



**KONTINUASJONSEKSAMEN I FAG SIF8039  
GRAFIKK, BILDEBEHANDLING  
OG  
MENNESKE-MASKINGRENSESNIITT  
ONSDAG 31. JULI 2002  
KL. 09.00 – 14.00**

**LØSNINGSFORSLAG**

**OPPGAVE 1            Bildebehandling – Konvolusjon            (150 poeng)**

a) Definer 2D konvolusjon (folding).

F.eks. ligning 4.2-30 (side 162 i ny bok)

b) Definer konvolusjonsteoremet.

Konvolusjonsteoremet (ordet nevnt på s. 161), relasjon: 4.2-31 (s. 162)

c) Diskuter anvendelser av, og algoritmer for a) og b) innen bildebehandling.

"Rett-frem" algoritme for konvolusjon: fire løkker

Smartere algoritme for konvolusjon: bruke konv.teoremet: 2x FFT og en bildemultiplikasjon.

Diskuter anvendelser: noe med lav-/høy-passfiltre (? vanskelig med 'løsn.forslag' på diskusjon, er det ikke?)

**OPPGAVE 2                      Bildebehandling - Hough-transformen                      (150 poeng)**

a) Gi eksempel på to anvendelser innen bildeanalyse.

To anvendelser av Hough = to kreative forslag (linje OG f.eks. sirkelgjenkjenning?)

b) Gi algoritme. Forklar alle frie parametre.

Hough-algoritme: s. 587-591. Frie parametre: dimensjon på parameterplan (2D for linjer, 3D for sirkler), "opløsning" på akkumulatorcellene, grenser for parametre uten naturlige grenser (f.eks. a og b i fig. 10.7, s. 587). En annen viktig parameter er terskelen når man etter transformasjonen leter etter 'sterke' punkter i parameterrommet.

c) Diskuter fordeler og ulemper.

Fordel: global/robust mønstergjenkjenning (i motsetning f.eks. til kantlenking)  
 Ulemper: beregningsintensiv (langsom) når parameterrommet blir stort (og nyttig), samt at den globale modellen er 'hard': man finner f.eks. ideelle heltrukne linjer der det egentlig er stiplede linjer, etc.

**OPPGAVE 3                      Grafikk – Fjerning av skjulte flater                      (150 poeng)**

a) Forklar hva metoden back-face removal (back-face culling) går ut på og vis hvordan metoden kan realiseres ved hjelp av enkle vektorer som alltid vil være definerte når en skal gjengi en scene.

Back-face removal er å fjerne polygoner som vender bort fra øyepunktet og som dermed ikke er synlige. Dette gjøres for å redusere mengden av polygoner som må underkastes behandling med mer beregningskrevende algoritmer. Eksempler på situasjoner der back-face removal kan være aktuelt, er:

- Fjerning av skjulte flater med avanserte algoritmer
- Fjerning av skjulte flater før avbildningsalgoritmer taes i bruk

I forbindelse med realistisk avbildning vil følgende to vektorer alltid være definerte:

- Utadrettet flatenormal  $\mathbf{n}$  for hvert polygon
- Vektor  $\mathbf{v}$  rettet fra polygonets midtpunkt til øyepunktet (det som vil være gitt, er polygonets midtpunkt og øyepunktet hvorav vektoren er gitt). For ortografiske projeksjoner vil projeksjonsretningen være vektoren  $\mathbf{v}$ .

Den enkle beregningen som skal til for å realisere back-face removal, er beregning av skalarproduktet  $\mathbf{n} \cdot \mathbf{v}$ . Dersom dette skalarproduktet er positivt, kan polygonet være synlig. Er det negativt, er polygonet definitivt ikke synlig.

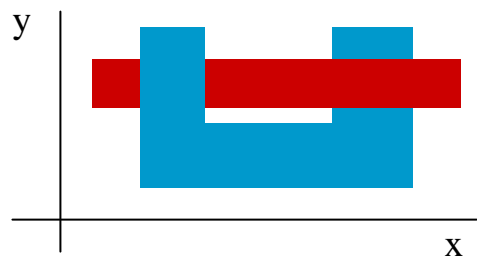
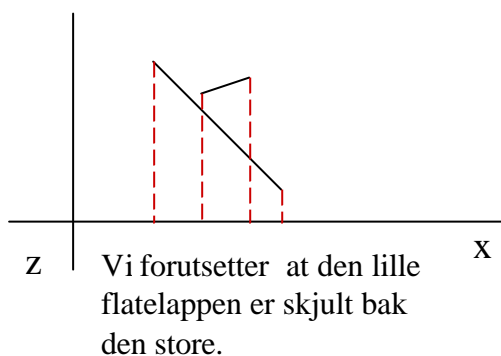
- b) Forklar painters algoritme og vis noen situasjoner der den vil gi feil i det resulterende bildet.

