
...		

Leverandøren har ændret overskriften på midterste søjle til *Tilbudt løsning*. De løsningseksempler som kunden har anført, men som ikke er relevante i den tilbudte løsning, har leverandøren streget over.



## Koder

I højre søjle bruges følgende koder:

- 1 Del af standardsystemet
- 2.x En udbygning af standardsystemet, men udbygningen er omfattet af den normale vedligeholdelsesaftale. Vil være til rådighed fra delleverance x.
- 3.x Specialudviklet software eller en udbygning af standardsystemet som ikke er omfattet af den normale vedligeholdelsesaftale. Vil være til rådighed fra delleverance x.
- 4.y Standard i en kommende version der leveres som en del af vedligeholdelsesaftalen. Vil være til rådighed fra version y.
- 5 Der tilbydes ikke en løsning til dette krav.
- alt.z Der tilbydes flere alternativer og denne løsning vedrører alternativ z.

I eksemplet har leverandøren angivet at delopgave 1 ikke støttes af systemet, men udføres af kunden som i dag (kode 5).

For delopgave 2 består løsningen af to dele, og leverandøren har derfor delt svaret i to dele. Første del er den standardløsning der vises i skærmbillede 12 i bilag x til tilbudet (kode 1). Anden del vedrører en systemdel der automatisk kan analysere en e-mail og overføre data til søgebilledet. Den er beskrevet i bilag x til tilbudet og leveres som en standarddel af version 18 (kode 4). Kundens eksempel på en løsning er ikke længere relevant og derfor streget over.

Variant 2a støttes gennem skærmbillede 13 (kode 1: standard). I princippet behøver leverandøren ikke skrive en løsning, men kan nøjes med at skrive kode 1. Erfaringen viser dog at det gør kunden usikker, især hvis mange løsningsfelter er blanke eller uændrede: Har leverandøren overhovedet forstået vores behov? En sådan tvivl mindsker leverandørens chancer betydeligt.

Problem 2p illustrerer endnu en kompleksitet i tilbuddet. Leverandøren har tilbudt to alternative leverancer, som kunden kan vælge mellem. Alternativ A indeholder bl.a. en traditionel søgemekanisme, som kan leveres i første delleverance. Alternativ B indeholder bl.a. en avanceret fonetisk søgning, som først kan leveres i delleverance 3. Begge funktionaliteter er omfattet af den normale vedligeholdelsesaftale (kode 2).

Sådanne alternativer, versioner, mv. kan være svære at overskue for kunden, så det må kraftigt anbefales at leverandøren vedlægger en oversigt over alternativerne og hvilke delleverancer de består af, samt af kommende versioner der leveres som en del af vedligeholdelsesaftalen. Priserne for de forskellige alternativer kan passende anføres samme sted, så kunden let kan sammenligne.

Her er et muligt svar på det andet eksempel ovenfor:

### G1. Brug af eksisterende hardware

...

Krav til kundens hardware og software:	Tilbudt løsning:	Kode:
1. Det tilbudte system skal i starten køre på det eksisterende udstyr uden udbygning, og overholde svartidskravene i L1.	Systemet kan under disse omstændigheder trække <u>10</u> brugere. (Kunden forventer 20 brugere).	1
2. ...		

Leverandøren har indsat det antal brugere hans løsning kan trække. Da det er færre end hvad kunden forventer, er han klar over at han på dette punkt bliver vurderet lavere end den der tilbyder 20. Havde han omvendt tilbudt 40 brugere, ville det ikke give ham fordele frem for den der tilbyder 20.

### A4. Praktiske forhold om formateringen

...

## B. Overordnede behov

### B1. Forretningsmæssige mål

Dette afsnit af skabelonen indeholder systemets forretningsmæssige mål, opstillet i en tabel så man kan se hvordan de tænkes nået. Første søjle indeholder målene, anden søjle hvordan man forestiller sig at målene nås, tredje søjle hvilke krav der skal muliggøre det. Sidste søjle giver kunden mulighed for at angive hvornår målet skal nås. Det præciseres at der ikke er tale om krav til leverandøren, men en hjælp til at vurdere hvad der er vigtigt for kunden. Selv hvor der er angivet en tidsfrist, er det sjældent nok at leverandøren opfylder kravene med den frist. Der skal også være tid til kundens organisatoriske implementering.

De forretningsmæssige mål tjener flere formål:

- a. De fortæller leverandøren hvad kunden vil opnå.
- b. De er væsentlige kriterier for kundens valg af løsning.
- c. De hjælper kunden med at kontrollere at han har de afgørende krav med.

I eksemplet er mål 1 (effektiv støtte til alle arbejdsopgaver) et meget bredt mål, som gør det muligt at fravælge en løsning der giver dårlig støtte til en eller flere arbejdsopgaver. Fx kan lægen have behov for en god oversigt over patientens situation for at kunne træffe den rigtige beslutning. Et system med et meget dårligt oversigtsbillede skal derfor kunne fravælges selvom oversigtsbilledet blot er én blandt 1000 detaljer i systemet. Andre detaljer kan være lige så vigtige, så systemet skal også kunne fravælges hvis en af dem har en væsentlig brist. Det kan få alvorlige konsekvenser hvis man af juridiske grunde ikke kan fravælge sådan en løsning. Mål 1 sikrer at man kan (se også B3).

I eksemplet havde kunden tidligt formuleret mål 3, løbende forbedring af arbejdsgangene, men havde ikke gjort sig klart at det krævede en ny organisatorisk struktur, en *forbedringskomité*, der skulle opbygge, teste, og udbrede nye standardplaner for behandling. Dette arbejde krævede it-støtte. Først da man udfyldte tabellen blev dette behov klart.

Tabellen er god til at afsløre hvor kunden mangler planer for at nå målene. Det er især vigtigt at tredje kolonne peger direkte på krav i kravspecifikationen. Tabellen er også en stor hjælp for leverandøren, der på den måde lettere kan se hvad der er særligt vigtigt.

Hold antallet af mål på et rimeligt niveau. Bliver der mere end 10 mål bør man kontrollere at det ikke bare er krav man skriver. Fx ser man "målsætninger" af denne art: *Der skal let kunne udskrives forbrugsopgørelser*. Dette er nærmere et krav til it-systemet end et forretningsmæssigt mål. Et forretningsmæssigt mål handler om hele organisationens resultater, ikke blot noget som it-systemet gør.

Det kan nogle gange være svært at skrive en overordnet løsning i kolonne 2, fx hvis det forretningsmæssige mål er meget tæt på at være et krav. Men hvis det er svært, kan det også være et tegn på at det ikke er et rigtigt forretningsmæssigt mål man har skrevet. I eksemplet ovenfor, *der skal let kunne udskrives forbrugsopgørelser*, vil det være svært at skrive en overordnet løsning.

## B. Overordnede behov

Dette kapitel forklarer hvordan kravene tænkes at tilgodesæ kundens forretningsmæssige mål og hvordan risikable krav ønskes afklaret tidligt.

### B1. Forretningsmæssige mål

Kundens formål med at anskaffe systemet er at opfylde en række forretningsmæssige mål. Kunden forventer at systemet bidrager til målene som anført nedenfor. Leverandøren kan sjældent opfylde målene alene, idet kundens medvirken også er afgørende. Målene er altså *ikke krav* til leverandøren, selvom de står i en tabel for overskuelighedens skyld.

Alle mål er vigtige og jo før de kan nås, desto bedre. For nogle mål er det kritisk at de nås på et bestemt tidspunkt, fx af forretningsmæssige eller lovgivningsmæssige grunde. Sådanne tidsfrister er anført i tabellen.

Formål med det nye EPJ-system	Overordnede løsninger	Relaterede krav	Evt. tidsfrist
1. Effektiv støtte til alle arbejdsopgaver.	Alt nødvendigt data er til rådighed ved arbejdet, specielt kan alle parter se journalen.	Støtte til alle arbejdsopgaver i kapitel C og alt data i kapitel D.	
2. Færre fejlmedicineringer.	Undgå manuelle mellemed-indtast ordinationen straks.  Systemet kontrollerer for rimelighed, interaktion af præparater, etc.	Støtte til arbejdsopgave C10 (klinisk session), især delopgave 2 (vurder patientens tilstand).  Støtte til opgave C11 (ordination), stort set alle delopgaverne.	
3. Løbende forbedring af arbejdsgange.	Let at opbygge og ændre standardplaner for behandling.  Let at integrere systemet med nye laboratoriesystemer, etc.	Støtte til arbejdsopgave C20 (forbedringskomité).  Krav i afsnit E3 og F10 (udbygning af systemet og integration med nye eksterne systemer).	
4. Lavere driftsomkostninger.	Erstatte flere systemer der hver kræver stor teknisk ekspertise med ét system.	Støtte til alle arbejdsopgaver fra de hidtil adskilte områder.	
5. Overholde de nye EU-regler om ...	...	...	1/1- 2008

## B2. Tidligt bevis for gennemførlighed (proof of concept)

Dette afsnit beskriver de dele af projektet som anses for mest risikable fordi mangler på disse områder normalt ikke kan udbedres sent i projektet. Der er typisk tale om kvalitetskrav for, som amerikanerne siger, *quality is not an add-on feature*. En tidlig test kan reducere risikoen meget betydeligt.

Afsnittet nævner at kontrakten giver hævebeføjelser til begge parter hvis de tidlige beviser ikke forløber tilfredsstillende. Selvom det kraftigt kan anbefales, er det selvfølgelig ikke sikkert at parterne er villige til at lave en sådan aftale.

Krav 1 til 5 præciserer hvad det er man vil have testet tidligt. Midterste søjle giver et eksempel på hvordan det kan testes. Leverandøren kan ændre det til sit eget forslag om test. Desuden angiver han hvornår beviset kan foreligge. Højre søjle, med en kode for hvornår det leveres, er irrelevant her.

Disse tests kan være dyre, og derfor er det ikke rimeligt at leverandøren skal have gjort det inden han giver tilbud. Leverandøren kan til gengæld opleve det som urimeligt at kunden uden begrundelse kan hæve kontrakten i denne periode. Tilsvarende kan kunden opfatte det som urimeligt hvis leverandøren hæver kontrakten. Man kan derfor i kontrakten indføre en kompensationspris der skal betales af den part der hæver.

## B3. Tildelingskriterier

Tildelingskriterier er de kriterier man bruger til at vælge en leverandør. For udbud efter EU-udbudsreglerne er der kun to mulige overordnede kriterier: *Udelukkende den laveste pris* eller *det økonomisk mest fordelagtige tilbud*. Skabelonen viser den sidste mulighed, som er den mest komplicerede.

Dette afsnit vil ofte være en del af kontrakten, så det ikke skal stå i kravspecifikationen. Hvis projektet ikke er underlagt EU-reglerne, kan man frit bruge andre tildelingskriterier eller helt udelade dem.

Afsnittet nævner de delkriterier der stort set altid er nødvendige og tilstrækkelige:

1. Den forretningsmæssige værdi af løsningen.
2. Risikoen.
3. Leveringstiden.
4. Prisen.

Delkriterierne kan opdeles yderligere, fx et tal for hvert forretningsmæssigt mål. Hvad skal man nu gøre ved alle disse kriterier? Skal man vægte kriterie 1 til 3, lægge dem sammen og dividere med prisen? Det er meget brugt, men kan gå helt galt hvis en leverandør vinder ved at score højt på de fleste kriterier, men lavt på fx "støtte til daglig brug". Projektet vil blive en katastrofe.

Måske skulle man hellere gå efter mini-max princippet, dvs. vælge den leverandør der scorer højest på sit dårligste delkriterie? Men det kan give en økonomisk ufordelagtig løsning.

Situationen kaldes en multi-kriterie beslutning, og ved sådan en findes der ikke en optimal måde at beslutte på. Det nytter ikke at fastsætte prioriteter eller vægte på forhånd. Det der virker bedst er at se på de foreliggende muligheder (tilbuddene) ud fra den konkrete situation, kassere de tilbud der er klart dårligere end resten, og

finde de argumenter der kan udpege vinderen blandt resten.

Desværre er det ikke tilladt efter EU-reglerne, fordi det giver kunden mulighed for at manipulere med vægterne, så yndlingsleverandøren kan fremstå som vinderen.

Skabelonen viser et rimeligt kompromis. Delkriterierne vægtes, fx med en vægt der afspejler delkriteriets indflydelse på den samlede økonomiske værdi for kunden. Desuden er det vigtigt at kunne afvise en løsning som klart er utilstrækkelig på et eller andet punkt. Det sker ved at indføre minimumsværdier for delkriterierne (for pris og leveringstid er der dog tale om maksimumsværdier).

## B2. Tidligt bevis for gennemførlighed (proof of concept)

Kunden ønsker at undgå den situation at de mere trivielle dele af systemet udvikles løbende, mens de risikable dele udskydes og til sidst viser sig umulige at gennemføre. Kunden kræver derfor en tidlig påvisning af at de risikable dele er gennemførlige.

*Ifølge kontrakten kan begge parter hæve aftalen hvis de tidlige beviser ikke er tilfredsstillende.*

Følgende krav betragtes som de mest risikable områder. Væsentlige mangler her kan næppe udbedres sent i projektet. Leverandøren skal i sit svar angive hvordan han vil udføre disse tidlige beviser, og hvornår det kan ske. Tidspunktet angives som antal arbejdsdage efter kontraktunderskrift. Kunden forventer 40 arbejdsdage.

Risikable krav hvor tidligt bevis kræves:	Eksempel på bevis:	Kode:
1. Effektiv støtte til klinisk session (arbejdsopgave C10).	En prototype af de nødvendige skærbilleder (gerne på papir) vurderes af ekspertbrugere. Kan ske efter __ arbejdsdage.	N/A
2. Brugervenlighed (alle krav i afsnit II).	En prototype (gerne på papir) usability-testes med typiske brugere. Kan ske efter __ arbejdsdage.	N/A
3. Svartider med det planlagte antal brugere (alle krav i L1).	Et testsystem sættes op med simulation af det forventede antal brugere. Svartiderne måles. Kan ske efter __ arbejdsdage.	N/A
4. Mulighed for tredjeparts udbygning (E3 og F10).	Dokumentation af dele af systemet og dele af de tekniske grænseflader vurderes af et uafhængigt software-hus med henblik på egnethed til udbygningsformålet. Kan ske efter __ arbejdsdage.	N/A
5. Integration med andre systemer.	En forsøgsopstilling hvor dataudvekslingen demonstreres. Kan ske efter __ arbejdsdage.	N/A

## B3. Tildelingskriterier

Kunden vælger det økonomisk mest fordelagtige tilbud. I vurderingen indgår følgende delkriterier:

1. Den forretningsmæssige værdi af løsningen. Vurderes som et tal for hvert af de forretningsmæssige mål (B1).
2. Risikoen. Vurderes ud fra hvornår leverandøren kan give et tidligt bevis (B2) og hvor stor en del af leverancen der er et eksisterende system.
3. Leveringstiden.
4. Prisen, inklusive kundens interne omkostninger til nødvendigt udstyr, konvertering og uddannelse.

Beregningen af de enkelte delkriterier og den samlede værdi er nøjere beskrevet i . . . For hvert delkriterium er der fastsat en minimumsværdi der skal være opfyldt for at tilbuddet kan komme i betragtning. Desuden er der fastsat en vægt med hvilken delkriteriet indgår i det samlede resultat.

