

## FORORD .....ERROR! BOOKMARK NOT DEFINED.

### 1. HVORFOR SKAL ET BACHELORPROSJEKT DOKUMENTERES? ..... 3

1.1 Hvem har bruk for dokumentasjonen for et bachelorprosjekt? ..... 3

1.2 Men dokumentasjon har så dårlig rykte -? ..... 4

1.3 Hvorfor må dokumentasjon følge en standard? ..... 4  
- men hvorfor akkurat denne standarden? ..... 5

1.4 Hvilke dokumenter skal leveres for bachelorprosjekt i data? (Høgskolen i Oslo, avdeling for Ingeniørutdanning) ..... 5

1.5 Planlegging av dokumentasjon ..... 6

Gjenbruk av tekst..... 7

### 2. STYRINGSDOKUMENTASJON ..... 8

2.1 Prosjektskisse ..... 8

2.2 Prosjektdagbok ..... 8

2.3 Forprosjektrapporten..... 10  
Oppbygging av forprosjektrapporten ..... 11

2.4 Arbeidsplan og fremdriftsplan ..... 12  
Er det virkelig mulig å planlegge?..... 13

2.5 Kravspesifikasjonen..... 14  
Datainnsamling..... 14  
Endringer av kravspesifikasjon ..... 15  
Oppbygging av kravspesifikasjon ..... 15

### 3. SLUTTDOKUMENTASJON ..... 16

3.1 Innbinding og rekkefølge ..... 16

3.2 Prosessdokumentasjonen ..... 17  
Prosessdokumentasjonen – et viktig dokument ..... 17  
To hovedtyper bachelorprosjekter ..... 18

3.3 Produktdokumentasjonen ..... 21  
Beskrivelsen av programmet ..... 22

3.4 Testdokumentasjonen..... 22

3.5 Brukerveiledning ..... 23  
To typer fakta ..... 24

3.6 Oppbygging av brukermanual..... 24

3.7 Gode råd for brukermanual ..... 25  
Rekkefølge og struktur ..... 25  
Språk ..... 25

Forskjellig .....	26
Dokumentasjon som er viktig for datasikkerheten .....	26
<b>4. MUNTlig PRESENTASJON AV ET PROSJEKT .....</b>	<b>27</b>
<b>4.1 Demonstrasjonen .....</b>	<b>28</b>
<b>4.2 Muntlig sluttrapportering i punkter .....</b>	<b>28</b>

## Hvorfor skal et bachelorprosjekt dokumenteres?

IT gjør tilværelsen både enklere og bedre for utallige mennesker. Datasystemer utfører oppgaver som man ellers aldri kunne fått utført, eller som det ville tatt uendelig mye tid å utføre. Datasystemer gir praktiske forbedringer i hverdagen, nytte og glede, ny kunnskap, innsparinger og god økonomi.

Men datasystemer, spesielt større datasystemer, er oftest ganske komplekse. Det krever kunnskap og kompetanse å skape dem, og det krever kunnskap og kompetanse å forstå, vedlikeholde og bruke dem.

Det er ofte stor avstand mellom dem som har designet og implementert et datasystem, og dem som skal bruke det. Avstanden kan være fysisk, uttrykt i kilometer og mil. Den kan uttrykkes i tid: et system kan brukes i mange år etter at det er laget. Avstanden kan også ha å gjøre med kunnskaper og kompetanse. Den som vedlikeholder eller bruker produktet, har alltid mye mindre kunnskaper om det spesielle systemet enn de som har utviklet det, og ikke så sjelden har de mindre kunnskaper om data generelt.

Dokumentasjonen skal bygge bru over de avstander som finnes, slik at datasystemet kan utnyttes fullt ut også av dem som ikke har utviklet det og som står fjernt fra utviklerne – på den ene eller den andre måten..

### **1.1 Hvem har bruk for dokumentasjonen for et bachelorprosjekt?**

Hvilken dokumentasjon som trengs for et dataprodukt, er avhengig av hvem som skal ha produktet og hva de skal gjøre med det.

For et dataprodukt finnes det vanligvis to typer lesere/brukere.

Den ene gruppen inneholder de personene som er ansvarlige for drift av det aktuelle produktet, de som skal installere, kjøre, vedlikeholde, modifisere og feilsøke programsystemet. Disse har bruk for ganske mye informasjon om produktet: om system, oppbygging, funksjoner, begrensninger, opsjoner, tidligere testing, osv. Denne gruppen vil normalt være datakyndig på tilnærmet samme nivå som utvikleren.

Den andre hovedgruppen er det vi kaller ”sluttbrukere”: de som skal bruke systemet til det det er designet for, kanskje til daglig. De trenger informasjon om hvordan produktet kan brukes: hva det kan brukes til, prinsipper for normal bruk, valgmuligheter, opsjoner. Deres fagfelt er ofte noe annet enn IT, og de trenger bedre forklaringer for å lykkes med bruken av et IT-system enn det driftspersonalet gjør.

Uansett hvem man utvikler programmer for, enten det er for næringslivet eller utdanningsinstitusjoner, må det lages dokumentasjon for disse to typene brukere. Uten slik dokumentasjon kan et datasystem i lengden ikke fungere. Kunnskap som bare finnes i hodet på en bestemt person, utvikleren, er uhyre utsatt og kan bli borte på mange ulike måter.

I utdanningssituasjoner kommer det til enda en type lesere/brukere: de som skal evaluere prosjektet. De trenger i tillegg til kunnskap om produktet også kunnskap om utviklingsprosessen for produktet. Bare hvis man kjenner denne prosessen, kan man vurdere kvaliteten på det produktet som er utviklet, samt utviklerens faglige kompetanse og evne til systematisk arbeid. (Kunnskap om prosessen kan i mange situasjoner være nyttig for driftspersonalet også.)

## **1.2 Men dokumentasjon har så dårlig rykte -?**

Det er et vanlig synspunkt at det er kjedelig å utforme dokumentasjon. Å utvikle dataprogrammer og -systemer oppfattes som spennende og interessant, og utvikleren føler etterpå at han/hun virkelig har gjort en jobb. Å utforme dokumentasjon for det samme systemet, oppfattes helt annerledes. Utvikleren forstår og kjenner sitt system. Problemstillingene og utfordringene som eksisterte under utviklingen, er vurdert, og løsningene funnet.. Det virker som en overflødig repetisjon, tungt og vanskelig, å beskrive og forklare alt sånn i etterhånd. Til manges fortvilelse krever dokumentasjon også svært mye tid. Man regner at mellom 10 % og 25 % av utviklingstiden går med til dokumentasjon.

Erfaringene fra dem som er avhengig av å bruke dokumentasjonen, er heller ikke alltid så gode. Egentlig håper man å klare seg **uten** å bruke dokumentasjon. ”Hvis alt annet svikter, les bruksanvisningen!” Det har sine grunner. Dokumentasjonen er ofte omfangsrik, og samtidig kanskje uhensiktsmessig strukturert, slik at det er vanskelig å finne fram til det man trenger. Kanskje finner man det ikke i det hele tatt. Etter slike opplevelser mister man tilliten til at det er noen vits i å bruke dokumentasjon.

Dette betyr nok at tiden som brukes til dokumentasjon, alt i alt blir dårlig utnyttet – det gjelder både tiden som går med til å utforme dokumentasjon og den tiden som går med til å finne fram i den. Dokumentasjon gir en mulighet til å overføre kunnskap fra den som har utviklet et program til dem som skal bruke programmet. Hvis dette ikke fungerer som det burde, kan det ha noe med kvalitet å gjøre: Dokumentasjonsarbeidet må være bra nok til at de som trenger informasjonen, kan finne den uten for mye strev. Samtidig må brukeren ha tillit til at han vil finne det han trenger hvis han setter inn rimelige anstrengelser. Noe arbeid må alltid kreves, for dokumentasjonen for et større system er så omfattende at det må kreve noe tid å finne det man er ute etter.

Hvis alle parter er kjent med samme forutsigbare standard for dokumentasjon, vil dette målet lettere oppnås.

Dokumentasjon er en nødvendighet for et datasystem. Utfordringen er å gjøre den så god at den fungerer for dem som har bruk for den.

## **1.3 Hvorfor må dokumentasjon følge en standard?**

I dette heftet vil du finne mange regler for hva som skal være med i dokumentasjonen og hvordan den skal struktureres. Men må det gjøres på akkurat denne måten? Kan ikke utviklere dokumentere slik de selv føler for?

Det er faktisk mange grunner til at det bør brukes en standardisert måte å dokumentere på.

En grunn er at hvis det brukes en standard, får den enkelte utvikler hjelp av standarden til å huske og få med seg alle relevante fakta. Ellers kan mye bli glemt bort i strevet med å fullføre et dataprodukt til tidsfristen.

Samtidig vil en standardisert struktur sikre at det blir lettere for lesere og brukere å finne fram til de informasjonene som er interessante. Leseren kan da gjenbruke sin egen kunnskap om hvor fakta finnes, og utnytte den.

I utdanningsinstitusjoner med prosjekter som skal evalueres, gjør en felles standard det mulig for lærere og sensorer både å finne greit fram i de forskjellige oppgavene og å sammenligne ulike prosjekter på en meningsfylt måte.

#### **- men hvorfor akkurat denne standarden?**

Den standarden du lærer her, er en vanlig standard, brukt i store trekk i ganske mange utdanningsinstitusjoner og mange bedrifter over hele verden (der de bruker standarder). Men den er ikke den eneste, og slett ikke den eneste mulige, og det er mange punkter der det kan finnes små avvik. Det kan godt hende at du vil treffe på bedrifter med en annen filosofi for dokumentasjon enn den som presenteres her. Likevel er det klokt å satse på å forstå og mestre denne standarden, selvfølgelig hvis ikke oppdragsbedriften din forlanger en annen standard. Det er viktig for bachelorprosjektet at du forstår denne standarden og kan bruke den. Den er for det første nokså utbredt, så den fungerer mange steder. For det andre: Hvis du senere treffer på andre måter å bygge opp dokumentasjon på, vil du forstå dem mye lettere nettopp hvis du forstår prinsippene i denne. Det vil gjøre det lettere for deg å lære en ny standard. Under enhver standard ligger grunnleggende prinsipper om forutsigbarhet, gjenkjennelighet og brukerorientering.