Painters algoritme går slik:

1. Sett hele billedlageret til bakgrunnsfargen
2. Sorter alle polygoner etter største avstand fra bildeplanet
3. For hvert polygon i sortert orden:  
Gjengi ("mal") hele polygonet

Virkningen er at polygoner som ligger nær bildeplanet males over polygoner som ligger lenger bak.

Eksempler på situasjoner der painters algoritme feiler, illustreres av følgende skisser:



- c) Forklar hvordan z-bufferalgoritmen for fjerning av skjulte flater virker. Forutsett at du kjenner koeffisientene til likningen for planet som inneholder flatelappen du for øyeblikket behandler. Vis hvordan du kan forenkle dybdeberegningen så lenge du beveger deg langs en scanlinje (utnyttelse av koherens).

Z-bufferalgoritmen behandler i prinsippet scenen polygon for polygon. Algoritmen går slik:

1. Initierer billedlageret med bakgrunnsfargen
2. Initierer z-bufferen med en z-verdi som ligger bak de z-verdiene som er mulige (for eksempel det negative tallet med størst tallverdi)
3. For hvert piksel:
  - Ser "gjennom" hvert piksel fra projeksjonssenteret (eller for parallellprojeksjon i projeksjonsretningen)
  - Dersom polygon-punktet synsstrålen treffer, har z-koordinat nærmere "øyet" enn den siste z-verdien som ble lagret, erstattes fargen i billedlageret med det nye polygonets farge og z-verdien i z-bufferen med det nye polygonets z-verdi.

Koherens kan utnyttes til å forenkle dybdeberegningen på grunn av følgende sammenheng:

Polygonet ligger i flaten med likning:

$$ax + by + cz + d = 0$$

To punkt i polygonet er  $P_1$  og  $P_2$  slik at:

$$\Delta x = x_2 - x_1$$

$$\Delta y = y_2 - y_1$$

$$\Delta z = z_2 - z_1$$

Innsetting i flatelikningen og subtraksjon gir:

$$a\Delta x + b\Delta y + c\Delta z = 0$$

Vi vil vanligvis behandle polygonet scanlinjevis. På en scanlinje gjelder:

$$\Delta y = 0$$

Målt i bildelageradresse øker  $x$ -koordinaten i trinn på 1:

$$\Delta x = 1$$

Dermed får vi for forflytning langs scanlinjen:

$$a + c\Delta z = 0$$

Det vil si at  $z$ -verdien inkrementeres i faste trinn:

$$\Delta z = -\frac{a}{c}$$

Dermed er beregningen av neste  $z$ -verdi redusert til addisjon av en konstant til forrige  $z$ -verdi:

$$z_{ny} = z_{gammel} + \Delta z = z_{gammel} - \frac{a}{c}$$

#### OPPGAVE 4

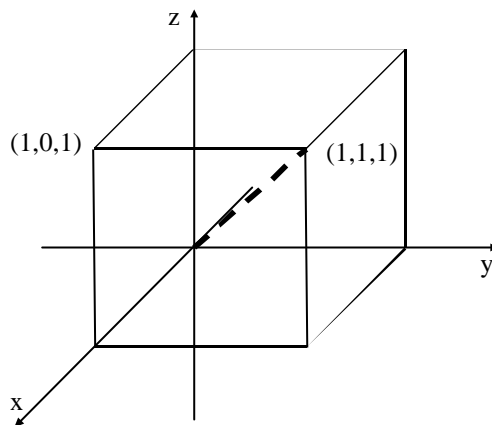
#### Grafikk – Geometriske transformasjoner

(150 poeng)

Gitt en terning med sidekant 1 som har ett hjørne i origo, ett i punktet  $(1, 0, 1)$  og ett i punktet  $(1, 1, 1)$ .