## C. Arbejdsopgaver systemet skal støtte

Dette kapitel beskriver de arbejdsopgaver systemet skal støtte. For at kunne vurdere hvor godt en arbejdsopgave støttes, skal man vide hvilken slags brugere der er tale om, i hvilke omgivelser opgaven udføres, osv. Det kan man anføre for hver enkelt opgave, men da oplysningerne ofte er fælles for flere opgaver, er det praktisk at samle dem et andet sted.

I eksemplet har man opdelt arbejdsopgaverne i *arbejdsområder* og beskrevet hvert område for sig. Man beskriver de brugerprofiler (roller) medarbejderne har og evt. også de omgivelser arbejdet foregår i. Brugerprofilen omfatter brugernes it-erfaring, deres kendskab til arbejdsområdet, deres motivation, mv. En medarbejder kan godt kan arbejde i flere områder, evt. med forskellige roller i dem.

### C1. Indskriv patient inden ankomst

Arbejdsopgaverne skal udføres af bruger og computer i fællesskab, og man ved ikke på forhånd hvem der gør hvad. Det afhænger af den valgte løsning. Man bruger bydemåde (*indskriv patient*) for at skjule hvem der gør hvad. Beskrivelsen af en arbejdsopgave består af disse dele:

1. Nummer og navn på opgaven. Arbejdsopgaver nummereres C1, C2 osv. For at undgå for meget omnummerering under arbejdet med kravene, kan man gruppere opgaverne i bundter og starte hvert bundt med et rundt nummer. I eksemplet er C10 fx første opgave i det næste bundt.
2. En kort indledning til hvad opgaven handler om.
3. Angivelse af hvornår opgaven starter og slutter. En arbejdsopgave bør være noget der udføres i ét stræk fra start til slut. Bemærk at opgaven kan starte af flere forskellige grunde.

Startsignalet bør være noget der sker i brugerens verden uden for computeren. Det kunne fx være at en patient indtager sin medicin, hvorefter sygeplejersken registrerer det i systemet. Undgå computer-fikserede startsignaler som: *Brugeren ønsker at registrere et medicinindtag*. Det giver ikke leverandøren nogen fornemmelse af hvad der bevirker dette ønske og om systemet måske kunne støtte brugeren bedre, fx med et stregkodearmbånd på patienten.

I eksempel C1 kan opgaven afsluttes (parkeres) fordi der mangler oplysninger. Selvom den ikke er logisk afsluttet endnu, er den afsluttet i arbejdsmæssig forstand - brugeren giver sig til at lave noget andet. Dette mønster er meget almindeligt, og det er vigtigt at systemet støtter det godt.

4. Hvor hyppigt opgaven udføres. Dels totalt for hele organisationen, dels pr. bruger der udfører mange af disse opgaver. Det totale antal hjælper leverandøren til at vurdere den nødvendige computerkapacitet. Antallet pr. bruger siger noget om hvor effektiv brugerstøtten skal være. Disse tal er dog til orientering, ikke krav. Egntlige krav på disse områder skrives i afsnit I1 og L1.
5. Situationer hvor opgaven er særlig vanskelig at udføre, fx fordi det sker under stress eller kræver særlig akkuratse. Bemærk at C1 ikke har nogen vanskelige situationer, mens C10 har en. Vanskelige situationer bør være sjældne. Dvs. at man ikke umiddelbart kan observere dem, men må spørge brugerne om dem.

## C. Arbejdsopgaver systemet skal støtte

Systemet skal støtte arbejdsopgaverne i dette kapitel, herunder alle delopgaver og varianter, samt reducere problemerne. Selvom delopgaverne er nummereret, skal de ikke nødvendigvis udføres i den rækkefølge, og de skal heller ikke nødvendigvis udføres alle sammen hver gang. En delopgave kan også udføres flere gange inden for samme arbejdsopgave.

### Arbejdsområde 1: Patientadministration

*Området omfatter indkaldelse af patienter, overvågning af ventelister . . .*

**Brugerprofil:** *Lægeselekretærer. De fleste lægeselekretærer er erfarne it- brugere og har et godt kendskab til deres arbejdsområde. De er gode til at kommunikere med klinikere.*

**Brugerprofil:** *Administrative medarbejdere. . . .*

#### C1. Indskriv patient inden ankomst

*Denne arbejdsopgave opretter et patientforløb eller fortsætter oprettelsen hvis sagen har været parkeret. De fleste henvendelser kan behandles færdigt i én arbejdsgang. Resten må parkeres, fx fordi der mangler oplysninger. Det er vigtigt at systemet hjælper med til at de parkerede sager ikke glemmes.*

**Start:** *Meddelelse fra egen læge, andet sygehus . . . Meddelelsen kan også være supplerende oplysninger der manglede.*

**Slut:** *Når patienten er indkaldt eller på venteliste, eller når sagen må parkeres mens der ventes på manglende oplysninger.*

**Hyppighed:** *Totalt: Ca. 600 henvendelser pr. døgn. Pr. bruger: Op til 40 pr. dag.*

**Vanskeligt:** *(aldrig)*

**Brugere:** *I første omgang lægeselekretær, men opgaven kan overdrages til andre.*

Delopgaver og varianter:	Eksempel på løsning:	Kode:
1. Registrér patient. (Se databeskrivelse D10).		
1a. Patient findes i systemet. Opdater data.		
1b. Indskriv også en rask ledsager.		
2. Opret forløb og registrér data, bl.a. den foreløbige diagnose. (Se databeskrivelse D1 og D2, herunder støtte til brug af SKS klassifikationen).		
2a. Overfør data elektronisk fra egen læge, etc.	<i>MedCom-formaterne bruges.</i>	
2p. Problem: Der kan være tale om afvigende codesystemer og strukturer i de elektroniske meddelelser.	<i>Støt den manuelle overførsel af data fra den elektroniske blanket til systemets form.</i>	
2q. Problem: Patienten kan være i gang med andre forløb samtidig og kan være indskrevet flere steder og i flere afdelinger. Det kan være svært at overskue hvem der har plejansvar og hvem der stiller seng til rådighed.		
3. Udskriv indkaldelsesbrev (eller anden form for indkaldelse).		
3a. Patienten overføres til venteliste.		
3b. Der mangler oplysninger og sagen parkeres midlertidigt med tidsovervågning.		
3c. Sagen overdrages til en anden, evt. med tidsovervågning.		
3d. Evt. afslås henvendelsen.		
4. Rekvirér tolk til indkaldelsestidspunktet.		

#### C2. Indskriv . . .

6. En angivelse af hvilken slags brugere der udfører disse opgaver, de omgivelser det sker i, etc. Hvis det ikke er beskrevet samlet under arbejdsområdet, må det beskrives her.
7. En liste af delopgaver, varianter og problemer. Delopgaverne nummereres fortløbende. Varianter af den enkelte delopgave anføres med bogstaverne a, b, osv. En variant betyder at samme delopgave kan udføres på en anden måde. Fx kan man enten finde sagen frem eller oprette en ny sag.

Bemærk at man også her bruger bydemåde (fx *opret patient*) for at skjule hvem der gør hvad.

Problemer der vedrører delopgaven anføres med bogstaverne p, q, osv. Selv om delopgaverne er nummereret, skal de ikke nødvendigvis udføres i den rækkefølge, og de skal heller ikke nødvendigvis udføres alle sammen hver gang.

Mange delopgaver består i at registrere og bruge data i systemet, men der er også delopgaver der går videre, fx advisere andre om noget, ophælde medicin, sende et brev. Det er vigtigt også at anføre disse delopgaver, selvom de i dag udføres manuelt, for måske har leverandøren en løsning man ikke har tænkt på.

**Eksempel på løsning.** Kunden har mulighed for at anføre eksempler på løsninger i midterkolonnen, men brug det sparsomt. Der er ingen grund til at tvinge sig selv til at skrive et eller andet "genialt" her. Skriv kun noget hvis det er en ikke-triviell mulighed.

**Brug af "problemer".** Det er vigtigt at bruge *problemer* på den rigtige måde. Et problem skal være noget der volder brugeren problemer med den nuværende løsning. Problem 2q er et godt eksempel. Kunden ønsker at leverandøren skal løse det problem. Man ser desværre ofte *problem* brugt om noget man mener vil give problemer for leverandøren eller brugerne af det nye system. Det er ikke tanken. Vil man nævne den slags, hører det hjemme i midtersøjlen under "løsning".

**Hvad er kravet?** Skabelonen anfører lige efter kapiteloverskriften, at kravet er at støtte delopgaver og varianter, og så vidt muligt løse problemerne. Dvs. at kun punkt 7 ovenfor er krav. Punkterne 1 til 6 er forudsætninger for kravene, dvs. de omstændigheder under hvilke kravene skal opfyldes.

## Opgaver kontra delopgaver

En arbejdsopgave bør afspejle en faktisk arbejdsituation, dvs. noget en bruger udfører i ét stræk fra start til slut, hvorefter brugeren kan begynde på noget andet. Man bruger udtrykket at *opgaven er afsluttet når brugeren har fortjent en kaffepause*. Systemet bør effektivt støtte brugeren indtil "kaffepausen".

En delopgave har ikke samme form for afslutning. Når delopgaven er udført, vil brugeren straks fortsætte med en anden delopgave i samme arbejdsopgave (med mindre hele arbejdsopgaven nu er afsluttet).

Det kan ske at der er så mange mulige delopgaver i en arbejdsopgave, at beskrivelsen bliver for lang og uoverskuelig. Man kan så tage nogle af delopgaverne ud og give dem et separat C-nummer. Skabelonen viser et eksempel i C10-6. Ordination



af medicin sker ideelt under den kliniske session, men der er mange delopgaver i ordinationen. Ordination er derfor blevet til en lang delopgave, C11. Der optræder andre lange delopgaver i en klinisk session, fx booking af tid til scanning eller operation. De kan beskrives som C13, booking. Bemærk at en lang delopgave ikke skal have start og slut - den er blot en del af en egentlig arbejdsopgave. Derimod har det mening at anføre hyppigheden fordi ikke alle kliniske sessioner har ordinationer, mens nogle kan have mange.

## Arbejdsområde 2: Patientbehandling

Området omfatter . . .

### C10. Klinisk session

*En klinisk session kan omfatte diagnose, planlægning, udførelse, vurdering, mv. Som regel udføres der lidt af det hele, men det kan også ske at der fx kun udføres planlægning.*

**Start:** Kontakt med patienten eller konference om patienten.

**Slut:** Når der ikke skal gøres mere med patienten lige nu.

**Hyppighed:** Totalt: Ca. 15.000 pr. døgn. Pr. bruger: Op til 20 pr. døgn.

**Vanskeligt:** Katastrofesituationer med mange tilskadekomne.

**Brugere:** . . .

Delopgaver og varianter:	Eksempel på løsning:	Kode:
1. Identificér patient.	Mulighed for armbånd der scannes, fx til bevidstløse patienter.	
2. Vurdér patientens tilstand. Se åbne diagnoser og tilhørende fokuserede oplysninger. Se notater. Se resultater fra ydelser der tidligere er rekvireret og sammenhold dem med operationelle mål. (Det data der skal overskues fremgår af databeskrivelserne D1, D3, D4 og D5).	Overblik over det hele på ét skærbillede, gerne så man også kan overskue det tidsmæssige forløb, fx med Gantt diagram. Valg af detaljer fra oversigtsbilledet.	
3. Giv ydelser der kan gives med det samme (lokale ydelser, fx blodtryk og SAT).	Registrér resultater straks.	
4. Følg op på planlagte ydelser og resultater. Kontrollér om tidsfrister er udløbet.	Overblik over bestilte ydelser og færdige resultater.	
5. Justér diagnoser (ret, tilføj, slet, prioritér). Afstem med vejledninger. Skriv notater.	Registrér ændringer straks. Der kan med det samme registreres notater.	
5p. Problem: Besværligt at få adgang til vejledninger.	Vejledninger og checklister er tilgængelige on-line ud fra en valgt diagnose eller ydelse. Standardpakker af ydelser angiver hvad der skal gøres.	
6. Planlæg og bestil nye ydelser. Afstem med ledige tidspunkter hos alle parter - også patienten. (Se delopgaver C11, C12 og C13 om medicinordination, rekvisitioner og booking).	Ydelsen rekvireres straks. For tidsbestemte bookinger fastlægges tidspunktet straks.	
6p. Problem: Man glemmer dele af bestillingen.	Standardpakker af ydelser. Gældende praksis (vejledning) ses on-line.	
6q. Problem: Fejl når data først noteres på papir, senere tastes ind.	Data indtastes straks.	
7. Evt. afslutning af forløb. (Se opgave C6).		

Ved kundehenvendelser sker det ofte at sagen ikke kan gøres færdig med det samme. Dvs. at den faktiske arbejdsopgave kun tager sig af en del af den logiske opgave - at håndtere henvendelsen. Både C1 og C10 i eksemplet har denne karakter. Ved en indskrivning kan det ske at der mangler oplysninger. Så må sagen midlertidigt parkeres, hvilket sker i C1-3b. På samme måde er en klinisk session (C10) kun en del af behandlingen. Arbejdsopgaven er forbi når man ikke kan gøre mere for patienten lige nu. Til gengæld vedrører de første delopgaver at få et overblik over patientens status - hvor langt er vi kommet i indskrivningen eller behandlingen.

## Kun få arbejdsopgaver

Bruger man arbejdsopgave-princippet rigtigt, bliver der ret få arbejdsopgaver, selv i meget store systemer. Typisk er der 10-30 arbejdsopgaver. Hertil kan komme et tilsvarende antal lange delopgaver. Det er en fordel med få arbejdsopgaver fordi det giver bedre overblik og reducerer arbejdet med at skrive kravene.

Oftest ser man kravspecifikationer hvor 10 arbejdsopgaver er blevet formuleret som 100 *use cases*, der hver fylder en eller flere sider, selvom der ikke sker ret meget i hver af dem. Det skyldes at man gør hver delopgave til en separat lille "arbejdsopgave" med start og slut, hyppighed, osv. I virkelighedens verden er disse delopgaver ikke adskilte opgaver, men læseren får ingen fornemmelse af hvordan de hænger sammen. Her er et eksempel fra EPJ-verdenen:

### (En uhensigtsmæssig form for krav)

#### Use case 2.1. Vis diagnoser

Den sundhedsfaglige person ønsker at opnå et overblik over patientens samlede diagnoser og disses indbyrdes relationer i diagnosehierarkiet.

**Start:** Brugeren ønsker at orientere sig om udviklingen i patientens helbredstilstand.

**Slut:** . . .

Trin:	Eksempel på løsning:	Kode:
1. Vis diagnosehierarki		
2. Vælg visning	Fx et hierarki, Gantt-diagram . . .	
3. Vælg detaljeringsniveau	Fx udvid og indskrænk med plus og minus	
4. Vis notater om en udpeget diagnose		
5. Vis dato og forfatter for notatet		
6. Vis eventuelle eksterne årsager for diagnosen		
. . . osv. i alt 8 trin		

Dette er ikke en sand arbejdsopgave fordi den som regel ikke er afsluttet i "kaffe-pause" forstand. Den vil være en delopgave i en større arbejdsopgave. I dette tilfælde er den blot en lille del af opgave C10. Den er i øvrigt også alt for detaljeret, så den nærmest har foreskrevet en ganske bestemt brugerdialog.