Av og til vil man komme i situasjoner der man er nødt til å gjøre tilpasninger til standarden. Da må man forsøke å la seg lede av de grunnleggende prinsipper som ligger under standardene og velge løsninger som er tilpasset oppgave og bruker, samt bruke sin sunne fornuft.

Å skrive løs etter innfallsmetoden vil aldri fungere godt. Uansett, du må tilpasse deg en standard. Den gir rammen, og derved en nødvendig støtte. Rammen fylles ut med det produktet du har utviklet og prosessen omkring det.

### **1.4 Hvilke dokumenter skal leveres for bachelorprosjekt i data? (Høgskolen i Oslo, Avdeling for ingeniørutdanning)**

Selv et middels stort bachelorprosjekt krever mange sider dokumentasjon, med ganske store krav til både kompleksitet og nøyaktighet. Papirutskriftene av dokumentasjonen for bachelorprosjekter blir ofte imponerende tykke.

Det kan være nyttig først å skille mellom to hovedtyper dokumentasjon:

**styringsdokumentasjon** og **sluttdokumentasjon**. Styringsdokumentene er dokumenter som lages under prosessen, og som først og fremst skal støtte den. Styringsdokumentene skal hjelpe utviklerne til å fullføre prosjektet. Når prosessen er over og prosjektet fullført, har disse dokumentene gjort sin nytte og har ingen interesse lenger for andre enn eventuelt utviklerne selv.

Sluttdokumentasjonen er den samlingen av dokumenter som leveres inn til bedømmelse til slutt, som både beskriver hvilket produkt man skulle lage (kravspesifikasjon), det produktet som er laget (eventuelt med egen brukerdokumentasjon), prosessen som førte fram til det og eventuell testing som er utført.

Nedenfor følger en oversikt over de dokumenter som skal leveres i forbindelse med bachelorprosjekt i data, HIO-IU.

Dette betraktes som **styringsdokumenter**:

- Prosjektskisse
- Prosjektdagbok
- Forprosjektrapport
- Arbeids- og fremdriftsplan
- Eventuell muntlige eller skriftlig rapportering, f. eks. til veileder
- Kravspesifikasjon\*

\*Kravspesifikasjonen er både en del av styringsdokumentene og en del av sluttdokumentasjonen. Den leveres sammen med produktet til slutt for at sensor og veileder skal kunne se i hvilken grad denne spesifikasjonen er oppfylt.

Dette betraktes som **sluttdokumentasjon**

- Kravspesifikasjon
- Prosessdokumentasjon
- Produktdokumentasjon
- Testdokumentasjon
- Brukerdokumentasjon
- Muntlig presentasjon av prosjekt

Før vi tar for oss disse ulike dokumentene etter tur, skal vi snakke litt om planlegging av dokumentasjon. Dokumentasjonen krever ofte ganske mye tid. Med planlegging og forsiktig gjenbruk kan den tiden forminskes.

### **1.5 Planlegging av dokumentasjon**

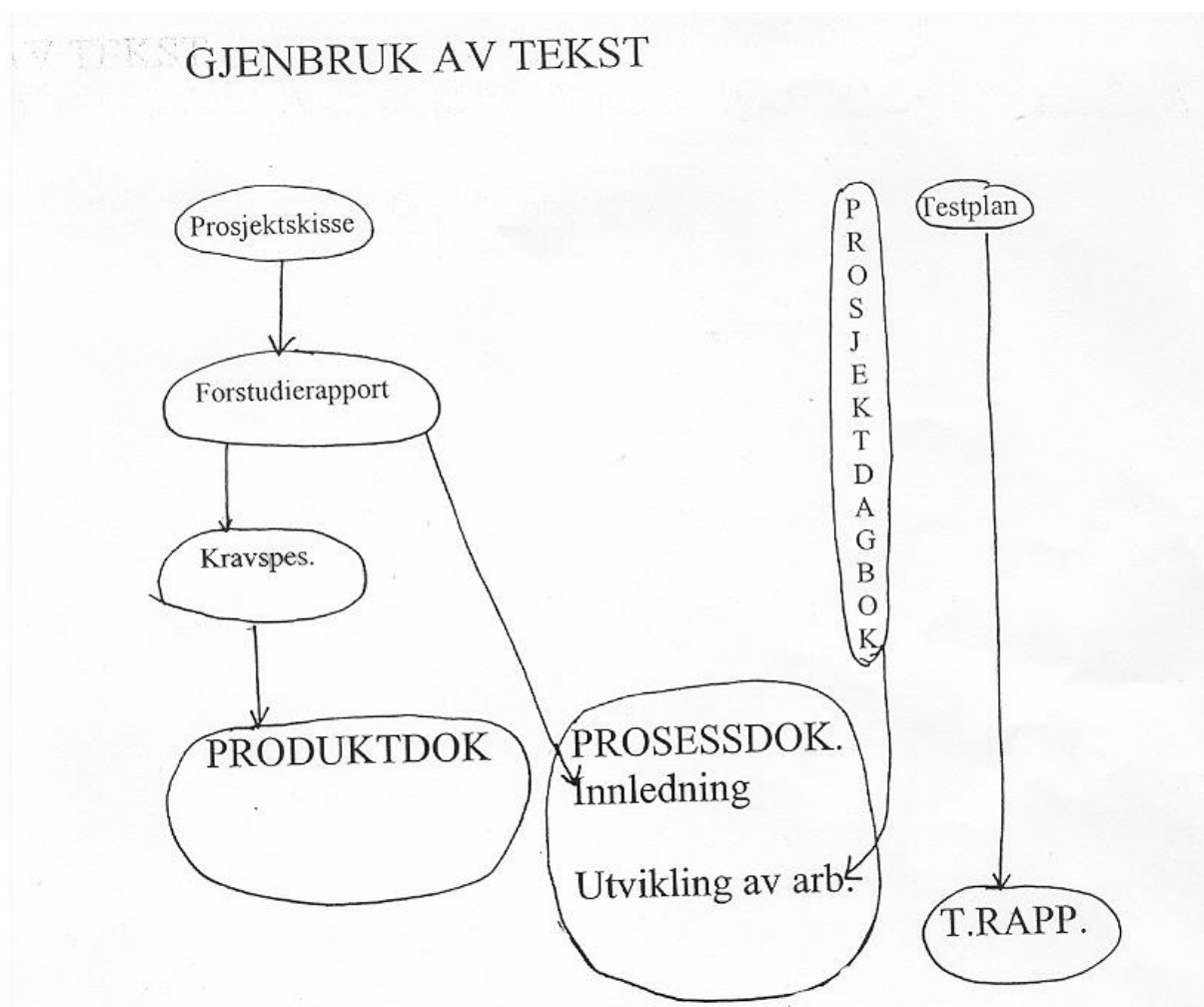
Tid til utarbeiding av dokumentasjon må planlegges inn fra begynnelsen. Hvis man arbeider med dokumentasjon parallelt med det øvrige arbeidet, og planlegger godt, vil man utnytte tiden maksimalt, og dokumentasjonen vil sannsynligvis bli bedre.

Det er nyttig å gjøre notater om arbeidet hele tiden, i prosjektdagbok og i separate dokumenter. Samtidig er det viktig å ha oversikt over hvordan de enkelte dokumenter bygger på informasjon i andre dokumenter. F. eks. gir forprosjektrapporten en del informasjon som det er bruk for både i kravspesifikasjonen og i sluttrapporten - i kapitlet Om arbeidet. Hvis man har gjort notater til forprosjektrapporten med tanke på at stoffet også skal brukes i sluttrapporten, går det lettere å lage den siste. Kanskje man kan utforme deler av dokumentasjonen slik at man kan ha **tilnærmet gjenbruk** av tekst.

Når man arbeider med oppbyggingen av programmet, og med den dokumentasjonen som skal integreres i systemet - oppbygging, navn på konstanter, prosedyrer osv.- bør man også notere momenter for bruk i produktdokumentasjonen.

### Gjenbruk av tekst

Figuren viser hvilke muligheter for gjenbruk av tekst som finnes i et bachelorprosjekt.



Legg merke til at det går en forbindelse helt fra prosjektskisse gjennom forprosjektrapport og så kravspesifikasjon til den endelige produktdokumentasjonen.

Forprosjektrapporten har videre forbindelse til prosessdokumentasjonen (det vil særlig være innledning til prosessdokumentasjonen som har nytte av forprosjektrapporten), mens prosjektdagboka, som bør føres hele tida, er hovedkilden for resten av prosessdokumentasjonen.

Testplan med testdokumentasjon ser ut til å utgjøre en egen, isolert gren av det hele. Men det er klart at testingen vil påvirke produktet og derved produktdokumentasjonen - og den vil påvirke prosessen Selv om dokumentene er meget forskjellige, handler de om samme prosjekt, og derfor henger de også nært sammen. Pilene mellom dem betyr også at man kan forsøke å bruke deler av teksten på nytt i det dokumentet pilen peker mot. Vær klar over at konteksten endrer seg fra dokument til dokument, så man kan ikke skrive av ordrett det som

sto i det forrige dokumentet. Men det trenges ofte bare små endringer for å kunne bruke et stoff om igjen, hvis man bare er oppmerksom på hvordan det bør gjøres.

## 2. Styringsdokumentasjon

### 2.1 Prosjektskisse

Prosjektskissen er vanligvis det første som skrives om prosjektet. En oppdragsgiver har sagt seg interessert i å gi dere et prosjekt, men det er selvfølgelig bare begynnelsen. Nå må problemet defineres så presist at seksjonen kan ta stilling til om dette er et passende prosjekt, og det er viktig at det blir så utfyllende som mulig, både for studentenes egen del og for skolens. Prosjektskissen skal være et vedlegg til den samarbeidsavtalen som skal inngås mellom oppdragsgiver og Høgskolen.