- a) Forutsett at du skal rotere terningen en vinkel  $\varphi$  om diagonalen gjennom origo og punktet  $(1, 1, 1)$ . Du skal benytte en serie av rotasjoner om koordinataksene (se bort fra bruk av de spesielle egenskapene til ortogonale matriser). Svar på følgende spørsmål:

1. List opp en serie av rotasjoner om koordinataksene som realiserer denne rotasjonen. Det kreves her **ikke** at du skal skrive opp matrisene eller finne rotasjonsvinklene.



Terningen ligger i utgangsposisjonen som vist i ovenstående figur. En serie av rotasjoner som vil rotere terningen vinkelen  $\varphi$  om diagonalen, kan være:

1. Rotere om z-aksen slik at diagonalen blir liggende i planet  $x = 0$
2. Rotere om x-aksen slik at diagonalen faller sammen med z-aksen
3. Rotere vinkelen  $\varphi$  om z-aksen
4. Utføre den inverse rotasjonen av den i punkt 2
5. Utføre den inverse rotasjonen av den i punkt 1

2. Hvilke andre valg kan du gjøre som ville gi det samme resultatet? Svar kort.

En rekke andre serier av rotasjoner ville samme resultat.

- I punkt 2 i serien ovenfor kunne rotasjonen alternativt bli gjort slik at diagonalen falt sammen med y-aksen
- I punkt 1 i serien ovenfor kunne rotasjonen alternativt bli gjort slik at diagonalen ble liggende i planet  $y = 0$
- I punkt 1 kan en velge mellom å rotere enten slik at enden av diagonalen blir liggende i y-rommets positive halvrom eller i det negative halvrommet. Tilsvarende valg må gjøres i punkt 2.

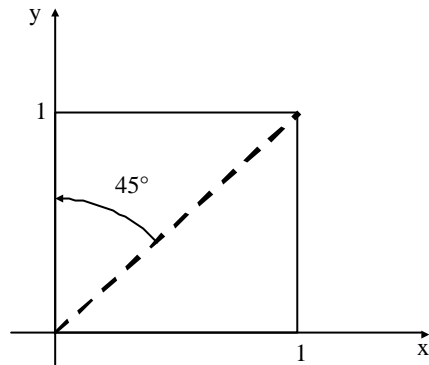
I punkt 1 kunne en også velge rotasjon om x-aksen eller y-aksen med tilsvarende etterfølgende valg. (I alt 48 varianter av rotasjonsserier er mulige.)

- b) Du skal rotere terningen slik at diagonalen som opprinnelig går gjennom origo og punktet  $(1, 1, 1)$ , faller langs den positive y-aksen og slik at punktet som opprinnelig hadde koordinater  $(1, 0, 1)$  blir liggende i x-y-planets første kvadrant. Beskriv denne rotasjonen ved å svare på spørsmålene 1 – 3 nedenfor. Det ventes **korte** og **konsise** svar gjerne støttet av en skisse eller to.

Alternativt kan du, dersom du finner det mer hensiktsmessig, beskrive rotasjonen ved å sette opp en rotasjonsmatrise basert på de spesielle egenskapene til ortogonale matriser.

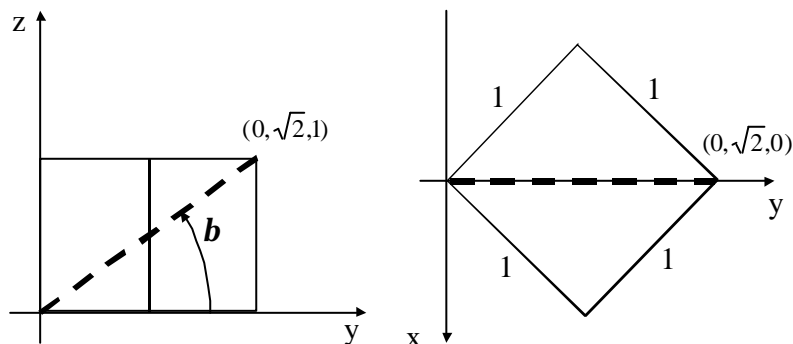
Dersom du velger denne måten å løse oppgavene på, skal du **ikke** svare på spørsmålene nedenfor.