Man kan ikke se det egentlige formål med sådan en use case. Hvorfor skal brugeren have et overblik over diagnoserne? Er det for at finde en behandling, for at forklare et nyt symptom, eller for at skrive en rapport om patienten? Selv om man får støtte til denne lille use case, er det ikke sikkert at der er god støtte til den samlede ar-

bejdsopgave. I værste fald kan man risikere at brugeren må skrive diagnoserne på papir for at kunne udføre næste trin af den samlede arbejdsopgave.

## Undgå at beskrive data som delopgaver

En anden ting der får beskrivelserne til at svulme op, er når man forsøger at beskrive data som delopgaver. Use case 2.1 ovenfor er også et eksempel på det. Notater, dato og eksterne årsager beskrives som separate delopgaver. Da det samme data ofte bruges i flere arbejdsopgaver, bliver det hurtigt svært at holde beskrivelserne konsistente.

Løsningen er at beskrive data separat. Fra delopgaverne kan man kort referere til det data der er relevant i denne delopgave. Se fx skabelonens krav C1-2 og C10-2. I nogle tilfælde kan det være nødvendigt med en mere præcis opremsning af det relevante data, men man behøver ikke bruge en delopgave til hvert dataelement. Det fylder bare og man taber overblikket.

### C11. Ordiner medicin til patient

*Dette er ikke en selvstændig arbejdsopgave, men en længere delopgave der altid indgår i en klinisk session. (Derfor er der ingen start- og slutangivelse).*

**Hyppeghed:** Totalt: Ca. 30.000 pr. døgn. Pr. bruger: Op til 20 pr. døgn.

Delopgaver og varianter:	Eksempel på løsning:	Kode:
1. <i>Vurdér patientens samlede medicineringssituation, både i dette forløb og andre forløb.</i>	<i>Overblik over alle oplysningerne på en gang, helst inkl. CAVE, diagnoser, mv.</i>	
1p. <i>Problem: Besværligt at få adgang til vejledninger.</i>	<i>Vejledninger og tjeklister er tilgængelige on-line ud fra en diagnose eller medicintype.</i>	
...		
6. <i>Beregn dosis. Man skal senere kunne se hvordan dosis er beregnet.</i>	<i>Systemet tilbyder en beregning baseret på fx legemsvægt og kan evt. selv hente vægten i journalen.</i>	
6p. <i>Problem: Omregning mellem enheder. Der kan være forskel på den enhed man doserer i (fx mg) og den enhed der dispenseres i (fx antal tabletter).</i>	<i>Både doseringsenheder og dispenseringsenheder vises.</i>	
...		

## Højniveau opgaver - oversigt, ikke krav

Det kan være nyttigt at vise det totale forløb af et eller andet, fx behandlingsforløbet for en patient. Man kan også vise livscyklus for et eller andet, fx en diagnose-type fra den overvejes af en lægefaglig kommission til den mange år senere nedlægges af en anden kommission. Her er behandlingsforløbet for en patient:

### Forløb 1: Behandlingsforløb

**Start:** Patienten henvises af egen læge eller kommer akut.

**Slut:** Patienten er helbredt.

Trin:	Løsning:
1. Indskriv patienten	Se C1 og C2.
2. Stil diagnoser	Se C10.
3. Planlæg behandling	Se C10.
4. Udfør behandling	Se C10.
5. Vurder resultat	Se C10.
6. Udskriv patient	Se C20 og C21.
7. Efterbehandling eller hjemmepleje	?

Det er det logiske, ideelle forløb. Med sådan en højniveau opgave opdager man tit nye behov for it-støtte, her fx behovet for samspil med hjemmeplejen (markeret med et spørgsmålstegn ved trin 7). Det kan være en god idé at beskrive højniveau opgaverne i kravspecifikationen, fx forrest i kapitel C, men det skal klart fremgå at der ikke er tale om krav. Det er kun til orientering for læseren. Derfor har vi i eksemplet udeladt kolonne tre, og i løsningskolonnen henviser vi til de krav der tager sig af de enkelte trin.

Det er fristende at udstykke højniveau opgaven i flere separate arbejdsopgaver: indskriv patient, stil diagnose, planlæg behandling, osv. Hver opgave støttes så af egne skærmbilleder.

Desværre bliver resultatet ofte en meget dårlig støtte til de faktiske arbejdsopgaver. De kan nemlig gå på tværs af de idealiserede trin i højniveau opgaven. I eksemplet har vi vist hvordan trinene udføres gennem de faktiske arbejdsopgaver. Trin 1 bliver udført under C1 og C2 (indskriv patient inden ankomst og indskriv akut). Alle trinene fra 2 til 5 bliver derimod udført under den samme arbejdsopgave, C10, klinisk session.

Er de faktiske arbejdsopgaver virkelig sådan? Ja, især i lidt mere komplekse sager. For ældre patienter stilles der fx ofte flere diagnoser og man prøver at behandle dem alle under samme indlæggelse. Det betyder at én enkelt klinisk session kan vedrøre en ny diagnose, behandling for en tidligere diagnose, vurdering af en tredje, osv. Det fanges udmærket af opgave C10, der netop lægger vægt på en samlet oversigt over det hele og muligheden for både diagnose og behandling i samme session.

(Blank med vilje)

## D. Data systemet skal anvende

Dette kapitel beskriver de data systemet skal opbevare. Der er først en oversigt, dels på verbal form, dels i form af en datamodel (E/R model) eller en UML klasse-model. Man vil ofte nøjes med at vise et udvalg af attributterne (datafelterne) i modellen.

Skabelonens datamodel er ikke UML, men en meget ældre notation. Den er valgt fordi den giver langt bedre overblik, fx bruger den typisk kun 1/5 så meget plads. Det skyldes flere ting: Attributterne står uden for kasserne, som derved bliver meget mindre. Forbindelsernes kardinalitet (en-til-mange, mange-til-mange, osv.) er vist som "hønsenben" i stedet for som tal og symboler. (I UML skriver man fx 0:1, 1:\*, osv.). Hønsenbenene gør det langt lettere at overskue sammenhængene mellem

## D. Data systemet skal anvende

Systemet skal anvende de data der er beskrevet i dette kapitel. Data skal kunne oprettes, ses og ændres gennem de relevante arbejdsopgaver. Data skal i mange tilfælde kunne udveksles med omgivende systemer som beskrevet i kapitel F.

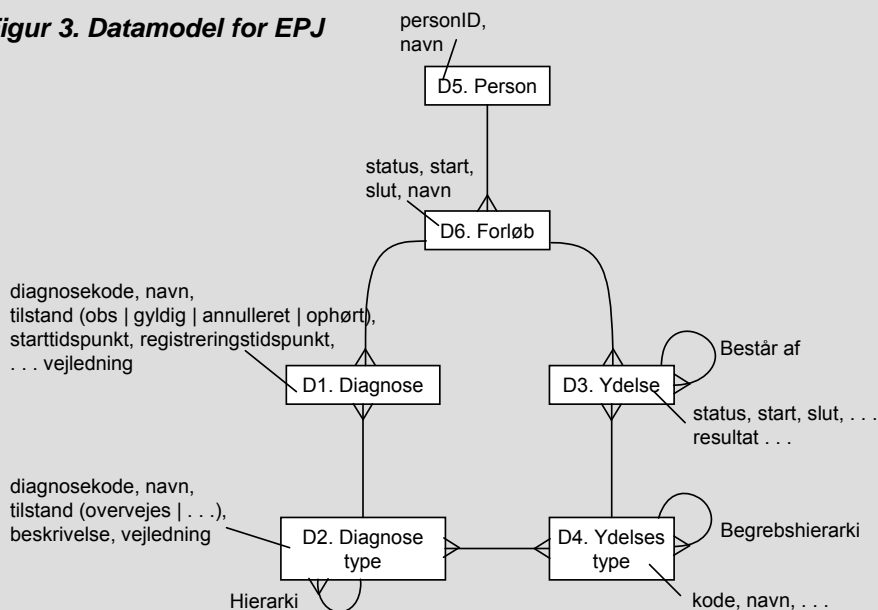
Figur 3 er et E/R diagram der giver en oversigt over data. Data skal ikke nødvendigvis struktureres på denne måde i systemet, men det skal håndteres på en eller anden måde. Her er en kort kommentar til figurens kasser.

**D1. Diagnoser:** Patientens sygdomme, såvel formodede som konstaterede. Svarer til SKS-klassifikationen af sygdomme, men der skal også være mulighed for at registrere sygdomme der ikke er i SKS eller som først senere klassificeres.

**D2. Diagnosetyper:** Et katalog over de forskellige typer diagnoser - uafhængigt af patienten. Brugere vil vælge diagnoser fra dette katalog. Diagnosetypen angiver diagnosens navn og SKS-kode (hvor muligt), vejledning, typiske standardpakker af ydelser. Ændres løbende af kunden ud fra erfaringer.

...

**Figur 3. Datamodel for EPJ**



flere kasser på én gang. Endelig er der brugt linier med bløde kurver i stedet for knækkede linier. Det gør det lettere at se hvad der er forbindelser og hvad der er kasser, især i større diagrammer.

## D1. Diagnoser

Data beskrives i detaljer klasse for klasse. (Man kan dog undertiden med fordel beskrive flere relaterede klasser under et). Beskrivelsen af en klasse består af disse dele:

1. Nummer og navn på klassen. Klasserne nummereres D1, D2 osv. For at undgå for meget omnummerering under arbejdet med kravene, kan man gruppere klasserne i bundter og starte hvert bundt med et rundt nummer.
2. En kort indledning til hvad klassen drejer sig om, meget gerne med eksempler på typisk og usædvanlig anvendelse af data.
3. Det forventede datavolumen. I kapitel D står datavolumen udenfor tabellen og

### D1. Diagnoser

*En diagnose er en sygdom eller et symptom hos en konkret patient. Der er en flydende overgang mellem sygdomme og symptomer. Fx er både kolera og hoste "diagnoser".*

**Datavolumen:** Der registreres op til 800.000 diagnoser pr. år.

Felter og relationer:	Eksempel på løsning:	Kode:
<p>1. <i>Diagnosekode: Givet via relationen til Diagnosetype (dvs. SKS-kode, ICPC-kode eller midlertidig kode). Diagnosekoden kan ændres under behandlingen. Den endelige diagnosekode bruges ved afregning og indberetning.</i></p>	<p>a. <i>Mulighed for at søge i et begrebshierarki (svarende til SKS over- og underklasser).</i>  b. <i>Reduceret hierarki tilpasset den enkelte afdeling så de normalt kun ser diagnosetyper relevante for dem.</i>  c. <i>"Live search" hvor brugeren indtaster dele af diagnosens navn og systemet for hvert tegn viser mulige match.</i></p>	
<p>1p. <i>Problem: Meget svært at finde den rigtige SKS-kode.</i></p>		
<p>2. <i>Navn: Normalt navnet fra Diagnosetype, men kan også være et navn indtastet specielt for denne patient.</i></p>		
<p>3. <i>Tilstand: En diagnose kan være i følgende tilstande: Obs, gyldig, annulleret, ophørt.</i></p>		
<p>4. <i>Starttidspunkt: Det tidspunkt hvorfra diagnosen er i denne tilstand. Er ofte det samme som registreringstidspunktet, men ikke altid, fx hvis man registrerer at patienten havde hoste i går.</i></p>	<p><i>Systemet gør det let at vælge registreringstidspunktet som Starttidspunkt.</i></p>	
<p>5. <i>Registreringstidspunkt: Alle ændringer af diagnosen registreres, og de tidligere udgaver af diagnosen bevares i systemet. Normalt er klinikerne kun interesseret i den sidste udgave af diagnosen.</i></p>	<p><i>Registreres automatisk af systemet.</i></p>	
<p>...</p>		
<p>17. <i>Vejledning: Den vejledning der var gældende på det tidspunkt hvor diagnosen blev oprettet.</i></p>		

er derfor ikke et krav. Egentlige krav til datavolumen står i afsnit L3.

4. En tabel med detaljer om data. Hver række angiver et attribut (et felt eller en forbindelse til en anden kasse). Attributterne nummereres fortløbende. Problemer der vedrører anvendelserne af et attribut anføres med bogstaverne p, q, osv. Listen er delt i tre søjler, helt analogt til beskrivelsen af arbejdsopgaver.

Det er en god idé at skrive lidt om hvor data kommer fra og hvad det bruges til. D1-1 er et godt eksempel. I skabelonen er der kun sparsomme forklaringer for de andre felter. I praksis bør der stå væsentligt mere.

I skabelonens eksempel er beskrivelsen på et højt niveau, men der kan være behov for også at angive detaljer om dataformat, tekstlængder, etc.

### **D10. Data i eksisterende systemer og standarder**

Det kan undertiden være svært for kunden at samle oplysningerne om data på den anbefalede form. Nødløsningen er at levere en udskrift af den eksisterende databases struktur og/eller print af eksisterende blanketter, skærbilleder og rapporter. D10-1 viser et eksempel hvor man henviser til skærbilleder for et eksisterende system. Denne form giver leverandøren en vis baggrund for at vurdere hvad kunden har brug for, men der er en risiko for misforståelser og senere konflikter.

D10-2 viser et eksempel hvor man henviser til en standard for adresseoplysninger. Standarder har det med at være meget omfattende, så man skal passe på ikke at bede om flere data end der reelt er brug for.



## D2. Diagnosetyper

En diagnosetype beskriver den del af diagnosen der er uafhængig af den konkrete patient. Samlingen af diagnosetyper udgør diagnosekataloget. Når klinikerer registrerer en diagnose for en patient, vælger han en diagnosetype fra kataloget.

**Datavolumen:** Der vil være op til 30.000 diagnosetyper. SKS har i dag ca. 20.000 typer.

Felter og relationer:	Eksempel på løsning:	Kode:
1. <i>Diagnosekode: SKS-kode, ICPC-kode eller midlertidig kode.</i>		
2. <i>Navn: Det fulde navn på diagnosen, fx "kolera uden specifikation".</i>		
3. <i>Tilstand: En diagnosetype kan være i følgende tilstande: Overvejes, gyldig, forældet.</i>		
4. <i>Hierarki: Diagnosetyperne er organiseret i et begrebshierarki, fx sådan som SKS er indrettet. En diagnosetype refererer således til den diagnosetype der er ovenover i hierarkiet.</i>	<i>Brugeren bør kunne vælge en diagnose på et lidt højere niveau, fx "Kolera, DA00" i stedet for "Kolera uden specifikation, DA009"</i>	
5. <i>Beskrivelse: En længere beskrivelse af diagnosen, dog ikke over 200 tegn. Endnu længere beskrivelser kan evt. findes i "Vejledning".</i>		
...		
10. <i>Vejledning: En vejledning der hører til diagnosen.</i>	<i>Kan fx være en URL.</i>	

...

## D10. Data i eksisterende systemer og standarder

Ud over data beskrevet ovenfor, skal systemet også behandle følgende data som findes i de eksisterende systemer eller standarder.