Prosjektskissen skal inneholde følgende elementer:

- Overskrift som sier at dette er et bachelorprosjekt i data /informasjonsteknologi ved Høgskolen i Oslo, våren/høsten xxxx
- Foreløpig tittel på prosjektet.
- Navn på medlemmene i prosjektgruppa
- Navn og øvrige data (adresse, tlf.osv. )for oppdragsgiver
- Navn og øvrige data (tlf., stilling osv.) på kontaktperson/er hos oppdragsgiver
- Beskrivelse /skisse av prosjektet  
Skissen skal inneholde en kort presentasjon av oppdragsgiver, av prosjektets plass i oppdragsgivers virksomhet og en mest mulig omfattende beskrivelse av prosjektet selv. Hvis det allerede nå eksisterer krav til maskinplattform, dataverktøy el l., så få med dette.

Prosjektskissen skal lages som en internettside, og skal ikke være større enn en A-4-side hvis den skrives ut.

### 2.2 Prosjektdagbok

Et meget nyttig hjelpemiddel i prosjektarbeidet er en projektdagbok. Projektdagbok bør føres fra første stund i prosjektet. Den er nødvendig til en rekke nyttige formål:

- Hjelp til å huske inngåtte avtaler innenfor prosjektperioden
- Hjelp til å fullføre påbegynte planer innenfor prosjektperioden
- Muligheter til å skrive en fornuftig prosessbeskrivelse, ut fra nøyaktige notater
- Hjelp til å huske problemer og hvordan de ble løst (viktig for å skrive en god prosessrapport)
- Dokumentasjon for hva som er gjort og av hvem. (Viktig hvis dere skulle bli uenige om deltagelse i gruppa eller om dere fortjener samme karakter)



- Underlag for timelister

Føring av prosjektdagbok er en slags loggføring.

Prosjektdagboka føres hver gang det skjer noe med prosjektet som det kan være nyttig å huske. Felles møter med beslutninger om arbeidsmåte og arbeidsfordeling, avtaler med kontakter, avtalt arbeidsfordeling, arbeid som er fullført, individuelt arbeid, problemer som har oppstått og hvordan de ble løst, milepæler, viktige data om avtaler og avtalepartnere, osv. Det er umulig å huske slikt i detalj flere måneder senere.

De fleste fører prosjektdagboka på data. Prosjektdagboka organiseres slik at alle i prosjektgruppa har tilgang til filen. Der føres det både individuelt arbeid og felles arbeid, vanlige møter, kontakt med oppdragsgiver, eventuell kontakt med andre hjelpere, hva som ble ferdig og hvem som utførte det, osv. Hver gang registreres dag, tid og sted, og så det som var verdt å huske. Når det gjelder anledninger der flere i gruppa er med, bør man sikre seg at alle er enige i den beskrivelsen som er gitt.

På den måten kan man følge utviklingen dag for dag med tenking, utvikling, feilslag, testing og resultater. Når man til slutt skal skrive prosessdokumentasjon, har man alt dette samlet og kan se etter helheten. Hvordan gikk utviklingen i store trekk, hva var de viktigste ideene, hva var de vanskeligste problemene? Man kan se hvilke avtaler som faktisk ble gjort og hvordan de ble fulgt opp. Ikke minst er det viktig med en god beskrivelse av problemer man fant løsning på, og hvordan. Dette viser kreativitet og problemløsningskapasitet i gruppa, og blir lagt merke til i prosessdokumentasjonen.

Det er nyttig å uttrykke seg spesifisert og konkret om hvilke problemer man arbeider med. "Hadde problemer med at harddisken ble full" er en nokså intetsigende beskrivelse. Følgende beskrivelse er mer informativ og dokumenterer både arbeidsmåte og kreativitet, samtidig som den problemløsningen man kom frem til, kan brukes en annen gang også:

"Harddisken var full ved lagringstider, og vi måtte isteden lagre den på diskett. Da dokumentet skulle hentes inn igjen, fikk vi feilmelding om disk error. Det var ikke mulig å kopiere fra disketten, men det var mulig å lese dokumentet i Word. Vi forsøkte å lure operativsystemet, leste inn dokumentet og lagret det tilbake til fil på harddisk. Dette medførte utnulling av det gamle dokumentet på harddisken. Vi trodde en stund at hele dokumentet var tapt. Men Ståle fikk en ide: siden det var mulig å lese dokumentet, kunne vi bruke ClipBoard. Vi klippet derfor ut de delene av dokumentet som ikke var ødelagt (bare bildene var ødelagt) og lagret dem på nytt dokument. Heldigvis var de fleste skjermbildene lagret, men menyene måtte tegnes på nytt."

En sjelden gang har man bruk for prosjektdagbok i en helt annen sammenheng: Det hender at et av medlemmene i en gruppe ikke svarer til de andres forventninger om arbeidsinnsats og kompetanse, men opptrer mer som gratispassasjer. Det blir en belastning på de andre, og det er dessuten demotiverende at noen skal kunne få en god karakter uten å ha fortjent det. Så ønsker gruppa å kvitte seg med vedkommende.

I slike tilfeller kan det være avgjørende om prosjektdagboka er godt ført.

Hvis det kan dokumenteres at det har vært en tydelig forskjell på arbeidsinnsats i gruppa, vil det være vesentlig enklere å få tillatelse til å skyve en person ut av gruppa. Hvis det ikke

finnes dokumentasjon om arbeidsinnsats, kan ord står mot ord, og gruppa kan bli kommandert til å holde sammen, fordi man ikke har noe bevis å komme med.

Noen grupper skriver en samarbeidskontrakt før de starter bachelorprosjektarbeidet for alvor. Dette er ofte studenter som har brent seg på dårlig samarbeid tidligere. Ingen tror i utgangspunktet at de skal få problemer med samarbeidet, men det er mange som får litt mer problemer enn de hadde trodd - selv om det er sjelden at det ender med ønske om å utelukke noen fra gruppa. Å skrive en samarbeidskontrakt kan være en måte å unngå problemer på.

## **2.3 Forprosjektrapporten**

Når en prosjektskisse er godkjent, kan arbeidet begynne for alvor. Man må foreta en analyse av hva som skal gjøres og avgrense oppgaven mer presist. Utviklerne må foreta en grovvurdering av hvilke elementer oppgaven bør inneholde - hva det er mulig for gruppa å rekke – og hva som ikke er aktuelt. Videre må de vurdere hvilke andre praktiske eller økonomiske konsekvenser oppgaven innebærer, og ofte - hvis dette ikke er gitt som en del av rammebetingelsene - hvilke dataverktøy som skal brukes.

Samtidig må det avklares med oppdragsgiver at den oppgaven som trer frem på denne måten, virkelig er av interesse. Det kan hende at den må begrenses så sterkt at den ikke kan tjene oppdragsgivers formål lenger. Det kan også hende at det prosjektet oppdragsgiver ønsker seg, virker så litt lite, så man er redd for at det ikke gir nok arbeid og utfordring for den aktuelle studentgruppen. Men det lar seg nesten alltid gjøre å utvide et prosjekt, i samarbeid med oppdragsgiver: funksjoner og egenskaper og samspill som oppdragsgiver ikke har tenkt seg på forhånd, men som han kanskje ønsker når han blir tilbudt det. På den måten kan man ofte utvide et prosjekt som i utgangspunktet ser litt smått ut..

Selv om det virker mer tillokkende å kaste seg ut i arbeidet enn å regne og beregne prosjekttid, bør man gi seg tid til realistiske vurderinger på dette området.

Selvfølgelig lar det seg gjøre å begrense oppgaven også senere i prosjektperioden. Men det vil da være vanskeligere å finne en avgrensing som både passer gruppa og oppdragsgiver: forventninger er skapt, og noe skuffelse vil lett komme inn. Det er også mer sannsynlig at sluttresultatet blir dårligere enn det ellers kunne blitt. Når man til slutt likevel **må** kutte noe, er det lite trolig at det som kuttes, virkelig er det minst viktige. I en slik situasjon er det også sannsynlig at det er gjort en del arbeid som ikke kan brukes og som studentene ikke får uttelling for - verken hos oppdragsgiver eller på høgskolen.

Det dokumentet som skal sørge for et ryddig utgangspunkt, er forprosjektrapporten.

I et større prosjekt i en bedrift skal forprosjektrapporten fremlegge nok informasjon til at styret skal kunne vurdere om prosjektet skal settes i gang. For en mindre oppgave, som et bachelorprosjekt, er hensikten først og fremst at studentene skal klargjøre for seg selv og oppdragsgiver de handlingsrammer som de skal arbeide innenfor.

Forprosjektrapporten for et stort administrativt dataprojekt kan være svært omfattende, og inneholder analyse av konsekvenser på en rekke områder, ikke minst økonomiske og organisatoriske konsekvenser. "Null-alternativet" - det ikke å foreta seg noe med det aktuelle problemet - skal alltid vurderes.

Oppgaver av størrelse som bachelorprosjekter representerer imidlertid små prosjekter. Tidsperspektivet blir entydig: oppgaven skal løses til gitt frist. Gruppa har fått i oppdrag å finne en dataløsning på et mindre problem, og det er løsningen av problemet som utgjør grunnlaget for studentenes karakter. Det vil ikke være riktig av studentene å bruke ressurser på å utrede null-alternativet. Det er vanligvis heller ikke riktig å bruke ressurser på å utrede i detaljer økonomisk/administrative eller organisatoriske sider ved løsningen.

En forprosjektrapport for et bachelorprosjekt blir altså en svært forenklet forprosjektrapport i forhold til et større prosjekt i arbeidslivet. Rapporten er beregnet på studentene selv, veileder på høgskolen, og oppdragsgiver. Studentene skal presisere for seg selv hva prosjektet dreier seg om, og hvilket omfang det har. Veileder skal vurdere om studentenes presisering av oppgaven har slike kvaliteter at de kan arbeide videre på grunnlag av den, og oppdragsgiver må vurdere om dette er det arbeidet som ønskes utført.