1. Første skritt skal være en rotasjon om z-aksen slik at diagonalen faller i planet  $x=0$ . Hva blir rotasjonsvinkelen?



Ovenstående figur viser en ortografisk projeksjon av teringen i utgangsposisjonen i planet  $z = 0$ . Her ser vi at rotasjonsvinkelen det spørres etter er  $\pi/4$ .

2. Neste skritt blir en rotasjon om x-aksen slik at diagonalen blir liggende i den spesifiserte sluttposisjonen. Hva blir rotasjonsvinkelen? Det er tilstrekkelig å angi sinus og cosinus til denne vinkelen.



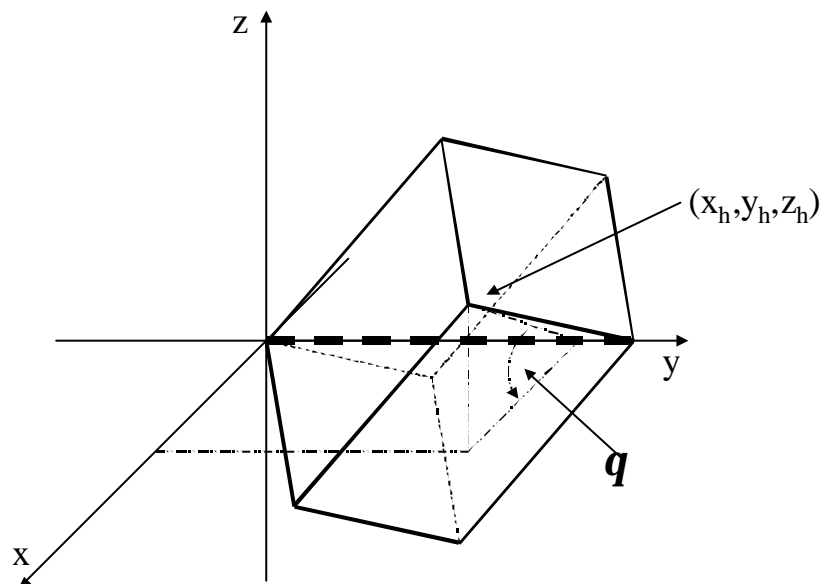
Situasjonen i planene  $x = 0$  og  $z = 0$  etter rotasjonen i punkt 1 er som vist i ovenstående figurer. Enden av diagonalen befinner seg i punktet  $(0, \sqrt{2}, 1)$ . Rotasjonsvinkelen som bringer diagonalen til å falle langs den positive y-aksen, er vinkelen  $-\beta$ .

Sinus og cosinus til vinkelen er:

$$\underline{\underline{\sin(-\mathbf{b}) = -\sin(\mathbf{b}) = \frac{1}{\sqrt{(\sqrt{2})^2 + 1}} = \frac{1}{3}\sqrt{3}}}$$

$$\underline{\underline{\cos(-\mathbf{b}) = \cos(\mathbf{b}) = \frac{\sqrt{2}}{\sqrt{(\sqrt{2})^2 + 1}} = \frac{1}{3}\sqrt{6}}}$$

3. Hvordan vil du finne ut hvor hjørnet som opprinnelig befant seg i punktet  $(1, 0, 1)$ , befinner seg etter at ovenstående to rotasjoner er utført, og hvordan vil du bruke resultatet til å realisere den siste rotasjonen som bringer dette hjørnet til den spesifiserte sluttposisjonen?



Etter rotasjonene i punktene 1 og 2 vil situasjonen være som vist i ovenstående figur. Hjørnet som opprinnelig lå i  $(1, 0, 1)$  har nå koordinatene  $(x_h, y_h, z_h)$ . De nye koordinatene kan finnes ved å anvende transformasjonsmatrisen som kan settes opp på grunnlag av punktene 1 og 2 på hjørnets opprinnelige koordinater. Hjørnet bringes så til sin sluttposisjon ved å rotere vinkelen  $\theta$  om y-aksen:

$$\sin(\mathbf{q}) = \frac{z_h}{\sqrt{x_h^2 + z_h^2}}$$

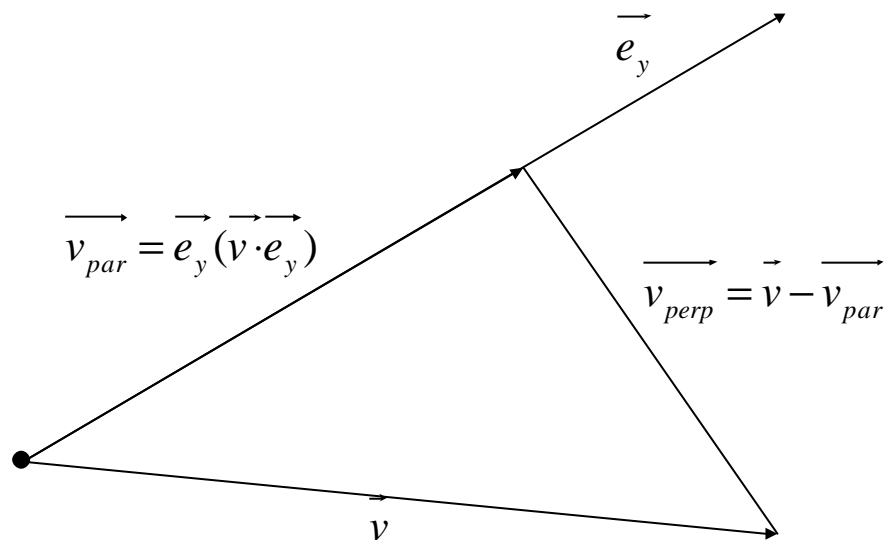
$$\cos(\mathbf{q}) = \frac{x_h}{\sqrt{x_h^2 + z_h^2}}$$

Løsning ved hjelp av ortogonale matriser:

Vektoren  $(1, 1, 1)$  langs diagonalen skal som et resultat av rotasjonen falle i samme retning som den positive y-aksen. Vi definerer enhetsvektoren  $\vec{e}_y$ :

$$\vec{e}_y = \frac{1}{\sqrt{3}}[1 \ 1 \ 1]$$

Planet definert av diagonalen og punktet  $(1, 0, 1)$  vil etter rotasjonen falle sammen med planet  $z = 0$ . Vi trenger en enhetsvektor  $\vec{e}_x$  i dette planet som vi ved rotasjonen kan bringe til å falle i samme retning som den positive x-aksen. Vi definerer en vektor  $\vec{v}$  fra origo til punktet  $(1, 0, 1)$ . Den normaliserte komponenten av vektoren  $\vec{v}$  ortogonal til enhetsvektoren  $\vec{e}_y$  er den enhetsvektoren  $\vec{e}_x$  vi er på jakt etter.



Av figuren ovenfor ser vi:

$$\begin{aligned} \vec{v}_{perp} &= \vec{v} - \vec{e}_y(\vec{v} \cdot \vec{e}_y) = [1 \ 0 \ 1] - \frac{1}{\sqrt{3}}[1 \ 1 \ 1]([1 \ 0 \ 1] \cdot \frac{1}{\sqrt{3}}[1 \ 1 \ 1]) = \\ &= [1 \ 0 \ 1] - \frac{2}{3}[1 \ 1 \ 1] = \frac{1}{3}[1 \ -2 \ 1] \end{aligned}$$



Normalisering gir:

$$\vec{e}_x = \frac{\vec{v}_{perp}}{|\vec{v}_{perp}|} = \frac{1}{\sqrt{6}} [1 \quad -2 \quad 1]$$

Den tredje enhetsvektoren  $\vec{e}_z$  får vi ved å ta vektorproduktet av  $\vec{e}_x$  og  $\vec{e}_y$ :

$$\vec{e}_z = \vec{e}_x \times \vec{e}_y = \frac{1}{\sqrt{2}} [-1 \quad 0 \quad 1]$$

Rotasjonsmatrisen på homogen form blir når vi setter inn de tre enhetsvektorene som rader i matrisen:

$$\underline{\underline{M_{rot}}} = \begin{bmatrix} \vec{e}_{xx} & \vec{e}_{xy} & \vec{e}_{xz} & 0 \\ \vec{e}_{yx} & \vec{e}_{yy} & \vec{e}_{yz} & 0 \\ \vec{e}_{zx} & \vec{e}_{zy} & \vec{e}_{zz} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{6}} & \frac{-2}{\sqrt{6}} & \frac{1}{\sqrt{6}} & 0 \\ \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & 0 \\ \frac{-1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

## OPPGAVE 5 MMI

(300 poeng)

Du er gitt i oppgave å lage et grafisk brukergrensesnitt for en ny nettbasert tjeneste for spesialbestilling av mobiltelefoner. En kjent mobiltelefonfabrikant har funnet ut at kundene burde kunne individualisere ikke bare deksel, men også mobiltelefonenes form. Den nye tjenesten er gitt tittelen "Connecting designers". Tjenesten skal foreløpig bare tilbys for én modell.