**Datavolumen:** ...

System:	Eksempel på løsning:	Kode:
1. <i>Patientadministrativt system (PAS). Se felterne markeret i skærbillede x og y.</i>		
2. <i>Adresseoplysninger fra Katalog over offentlige it-standarder, OIOXML, AddressSpecific.</i>		
...		

## E. Andre funktionelle krav

De fleste systemfunktioner er simple oprettelser, sletninger, editeringer og forespørgsler der implicit fremgår af arbejdsopgaverne og datakravene. Dette kapitel beskriver funktionalitet der er mere kompleks.

### E1. Komplekse beregninger og regler

Beregninger og regler kan beskrives på flere måder.

Funktion 1 i eksemplet viser en beregning der er beskrevet i et separat bilag (ventelisteberegning). Bilaget kan fx indeholde en algoritme skrevet som et lille program, et *flow chart*, et *activity diagram*, eller en tabel over mulighederne.

Funktion 2 viser en regel udtrykt som et tilstandsdiagram.

Funktion 3 henviser til et offentligt dokument hvor reglerne er beskrevet (lønoverenskomster).

Man kan også indirekte beskrive en funktion ved et nøjagtighedskrav, fx at systemet skal kunne genkende menneskelig tale med en baggrundsstøj på 30 dB. Eller at systemet skal kunne beregne en vagtplan der højst er 3% dyrere end den optimale plan.

### E2. Udskrifter og rapporter

Ofte er man i den situation at det eksisterende system kan udskrive et hav af rapporter, men for de fleste ved man ikke om de bruges og til hvad. Skabelonen viser flere muligheder for at omsætte det til krav.

Udskrift 1 har en veldefineret brug og man kan beskrive formatet præcist, fx ved at vedlægge et eksempel.

Udskrift 2 forklarer behovet, men giver ikke et præcist format. Ofte kan det være en fordel at henvise til den eller de arbejdsopgaver hvor rapporten skal bruges, så leverandøren bedre kan vurdere hvad der er hensigtsmæssigt.

Udskriftskrav 3 omgår problemet ved at bede leverandøren give en fast stykpris for at lave en rapport. På den måde kan man vente med at finde ud af hvilke rapporter man har brug for, til man har valgt en leverandør. Den faste pris forhindrer leverandøren i at misbruge det monopol han har efter indgåelse af kontrakten. Prisen kan evt. være afhængig af rapportens kompleksitet, fx udtrykt ved en pris pr. *function point*.

Udskriftskrav 4 omgår problemet på en anden måde, nemlig ved at bede om en rapportgenerator så kunden selv kan udvikle rapporterne. I eksemplet beder man leverandøren specificere hvor let det er at udvikle rapporterne, fx ved at angive hvilke typer brugere der kan gøre det, og hvor lang uddannelse det kræver. (Se også kravene om brugervenlighed i afsnit I).

Udskriftskrav 5 præciserer at alle rapporter skal kunne vises både på skærm og print.

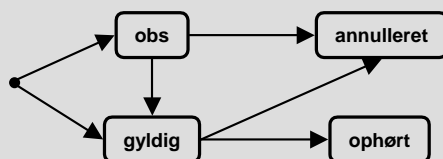
## E. Andre funktionelle krav

### E1. Komplekse beregninger og regler

De fleste systemfunktioner er simple oprettelser, sletninger, editeringer og forespørgsler, som ikke er specificeret yderligere. Funktionaliteten fremgår implicit af arbejdsopgaverne (kapitel C) og databeskrivelserne (kapitel D). Systemet skal også kunne udføre følgende mere komplekse funktioner.

Funktion:	Eksempel på løsning:	Kode:
1. Ventelisteprioritering skal beregnes som beskrevet i . . .		
2. En diagnose må normalt kun skifte tilstand som beskrevet i figur 4. I tilfælde af fejltagelser skal brugeren dog have mulighed for at afvige fra reglerne (se også H4-2).	En bruger der forsøger at afvige fra reglerne, spørges om det er hensigten. I så fald udføres ændringen og den logges i . . .	
3. Lønberegningen skal følge de til enhver tid gældende overenskomster (se også vedligeholdskraverne i . . .).		

Figur 4. Diagnosetilstande



### E2. Udskrifter og rapporter

Nogle udskrifter og rapporter skal bruges i forbindelse med de konkrete arbejdsopgaver beskrevet i kapitel C. Udskriftsformaterne er ikke afgørende, når blot arbejdsopgaven støttes effektivt. Disse rapporter beskrives ikke yderligere her. Der er dog også behov for udskrifter der bruges til ad hoc formål, på tværs af arbejdsopgaver, og udskrifter med et præcist format. De beskrives her.

Udskriftskrav:	Eksempel på løsning:	Kode:												
1. Checks skal udskrives på fortrykte blanketter med formatet vist i . . .														
2. Systemet skal kunne vise en oversigt og prognose over sengebelægningen (bruges bl.a. i arbejdsopgave . . .).	Figur . . . viser et eksempel på en sådan udskrift.													
3. Leverandøren skal påtage sig at levere op til 100 nye rapporter til en fast stykpris som et led i vedligeholdelsen.	Leverandøren bedes specificere stk. prisen, evt. afhængig af rapportens kompleksitet.													
4. Systemet skal indeholde en rapportgenerator der er let at bruge.	Hvor stor en del af medarbejderne i følgende grupper ville efter et kursus på ___ dage kunne udvikle rapporttyperne i bilag x: <table style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th></th> <th>type 1</th> <th>type 2</th> </tr> </thead> <tbody> <tr> <td>almindelige brugere</td> <td>___%</td> <td>___%</td> </tr> <tr> <td>superbrugere</td> <td>___%</td> <td>___%</td> </tr> <tr> <td>kundens it-afdeling</td> <td>___%</td> <td>___%</td> </tr> </tbody> </table>		type 1	type 2	almindelige brugere	___%	___%	superbrugere	___%	___%	kundens it-afdeling	___%	___%	
	type 1	type 2												
almindelige brugere	___%	___%												
superbrugere	___%	___%												
kundens it-afdeling	___%	___%												
5. Alle udskrifter og rapporter skal kunne vises på skærm såvel som i print.														
. . .														

### **E3. Udbygning af systemet**

I nogle tilfælde er der behov for at kunden selv kan udbygge systemet på et eller andet område. Det kan fx skyldes at man vil have mulighed for at eksperimentere med nye skærbilleder, eller at man frygter at leverandøren ikke vil udbygge systemet til en rimelig pris og på en rimelig tid.

Dette afsnit stiller krav om funktionalitet der gør det muligt for kunden at foretage visse udbygninger. Tidligere var leverandører meget utilbøjelige til at åbne for den slags, fordi man ikke mente at kunne indestå for systemets korrekthed og stabilitet efter sådanne udbygninger. Dette har dog ændret sig, og selv i ERP-systemer som SAP og Axapta kommer der bedre og bedre muligheder for at kunden kan udbygge.

I EPJ-eksemplet er behovet meget udtalt fordi der er over 20.000 ydelser med hver deres datafelter, og nye kommer til hele tiden. Man vil ikke kunne leve med at leverandøren skal rette i systemet hver gang en ny slags ydelse kommer til. Desuden er der en lang række lægefaglige specialer som har brug for oversigter på hver deres facon.

Bemærk at der ikke alene stilles krav om funktionalitet, men også om ret til at bruge den. Det skyldes tidligere dårlige erfaringer hvor leverandøren har eneret til at bruge funktionaliteten og de data systemet opbevarer.

### E3. Udbygning af systemet

Systemet viser og vedligeholder data gennem brugerens skærbilleder. Kunden forventer at hans egen stab eller tredjepart bemyndiget af ham kan modificere skærbillederne og tilføje nye, sådan at der kan skabes overblik for lægefaglige specialer, nye arbejdsgange, etc. Systemet håndterer mange slags ydelser, mange af dem med særlige kombinationer af data. Kunden forventer at der kan tilføjes nye typer ydelser uden at leverandøren skal indblandes. I dette afsnit betyder "kunden" hans egen stab eller tredjepart bemyndiget af ham. Nedenstående krav konkretiserer behovene.

Krav til udbygningsmuligheder:	Eksempel på løsning:	Kode:
1. Kunden kan definere nye ydelsestyper og tilhørende skærbilleder, jvf. databeskrivelsen i kapitel D.		
2. Kunden kan definere skærbilleder der kombinerer data fra hele datamodellen i kapitel D (views på data).		
3. Skærbillederne kan aktivere funktionalitet i EPJ-systemet og i eksterne systemer forbundet med EPJ-systemet.	Fx bestilling af en service, notifikation, udskrifter.	
4. Skærbillederne kan sammensættes af mange typer komponenter og deres farve kan afspejle datas værdi.	Fx tekstfelter, tabeller, knapper, grafer, billeder.	
5. Kunden kan tilføje nye typer komponenter til brug i skærbillederne.		
6. Der kan defineres skærbilleder for flere slags udstyr.	Fx PC, PDA, Smartphone.	

Dokumentation og rettigheder:	Eksempel på løsning:	Kode:
7. Værktøjerne til opbygning af billeder, tilføjelse af komponenttyper, mv. skal dokumenteres så kundens stab eller tredjepart kan forstå dem og bruge dem til formålet.	Leverandøren bedes beskrive hvilke målgrupper der kan bruge værktøjet og hvor lang tids uddannelse der kræves.	
8. Kunden skal have ret til at bruge værktøjerne og systemets data til formålet.		

## **F. Systemets integration med eksterne systemer**

Der er som regel behov for integration med andre systemer - *eksterne* systemer. Skabelonen giver en verbal oversigt over de eksterne systemer samt en oversigt i form af et kontekstdiagram. Det system der skal leveres er vist i midten af diagrammet, omgivet af de eksterne systemer. Man viser som regel også nogle af de arbejdsområder eller brugerprofiler der direkte anvender systemet.

### **F1. SKS**

Dette afsnit er et eksempel på en meget løs integration med et eksisterende system, Sundhedsstyrelsens klassifikationssystem, SKS, der blot består af nogle tabeller enhver kan downloade. Der er en indledning der giver forudsætningerne for kravene, helt analogt med kravene om arbejdsopgaver der skal støttes. Bemærk at der henvises til arbejdsopgaver hvor integrationen skal benyttes. De tekniske grænseflader er beskrevet tilstrækkeligt detaljeret til at leverandøren kan vurdere hvor komplekst problemet er.

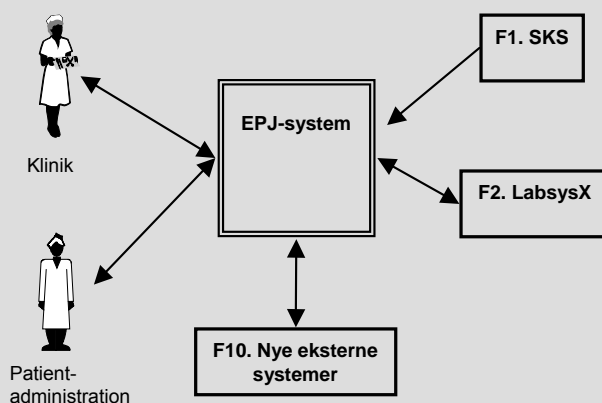
Kravene viser at behovet for dataaktualitet ikke er stort på dette område, så ambitionerne kan være lave. Der nævnes et problem man har i dag om konflikt med lokale løsninger og foreslår en løsning.

## F. Systemets integration med eksterne systemer

Systemet skal i større eller mindre grad integreres med de systemer der er vist i figur 5 (kontekst diagram). Integrationen består dels i datadeling, dels i at brugeren via systemet kan aktivere funktionalitet i andre systemer (eksterne systemer). Her er en kort kommentar til figurens kasser:

- F1. SKS: Sundhedsstyrelsens klassifikationssystem. Opdateres regelmæssigt af Sundhedsstyrelsen.  
 F2. LabsysX: Kundens eksisterende laboratoriesystem til . . .  
 F3. . . .  
 F10. Et nyt eksternt system som kunden senere vil indkøbe.

**Figur 5. Kontekstdiagram**



### F1. SKS

SKS-tabellerne omfatter koder for såvel diagnoser som ydelser og bruges derfor ved de fleste arbejdsopgaver. Det er normalt tilstrækkeligt at man har adgang til de nyeste tabeller med en uges forsinkelse. Tabellerne er bl.a. tilgængelige som zip-tekstfiler med fast feltbredde fra Sundhedsstyrelsens web-site.

**Datavolumen:** SKS-tabellerne udgør op til 100.000 records a ca. 100 tegn.

Integrationsbehov:	Eksempel på løsning:	Kode:
1. EPJ-systemet skal anvende SKS-data der er rimeligt aktuelt.	Systemet overfører data hver__ dag. (Kunden forventer 7 dage). Eller: EPJ-systemet overfører nye data når de annonceres af Sundhedsstyrelsen.	
1p. Det sker at de nye SKS-data giver problemer, fx at de er i konflikt med lokale koder.	Driftsafdelingen kan rulle SKS-data tilbage til forrige version.	
2. I særlige tilfælde kan der være behov for større dataaktualitet.	Driftsafdelingen kan starte en dataoverførsel.	

## **F2. LabsysX**

Dette afsnit er et eksempel på en tæt integration med et eksisterende system. Igen henvises der til arbejdsopgaver hvor integrationen skal benyttes.

Undertiden kan man helt undvære særlige integrationskrav fordi fuld støtte af arbejdsopgaverne kræver integration (det har vi fx gjort i C1-2a, brug af MedCom). Men som regel er integration en meget kompliceret affære, og det vil være uhyre vanskeligt at vurdere en leverandørs integrationsløsning blot ved at se arbejdsopgaverne blive udført. Det gælder især hvis man tidligt vil vurdere hvor godt arbejdsopgaverne støttes (B2). Derfor er der behov for særskilte integrationskrav.

Bemærk at der henvises til en beskrivelse af de tekniske grænseflader. Det er desværre almindeligt at kunden ikke dokumenterer de tekniske grænseflader, men det gør det meget dyrt for en leverandør at give et tilbud. Ofte skal han skaffe kernekompetence der kender de pågældende eksterne systemer. Samtidig er det meget risikable krav at skulle opfylde. Resultatet kan blive at gode leverandører slet ikke byder.

Leverandøren beskriver graden af integration gennem en gruppe krav. Nogle af kravene angiver en lav grad af integration, som kunden helst vil undgå. Det er altså unødvendigt at leverandøren tilbyder en løsning her.

Krav 1 er den laveste form for integration, hvor det tilbudte system blot giver mulighed for at logge ind på det eksterne system. Det er hvad kunden har i dag, og det er selvfølgelig et problem fordi identifikationer skal testes ind igen. En løsning på problemet kunne være en automatisk overførsel af identifikationer.

Ved krav 2 kan brugeren indtaste data til det eksterne system og se resultater fra det gennem de skærmbilleder som brugeren alligevel anvender i EPJ-systemet. Det eksterne system er således næsten usynligt for brugeren.

Ved krav 3 er der yderligere funktionalitet fælles for de to systemer.

Krav 4, 5, 6 og 7 angiver graden af datadeling mellem de to systemer. Ved krav 4 overføres data periodisk. Ved krav 5 kan data overføres på brugerens initiativ. Ved krav 6 overføres data automatisk når nye data foreligger. Ved krav 7 deler de to systemer deres database. Hvis leverandøren ikke deler data på præcis en af disse måder, kan han beskrive sin løsning i midterste søjle.