Fordi man må gå en del inn i oppgaven og gjøre noe innledende vurderingsarbeid for å kunne lage en forprosjektrapport, vil man kunne gli nesten umerkelig over i det faglige arbeidet med oppgaven. Det lønner seg likevel å satse på å skrive en solid forprosjektrapport før man går løs på det faglige innhold for alvor. Det vil klargjøre egne tanker, og gi muligheter for presis tilbakemelding fra veileder og oppdragsgiver.

## Oppbygging av forprosjektrapporten

Vanligvis inneholder en forprosjektrapport følgende deler:

- Presentasjon  
Kort formell beskrivelse av gruppe, oppgave, oppdragsgiver med kontaktperson(er), veileder
- Sammendrag  
Sammendraget skrives for den travle oppdragsgiver/veileder slik at vedkommende skal komme raskt inn i saken og få oversikt over resultatene av det analysearbeid som er utført. Sammendraget skrives til sist, etter de andre delene av forprosjektrapporten, men plasseres foran i rapporten.

Her beskrives kort hovedtrekkene i den løsning som foreslås, sammen med hovedpunkter i de øvrige deler av forprosjektrapporten.

- Dagens situasjon

Dette kapitlet er særlig beregnet på veileder. Oppdragsgiver er oftest kilden for disse opplysningene, men det kan være nødvendig å snakke med flere i bedriften for å få tilstrekkelig oversikt. Ved prosjekter i større bedrifter, der ingen har full oversikt over hva som foregår i alle avdelinger, kan et slikt kapittel tjene til å klargjøre situasjonen f. eks. for ledelsen.

I kapitlet om Dagens situasjon beskrives bedriften, **med fokus på den situasjonen som har gjort dette prosjektet aktuelt**. I denne sammenhengen beskrives rutiner og praksis som ønskes endret ved hjelp av prosjektet.

- Mål og rammebetingelser

Her beskrives sentrale mål for systemet som skal lages, og samtidig de rammebetingelser som eksisterer - enten de er bestemt av oppdragsgiver, eller ligger i oppgave, situasjon og miljø, i valg av teknisk utstyr eller programvare - osv.

En slik klargjøring er grunnleggende for arbeidet med oppgaven. Den er også viktig for oppdragsgiver - som kan kontrollere om det arbeides med de riktige mål og forutsetninger - og for veileder, som kan vurdere analysearbeidet så langt og eventuelt styre i riktig retning.

- Løsninger /alternativer

Her beskrives hovedtrekkene i den løsningsmodellen som foreslås. Den må samsvare med mål og rammebetingelser. Beskriv også eventuelle fordeler og ulemper ved denne løsningen.

Ofte vil det eksistere flere mulige alternativer til løsning på dette problemet. Det er i så fall viktig for oppdragsgiver å vite hvilke valgmuligheter som eksisterer og hvilke konsekvenser det vil ha å velge hvert enkelt.

Beskriv hovedtrekk ved de aktuelle alternativer. Nevn fordeler og ulemper for hvert. Sammenligningen kan gjerne gis i tabellform.

- Analyse av virkninger

Forklar hvilke virkninger det (de) omtalte alternativene vil få.

Forsyn forprosjektrapporten med en **innholdsfortegnelse** hvis omfanget tilsier det.

Forprosjektrapporten skal dessuten **dateres** og **underskrives**, for hånd.

Denne ovennevnte rekkefølgen vil oftest passe rimelig bra. Hvis den derimot ikke passer, er det selvfølgelig mulig å velge en annen rekkefølge. Det viktige er å få med **innholdet** i disse punktene, og i en rekkefølge som passer med prosjekt og oppdragsgiver.

Klargjøring av oppgaven må følges av en plan for hvilket arbeid som skal utføres og hvordan fremdriften skal være.

## **2.4 Arbeidsplan og fremdriftsplan**

I et større administrativt prosjekt er arbeidsplan og fremdriftsplan en del av forprosjektrapporten, som skal vurderes av styre/ledelse for å se om prosjektet skal settes i gang. Disse planene skal hjelpe ledelsen til å vurdere ressursbehov og fremdrift, og til å holde styring med fremdriften

I bachelorprosjektsammenheng er disse dokumentene aller viktigst for studentene selv, men hjelper også veileder til å følge opp arbeidet og fremdriften.

I **arbeidsplanen** planlegges arbeidet med oppgaven. Hensikten er å få en oversikt over det arbeid som må gjøres for å fullføre prosjektet. Det totale arbeidet deles opp i naturlige deler,

som igjen detaljeres så godt det er mulig på dette tidspunkt. Det er viktig å komme ned på et konkret og praktisk plan, så det kan vurderes hvor mye tid som medgår til hvert enkelt punkt.

Det er klokt å bruke en del tid på arbeidsplanen. Dess mer gjennomtenkt denne er, dess bedre oversikt oppnås. Man risikerer ikke at viktige ting blir glemt til i siste liten. Det kan være lett å overse at også planlegging tar tid, at testing tar tid, og ikke minst: at det tar mye tid å utforme sluttokumentasjonen. Selv om man har gjort et godt arbeid med å notere og skrive deler av dokumentasjonen underveis, er det et stort arbeid å sette alt sammen. Har man ikke satt av nok tid, blir det et enormt arbeidspress mot slutten. Mange studenter har brent seg på det!

Mens arbeidsplanen utdyper **hva** som skal gjøres, forteller **fremdriftsplanen** hvor lang tid man regner med å bruke på de enkelte hoveddeler som arbeidet omfatter. Den angir når arbeidet med ett punkt skal være avsluttet, hva som skal være gjort før noe annet, eller hvilke arbeider som løper parallelt.

Det er ulike måter å bygge opp en fremdriftsplan på.

Den kan bygges opp omkring **milepæler**.

Den kan bygges opp etter **faser**, som eventuelt har avslutningsdato.

Den kan bygges opp etter **beslutningspunkter**.

I mange organisasjoner - spesielt offentlig administrasjon - må ulike forslag til tiltak godtas av ulike personer eller ulike beslutningsorganer - sikkerhetssjef, styremøte, generalforsamling, tilsettingsråd, fagforeninger osv. Prosjektet kommer ikke videre før det er gitt godkjenning for de beslutninger som foreslås. - Hvis et prosjekt har mange slike beslutningspunkter som er avgjørende for fremdriften, kan det være naturlig å bygge opp framdriften om disse.

Fremdriften for de fleste bachelorprosjekter kan med fordel utformes som et **Gantt-diagram**: her det lett å få oversikt over i hvilken tidsperiode man arbeider med denne delen av prosjektet, og hvilke ting man må arbeide med parallelt, og hvordan man totalt ligger an med hensyn på tidsbruk.

For noen prosjekter kan det lønne seg å få tak i et profesjonelt Gantt chart eller eventuelt et prosjektstyringsprogram som Microsoft Project.

### **Er det virkelig mulig å planlegge?**

Det er selvfølgelig ikke mulig å planlegge alt i detalj på et såpass tidlig stadium.

En del arbeid faller ut annerledes enn planlagt, og den tiden som medgår, vil avvike fra det som er planlagt. Fremdriftsplanen vil sjelden komme til å stemme helt med det som blir gjort.

Likevel er arbeidsplan og fremdriftsplan til stor nytte. Planleggingen bidrar til en gjennomtenking av problemene på forhånd og sikrer en oversikt som gjør det mye lettere å innhente eventuelle forsinkelser og fullføre arbeidet i tide.

Den arbeidsplan man klarer å lage på et tidlig tidspunkt, representerer en **grovplan**. Etter hvert som man får bedre oversikt og grundigere kjennskap til det man arbeider med, bør man utdype arbeidsplanen og fremdriftsplanen så langt det er mulig. Det er viktig hele tiden å ha

skikkelig, konkret oversikt over hvilke arbeider som skal utføres. Grovplanen bør for hver fase utfylles med detaljer for denne fasen.

## **2.5 Kravspesifikasjonen**

er en beskrivelse av de krav som oppdragsgiver/bruker har til det systemet som skal lages, uttrykt i et symbolsystem som er felles for oppdragsgiver og studenter. Kravspesifikasjonen må godkjennes av begge parter som den modellen det skal arbeides videre med, og fungerer derved som en kontrakt mellom partene. Når den er formulert, tjener den som rettesnor for utviklerne gjennom hele prosjektet.

Forprosjektrapporten gir en god oversikt over prinsippene ved den løsningen man har tenkt seg. Etter at det er gitt tilbakemelding på forprosjektrapporten, er det nødvendig å spesifisere prosjektet mer i detalj. Dette gjøres med kravspesifikasjonen.

### **Datainnsamling**

For å lage en god kravspesifikasjon trenger man en rekke data.

Viktige data i dette arbeidet er selvfølgelig oppdragsgivers egen beskrivelse av krav og ønsker til det produktet som skal lages. Kanskje er denne beskrivelsen så klar og utvetydig at grunnarbeidet er gjort - men den kan også være forholdsvis vag og uspesifisert ( det siste er ganske vanlig ved bachelorprosjekter.) Oppdragsgiver vet kanskje sånn omtrent hva produktet skal kunne gjøre - men har ikke noe særlig detaljerte synspunkter, og oppdager først etter hvert hva som virkelig ønskes. Det betyr at både den første og en eventuelt senere (endelig) kravspesifikasjon må skapes i samarbeid mellom utvikler og oppdragsgiver. Hvis det ikke er oppdragsgiver selv som skal bruke produktet, men noen som er ansatt hos oppdragsgiver, må de aktuelle brukerne trekkes med i arbeidet så snart som mulig, hvis da ikke oppdragsgiver insisterer på å være eneste informatør om brukernes situasjon og behov.

Svært ofte skal et datasystem hjelpe en bruker til å utføre det samme arbeid som før på en enklere måte, eller å få utført arbeid tilknyttet de gamle arbeidsoppgavene – arbeid som før kanskje var for stort eller umulig.

Når et nytt system skal designes, må man både ha klart for seg hvordan det aktuelle arbeidet blir utført i dag, og hvordan det nye systemet vil endre arbeidet.