For denne modellen skal kunden kunne gjøre følgende individuelle tilpasninger:

- Form
  - Kunden kan tilpasse mobiltelefonens form på høyre side, venstre side og bunn.
  - For hver av disse finnes 5 valg: "Rett", "Buet inn", "Buet ut", "Små bølger", "Store bølger", "Sagtann".
- Deksel
  - Deksler vil kunne bestilles spesiallaget ut fra mobiltelefonens form.
  - Deksler finnes i et antall navngitte kategorier (for eksempel "Kunst", "Filmer", "Natur",,,).
  - For hver kategori finnes et antall navngitte deksler (for eksempel i kategorien "Kunst": "Munch's skrik", "Picasso's dame", "Dali's klokke",,,).
  - Hvert deksel kan spesialtilpasses med en bakgrunnsfarge fra en gitt liste av farger.
  - Det kan bestilles max. 6 deksler for hver mobiltelefon.

Den nettbaserte tjenesten skal gjøre det mulig å bestille spesiellagede mobiltelefoner med parametrene angitt over. Kunden skal kunne få en visuell forhåndsvisning før bestilling gjøres.

**a) Interaksjonsteknikker**

Velg deg to forskjellige interaksjonsteknikker, og skisser en løsning basert på hver av disse. Angi fordeler og ulemper ved de to løsningene. Du skal ikke la løsningene være begrenset av for eksempel HTML. Du skal heller ikke skissere dialogene for å angi navn, VISA nr. etc., kun de delene som har å gjøre med valg av parametere som angitt over.

Her kan man for eksempel velge seg ikon/meny basert vs. direktemanipulasjon. Jeg forventer noen enkle skisser av skjermlayout for hver av eksemplene.

Fordelene med et knappe og menybasert grensesnitt er at dette er kjent for de fleste brukere, og at det er relativt enkelt å implementere.

Ulempene er at det lett kan bli mange knapper, og at det kan oppleves som litt kjedelig.

Fordelen med et direktemanipulasjon-grensesnitt er at det kan gi færre elementer på skjermen, og at det kan oppleves som mer interessant.

Ulempen er at det ikke på samme måte som for ikon/meny baserte grensesnitt er innarbeidet konvensjoner for "affordance" og "constraints". Det ligger derfor en spesiell utfordring i å utforme grafikken slik at brukeren forstår hva som skal gjøres for å få den ønskede effekt. Dette vil kunne kreve flere iterasjoner med brukbarhetstesting/evaluering.

**b) Evaluering**

Hvilke evalueringsteknikker vil du anbefale for empirisk å kunne måle brukbarheten av brukergrensesnittet. Begrunn svaret.

I dette eksemplet er brukbarhet i en snever forstand knyttet til i hvilken grad brukergrensesnittet tillater brukeren enkelt å utføre de ønskede handlingene. En empirisk evaluering av brukbarheten vil her kunne gjøres ved kontrollerte brukbarhetstester der for eksempel 6 forsøkspersoner får samme konkrete veldefinerte oppgave. Oppgaven kunne for eksempel være: "Velg en telefon med buer til venstre, rett til høyre og sagtann under. Vel deksel med.....".

Brukbarheten vil så kunne måles utfra for eksempel hvor mange feil brukeren gjør og hvor lang tid de bruker på å løse oppgavene.

Det er også mulig å se litt videre på selve produktideen, og dens "salgbarhet". Er dette noe kundene ønsker? En fare med en for ustrukturert test er følgelig at vi tester produktideen og ikke grensesnittet. En fare den andre veien er såklart at oppdragsgiver ikke forstår forskjellen, og leverer en tjeneste med perfekt brukbarhet, men som ingen egentlig vil ha.