Man ser ofte at kunden prioriterer tæt datadeling meget højt, dvs. som grad 7, i den hensigt at brugeren altid skal se det allerseneste data. Det kan blive en rigtig dyr løsning, især hvis leverandøren allerede har et standardsystem. Han kan blive nødt til at ændre sit system overordentlig mange steder - selv hvis det er nydeligt opbygget med en lagdelt arkitektur. Og et system der er ændret så mange steder kan ikke vedligeholdes sammen med grundversionen, så vedligehold bliver også meget dyrt.



Tæt datadeling er ikke helt så vigtigt som man forestiller sig. Man skal nemlig huske på at der under alle omstændigheder går en væsentlig tid fra det øjeblik brugeren ser data til han har handlet ud fra det, fx gjort noget ved patienten. Dataændringer der sker imens brugeren handler, vil således ikke blive opdaget, selv ved grad 7. En lille forsinkelse mellem de to systemers dataoverførsel er altså relativt ligegyldig. En løsning som grad 5 eller 6 vil ofte være fuldt tilstrækkelig.

## F2. LabsysX

*LabsysX bruges i forbindelse med arbejdsopgave C10. De tekniske grænseflader til LabsysX er beskrevet i ...*

**Hypighed:** Totalt leveres der op til 400 prøveresultater pr. døgn.

Leverandøren bedes beskrive den tilbudte grad af integration gennem følgende gruppe krav. Hvis leverandøren som standard tilbyder en høj grad af integration (fx krav 2), er der ikke behov for en lavere grad (fx krav 1), og leverandøren kan for den lave grad angive kode 5 (tilbydes ikke).

Grad af integration:	Eksempel på løsning:	Kode:
1. Brugeren starter LabsysX gennem EPJ-systemet, logger dernæst ind på LabsysX hvor han indtaster patientens cpr og bestiller ydelsen gennem LabsysX's skærbilleder.	(En lav grad af integration)	
1p. Problem: Det er besværligt og risikabelt at skulle logge ind igen og angive patientens cpr.	Bruger-ID og cpr overføres automatisk.	
2. Brugeren bestiller LabsysX ydelsen gennem EPJ-systemets skærbilleder på samme måde som for andre ydelser.	EPJ-systemet bruger API grænsefladerne til LabsysX.	
3. EPJ-systemet kan advisere brugerne om færdige eller udeblevne resultater fra LabsysX på samme måde som for andre ydelser.		
4. Data fra LabsysX overføres periodisk til EPJ-systemet (replikerede databaser).		
5. Data for en enkelt patient kan overføres på brugerens initiativ.		
6. Data fra LabsysX overføres på LabsysX's initiativ når de er til rådighed.		
6p. Problem: Når et eller begge systemer har været ude af drift, kan en overførsel let gå tabt.		
7. EPJ-systemet deler database med LabsysX således at data altid er identisk i de to systemer.		

...

## **F10. Integration med nye eksterne systemer**

Når man først har anskaffet systemet, kan det blive meget dyrt at få det integreret med nye eksterne systemer. Leverandøren vil nemlig normalt have monopol på at udføre den slags ændringer. Dette afsnit af skabelonen viser hvordan man kan beskytte sig mod et urimeligt monopol ved at tredjepart kan udføre den slags ændringer.

Den første gruppe krav beskriver hvordan systemet kan fungere som server for nye eksterne systemer. Krav 1 angiver at data kan hentes og opdateres. I en traditionel database-orienteret arkitektur, kan et eksternt system (eller et adapter-program) hente data gennem SQL-forespørgsler, der har en høj grad af fleksibilitet. I en service-orienteret arkitektur (SOA) hentes data gennem services der ikke har samme grad af fleksibilitet. Her bliver der behov for let at kunne tilføje nye services for at sikre en rimelig svartid. Det beskrives som problem 1p, med tilhørende løsnings-eksempler.

Krav 2 angiver behovet for anden funktionalitet der skal være til rådighed for det eksterne system.

Den anden gruppe krav beskriver hvordan systemet kan fungere som klient for et nyt eksternt system. Krav 3-5 beskriver graden af datadeling ligesom for F2. Krav 6 angiver behovet for at det tilbudte system kan trække på anden funktionalitet i det nye system.

Endelig angiver den sidste gruppe krav den dokumentation der skal foreligge af de tekniske grænseflader (krav 7 og 10), og de rettigheder tredjepart skal have til at bruge dem (krav 8 og 9). Bemærk hvordan man kan verificere dokumentationens kvalitet ved at lade et typisk tredjeparts softwarehus afprøve dokumentationen. Det bør ske tidligt så denne form for dokumentation med stor sandsynlighed bliver tilstrækkelig for tredjeparts udbygning af systemet (se afsnit B2).

## F10. Integration med nye eksterne systemer

Kunden forventer at nye eksterne systemer kan integreres med *EPJ-systemet* af tredjepart, fx af et uafhængigt softwarehus. Tredjepart vil typisk levere adapter-moduler der oversætter data og beskeder mellem de to systemer.

Systemets grænseflader i serverrollen:	Eksempel på løsning:	Kode:
1. <i>Det nye eksterne system kan hente og opdatere data i EPJ-systemet. Data omfatter det der er beskrevet i kapitel D.</i>		
1p. Problem: Hvis der er tale om en service-orienteret arkitektur, er der ofte behov for nye services for at sikre en rimelig svartid.	Tredjepart kan tilføje en ny service. Eller leverandøren kan gøre det til en fast pris.	
2. <i>Det nye eksterne system kan anvende funktionalitet i EPJ-systemet, såsom advisering til brugeren og print på printere der administreres af EPJ-systemet.</i>	<i>Leverandøren bedes beskrive den funktionalitet der tilbydes.</i>	

Systemets grænseflader i klientrollen:	Eksempel på løsning:	Kode:
3. <i>EPJ-systemet kan periodisk overføre data fra det nye system (replikerede databaser).</i>		
4. <i>EPJ-systemet kan på brugerens initiativ overføre data for en enkelt patient fra det nye system.</i>		
5. <i>EPJ-systemet kan bruge det nye systems database således at data altid er identisk i de to systemer.</i>		
6. <i>EPJ-systemet kan anvende funktionalitet i det nye system, såsom bestilling af ydelser og advarsler om udeblevne eller ændrede ydelser.</i>	<i>Leverandøren bedes beskrive den funktionalitet der kan anvendes.</i>	

Dokumentation og rettigheder:	Eksempel på løsning:	Kode:
7. Den tekniske grænseflade til <i>EPJ-systemet</i> skal dokumenteres. Dokumentationen skal kunne forstås af et typisk tredjeparts software-hus og findes egnet til integrationsformålet.	Et kursus på ___ dage er nødvendigt for at kunne anvende dokumentationen i praksis. Et uddrag af dokumentationen bedes fremsendt sammen med tilbuddet (jvf. B2-4).	
8. Kunden og tredjepart bemyndiget af kunden skal have ret til at bruge dokumentationen og selve grænsefladerne.		
9. Kunden og tredjepart bemyndiget af kunden skal have mulighed for og ret til at udtrække og bruge alt data beskrevet i kapitel D, fx ved konvertering til et andet system.		
10. Svartiderne for de forskellige funktioner på grænsefladen skal specificeres.		

## G. Teknisk it-arkitektur

Ordet *it-arkitektur* har efterhånden fået to helt forskellige betydninger. Den klassiske er konfigurationen af hardware, software, datatransmission, mv. Det er den *tekniske it-arkitektur*. Den nye betydning omfatter både den tekniske arkitektur, drift, support, datamodel, brugervenlighed, mv. Kort sagt det meste af hvad denne kravskabelon allerede handler om.

Dette kapitel af skabelonen beskriver den tekniske it-arkitektur, dels kundens eksisterende it-udstyr (hardware, software, mv.), dels behovet for nyt it-udstyr. Eksemplet viser kun situationer hvor kunden selv står for anskaffelse af udstyret. Hvis kunden ønsker total-leverancer af hardware, software, mv. er krav om svar-tider (L1), datalagring (L3), mv. tilstrækkelige.

### G1. Brug af eksisterende hardware og software

Her beskriver kunden dels det eksisterende udstyr, dels de opgaver der allerede kører på dette udstyr. Uden disse oplysninger, kan leverandøren ikke vurdere om hans system kræver ekstra ressourcer. Det vil være endnu bedre hvis kunden også kan give driftsstatistik, specielt for spidsbelastningsperioder.

Krav 1 og 2 beder leverandøren oplyse hvor mange brugere systemet kan trække med det nuværende udstyr, og hvad der kræves derudover for nye brugere.

### G2. Nyt hardware og software

Her bedes leverandøren oplyse hvilket udstyr kunden skal anskaffe og hvordan det skalerer op med antallet af brugere.

Krav 3 specificerer at der kun bør anvendes udstyr fra kundens liste af foretrukne teknologier. Det kan fx være vigtigt hvis kunden allerede har ekspertise på disse områder.

## G. Teknisk it-arkitektur

### G1. Brug af eksisterende hardware og software

Kunden har i øjeblikket følgende it-udstyr der forventes brugt til drift af systemet:

1. 20 servere af typen . . .
2. 300 PC'er med Windows XP og mindst 100 GB disk.
3. Fibernet . . .
4. Oracle database . . .

Belægningen af udstyret til andre opgaver i spidsbelastningsperioder forventes i fremtiden at være:

5. Servere . . .
6. Der kører ikke andre opgaver på PC'erne når de kører EPJ-systemet.
7. Fibernet belastes ingen steder mere end 50% i gennemsnit over en periode på 60 sekunder.
8. . . .

Krav til kundens hardware og software:	Eksempel på løsning:	Kode:
1. Systemet skal i starten køre på det eksisterende udstyr uden udbygning, og overholde kravene i L1 og L2 for et begrænset antal brugere.	Systemet kan under disse omstændigheder trække ____ brugere. (Kunden forventer 20 brugere).	
2. Når antallet af brugere vokser, skal systemet gradvis udbygges for at overholde kravene i L1 og L2 indtil det fulde antal brugere.	Systemet skal udbygges med ____ for hver ____ brugere.	

### G2. Nyt hardware og software

For at anvende systemet, skal kunden anskaffe følgende hardware og software:

Krav til kundens hardware og software:	Eksempel på løsning:	Kode:
1. Kunden skal anskaffe nyt udstyr for at systemet kan overholde kravene i L1 og L2 for et begrænset antal brugere.	Kunden skal anskaffe ____. Systemet kan under disse omstændigheder trække ____ brugere.	
2. Når antallet af brugere vokser, skal systemet gradvis udbygges for at overholde kravene i L1 og L2 indtil det fulde antal brugere.	Systemet skal udbygges med ____ for hver ____ brugere.	
3. Der skal så vidt muligt kun bruges hardware og software fra positiv-listerne i bilag x.		

## **H. Sikkerhed**

Sikkerhed omfatter mange områder. Skabelonen viser de vigtigste. Hvis systemet ikke blot kan basere sig på de sikkerhedsteknikker der allerede er i drift, må det tilrådes at inddrage en sikkerhedsekspert i arbejdet med sikkerhedskravene.

### **H1. Adgangsret for brugere**

Dette afsnit beskriver kontrollen af at kun autoriserede brugere får adgang. Kravene beskrives som en slags arbejdsopgave der skal støttes, samt problemer der bør fjernes.

Resultatet er at de velkendte teknikker, såsom password, time-out og single sign-on, nu optræder som eksempler på støtte af et behov. Det giver leverandøren mulighed for at tilbyde andre løsninger.

En vigtig ting er den liste af rettigheder der står under kravtabellen. Det er et punkt der næsten altid forsømmes og som er vigtigt for leverandørens vurdering af kompleksiteten. Problemet med at give de rigtige brugere de rigtige rettigheder er ikke så stort, men kontrollen af de mulige rettigheder skal håndteres dybt inde i systemet.

## H. Sikkerhed

### H1. Adgangsret for brugere

Login, mv. er ikke selvstændige arbejdsopgaver for brugerne, men delopgaver der optræder i mange forskellige arbejdsopgaver. Systemet skal støtte følgende delopgaver i forbindelse med brugernes adgang til systemet.

Delopgaver vedr. brugeres adgang:	Eksempel på løsning:	Kode:
1. Identificér brugeren.	Hver bruger har et brugernavn og et password. Gerne også en alternativ form for identifikation, fx stemmegenkendelse eller fingeraftryk.	
2. Brugeren har været væk fra systemet i et stykke tid.		
2p. Problem: En anden bruger kan anvende systemet med den førstes rettigheder.	Systemet timer ud efter 5 min.	
2q. Problem: Hvis systemet selv logger ud, er det besværligt at logge på igen.	Systemet kræver kun password og er så parat til brug. Timeout perioden er sted-afhængig.	
3. Kontrollér at kun autoriserede brugere har adgang til systemer/data. (Se listen af rettigheder nedenfor).	Rettighederne kontrolleres i database-systemet. Eller: Brugers skærbilleder viser kun de funktioner han har ret til at bruge.	
3p. Problem: I dag har brugerne password for hvert system. Tungt at skifte system. Svært at skifte password regelmæssigt.	Hver bruger har kun ét brugernavn og ét password (single sign-on).	

#### Rettigheder

1. Ret til at ordinere medicin på afdeling M.
2. Ret til at se patientdata på afdeling M.
3. Ret til at registrere klinisk data (ydelser og diagnoser) på afdeling M.
4. . . .

*En læge på afdeling M kan fx have rettighederne 1, 2 og 3, mens en tilsynsførende på afdeling M kun har rettighed 2 og 3.*

## **H2. Sikkerhedsadministration**

Sikkerhedsadministrationen beskrives som en arbejdsopgave der skal støttes. En række nuværende problemer med opgaverne nævnes, fx at der ved månedsskift er et stort antal brugere der skal oprettes.

Igen bliver flere velkendte teknikker nu til mulige løsninger, fx rollebaseret tildeling af rettigheder og tidsbegrænsede rettigheder.

## **H3. Sikring mod tab af data**

Her er nævnt nogle typiske risici for tab af data, og leverandøren bedes angive sin løsning. Der kunne tilføjes mange andre risici for tab af data, fx disk-nedbrud og transmissionsfejl.



## H2. Sikkerhedsadministration

Sikkerhedsadministrationens arbejde omfatter nedenstående delopgaver.

**Vanskeligt:** Når 120 nye medarbejdere er ansat og skal have rettigheder fra dag 1.

Delopgaver vedr. sikkerhedsadministration:	Eksempel på løsning:	Kode:
1. Tildel eller fjern rettigheder for en bruger.		
1a. Brugeren skal oprettes.		
1p. Problem: En lang række brugere skal oprettes, fx lige inden de tiltræder ved månedsstart.	Automatisk overførsel af data fra personalesystemet.	
1q. Problem: En hasteindkaldt vikar er endnu ikke kommet ind i personalesystemet, men skal have adgangsret.	Mulighed for midlertidig oprettelse på afdelingen udenom sikkerhedsadministrationen.	
1r. Problem: Sikkerhedsadministrationen skal holde styr på sammenhængen mellem 4000 brugere og 300 rettigheder.	Hver bruger tildeles en eller flere roller, fx læge i afdeling M og tilsynsførende i afdeling N. Hver rolle har en eller flere rettigheder, fx ordination og diagnostik.	
1s. Problem: Sikkerhedsadministrationen glemmer at oprette og fjerne rettigheder på de rigtige tidspunkter, fx i forbindelse med ansættelse og fratræden af medarbejdere.	Rettigheder kan defineres i god tid så de gælder for en periode, fx fra den dag personen ansættes.	
2. Opret nye typer roller.		
3. Få oversigt over hvem der har ret til hvad og om der fx er en rettighed som ingen har.		