For å skaffe den nødvendige informasjon kan det brukes mange metoder. Kvaliteten på det endelige resultatet vil være avhengig av at denne informasjonsinnsamlingen er solid nok. Nødvendig informasjon om bedriften/institusjonen fås vanligvis via oppdragsgiver. Ellers kan man bruke teknikker som intervjuer, spørreskjemaer, observasjoner og analyser av dokumenter som er sentrale i bedriften i den aktuelle saken. Man intervjuer og observerer potensielle brukere - de som i dag utfører det arbeidet som det bestilte dataproduktet skal endre – eller, hvis det dreier seg om funksjoner ingen har utført før – personer som kan komme til å bruke det nye produktet. Man undersøker deres synspunkter på arbeidet og på eksisterende produkter. Man studerer deres vanlige fysiske og psykiske arbeidsmiljø og måten de arbeider på.

## Endringer av kravspesifikasjon

Kravspesifikasjonen er, som tidligere sagt, både et styringsdokument og en del av sluttdokumentasjonen, samtidig som den er en kontrakt med oppdragsgiver om hva som skal gjøres. Den er altså et meget sentralt dokument.

Selv om en kravspesifikasjon er akseptert både av oppdragsgiver og prosjektgruppe, vil den ofte komme til å måtte endres. Både gruppa og oppdragsgiver kan komme fram til at noe kan gjøres enda bedre på en annen måte, så endringer i kravspesifikasjonen representerer vanligvis noe positivt. Er det mye endringer, bør det være den siste versjonen av kravspesifikasjonen som leveres med sluttdokumentasjonen.

Den kravspesifikasjon som leveres som sluttdokumentasjon, må være den som til syvende og sist uttrykker det som prosjektet skal bli. Kravspesifikasjonen skal være med for at veileder og sensor skal kunne sammenligne det spesifikke produktet med den generelle kravspesifikasjonen. Studentene skal i prosessdokumentasjonen forklare hvordan de har brukt kravspesifikasjonen i arbeidet med produktet, og i produktdokumentasjonen hvordan samsvaret er mellom kravspesifikasjon og produkt.

Hvis det derimot er gjort bare noen endringer, ikke så mange eller viktige, kan gruppa nevne disse, og årsaken til disse, i prosessdokumentasjonen.

Selvfølgelig må både oppdragsgiver og utviklere være enige om endringer i kravspesifikasjonen.

En kravspesifikasjon skal **ikke** være en beskrivelse av det produktet gruppa har laget. Selv om gruppa etter hvert som kravspesifikasjonen endres, har fått et klart bilde av hva de skal lage, skal ikke dette bildets spesielle detaljer komme fram i kravspesifikasjonen. En kravspesifikasjon er et **generelt** dokument. Det skal si **hvilken funksjonalitet** som kreves - ikke hvordan produktet skal se ut i detalj. Kravspesifikasjonen kan inneholde et krav om at et skjermbilde skal ha de og de egenskaper - men det er **ikke** en del av kravspesifikasjonen akkurat hvordan skjermbildet skal se ut. Kanskje man i løpet av prosessen er blitt enige med oppdragsgiver om det eksakte utseendet på skjermbildene, men likevel er ikke dette utseendet en del av **kravspesifikasjonen**. Kravspesifikasjonen skal være utformet slik at den ikke peker på et bestemt produkt – den kan tilfredsstilles av mange ulike produkter med ulike skjermbilder. Gruppas produkt skal være ett av disse produktene.

## Oppbygging av kravspesifikasjon

I og med at kravspesifikasjonen er et selvstendig dokument, vil den inneholde stoff som også finnes i andre deler av dokumentasjonen. Kravspesifikasjonen vil ofte fungere som en utdyping av stoff i forprosjektrapporten f. eks., og noe stoff vil være felles for disse to. Samtidig vil krav fra kravspesifikasjonen gå igjen i produktdokumentasjonen, kanskje i brukerdokumentasjonen, eller kanskje fremkomme som utfordringer nevnt i prosessdokumentasjonen. Det gjelder å holde fast ved det som er hensikten med kravspesifikasjonen, og den funksjon den har i dokumentasjonen, og formulere kravspesifikasjonen etter dette.

Det er vanlig å bygge opp kravspesifikasjonen på denne måten:

## Presentasjon

Som for forprosjektrapport

## Om bakgrunnen

Kort om bakgrunnen - litt om bedrift, situasjon og hensikten med produktet, men kortere enn i forprosjektrapport. Eventuelle hovedtrekk ved kravspesifikasjonen

## Forord

Hensikten med kravspesifikasjonen. Hvem kravspesifikasjonen er beregnet for. Kort om hvilken rolle den har spilt i prosjektet. (Mer om dette må plasseres i prosessdokumentasjonen)

## Leserveiledning

Hvordan kravspesifikasjonen er organisert

## Kort systembeskrivelse

med de forbedringer en venter å oppnå

Systemets funksjonelle egenskaper, beskrevet ved diagrammer og prosess-spesifikasjoner osv.

## Rammekrav i systemet, f.eks.

sikring mot tap, ødeleggelse, tyveri og misbruk av data  
kapasitet  
fremtidig utvidelse av systemet  
bruk og brukervennlighet

## Logisk datamodell

Eventuelle krav til systemkonstruksjon

Eventuelle krav til dokumentasjon

Eventuelle krav til manuelle funksjoner

Dataordbok

En del av dette lar seg generere ved dataverktøy.

## 3. Sluttdokumentasjon

### 3.1 Innbinding og rekkefølge

Som tidligere nevnt, består sluttdokumentasjonen av flere deler: produktdokumentasjon, testdokumentasjon, prosessdokumentasjon og brukerdokumentasjon. Disse delene leveres vanligvis i et og samme hefte. Det er et naturlig spørsmål hvordan disse delene skal organiseres sammen, og hvilke som skal stå først og sist.

Det er ofte greit å gi denne samlingen av dokumenter en felles forside, et forord som dekker hele prosjektet og en, kort, felles innholdsliste. Denne kan inneholde bare de store delene av prosjektet. Så kan man detaljere innholdslisten for hver av delene når man kommer til dem. Det bør være lett for leseren av dokumentasjonen å finne nøyaktig den hoveddelen av dokumentasjonen vedkommende er interessert i.

Hva som leses først og sist, vil være avhengig av hvem leseren er. Kjenner leseren litt til prosjektet eller produktet fra før, eller hvis leseren skal drifte systemet, vil vedkommende kanskje gå først til kravspesifikasjonen, for å se hvilket produkt dette prinsipielt er, eller til



produkt dokumentasjonen, for å se hvordan spesifikasjonene er blitt utformet i praksis. Vet man mindre om prosjekt og produkt, er prosessdokumentasjonen en god begynnelse. Ved å lese den forstår man bakgrunnen for prosjektet, og det forklarer både vektlegging og detaljer i den øvrige dokumentasjonen. Sensorer for bachelorprosjekt, som kommer utenfra, leser ofte prosessdokumentasjonen først.

I en del tilfelle er det prosessdokumentasjonen som er den egentlige prosjektoppgaven. Prosjektet er ikke å lage et dataprodukt, men å vise en (ønsket) prosess, beskrive og forklare et større system eller presentere et resonnement. Også i disse tilfellene er det naturlig å starte med å lese prosessdokumentasjonen.

Derfor starter vi gjennomgangen her med å beskrive prosessdokumentasjonen.

### **3.2 Prosessdokumentasjonen**

#### **Prosessdokumentasjonen – et viktig dokument**

Man kan selvfølgelig ikke fullføre prosessdokumentasjonen før prosessen er tilnærmet avsluttet. Men man bør arbeide med prosessdokumentasjonen gjennom hele prosjektperioden, fra første stund. En del av dette arbeidet foregår ved at man skriver prosjektdagbok, det er helt nødvendig. Men man kan gjøre mer: Man kan samle opp notater om sentrale punkter i utviklingen, og beskrive ulike faser av arbeidet ettersom de blir avsluttet. Egentlig bør det være bare "slutten" igjen når dere kommer så langt at hele prosessdokumentasjonen skal skrives.

Hvis man noterer og gjør seg ferdig med de enkelte deler etter hvert, sparer man tid. Det er når noe er ferskt, at det er lettest å beskrive det. Mot slutten av prosjektperioden er det så mange andre tråder som skal samles, så hvis prosessdokumentasjonen da er bortimot ferdig, vil det være en stor lettelse. Men avslutningen, vurderingen og evalueringen, må naturligvis fullføres mot slutten.

En rapport om et arbeid som er utført er et vanlig dokument så vel i arbeidslivet som i utdanning og forskning. Ved en slik rapport kan man beskrive innholdet i arbeidet til andre. Rapporten forteller sine lesere hvordan arbeidet er utført, og hvilket fundament det bygger på.

Når bachelorprosjektet evalueres, teller både produkt, prosess og dokumentasjonen av disse med. Prosessdokumentasjonen forteller i hvilken grad studentene har tilegnet seg kvalifiserte problemløsningsmetoder og arbeidsmetoder, og hvilken faglig utvikling de har gjennomgått.

I prosessdokumentasjonen legger studentene fram **bakgrunn og underlag** for det produktet de har laget. De forteller hvilke forhold gruppa har arbeidet under, hvilke arbeidsmåter de har valgt, hvilke rammebetingelser de hadde, hvilket arbeid som er utført (det er mye som ikke kommer direkte til syne i produktet), hvilke verktøy som er brukt, hvilke utfordringer arbeidet har bydd på og hvilke problemer og utfordringer man har funnet gode løsninger på.

Ofte ligger mye av det fortjenstfulle ved arbeidet nettopp i fakta som bare hører hjemme i prosessrapporten. Riktig brukt gir prosessdokumentasjonen en god mulighet for markedsføring for studentene og deres arbeid - uten at det trenger å være formet som skryt.