## H3. Sikring mod tab af data

Data kan utilsigtet tabes eller fejltypes på flere måder.

Systemet skal beskytte mod:	Eksempel på løsning:	Kode:
1. Tab eller dublering af data ved overførsel mellem to systemer, fx i forbindelse med at ét eller begge systemer går ned.		
2. Udelelighedsproblemer, fx at bruger A træffer beslutning om en medicin-ordination, men inden systemet registrerer beslutningen, har bruger B ordineret en medicin der interagerer. Hverken bruger A eller B vil opdage konflikten.		

#### **H4. Sikring mod utilsigtet brugeradfærd**

Her tænkes dels på sikring af datakvalitet ved at systemet kontrollerer for konsistens, mv., dels på beskyttelse mod ting som brugeren kommer til at gøre af en fejltagelse.

#### **H5. Sikring mod trusler**

Her tænkes på den lange række af trusler fra virus, hacking, etc. Det er en god idé at lave en risikoanalyse. Her gennemgår man de mulige trusler, vurderer sandsynligheden for at de indtræffer og konsekvensen hvis det sker (helst i kroner og øre). Ud fra det kan man finde den "gennemsnitlige" skade på et år for hver trussel.

Dernæst vælger man de alvorlige trusler og beder leverandøren foreslå en beskyttelse mod dem. Krav 1 og 2 viser hvordan det kan skrives for hacking og for wire-tapping af password.

Krav 3, at uvedkommende ikke må få adgang til personoplysninger, lyder enkelt men omfatter i virkeligheden en lang række trusler, fx at nogle kan tappe oplysningerne under datatransmission, at it-driftspersonalet kan se oplysningerne på disken, at sikkerhedsadministrationen kommer til at give en ny medarbejder forkerte rettigheder, osv. Leverandørens svar kan let blive til en hel roman - og to leverandørers romaner er svære at sammenligne. Man bør i stedet nedbryde problemet til en række separate trusler.

Krav 4 prøver at løse problemet ved at henvise til en lov, *Lov om behandling af personoplysninger*. Her opstår ofte et interessant spil. Kunden har ikke læst den pågældende lov, men forestiller sig at den sikrer mod problemet (det gør den kun delvis). Hvis han nu giver leverandøren besked på at overholde loven, så har leverandøren vel overtaget ansvaret?

Leverandøren kender muligvis loven og ved at den ikke dækker tilstrækkeligt. Han ved også at kundens formål med kravet bare er at fraskrive sig ansvaret, og at overholdelse af loven ikke kan verificeres ved afleveringsprøven. Hvorfor skulle han dog gøre opmærksom på det? Resultat? Det egentlige behov bliver ikke dækket.

#### H4. Sikring mod utilsigtet brugeradfærd

Ved utilsigtet brugeradfærd forstås at brugeren kommer til at gøre noget han ikke havde til hensigt, fx ved at ramme den forkerte tast eller bruge en kommando der har en anden virkning end han troede.

Krav:	Eksempel på løsning:	Kode:
1. Utilsigtede brugerhandlinger må ikke kunne få systemet til at bryde sammen, hverken på klienten eller serveren.		
2. Alt indtastet data skal kontrolleres for format, konsistens og rimelighed. I tvivlstilfælde skal brugeren advares og tage stilling.		
3. Brugeren skal let kunne rette fejltagelser.	Udbredt brug af <i>undo</i> .	

#### H5. Sikring mod trusler

En risikovurdering har vist at følgende trusler er alvorlige, og systemet skal beskytte mod dem.

Systemet skal beskytte mod:	Eksempel på løsning:	Kode:
1. At uvedkommende skaffer sig administratorrettigheder via internettet (hacking).	Rettighederne kan kun benyttes internt.	
2. Wire-tapping af password.	Kryptering af password.	
3. At uvedkommende får adgang til personoplysninger. [Utilstrækkeligt, se vejledning.]		
4. Systemet skal overholde Lov om behandling af personoplysninger (Lov nr. 429 af 31. maj 2000). [Utilstrækkeligt, se vejledning.]		

## I. Brugervenlighed og design

Brugervenlighed betyder at systemet er let at lære, effektivt i daglig brug, let at huske for lejlighedsvis brugere, let at forstå også i usædvanlige situationer, og behageligt at bruge. Disse såkaldte usability-faktorer er ikke alle lige vigtige - det afhænger af hvad det er for et system.

Når vi taler om brugervenlighed, forudsætter vi at systemet teknisk set virker korrekt, og at man faktisk kan udføre de opgaver der er behov for. Alligevel har brugere problemer med at udføre opgaverne.

Brugervenlighed kan måles på mange måder. En af de vigtigste er at observere brugere udføre nogle realistiske arbejdsopgaver med systemet eller en primitiv prototype af det. Man noterer ned når brugeren ikke kan udføre opgaven uden hjælp udefra, når brugeren er lang tid om at finde løsningen, mv. Dette kaldes en *usability test*. De problemer man noterer ned, kaldes usability problemer. De kan rimeligt objektivt klassificeres som kritiske og mindre alvorlige. Man kan så stille krav om at der kun må være et vist antal kritiske usability problemer.

Et kritisk problem er en situation hvor brugeren:

- a. ikke på egen hånd kan udføre arbejdsopgaven,
- b. eller tror den er udført uden at den faktisk er udført,
- c. eller giver udtryk for at *det her er dog besværligt*,
- d. eller testlederen kan se at brugeren ikke bruger systemet på en effektiv måde.

Hvis man beder brugeren tænke højt undervejs, kaldes det et tænke-højt forsøg. Det giver bedre muligheder for at forstå hvorfor brugeren ikke fandt løsningen, og dermed bedre muligheder for at leverandøren kan fjerne problemerne.

Erfaringen viser at usability problemer skal opdages og rettes meget tidligt, det vil sige inden der er programmeret noget. Ellers vil det være alt for dyrt at rette problemerne. I praksis sker det ved at lave prototyper af skærbillederne med papir og blyant eller simple computer-værktøjer. Man bruger så prototypen til tænke-højt forsøg. De fleste usability problemer kan faktisk afsløres på den måde. Disse erfaringer er baggrunden for de tidlige beviser i B2-2.

### I1. Indlæring og effektivitet i daglig brug

Eksemplet i dette afsnit stiller krav om brugervenlighed på sådan en måde, at man tidligt kan vurdere om systemet kan blive tilstrækkeligt brugervenligt. Samtidig giver det leverandøren hjælp til at levere den nødvendige brugervenlighed.

Det kan være svært på forhånd at stille krav om antallet af usability problemer og hvordan de skal observeres. I skabelonens eksempel kunne man fx forestille sig at udelade krav 1 og nøjes med krav 2 til 6, dvs. kræve at brugerne selv kan udføre alle arbejdsopgaver efter en kort instruktion, kan forstå fejlmeddelelserne, osv.

Leverandøren skulle så i tilbudet svare på hvor mange usability problemer der måtte være, hvilke testopgaver der skulle bruges, osv. Det er stort set umuligt, især hvis systemet slet ikke findes endnu. Det gælder altså om at finde frem til nogle rimelige krav lidt inde i projektet.

Derfor er krav 1 indsat. Man skal tidligt udføre usability tests, rette de mest alvorlige usability problemer, teste igen, osv. Kunden har i midterkolonnen givet et eksempel på hvordan testen kan udføres. Leverandøren kan som sædvanlig erstatte det med sit eget tilbud. Desuden skal man fastsætte detaljerede usability krav, fx hvor mange usability problemer der kan accepteres. Disse detaljerede krav skal testes når systemet leveres.

Krav 2 til 6 er eksempler på usability krav der kan være givet på forhånd, men skal præciseres i den tidlige fase. Fx skal midterkolonnen udfyldes med tal.

Krav 2 kontrollerer at brugerne efter den planlagte introduktion kan udføre deres daglige arbejdsopgaver med minimal hjælp fra andre.

## I. Brugervenlighed og design

### 11. Indlæring og effektivitet i daglig brug

Det skal sikres at systemet kan opnå den fornødne brugervenlighed. Det bør ske ved tidlige usability-tests. Efter de tidlige tests, aftaler parterne de detaljerede krav der skal verificeres ved overtagelsen. Det kan fx være præcisering af de arbejdsopgaver der skal udføres og de tal der skal indsættes i midterkolonnen

*Kan man ikke enes om de detaljerede krav, kan aftalen hæves (jvf. afsnit B2-2).*

Krav til tidligt bevis:	Eksempler på løsning:	Kode:
1. Brugergrænsefladen skal tidligt usability-testes af leverandøren. De mest alvorlige usability problemer skal rettes indtil usability testen giver tilfredsstillende resultat. Desuden aftales de detaljerede usability krav.	For dele af systemet der allerede findes, udføres der tænke-højt test i en passende systemopsætning. For dele der ikke findes endnu, udføres der tænke-højt test af papirprototyper. Der testes med tre nye brugere i hver runde.	
Krav der skal aftales i detaljer efter det tidlige bevis, og verificeres ved overtagelsen:		
2. Efter en kort instruktion fra superbrugerne, skal brugerne kunne udføre alle arbejdsopgaverne i kapitel C, inden for deres respektive arbejdsområder, uden væsentlige usability-problemer.	Inden for hvert arbejdsområde udføres tænke-højt test af det færdige system med 3 medarbejdere. Der må højst observeres __ kritiske usability-problemer.	
3. Fejlmeddelelserne skal være forståelige og hjælpsomme.	Under usability-testen vises et udvalg af fejlmeddelelser for brugeren, som skal forklare hvad meddelelsen betyder og hvad han skal gøre. __% af forklaringerne skal være acceptable.	
4. Systemet skal kunne betjenes uden brug af mus, og brugerne skal selv kunne lære at bruge tastaturet i stedet.	Ved usability-testen kontrolleres det at brugerne selv lærer det. __% af brugerne skal selv finde ud af det.	
5. Superbrugere skal hurtigt kunne lære at betjene systemet så de kan instruere andre brugere i det (jvf. J2-1).	Uddannelsen af superbrugere kan ske på __ dage.	
6. <i>En bruger der har brugt systemet dagligt i en uge skal hurtigt kunne bestille en række ydelser, fx . . . , til samme patient.</i>	<i>En typisk bruger kan bestille sådanne 5 ydelser på __ minutter.</i>	

Krav 3 kontrollerer at fejlmeddelelserne er brugbare. Under en usability test kommer brugeren kun i berøring med en lille del af systemets fejlmeddelelser. Krav 3 kan teste en større del af meddelelserne, også de der kun vil forekomme sjældent.

Krav 4 siger at systemet skal kunne betjenes uden mus, og at brugerne selv skal kunne lære det. Dette krav er selvfølgelig ikke altid relevant, fx hvis der er tale om et web-system som kun bruges lejlighedsvis.

Krav 5 handler om store systemer som en typisk bruger ikke på egen hånd kan finde ud af. Man kunne her stille krav om at der skal være kurser til rådighed (se afsnit J2). Men det er ofte en dyr og dårlig mulighed. Krav 5 kræver derfor at kun superbrugere skal på kursus, og at de herefter skal kunne træne de andre brugere.

Krav 6 handler om hvor effektivt systemet er til hyppige arbejdsopgaver. Under den tidlige usability test kan man kun få en fornemmelse af hvor hurtigt opgaven burde kunne udføres, men det kan reelt først måles når systemet er næsten færdigt.

### **Web-systemer**

Eksemplets krav er beregnet til produktionssystemer i daglig brug. Krav 2, 5 og 6 er fx ikke egnede til web-sites der kun bruges lejlighedsvis. Dels er der ingen superbrugere der kan give instruktion, dels er effektivitet i daglig brug ikke væsentligt. Krav 4 er stadig aktuelt hvis web-sitet skal kunne bruges af svagsynede.

## 12. Tilgængelighed og Look-and-Feel

Nogle aspekter af brugervenlighed er svære at udtrykke gennem usability tests. Det gælder fx standarder der bør overholdes og tilgængelighed for særlige brugergrupper.

Krav 1 siger at brugergrænsefladen skal følge MS-Windows standarden. Bemærk begrundelsen om at brugerne er vant til den, underforstået at systemet så vil være lettere at lære. Har man ikke en god begrundelse, er der ingen grund til at kræve en standard. En standard kan højst bidrage lidt til brugervenligheden. Den sikrer ikke i sig selv god brugervenlighed. Undertiden kan standarden endda være en hindring for brugervenligheden. Desuden er det i praksis forbløffende svært at afgøre om en sådan standard er overholdt.

Krav 2 siger at brugergrænsefladen skal være egnet for blinde, svagsynede, mv. En mulig løsning er at følge HTML principperne, der netop er udviklet til det formål. De siger fx at man skal bruge de standardiserede overskriftsmarkeringer i stedet for selvdefinerede, mere æstetiske formater. Det giver højt-læsningsprogrammerne mulighed for at bruge intonation til at angive hvad der er overskrifter. På samme måde skal man undgå faste kolonnebredder og bogstavstørrelser, så svagsynede kan forstørre teksten kraftigt.

Nogle kravspecifikationer erstatter hele krav 2 med et krav om at web-siderne skal kunne passere en W3C Markup Validation test (<http://validator.w3.org>). Testen analyserer en web-side og påpeger fejl. Her er atter et eksempel på at man henviser til en standard i den tro at så er man dækket. Testen finder kun formelle fejl, fx at der mangler anførelstegn eller slutelementer. Den siger intet om fx egnethed til blinde eller svagsynede. Det gør derimod de andre retningslinier i WCAG10.

Krav 3 er et eksempel på at man har brug for at præcisere hvilket sprog der skal bruges på grænsefladen.

### 12. Tilgængelighed og Look-and-Feel

Krav:	Eksempel på løsning:	Kode:
1. Brugergrænsefladen skal følge MS-Windows standarden, som de fleste brugere er vant til.		
2. Web-siderne skal kunne læses højt, skalere til svagsynede, samt udnytte den fulde skærmstørrelse på såvel små som store skærme.	Siderne følger HTML-principperne for Accessibility (WCAG10 fra W3C).	
3. <i>Brugergrænsefladen skal være på dansk. Web-delen om åbningstider, telefonnumre og adresser skal dog være tilgængelig på både dansk, engelsk, tyrkisk og urdu.</i>		

## J. Andre krav og leverancer

Dette kapitel samler en række krav af anden art.

### J1. Andre standarder der skal følges

De fleste standarder der skal følges, hører hjemme under et af de andre kapitler, fx om data eller sikkerhed. I dette afsnit kan man anføre de resterende standarder.

I praksis ser man at kunden lister den ene standard efter den anden, ofte uden at vide hvad den egentlig omfatter og hvilket behov den dækker. Det er som regel meget besværligt at kontrollere om en standard er overholdt, eller endda umuligt fordi standarden er for vag. For en ansvarsbevidst leverandør betyder det en forhøjelse af prisen, mens en mindre omhyggelig leverandør regner med at kunden alligevel ikke vil kontrollere om standarderne er opfyldt (se eksemplerne i H5 og I2).

Eksemplet viser kun et enkelt eksempel på en standard (af den bløde slags). Det præciseres at leverandøren skal sørge for certificeringen, dvs. en uafhængig kontrol af at systemet overholder standarden.

### J2. Uddannelse

Ofte glemmes uddannelse af brugerne, eller der kræves et urealistisk omfang. Man ser også at uddannelsen kommer på et forkert tidspunkt, fx længe inden overtagelsen så brugerne har glemt det hele når systemet endelig kommer i drift.

Krav 1 er et eksempel hvor man har indset at leverandøren kun behøver at uddanne superbrugere. De skal til gengæld blive i stand til at uddanne de andre brugere. Det er en erkendelse af at de fleste leverandørkurser er for fjernt fra brugerne egentlige arbejdsopgaver. Man bruger derfor superbrugere som mellemlid. Det præciseres også hvad man skal kunne efter uddannelsen.

Krav 2 stiller tilsvarende krav for medarbejderne i drift og support.

Krav 3 præciserer hvornår uddannelsen skal finde sted i forhold til overtagelsen af systemet.



## J. Andre krav og leverancer

### J1. Andre standarder der skal følges

Krav:	Eksempler på løsning:	Kode:
1. <i>Systemet skal overholde god regnskabskik. Leverandøren skal sørge for certificeringen.</i>		
2. ...		

### J2. Uddannelse

*Kunden ønsker at varetage en stor del af uddannelsen selv. Det tænkes gjort ved først at uddanne superbrugere, der så kan uddanne andre.*

Krav:	Eksempler på løsning:	Kode:
1. <i>Leverandøren skal uddanne 50 superbrugere, så de kan varetage uddannelsen af andre medarbejdere. Uddannelsen skal gøre superbrugerne i stand til at udføre alle arbejdsopgaverne i kapitel C, inklusive alle varianter, inden for deres respektive arbejdsområder.</i>	<i>Uddannelsen af en superbruger kan ske på ___ dage. (Kunden forventer at tre dage er nok).</i>	
2. <i>Leverandøren skal uddanne 10 it-medarbejdere så de kan varetage kundens del af drift og support.</i>	<i>Uddannelsen af en it-medarbejder kan ske på ___ dage. (Kunden forventer at 10 dage er nok).</i>	
3. <i>Uddannelserne skal gennemføres inden for den sidste måned før overtagelsen, således at medarbejderne straks kan bruge systemet og ikke allerede har glemt hvordan. Om nødvendigt må uddannelsen gentages og overtagelsen udskydes.</i>		
4. ...		

### **J3. Dokumentation**

Også dokumentation glemmes ofte. I eksemplet præciseres det at der for dette system ikke behøver være fuld dokumentation for alle. Det er i erkendelse af at meget få brugere anvender dokumentation eller on-line hjælp, selv hvis den er til rådighed og i rimelig brugbar form. Denne erkendelse kan spare mange omkostninger og frustrationer hos begge parter.

Krav 1, 2 og 5 angiver at der skal være kursusmateriale som superbrugerne kan anvende når de underviser. Det skal foreligge i sådan en form at superbrugerne selv kan tilpasse det. Der skal desuden være fuld dokumentation til superbrugerne selv. Det præciseres også hvornår dokumentationen skal foreligge.

Krav 3 og 4 angiver på samme måde dokumentation til drift, support og videreudvikling. Nogle gange vil man også kræve dokumentation af specielt udviklede programmer så kunden eller tredjepart om nødvendigt kan overtage vedligeholdelsen af dem. For at sikre at leverandøren rent faktisk kan levere den nødvendige kvalitet, kan man i B2 bede om et tidligt bevis.

### **J4. Datakonvertering**

Datakonvertering er ofte en meget betydelig del af leverancens pris. I dette afsnit præciseres det hvad der skal konverteres. Det er vigtigt at kunden kan dokumentere dataformaterne, da leverandøren ellers må skaffe oplysningerne selv inden han kan give et forsvarligt tilbud. Det kan afholde gode leverandører fra at byde.

Validering af datakonverteringen er et stort problem, som leverandøren normalt har langt mere erfaring med end kunden. Derfor beder krav 3 leverandøren beskrive hvordan han vil validere konverteringen.

### **J5. Installation**

Dette afsnit præciserer hvem der installerer hvad, herunder de konverterede data.

### J3. Dokumentation

Det forventes at kun superbrugere, driftsfolk og systemudviklere vil læse dokumentationen. Der er derfor ikke behov for begynderdokumentation, bortset fra kursusmateriale.

Krav:	Eksempler på løsning:	Kode:
1. Der skal senest ved overtagelsen være kursusmateriale til rådighed som superbrugerne kan anvende når de underviser andre brugere. (Kunden bidrager selv med vejledning om hvordan de eksisterende arbejdsgange skal udføres i fremtiden, se K-10).		
2. Der skal senest en måned efter overtagelsen være dokumentation til rådighed af alle systemets funktioner set fra et brugerperspektiv. Dokumentationen skal være anvendelig for superbrugerne.		
3. Der skal senest ved overtagelsen være tilstrækkelig dokumentation til at kunden kan varetage sin del af drift og installation.		
4. For specielt udviklet software (kode 3) og tekniske grænseflader som tredjepart kan benytte, skal der senest to måneder efter overtagelsen være dokumentation til rådighed der er egnet til videreudvikling.		
5. Al dokumentation skal leveres på maskinlæsbar form, og kunden skal frit kunne modificere den og kopiere den til eget brug.		
6. ...		

### J4. Datakonvertering

Leverandøren skal konvertere følgende data fra de eksisterende systemer:	Eksempler på løsning:	Kode:
1. De data fra PAS systemet som kan opbevares i systemet. Formatet er beskrevet i ...		
2. De data fra XY systemet som kan opbevares i systemet. Data må overføres ved IBM 3270 emulering (se skærbilledformatet i ...)		
3. Alt konverteret data skal valideres.	Leverandøren bedes beskrive hvordan.	
4. ...		

### J5. Installation

Krav:	Eksempler på løsning:	Kode:
1. Leverandøren skal installere alt hvad leverancen omfatter, såvel hardware som software.		
2. Leverandøren skal ligeledes installere de konverterede data.		
3. ...		

## **K. Kundens leverancer**

Det meste af kravspecifikationen handler om hvad leverandøren skal yde, men et it-system er ikke bare noget man "ruller ind og sætter i stikkontakten". Kundens medarbejdere skal deltage på forskellig vis, og leverandørens personale skal måske have kontor og faciliteter under udvikling og idriftsættelse. Dette kapitel beskriver hvad kunden skal levere. Leverandøren kan evt. i midterkolonnen præcisere hvad han forventer at kunden skal levere. Desuden kan han tilføje nye punkter på listen.

I mange kontrakter er dette afsnit flyttet ud i et separat bilag.

Ligesom i de andre afsnit er kravene kun eksempler og ikke en udtømmende liste. Efter juridisk praksis skal alt hvad kunden skal levere være specificeret. Leverandøren kan ikke efter kontraktindgåelse kræve at få kontor til rådighed eller ekspertise på et eller andet område, med mindre det er specificeret i kontrakten eller bilagene. Leverandøren skal derfor have mulighed for at tilføje til listen.

## K. Kundens leverancer

Nedenstående liste af kundens leverancer skal være udtømmende, og leverandøren kan ikke forvente yderligere leverancer. Leverandøren må derfor tilføje til listen om nødvendigt.

Kunden leverer:	Eksempler på løsning:	Kode:
1. <i>Hardware, software og eksterne systemer som det nye system forudsætter (se detaljer i kapitel G). Udstyret skal være til rådighed når installationsprøven starter.</i>		N/A
2. <i>Kontor med tre it-arbejdspladser fra en måned før planlagt installationsprøve til en måned efter overtagelsen.</i>		N/A
3. <i>Uddrag af produktionsdata til testformål og det fulde produktionsdata til konvertering, med undtagelse af data fra XY systemet.</i>		N/A
4. <i>Testcases til anvendelsestest.</i>		N/A
5. <i>Ekspertise på anvendelsesområdet, svarende til en halv medarbejder over hele projektets løbetid.</i>		N/A
6. <i>Testpersoner til usability tests.</i>		N/A
7. <i>En halvtids projektleder og en halvtids sekretær.</i>		N/A
8. <i>Instruktører/superbrugere der selv lærer systemet så de kan uddanne menige brugere.</i>		N/A
9. <i>Ekspertise til validering af konverteret data.</i>		N/A
10. <i>Bidrag til kursusmateriale om fremtidige arbejdsgange (jvf. J3-1).</i>		

## L. Drift, support og vedligehold

Dette kapitel stiller krav til ydelser som leverandøren erlægger efter at selve produktet er leveret. Kravene kan kun delvis verificeres (testes) ved overtagelsen. Den egentlige verifikation må fx ske ved en senere driftsprøve eller ved granskning af logs og driftsstatistikker.

I mange kontrakter er dette afsnit flyttet ud i separate bilag.

### L1. Svartider

I eksemplet beskriver kunden den planlagte belastning af systemet i de mest travle perioder (spidsbelastning). Uden disse tal kan leverandøren ikke vurdere svartiderne og det nødvendige it-udstyr. Der kan være behov for at specificere mere om belastningen, fx at man kan acceptere forskellige svartider i forskellige perioder og for forskellige brugergrupper.

Belastningen er i eksemplet udtrykt på to alternative måder:

1. Antal brugere der udfører de forskellige arbejdsopgaver. Hvis leverandøren har et dybtgående kendskab til arbejdsområdet, kan han omsætte disse tal til antallet af it-transaktioner, som så tillader ham at vurdere svartiderne. Uden dette kendskab er det meget risikabelt at tilbyde en svartid.
2. Antallet af it-transaktioner. Hvis leverandøren ikke kender arbejdsområdet, bør kunden specificere antallet af it-transaktioner i stedet for antallet af brugere. Hvis kunden ikke kender tallet, bør kravet alligevel udtrykkes ved et antal transaktioner, da kunden er den nærmeste til at vurdere det.

Det præciseres hvornår og på hvilken konfiguration svartiderne skal måles. Bemærk forskellen på det daglige produktionsnetværk og web-baseret anvendelse. I eksemplet er der en web-del for den brede offentlighed, hvor mange stadig har en langsom forbindelse. Her skal leverandøren fx undgå en masse billeder, som vil forsinke visningen af siderne. Det præciseres også hvor mange af de målte svartider der skal være som anført i kravene (mere om det nedenfor).

Krav 1 siger blot at der er behov for at målingerne udføres regelmæssigt. Det overlades til leverandøren at angive hvor hyppigt og hvordan, fordi det afhænger af de praktiske muligheder.

Krav 2-8 angiver de egentlige svartidskrav. De nævnte svartider er baseret på ergonomiske målinger af hvor hurtigt brugere arbejder ved en computer (den såkaldte *key-stroke level model*). En hurtig bruger taster 5-10 tegn pr. sekund, så 0,2 sekunder for at gå fra ét felt på skærmen til det næste, vil ikke sinke brugerne.

Når en bruger skifter opmærksomhed fra én delaktivitet til en anden, er den mentale skiftetid omkring 1,3 sekunder. Hvis skærbillederne er fornuftigt organiseret, vil det derfor ikke sinke brugeren at det tager 1,3 sekunder at skifte.

## L. Drift, support og vedligehold

Dette kapitel stiller krav til ydelser som leverandøren erlægger efter at selve produktet er leveret. Kravene kan kun delvis verificeres (testes) ved overtagelsen. Den egentlige verifikation sker senere, ved driftsprøven.

### L1. Svartider

Systemets belastning varierer gennem døgnet og ugen. Svartiderne er især vigtige i de travleste timer, spidsbelastningsperioderne.

**Spidsbelastning** (Alternativ 1: Kun egnet hvis leverandørerne kender arbejdsområdet indgående):

1. Indtil 2000 brugere arbejder med klinisk session (C10)
2. Indtil 1000 brugere arbejder med administrative opgaver (C1 til C4)

**Spidsbelastning** (Alternativ 2: Egnet hvis leverandørerne ikke kender arbejdsområdet indgående):

1. Simple forespørgsler i klinisk session (C10): 10 pr. sekund
2. Opdateringer i klinisk session (C10): 2 pr. sekund
3. Simple forespørgsler i administrative opgaver (C1 til C4): 3 pr. sekund
4. ...

#### Måling af svartid

Svartiden er tidsintervallet fra brugeren afgiver sin kommando til resultatet er synligt og brugeren har mulighed for at afgive en ny kommando. Ved en kommando forstås et tastetryk eller et museklik. Alle målinger udføres i perioder med spidsbelastning. De anførte tider skal gælde for 95% af svartiderne i disse perioder.

**Brug via lokalnet:** Målingerne udføres på et lokalnet der svarer til daglig drift.

**Den offentlige web-del:** Målingerne udføres på en PC forbundet til Internettet med et 56kB modem og lav trafik på vejen til serverne (men spidsbelastning af serverne selv).

Det er vigtigt at systemets svartider er så korte at brugerne ikke sinkes i deres arbejde. Det søges opnået gennem følgende krav.

Krav til svartider:	Eksempel på løsning:	Kode:
1. Målingerne skal udføres regelmæssigt.	Der måles en gang ugentligt med stopur. Eller: Systemet måler selv løbende.	
2. Når man går fra et felt til det næste, skal brugerens tastehastighed ikke nedsættes.	Man kan taste inden ____s. (Kunden forventer 0,2 s).	
3. Når man går fra et skærbillede til det næste, skal man kunne se billedets data og taste inden for den mentale skiftetid.	Man kan se data og taste inden ____s. (Kunden forventer 1,3 s).	
4. Opslag i dropdown-lister skal tillade valg fra listen inden for den mentale skiftetid.	Man kan starte valget inden ____s. (Kunden forventer 1,3 s).	
5. Visning af simple rapporter på skærmen skal ske inden brugeren taber tålmodigheden.	Rapporten skal være synlig inden ____s. (Kunden forventer 20 s).	
6. Data skal kunne overføres fra eksterne systemer og vises på skærmen inden brugeren taber tålmodighedens. Herefter skal de kunne vises lige så hurtigt som andet data i en vis periode.	Man kan se data og taste inden ____s. (Kunden forventer 20 s). Data kan fx være lokalt tilgængeligt indtil patientens forløb er afsluttet.	
7. Login skal kunne gennemføres inden brugeren taber tålmodigheden.	Man kan starte arbejdet inden ____s udover den tid det tager brugeren at taste navn og password. (Kunden forventer 10 s).	
8. Gentagen login når brugeren midlertidigt har været væk fra systemet, skal kunne ske uden væsentlig ventetid.	Man kan starte arbejdet inden ____s udover den tid det tager brugeren at taste navn og password. (Kunden forventer 4 s).	