Det er viktig at prosessen og utviklingen er solid dokumentert. Veileder kjenner prosessen langt på vei fra veiledningen, men i møter med veileder snakker man oftest om dagens engasjementer og problemer. Full oversikt kan bare komme ved en samlet oppsummering. **Sensor** er helt avhengig av den skriftlige prosessdokumentasjonen. Hvis studentene klager på evalueringen, har en nyoppnevnt sensor ingen andre mulige kilder til informasjon om prosessen enn den skriftlige prosessdokumentasjonen.

Av og til skal en gruppe studenter/utviklere fortsette der en annen slapp, eller de skal implementere en oppgave som er systemert/designet av andre, de skal løse lignende problemer og/eller bruke samme programvare/utstyr som en tidligere gruppe. De vil kunne ha stor glede av en skikkelig dokumentasjon av arbeidet, både arbeidsmåten, problemer som er løst, og utviklingsgangen.

Ulike oppdragsgivere vil som nevnt ha ulikt forhold til prosessrapporten. Av og til er det nettopp analysen og prosessen oppdragsgiver ønsker. De resonnementer som ble brukt, de valg som ble gjort, begrunnelsen for valgene og de følger de fikk - er det viktigste for oppdragsgiver. Kanskje skal oppdragsgiver bruke prosessrapporten til videre arbeid med et problem. Da er prosessrapporten helt sentral, og oppdragsgiver vil lete her for å finne det som interesserer mest..

For mange oppdragsgivere vil imidlertid produktdokumentasjonen stå som det vesentligste. Men også disse vil likevel kunne dra nytte av en solid prosessdokumentasjon. I denne dokumentasjonen samles fakta om historikk, utvikling, problemer og utprøvde, ikke-fungerende løsninger - alt sammen informasjon som har klar nytteverdi. Slik informasjon er nyttig når man skal modifisere, feilsøke og vedlikeholde systemet, og vil kunne utgjøre en erfaringsbank for andre som skal arbeide med lignende problemstillinger. Sannsynligvis ville mye dobbeltarbeid kunnet unngås hvis man vente seg til å utnytte allerede dokumentert informasjon.

Problemet her er at det ikke eksisterer noen helt etablert tradisjon for bruk av slik informasjon, kanskje først og fremst fordi slik informasjon ofte er av veldig varierende kvalitet.

## To hovedtyper bachelorprosjekter

Den typen rapport som beskriver en prosess, er bygd opp etter grunnregler som er felles i hele den vestlige verden, med mindre variasjoner. Grunnideen i en slik rapport kommer fra matematisk/naturvitenskapelige forskningsmiljøer: Rapporten skal sette en person med **samme faglige grunnlag** i stand til å forstå, vurdere, kontrollere, eventuelt gjenta og etterprøve det arbeid som er utført.

For en del dataprojekter passer denne klassiske rapporten bra. Det er de prosjektene som ikke utvikler et datasystem og dokumenterer det, men som gjennomfører en undersøkelse, utreder et resonnement eller prøver ut en sekvens. Vi kaller dem litt upresist "utredningsoppgaver." For utredningsoppgavene blir prosessdokumentasjonen svært viktig, for det er den som inneholder de svarene oppdragsgiver har ønsket å få utredet.

Men de fleste bachelorprosjekter i data dreier seg først og fremst om å lage programmer eller datasystemer, og dokumentere dem. Da fungerer ikke den tradisjonelle rapporten lenger. Istedenfor at hoveddelen er et utredningsresultat, er det et programsystem og dokumentasjonen av dette utgjør hoveddelen.

De to rapporttypene er her satt sammen i en tabell. Det skilles mellom "rapport med program" og rapport med utredning". Det er lett å se at det er store likheter, men også store forskjeller.

Uansett må man bruke sunn fornuft ved utformingen av rapporten. Det er vanskelig å lage et skjema som dekker alle muligheter samtidig.

<b>RAPPORT MED PROGRAM</b>		<b>RAPPORT MED UTREDNING</b>	
<b>Presentasjonsdel</b>		<b>Presentasjonsdel</b>	
Forside		Forside	
Tittelside	Dett ligger en standard tittelside på nettet	Tittelside	Det ligger en standard tittelside på nettet
	Vanligvis ikke eget sammendrag for rapport med dataprogram	Sammendrag	Litt større enn det på tittelsiden. Det forteller kort hvilket arbeid som er gjort, hvilket spørsmål som skulle besvares, hvilken arbeidsmåte som er brukt, og hvilket svar på spørsmålene denne rapporten har funnet
Forord	Inneholder opplysninger om rapporten, ikke om faglig innhold. Hva slags dokument dette er, hvordan gruppa kom i kontakt med prosjektet, hva hensikten med arbeidet har vært, hvem som har vært til hjelp og skal takkes. Hvilket medium er rapporten optimalisert for?	Forord	Inneholder opplysninger om rapporten, ikke om faglig innhold. Hva slags dokument dette er, hvordan gruppa kom i kontakt med prosjektet, hva hensikten med arbeidet har vært, hvem som har vært til hjelp og skal takkes. Hvilket medium er rapporten optimalisert for?
Innholdsliste	Samtlige hoved- og deloverskrifter skal være med, og det skal fremgå visuelt hvilket nivå hver enkelt overskrift befinner seg på. Få med liste over referanser og liste over eventuelle vedlegg		Samtlige hoved- og deloverskrifter skal være med, og det skal fremgå visuelt hvilket nivå hver enkelt overskrift befinner seg på. Få med liste over referanser og liste over eventuelle vedlegg
<b>Hoveddel</b>		<b>Hoveddel</b>	
Innledning	Innledningen skal forklare den faglig/bedriftsmessige	Innledning	Innledningen skal forklare den faglig/bedriftsmessige

	bakgrunnen for problemet: bedriftens situasjon, oppgavens mål, rammebetingelser og begrensinger. Det meste av stoffet kan hentes fra forprosjektrapporten		bakgrunnen for problemet: bedriftens situasjon, oppgavens mål, rammebetingelser og begrensinger. Det meste av stoffet kan hentes fra forprosjektrapporten
Planlegging og metode	Hvordan planla dere- og hvordan fungerte planleggingen i prosessen? Hvilke verktøy ble brukt - og hvorfor? Hva var nødvendig å lære av nye ting? Hvordan arbeidet dere, og hvordan fikk dere tilbakemelding fra oppdragsgiver?	Faglige forutsetninger	Hvilke teorier og hvilke forutsetninger bygde dere på? Hvordan fremkom disse? Hvilke metoder ble brukt?
Om utviklingsprosessen	Hvilke utviklingsfaser har prosjektet hatt, og hvordan har dere reflektert over de faglige utfordringene?. Hvilke viktige valg om oppbygging og funksjon i programmet har dere gjort? Hva var ekstra vanskelig å få til? Hvordan utviklet forholdet til oppdragsgiver seg under prosessen?	Om utviklingsprosessen	Hvilken problemformulering startet dere opp med? Hvordan tenkte dere i starten om prosjektet, og hvordan har dette utviklet seg? Faser, problemer og løsninger.
Kravspesifikasjonen og dens rolle	Er kravspesifikasjonen endret fra første versjon? Hvordan - hvorfor? Hva betydde kravspesifikasjonen for utviklingen i design og implementering? Hvordan samsvarer kravspesifikasjonen med det produktet som beskrives i produkt-dokumentasjonen? .		
Om resultatet	Dataprogram på CD, dokumentasjon om dette. Dette finnes i Kravspesifikasjon,	Om resultatet	Resultatet av undersøkelsene som er gjort. Det er ofte oppsummeringer,

	Produktdokumentasjon og Brukerdokumentasjon		oversikter, konklusjoner. En del kan gjerne gjengis i tabellform
		Tolking av resultatene	Hvordan kan/skal resultatene tolkes? Hvilke faglige/praktiske konsekvenser har de?
<b>Avsluttende del</b>	Det bør alltid være en avslutning. Man kan runde av på en eller flere følgende måter 1) snakke om eget utbytte 2) gi oppsummering og konklusjoner 3) fortelle hva man ville gjort annerledes hvis man skulle begynt på nytt, 4) peke mot fremtiden ved å fortelle hva produktet kan brukes til, og hvordan det vil komme til nytte. Hva sier oppdragsgiver om produktet? Skal det snart taes i bruk/settes i produksjon? Skal det tjene som utgangspunkt for noens videre arbeid?	<b>Avsluttende del</b>	Det bør alltid være en avslutning. Den utgjør oftest en faglig oppsummering eller konklusjon. Videre kan man fortelle hvilken betydning resultatet har for oppdragsgiver, eller for fagfeltet generelt.

I begge typer rapporter må referanser og vedlegg være med.

### **3.3 Produktdokumentasjonen**

er beregnet på dataansvarlig, den som skal installere, vedlikeholde og modifisere systemet. Produktdokumentasjonen er også nyttig for dem som eventuelt skal markedsføre systemet eller drive brukerstøtte på det.

Dataansvarlig eller supporter trenger informasjon om installasjon, drift og feilfinning. For å kunne utføre sine oppgaver trenger den dataansvarlige informasjon om programmets oppbygging, virkemåte og funksjoner, og om hvilken testing som er gjennomført.

Produktdokumentasjonen, sammen med programmets selv, er dessuten en viktig del av grunnlaget for evaluering av et bachelorprosjekt. Her får sensor og veileder oversikt over systemets egenskaper og funksjon.

Produktdokumentasjonen er vanligvis den klart største delen av en bachelorprosjektdokumentasjon. Den består egentlig av fire deler: den endelige kravspesifikasjonen, CD-en med programmet, beskrivelsen av programmet og testdokumentasjonen for programmet.

Kravspesifikasjon er omtalt andre steder i dette kompendiet, og trenger ikke gjentas. Beskrivelsen av programmet og testing av det er helt sentrale deler av dokumentasjonen. Derimot har vi ikke tidligere omtalt verken beskrivelsen av programmet eller testingen av det.

### **Beskrivelsen av programmet**

kan og bør henvise til program-CD-en, men skal også kunne leses alene.

Det finnes mange typer programprodukter, som alle skal dokumenteres forskjellig. Her må man bruke mye skjønn, og tenke nøye over situasjonen til den som skal drifte og vedlikeholde akkurat dette programmet. Nedenfor er en opplisting av prinsipielle elementer i en vanlig type produkter. Det passer ikke på langt nær alle, men kan være en antydning av hva slags type stoff om produktet som kan være viktig.