Der vil i praksis være tilfælde hvor systemet skal bruge lang tid til at svare, og hvor brugeren er forberedt på det. Her optræder en ergonomisk konstant på 20 sekunder. Selvom brugeren ved at det vil tage tid, vil han ubevidst vente omkring 20 sekunder og så begynde at lave noget andet. Skift fra én arbejdsopgave til en anden tager tid - tid som er spildt. For komplekse arbejdsopgaver kan den mentale skifte-tid til en anden opgave fx være 10-20 minutter. Krav 4 og 5 er eksempler hvor man forventer 20 sekunders svartid for at undgå dette spild.

Endelig kan der være funktioner hvor man af tekniske grunde må forvente svartider større end idealerne ovenfor. Hvis funktionerne bruges forholdsvis sjældent, giver det ikke de store problemer. Krav 6 og 7 er eksempler på det.

Leverandøren kan i midterkolonnen anføre særlige funktioner der ikke overholder de fælles regler om svartid, fx at et bestemt oversigtsbillede kan tage 3 minutter at vise.

### **Web-systemer**

Eksemplets krav er beregnet til produktionssystemer i daglig brug. For web-sites der kun bruges lejlighedsvis, er det alt for strenge krav, som vil være urimeligt dyre at opfylde.

### **Gyldig i X% af tilfældene**

Bemærk at de krævede svartider kun skal gælde i 95% af tilfældene (eller en anden brøkdelen). Hvorfor skal de ikke gælde i 100% af tilfældene? Fordi det ikke er realistisk for et flerbrugersystem.

Lad os se på et eksempel: Systemet er 0,2 sekunder om at behandle én forespørgsel, og der er i *gennemsnit* 2 forespørgsler pr. sekund. Systemet vil altså i gennemsnit arbejde med en forespørgsel i 40% af tiden, og når der tilfældigvis kommer endnu en forespørgsel imens, vil den blive sat i kø. Resultatet bliver at 95% af forespørgslerne får svar inden for 0,6 sekunder og 99% inden for 0,8 sekunder (en såkaldt M/D/1 kø-model).

Det lyder ikke så slemt, men hvad med worst-case? Hvis 2000 brugere næsten samtidig stiller en forespørgsel, vil den sidste bruger først få svar efter 400 sekunder. Hvis vi kræver at alle skal have svar inden for fx 2 sekunder, også i dette ekstreme tilfælde, skal man anskaffe et system der er så hurtigt at det kan behandle 1000 forespørgsler i sekundet. Det vil være urimeligt dyrt. Derfor må man aldrig stille krav om en maksimal svartid i et system med mange brugere.



(Blank med vilje)

## **L2. Tilgængelighed (driftseffektivitet)**

Tilgængeligheden angiver i hvor stor en del af tiden systemet skal være i drift set fra brugernes synspunkt. Hvad det præcis betyder at systemet er ude af drift må præciseres, og der må også tages stilling til hvordan man håndterer at nogle brugere oplever et driftsstop mens andre ikke gør. Fx vil man næppe regne det for et driftsstop hvis en enkelt brugerterminal ikke kan bruges.

Et driftsstop kan have forskellige årsager, og skabelonen nævner 5. Det er ikke dem alle der kan tillægges leverandøren. I eksemplet er det kun årsag 3 (fejl i software eller konfiguration), men hvis leverandøren har driftsansvar, kan også strømsvigt, hardware-nedbrud, kapacitetsproblemer, mv. tillægges ham.

I princippet kan kunden stille alle mulige krav om måling af tilgængeligheden, men i praksis må han acceptere de muligheder driftsoperatøren rent praktisk har - så længe det dækker kundens egentlige behov.

Skabelonen giver en række forslag. Fx foreslår den at et driftsstop altid regnes for mindst 20 minutter, og en driftsperiode skal vare mindst 60 minutter. Det skyldes at brugerne ikke får genoptaget deres afbrudte opgaver førend der er gået 20 minutter efter et driftsstop, og at de har svært ved at få noget ud af en driftsperiode på mindre end en time.

Skabelonen giver også forslag til beregning af driftstid og tilgængelighed når der skal tages hensyn til hvor mange brugere der er berørt af et driftsstop.

Bemærk krav 1 om at tilgængeligheden skal opgøres periodisk. Der kan således ikke spares tilgængelighed op fra én periode til den næste. Krav 1 giver leverandøren mulighed for at præcisere sin egen beregningsmåde, evt. ved at henvise til et bilag.

Krav 2-3 angiver hvilke driftsperioder der er tale om, og hvor stor en del af hver periode der må være driftsstop.

## L2. Tilgængelighed

Systemet er ude af drift når en del af brugerne ikke kan få støttet deres arbejdsopgaver som normalt. Årsagen til driftsstopet kan være:

1. Kundens forhold, fx fejl i kundens udstyr
2. Udefra kommende fejl, fx strømsvigt
3. Leverandørens forhold, fx fejl i software eller fejlagtig konfiguration
4. Planlagt vedligehold
5. Utilstrækkelig hardware kapacitet

### Måling af tilgængelighed

Et **driftsstop** regnes altid for at vare i mindst 20 minutter, selv hvis driften reelt genetableres inden. Hvis en driftsperiode varer mindre end 60 minutter, regnes den for en del af driftsstopet.

*I opgørelserne medregnes kun de driftsstop som skyldes leverandørens forhold (årsag 3). Hvis leverandøren har driftsansvar, kan han også have ansvar for årsag 2, 4 og 5.*

**Driftstiden** i en periode beregnes som periodens samlede længde minus den samlede længde af de driftsstop som leverandøren er ansvarlig for. **Tilgængeligheden** beregnes som driftstiden i forhold til periodens samlede længde. Der kan kompenseres for at kun nogle brugere er ramt af et driftsstop, fx ved at tilgængeligheden udregnes pr. bruger hvorefter den gennemsnitlige tilgængelighed beregnes.

Krav til driftstid:	Eksempel på løsning:	Kode:
1. Tilgængeligheden skal opgøres periodisk og der kan tages hensyn til hvor stor en del af brugerne der oplever driftsstoppe.	<i>Tilgængeligheden opgøres månedligt pr. brugersegment eller gruppe af arbejdsstationer, og det vægtede gennemsnit beregnes.</i>	
2. <i>I tidsrummet 8:00 til 18:00 på alle hverdage skal systemet have høj tilgængelighed.</i>	<i>I disse tidsrum er tilgængeligheden mindst ___%. (Kunden forventer 99%).</i>	
3. <i>I andre tidsrum behøver tilgængeligheden ikke være så høj.</i>	<i>I disse tidsrum er tilgængeligheden mindst ___%. (Kunden forventer 95%).</i>	

### **L3. Datalagring**

Dette afsnit beskriver hvor store datamængder der skal opbevares. I skabelonen fremgår de detaljerede datamængder af kapitel D, hvor der for tabellerne er angivet en samlet størrelse, og evt. en årlig tilvækst. Krav L3-1 og L3-2 angiver hvor mange års data der skal være umiddelbar adgang til og hvor mange der skal være adgang til med en længere svartid.

### **L4. Support**

Dette afsnit beskriver leverandørens support-ydelser, fx hjælp til brugere (hotline), overvågning af drift, og konfigurationsændringer. Der kan stilles krav om ydelser, fx fjerndiagnose eller udsendelse af en support-medarbejder. (Leverandøren har, som for andre krav, mulighed for at svare med kode 5, tilbydes ikke). Der kan også stilles kvalitetskrav om hvor hurtigt ydelsen skal ske. Ligesom for svartider, skal man undgå at stille krav om maksimal behandlingstid, da worst-case, hvor alle henvender sig samtidig, vil være meget dyrt at håndtere inden for en rimelig maksimal tid.

Kravene her må betragtes som typiske eksempler, men er bestemt ikke udtømmende. Mange SLA'er (Service Level Agreements) stiller krav om at en vis procentdel af henvendelserne skal klares med det samme. Det har desværre den ulempe at leverandøren er interesseret i at få en masse trivielle henvendelser som let kan besvares. Han er altså ikke motiveret til at forebygge disse henvendelser, som jo er tidsspild set fra brugernes side. Derfor stiller eksemplet kun krav om tiden for en besvarelse, og betragter konstateringen af at det er en systemfejl, som et gyldigt svar. Afhjælpningen af systemfejlen håndteres så via vedligeholdskravene.

### L3. Datalagring

Datavolumen fremgår af datakravene i kapitel D. Data skal opbevares som følger:

Krav til datalagring:	Eksempel på løsning:	Kode:
1. <i>Systemet skal give adgang til de sidste 5 fulde kalenderårs data med de svartider der er anført i LI.</i>		
2. <i>Systemet skal give adgang til ældre data for de sidste 20 fulde kalenderår med svartider som de eksterne systemer (LI-6).</i>		

### L4. Support

Support omfatter hjælp til brugerne, overvågning af driftssituation, samt ændring af den tekniske konfiguration. I dette afsnit skal "leverandør" forstås som leverandørens driftsorganisation. Alle de anførte tider skal gælde i 95% af tilfældene.

*Superbrugerne er de almindelige brugeres første kontakt om hjælp. Leverandøren skal derfor kun yde hjælp når superbrugerne ikke kan løse problemet.*

Krav til support:	Eksempler på løsning:	Kode:
1. Leverandøren yder hjælp til superbrugerne når de ikke selv kan løse problemet. Hjælpen omfatter alt udstyr og programmel leveret under denne kontrakt.		
1p. Problem: Det er vanskeligt for superbrugerne at afgøre hvilket produkt et problem hører under. Det er endnu vanskeligere at fungere som bindeled mellem flere leverandører.	Leverandøren inddrager på egen hånd andre parter for at løse problemerne.	
2. <i>I perioden 8:00 til 18:00 på alle hverdage kan superbrugerne hurtigt få telefonkontakt med leverandørens serviceorganisation.</i>	<i>I denne periode kan man få telefoniske kontakt inden for ___ minutter. (Kunden forventer 10 minutter).</i>	
3. Hvad enten problemet er modtaget telefonisk eller via email, giver leverandøren hurtigt superbrugeren et svar. Svaret er hjælp til superbrugeren indtil problemet er løst, eller en besked om at det er en fejl i systemet.	<i>Leverandøren besvarer henvendelserne inden for ___ timer i perioden 8:00-18:00, alle hverdage. (Kunden forventer 3 timer).</i>	
4. I tilfælde af en fejl i systemet, vil leverandøren dels hjælpe superbrugeren med så vidt muligt at omgå problemet, dels rapportere fejlen til vedligeholdsorganisationen.		
5. Leverandøren udsender en medarbejder hvis det er nødvendigt for at løse problemet.		
6. Leverandøren udfører fjerndiagnose for at løse problemet.		
7. Leverandøren registrerer data således at der dels kan beregnes svartider, dels uddrages data til identifikation og forebyggelse af hyppige problemer.	Leverandøren fører en log over alle henvendelser fra superbrugere med angivelse af fejlårsag og tidspunkt for fejlens afhjælpning.	
8. Leverandøren overvåger driftssituationen for at forudse problemer med tilgængelighed, og tager skridt til at ændre den tekniske konfiguration så tilgængeligheden sikres.		

## L5. Vedligehold

Dette afsnit viser typiske eksempler på krav til vedligehold. Der er både eksempler på afhjælpning af fejl, installation af nye versioner og forbedring af systemet.

Bemærk at leverandøren i første omgang er nødt til selv at vurdere hvor alvorlig en fejl er (krav 2). Ved regelmæssige møder vurderer parterne om der træffes rimelige afgørelser, hvad der skal rettes og hvad det skal koste (krav 4).

Når et system skal udbygges, har leverandøren monopol på at gøre det, og kan tage sig betalt derefter. Eksemplet viser hvordan det kan undgås, nemlig ved at måle ændringernes størrelse i *function points* og få en fast pris pr. function point (krav 6). Det kræver ekspertise at bruge function points, og der kan let blive uenighed om størrelsen. Der findes en dansk function point gruppe, som kan foretage en uafhængig beregning af størrelsen. I eksemplet aftaler man at bruge denne gruppe ved tvister.

## 4. Litteratur

- Soren Lauesen: Software Requirements - Styles and Techniques, Addison-Wesley, 2002. En lærebog om hvordan man kan formulere krav, og hvordan man finder og kontrollerer dem. I alt forklarer den omkring 100 teknikker med realistiske eksempler. Der er også råd om hvordan man kan verificere krav (kontrollere om de er opfyldt), hvordan man som leverandør overbeviser kunden, og hvordan man som kunde sammenligner tilbuddene. Indeholder store uddrag af virkelige kravspecifikationer der er formuleret på forskellige måder.
- Dansk Dataforening (nu: Dansk IT) & Søren Lauesen (2000): IT-anskaffelse, Kravspecifikationen. Dansk IT, 75 sider. Et kort, let læst hæfte som indeholder det bedste af lærebogen ovenfor.
- Suzanne Robertson & James Robertson: Mastering the Requirements Process, Addison-Wesley, London, 1999. ISBN 0-201-36046-2. Bruger et konkret eksempel, et system til at styre saltning og snerydning af veje, til at forklare forfatterens Volere-metode. Metoden handler om at finde kravene og formulere dem ved hjælp af kravskabeloner. Dækker primært systemer der udvikles fra grunden. Medtager mange flere typer kvalitetskrav end skabelon SL-07. Robertsons skabeloner er tilgængelige på:

<http://systemsguild.com/GuildSite/Robb/Template.html>

- Web-site, Managing Requirements:

[http://www.jiludwig.com/Template\\_Guidance.html](http://www.jiludwig.com/Template_Guidance.html)

Indeholder adskillige kravskabeloner, mest baseret på IEEE 830, som er vidt udbredt. Hovedvægten er på formalia, fx målgruppe, ændringslog, godkendelse. Der er kun meget simple eksempler på krav.

## L5. Vedligehold

Vedligehold omfatter både afhjælpning af fejl og forbedring af systemet, fx med nye funktioner. Alle de anførte tider skal gælde i 95% af tilfældene.

Krav til afhjælpning af fejl:	Eksempler på løsning:	Kode:
1. Leverandøren fører en log over såvel rapporterede fejl som ændringsønsker.		
2. For alle rapporterede fejl vurderer leverandøren hurtigt om fejlen er kritisk for kundens arbejdsopgaver, om den midlertidigt kan omgås, eller om den permanent kan omgås (afvises) .	<i>Leverandøren udfører vurderingen inden for ___ timer i perioden 8:00-18:00, alle hverdage. (Kunden forventer 3 timer).</i>	
3. Kritiske fejl afhjælpes hurtigt.	<i>Kritiske fejl afhjælpes inden for ___ timer. (Kunden forventer 24 timer).</i>	
4. Parterne mødes regelmæssigt for dels at kontrollere vurderingerne af fejlene, dels at afgøre hvad der skal rettes eller forbedres og hvad det skal koste.	<i>Parterne mødes ____ (Kunden forventer månedlige møder).</i>	

Krav til forbedring af systemet:	Eksempler på løsning:	Kode:
5. Leverandøren skal installere nye versioner og releases af det leverede software uden unødigt forsinkelse.	<i>Installationen sker inden for ___ dage efter at det er frigivet til brug i Danmark. (Kunden forventer 30 dage).</i>	
6. Inden for en periode af 3 år skal leverandøren tilbyde ændringer og udbygninger af software til en fast pris pr. function point.	<i>Prisen pr. function point er ____ Dkr.</i>	
7. Uenighed om beregningen af function points skal afgøres af . . .		