- Innholdsliste
- Forord (husk å få med hvilken kompetanse som forutsettes av den som skal bruke produktdokumentasjonen)
- Beskrivelse av programmet - hva som er hensikten med det, og hva det gjør rent prinsipielt
- Samsvar mellom kravspesifikasjon og produkt
- Sentrale datastrukturer i programmet
- Programmets oppbygging og virkemåte
- Hovedprogram og underprogrammer  
( gjerne fremstilt som en visuell oversikt av en eller annen type, eventuelt menyoversikt)
- Forhold til maskiner, lagerplass, operativsystemer o.l.(funksjonelt grensesnitt)
- For hver hoveddel av programmet :  
    Kort presentasjon med formål  
    oversikt over rutiner og subrutiner  
    særlige forhold

Det bør fremgå klart hva som er **egengenerert** kode og hva som er automatisk generert. Bruk tydelige kommentarer for den egengenererte koden.

Figurene brukt i teksten skal nummereres, de skal ha navn, det skal henvises til dem i teksten, og de skal om mulig plasseres nær henvisningen.

### **3.4 Testdokumentasjonen**

Testdokumentasjonen leveres vanligvis som en selvstendig del av sluttdokumentasjonen. Hvis det har vært lite testing, kan den utgjøre en del av prosessdokumentasjonen. Den er ellers nært knyttet til produktdokumentasjonen.

Testdokumentasjonen er viktig for den som skal drifte og vedlikeholde programmet. Solid og tilstrekkelig testing gjør programmet bedre. Skikkelig dokumentasjon av denne testingen gjør

arbeidet lettere for driftsansvarlig. Det viser sensor og veileder at gruppa satser seriøst på et hensiktsmessig program og forstår betydningen av testing.

Alle typer testing som er gjort, bør dokumenteres, enten det er det som kalles brukertesting, eller det dreier seg om feiltesting av programkoden, eller for den saks skyld funksjonstesting.

Det er lurt å legge opp en plan for testingen på forhånd.

Når det gjelder brukerdokumentasjon, er det særlig viktig å ha klart for seg hva man er interessert i å teste, og hvordan dette kan gjøres. Testplanen må utformes med tanke på disse to parametrene. Når det gjelder funksjonell testing, må man også bygge opp testplanen etter slike hensyn, men her kan det ofte bruke et prinsipielt utkast til testplan som vises nedenfor.

Testplanen bør også bygges opp slik at den med noen endringer (resultater, forklaring av resultatene og kommentarer til dem) kan blir til Testrapport.

Nedenfor er en prinsipiell testplan og en prinsipiell testrapport stilt ved siden av hverandre, for å vise hvor nær beslektet de kan være:

<b>Testplan</b>	<b>Testrapport</b>
Hva er formålet med testen?	Hva var formålet med testen?
Hvilket område av programmet gjelder den?	Hvilket område av programmet gjaldt den?
Hvilken metode skal brukes?	Hvilken metode ble brukt?
	Tid, sted og organisering av testen
Eventuelle testprosedyrer	Eventuelle testprosedyrer
Beskrivelse av testen: Hvilke inndata som vil bli gitt, hvilke resultater man venter	Hva som ble testet, og resultatene av testen
	Forklaringer og kommentarer til resultatene
	Eventuelle konklusjoner som ble trukket av testen

### **3.5 Brukerveiledning**

Brukerveiledning bør gis både som veiledning på skjermen (prompts, tilbakemeldinger, statusbeskrivelse, feilhåndtering og hjelpeinformasjon) og ikke minst som skriftlig brukerveiledning.

I de fleste bachelorprosjekter der det finnes et dataprodukt med brukere, er det nødvendig med skriftlig brukerinformasjon. Når produktet er en webside, må det riktignok være så selvforklarende at vanlige brukere kan ta det i bruk uten videre. Men ofte er det en administrator for systemet, som skal kunne endre og redigere den informasjonen som finnes på nettsiden, og da vil det trenge noe skriftlig brukerinformasjon.

Brukerveiledningen må alltid innrette seg etter de brukerne man regner med – deres behov, kunnskaper, brukersituasjon, erfaring. Regler for brukerdokumentasjon må derfor alltid vurderes i forhold til den/de brukere man har å gjøre med.

## To typer fakta

Brukerdokumentasjonen på papir beskjeftiger seg i hovedsak med to typer fakta: HVA kan systemet gjøre – og HVORDAN skal brukeren handle for å oppnå sine mål med systemet? Disse to typene fakta krever hver sin struktur, og bør ikke blandes alt for tett sammen. I store systemer plasseres de ofte i hver sin del av en manual eller får til og med to ulike manualer.

Den ene delen, referansedelen, gir en oversikt over systemets **muligheter** – hvilke systemer, funksjoner og opsjoner det er mulig å bruke, hvordan de fungerer, og hvordan de hører sammen. Det er ikke meningen at man skal lese denne delen fra begynnelse til slutt, den er organisert som et oppslagsverk, og er særlig beregnet på øvede brukere. Man skal her kunne finne fram til og fordype seg i den funksjonen eller den sammenhengen man har bruk for å vite mer om.

Den andre delen forteller hvilke **prosedyrer** som må følges for at man skal oppnå bestemte mål. Her der det brukerens mål som står i fokus. Denne delen er spesielt nødvendig for nye og uøvde brukere. Her bør stoffet være organisert som sekvenser av operasjoner som til sammen, og i angitt rekkefølge, fører til det ønskede resultatet.

Det er flere varianter av prosedural brukerveiledning. Ofte lager man en redusert utgave av brukerveiledningen med tanke på brukerens situasjon, sånn som miniinnføring eller referansekort.

Hvis omfanget av systemet ikke er for stort, legge man ofte både referansedel og veiledningsdel inn i samme manual, og med noen felles kapitler.

Nedenfor er en vanlig oppbygging av en slik manual, som ofte kan passe for bachelorprosjekter.

## 3.6 Oppbygging av brukermanual

Brukermanualen inneholder vanligvis disse elementene:

- tittelside
- forord
  - hvilket produkt?
  - hvem er leseren og hvilke forkunnskaper kreves?
  - finnes det beslektede manualer?
  - evt. litt om manualen selv
- innholdsliste
- notasjoner/terminologi. Symboler og spesialuttrykk forklares
- innledning: overordnet presentasjon av programmet
  - hvilke problemer tar programmet sikte på å løse?
  - hva kan programmet gjøre?
  - hvordan fungerer dialogen med skjermen i hovedsak?
  - fordelingen av skjerm-info og papir-info
  - hvilke typer meldinger gir programmet?
  - evt. liste over (typer) feilmeldinger
- hoveddel: veiledning i å bruke programmet



Både en instruksjonsdel med prosedyrestruktur og en referansedel med oppslagsstruktur

- feil og rettingsmuligheter
- stikkordliste
- vedlegg
- indeks/ordliste (datatekniske begreper som er ukjente for brukeren)

Som man ser, er det mange felles elementer her.

I referansedelen kan hver rutine forklares slik:

navn  
formål  
hva rutinen gjør  
eksempler på inndata, skjermer, rapporter  
meldinger, feilmeldinger  
hvordan rutinen brukes

Korte kapitler om **installasjon, vedlikehold og utvidelse/modifisering** kan naturligvis også innarbeides i brukermanualen som separate kapitler/enheter.

Hvis systemet er så stort at det er naturlig med separat innføring og referansehåndbok, gir den ovenfor beskrevne strukturen en grei mal for begge - bortsett fra at det blir bare en hoveddel i hver av disse to separate manualene.

### **3.7 Gode råd for brukermanual**

#### **Rekkefølge og struktur**

Det er viktig for bruker å få oversikt over systemet så fort som mulig. Bruk f. eks. et menykart.

Forutsetninger for det som skal gjøres, bør listes opp før man gir beskjed om å gjøre det.

Advarsler bør komme **før** man risikerer å gjøre det som det advares mot

Forklar operasjonene som skal gjøres i en rekkefølge som er naturlig for mottaker.

Er det mange likeartede operasjoner etter hverandre og i rekkefølge: nummerer dem

**Tempoet** kan skifte gjennom manualen. Vær langsom og forklar godt i de delene som **må** leses først. Dette gjelder både den overordnede presentasjon av programmet og i forklaringen av de første instruksjonene. Etter hvert både kan og bør tempoet økes - det som nå skal forstås, bygger på at det tidligere stoffet er forstått. Innenfor det enkelte punkt nevnes det **viktigste** først, og hovedregelen nevnes før unntaket.

Det er vanskelig for bruker å holde orden på mer enn fire nivåer.

#### **Språk**

Gi direktivene en **aktiv** og konkret form. Kommenter hvis nødvendig, men la det fremgå klart at det er kommentarer.

Bruk enkle, oversiktlige og entydige setninger. Del opp og omorganiser setningen hvis den blir for lang.

Legg arbeid i overskrifter og underoverskrifter. De bør korrespondere med tilsvarende menyvalg, men gjerne inneholde en informativ utdyping/forklaring.

## Forskjellig

Ikke tving leseren til å lese alt for mye før han kan starte med å bruke systemet

Hvis noe er vanskelig å forstå, bruk mer tid og plass på det enn på det som er enkelt.

Ulike illustrasjoner hjelper ofte på forståelsen og hukommelsen. Illustrasjoner skal ha en forklarende tekst (bildetekst), og være nummererte.

Figurer og tabeller kan nummereres for hver del for seg, eller fortløpende. Oftest er det enklest og mest oversiktlig å nummerere fortløpende. Blir det svært mange figurer eller tabeller, kan man begynne nummereringen på ny for hvert kapitel eller hver hoveddel.

Bruk eksempler i form av skjermbilder eller dialoger. Plasser disse riktig i forhold til den teksten som beskriver dem. Henvis til bildene i teksten. Det kan være hensiktsmessig å nummerere skjermbilder, slik at man kan henvisе til et spesielt nummer for å forklare en detalj man ikke ønsker å bruke et eget skjermbilde til. Er det mulig, bruk et gjennomgående eksempel.

Bruk tilstrekkelig store typer, men **versaler** (såkalt "store bokstaver") må bare brukes i korte tekstbiter på maksimum et par linjer av gangen. Versalene inneholder nemlig mindre informasjon enn de varierte "små bokstaver" og er mer anstrengende for brukeren å oppfatte (det går 15% langsommere å lese versaler enn vanlig tekst.).

Brukermanualer bør testes like grundig før de sendes ut som programmet/systemet for øvrig.

Instruksjonen i innføringsdelen og dokumentasjonen i referansehåndboka bør sees i sammenheng med en eventuell opplæringsdel, og med hjelpefunksjonen.

## Dokumentasjon som er viktig for datasikkerheten

Dokumentasjon er mer enn den dokumentasjon som leveres som del av et bachelorprosjekt. Også andre typer dokumentasjon er viktig i et datasystems livssyklus, og de som blir ansvarlige for en dataavdeling eller for datasikkerhet, må ikke glemme disse. Dokumentasjon er også

- Sikkerhetskopier, som sikrer mot katastrofer ved ulike maskinavbrudd og virusødeleggelser

- Systematisk merking av sikkerhetskopier
- Aktivitetslogger, kan avsløre inntrengere og gir mulighet for å avsløre dem
- Dokumentasjon av tester og rutiner kan være viktig i rettssaker om ansvar for sikkerhetsbrudd

Rent generelt er god dokumentasjon nyttig for datasikkerheten.

## 4. Muntlig presentasjon av et prosjekt

Når oppgaven er endelig skriftlig formulert og levert, skal den presenteres muntlig. Den skal enten presenteres for oppdragsgiver/samarbeidspartner eller for sensor, veileder og medstudenter på Høgskolen - eller begge deler.

Alt nødvendig stoff er nå samlet i sluttrapporten, og det som presenteres muntlig, må ta utgangspunkt i dette og få med det viktigste herfra. Likevel må man ikke binde seg til stoffmengden eller oppbyggingen i hovedoppgaven. Den muntlige presentasjonen må helt sikkert bli mindre omfattende enn den skriftlige, og fremføres i et annet språk. Dessuten må den tilpasses tilhører. Presentasjonen blir forskjellig avhengig av om den skal gis for oppdragsgiver alene, for veileder alene eller for en blandet forsamling med veileder, sensor, medstudenter og andre lærere, samt eventuelt representant for oppdragsgiver. Presentasjonene må innrettes for dem som er viktigst for funksjonen av den aktuelle presentasjon. I avsluttende presentasjon på høgskolen er vel dette veileder og sensor(er), ved presentasjon i bedrift er det bedriftsveileder, eventuelle ledere i bedriften, eventuelt fremtidige brukere, som er viktigst.

Vanligvis vil det være oppgitt en tid - f. eks. en halv time. Det er viktig å holde tiden. Hvis deres presentasjon foregår på bedriften, vil de som kommer, ofte ha begrenset tid. Det lønner seg å få sagt det man vil før de fleste er gått.

Foregår presentasjonen på høgskolen med blandet tilhørerskare, er det gjerne andre grupper som skal presentere sitt arbeid etter dere, mens veiledere og sensorer venter. Forskyvinger i timeplanen fører lett til kjedeligheter for noen av de andre.

Ved presentasjon hos **oppdragsgiver** vil det være størst interesse for resultat/program, altså produktet. Kontaktpersonene hos oppdragsgiver kjenner naturligvis en del av utviklingen og problemene. Er det også andre fra bedrift/samarbeidspartner til stede, kan dere redegjøre helt kort for prosessen - bakgrunn, mål og forutsetninger, utviklingen, problemer som har hatt betydning for arbeidet.

Men hovedsaken blir **produktet**: egenskaper, virkemåte, hvordan produktet overensstemmer med rammebetingelser og spesifikasjoner, hvordan det er testet og hvordan det kan brukes.

Ved muntlig presentasjon av bachelorprosjektet, som del av evalueringen, kan det ofte være medstudenter til stede, og veiledere for andre prosjekter - men iallfall vil både veileder ved høgskolen, og sensor, være til stede. De skal evaluere den læringsprosessen som har foregått, og har fått en del kunnskaper om denne fra den skriftlige rapporten. For dem begge vil det

likevel være interessant å få en samlet oversikt over utviklingen av prosjektet - med vekt på utvikling av ideer, planlegging, arbeidsmåte, hvordan dere har løst problemer, og prinsipielle trekk ved systemet som er utviklet.

De vil også være interessert i den løsningen dere har valgt, og produktets egenskaper, virkemåte, overensstemmelse med rammekrav og spesifikasjoner - og bruk.

Medstudentene er interessert i det emnet dere har og i hvordan dere presenterer.

Enten forsamlingen er slik eller slik, bør programmet naturligvis **demonstreres** - med vekt på det som er mest interessant for **denne forsamlingen**.

#### **4.1 Demonstrasjonen**

Demonstrasjonen tas til slutt, eller i forbindelse med orienteringen av produktet.

Å demonstrere et program ser lett ut når andre gjør det, hvis de gjør det riktig. Men en demonstrasjon har alltid begrenset tid, og skal man utnytte den best mulig, må det planlegges nøye.

Det er en kunst å demonstrere et program overbevisende.

Demonstrasjoner har alt for lett for å bli en temmelig kjedelig rekkefølge av skjermbilder med piler som peker på et alternativ, nytt skjermbilde med ny(lik) pil, nytt skjermbilde med ... og hastige kommentarer som er vanskelige å oppfatte. Demonstranten kjenner sitt program godt og vet selv hvor han er når han sier "her" og "der", men tilskuerne trenger lengre tid og nøyaktigere angivelser.

Det er vanligvis unødvendig å få presentert hvert eneste skjermbilde i programmet. Tilhørerne trenger ikke se alt, men må få et representativt inntrykk. Bruk gjerne kontekst- og målanalyse for å finne de riktige skjermbildene.

Velg forholdsvis få, viktige skjermbilder, og la dem se tilstrekkelig lenge på hvert enkelt bilde til å forstå hva det beskriver og hvordan det fungerer. Alle "veier" mellom skjermbilder som skal sees på, må gjøres så usynlige som mulig. Bruk gjerne et presentasjonsverktøy som bare viser akkurat de skjermbildene dere virkelig ønsker at forsamlingen skal se på. Knytt en del kommentarer og forklaringer til hvert bilde. Ved hjelp av et ikke for stort antall bilder og kommentarene til dem må dere få fram alle de viktige sidene ved brukergrensesnittet.

Velg **en hensiktsmessig rekkefølge** av skjermbildene som vises. Kanskje kan det greie seg med færre skjermbilder hvis rekkefølgen er fornuftig.

#### **4.2 Muntlig sluttrapportering i punkter**

Nedenfor følger en "mal" for presentasjon av et bachelorprosjekt for sensor/faglærer, eventuelt med andre lærere, medstudenter og representanter for oppdragsgiver til stede. Denne måten å bygge opp presentasjonen på, fungerer vanligvis godt. Det betyr naturligvis ikke at

presentasjonen **må** være akkurat slik: bruk sunt skjønn og avvik fra malen der det er gode grunner for det. Denne malen sikrer imidlertid at alle viktige punkter blir berørt.

- "Overskrift": fortell kort hva oppgaven deres handler om. Vis tittelen på et ppt-skjerm bilde e.l.
- Hvem har utført prosjektet? Presenter gruppa
- Hva var bakgrunnen for oppgaven: forklar hvem oppdragsgiver er og beskriv oppdragsgivers situasjon, så mye som må til for at tilhørerne skal få et klart bilde av hvilket behov som eksisterte.
- Kort om hvordan dere kom i kontakt med oppgaven/hvorfor valgte den
- Hva oppgaven helt presist går ut på. Problemformulering, mål og forutsetninger, om det var alternativer og hvorfor dere valgte dette.
- Kort omtale av planleggingen og hvilken rolle den har spilt for deres arbeid. Det viktigste er å få fram at dere har brukt planleggingsverktøy, og hvordan dere holdt fremdriftsplanen. Da kan dere også forklare eventuelle faglige/praktiske grunner til at dere ikke klarte å følge arbeidsplan/framdriftsplan.
- Om utviklingsprosessen.  
Dette kan være ganske avgjørende for den karakteren som gis. Ved å fortelle om utviklingsprosessen, kan man forklare hvorfor oppgaven mangler den eller den egenskapen (hvorfor den f. eks. ikke er fullført, eller hvorfor den har lagt vekt på dette eller hint, hvorfor den opsjonen mangler osv.) Man kan forklare hvilke problemer man har løst, som har tatt tid, eller hvordan programmet bygger på at man har løst problemer som det ikke kunne forventes at man skulle løse.  
Husk mulighetene for å få frem deres egen innsats, kreativitet og problemløsende evner!
- Beskriv det resultatet dere er kommet fram til - altså det produktet dere har laget.

#### DETTE ER DEN STØRSTE OG VIKTIGSTE DELEN AV PRESENTASJONEN!

Både dataløsning og brukergrensesnitt - slik det er naturlig. Hvordan svarer deres løsning til oppdragsgivers krav? Hva er nytt i deres program? Hvordan fungerer programmet? Hvordan kan det utvides og endres? Hva er oppdragsgivers reaksjon på produktet? (Få for all del med f. eks. informasjon om at produktet skal settes i drift - brukes som underlag for - osv.)

- Eventuelt demonstrasjon nå - eller etter avslutningen. Si gjerne litt om brukerdokumentasjonen dere har laget, iallfall hvis den er gjennomarbeidet.
- Avslutning. Oppsummering av arbeid - gruppas utbytte. Hva vil produktet deres bety for oppdragsgiver? For verden? Ville dere gjort noe annerledes hvis dere skulle gjort oppgaven på nytt? Pek gjerne mot fremtiden!
- Åpne for spørsmål

