

# INF3120 - Obligatorisk prosjektarbeid

## INNHOLD:

1	Bakgrunn.....	2
2	Læringsmål .....	2
3	Vurderingskriterier.....	2
4	Organisering av prosjektarbeidet .....	3
4.1	Grupper .....	3
4.2	Viktige aktiviteter og leveranser.....	4
4.3	Støtte til prosjektarbeidet.....	5
4.4	Prosjektgruppene.....	6
5	Kravbeskrivelse.....	6
5.1	Fullstendig kravspesifikasjon.....	6
6	Retningslinjer for leveransene .....	6
6.1	Generelt.....	6
6.1.1	Tittelside .....	6
6.1.2	Lagringsformat.....	6
6.1.3	Endringer.....	7
6.2	Leveranse I (innleveringsfrist: 24.09).....	7
6.2.1	Prosjekthjemmeside .....	7
6.2.2	Prosjektplan.....	7
6.2.3	Use Case Modell.....	8
6.3	Leveranse II (innleveringsfrist: 01.10) .....	10
6.3.1	UML Design v.h.a. klassediagram.....	10
6.4	Leveranse III (innleveringsfrist: 15.10) .....	11
6.4.1	”Reverse engineering” .....	11
6.4.2	Oppdatert prosjektplan (midtrapport) .....	11
6.5	Leveranse IV (innleveringsfrist: 19.11).....	11
6.5.1	Design av ny funksjonalitet .....	11
6.5.2	Kodegenerering og implementering av ny funksjonalitet.....	12
6.5.3	Oppdatert klassediagram.....	13
6.6	Leveranse V (innleveringsfrist: Prosjektet presenteres i perioden 22.11 - 26.11) 13	
6.6.1	Presentasjon av prosjektet.....	13

# 1 Bakgrunn

Et viktig prinsipp for INF 3120 er sekvensen: LÆRE - BRUKE - EVALUERE. Forelesningene og egenlæringen vil ofte ha fokus på tilegnelse av kunnskap (LÆRE). Øvelsene i grupper og, ikke minst, prosjektoppgaven har fokus på oppøvelse av ferdigheter (BRUKE). Prosjektoppgaven innebærer i tillegg øvelse i EVALUERING av eget og andres arbeid. Prosjektoppgavens rolle er altså i hovedsak å bidra til at kunnskapen fra forelesningen bidrar til ferdigheter og evalueringsevner innen systemutvikling. Dere vil, av ressursmessige grunner, ikke kunne gjennomføre et 100% realistisk prosjekt. Selv om læringseffekten har hatt størst fokus, har det likevel vært vårt mål å ha en stor grad av realisme i prosjektgjennomføringen. Graden av realisme avhenger imidlertid mye av dere!

For å kunne gå opp til eksamen i INF 3120 må prosjektgruppen du tilhører ha fått godkjent alle prosjektleveransene. Dette er kursets obligatorisk oppgave. 25% av oppgavene på eksamen vil omhandle prosjektarbeidet.

**Viktig:** Det er mye som skal klaffe for at gjennomføringen av prosjektoppgaven skal gå knirkefritt. Regn med at ikke alt er strømlinjeformet. Kom gjerne med forbedringsforlag og kritikk underveis.

## 2 Læringsmål

Gjennom prosjektarbeidet skal det bl.a. oppnås ferdigheter og evalueringskompetanse innen:

- 1) Prosjektplanlegging og gjennomføring
- 2) Beskrivelse av krav vha UML Use Cases
- 3) Utarbeidelse og gjennomgang av objektorientert design
- 4) Kodegenerering (Java) og "reverse engineering"

Ferdighetene skal i størst mulig grad være ihht prinsipper innen "software engineering".

**NB: Det forutsettes at dere har noe forkunnskaper om UML (f eks fra INF 102) eller skaffer dere elementærkunnskap om UML på egenhånd.**

## 3 Vurderingskriterier

For at en leveranse skal bli godkjent må den være **levert innen angitt tid** og ha **tilstrekkelig kvalitet**. Alle SKAL-krav i retningslinjene (beskrevet i dette dokumentet) MÅ være oppfylt.

Tilstrekkelig kvalitet vurderes av kursansvarlige og gruppelærere på bakgrunn av om leveransen:

- 1) er egnet grunnlag for videre arbeid i prosjektet, og
- 2) viser at studentene kan anvende kunnskapen fra INF 3120-forelesningene og andre relevante informasjonskilder.

Dersom en leveranse ikke blir godkjent i første omgang eller blir forsinket betyr dette at prosjektet må forbedre leveransen og får dårligere tid på de resterende leveransene. Prosessen underveis vil være et moment ved vurdering av godkjent/ikke godkjent av gruppens arbeid som helhet. Vi vil IKKE akseptere at noen "spekulerer" i å levere halvgode leveranser for at gruppelærer skal komme med innspill på hva som mangler for at leveransen blir godkjent. Dersom en leveranse ikke oppfyller alle SKAL-kravene vil prosjektgruppa få tilbake leveransen uten kommentarer med beskjed om at "leveransen mangler SKAL-krav". Utfyllende kommentarer til leveransene gis først når alle SKAL-krav er dekket.

**Viktig:** Det kan være gode grunner til at leveranser ikke blir fullført, f.eks problemer med verktøy eller sykdom. Dersom slike situasjoner inntreffer skal grunner til at leveransen ikke ble fullført beskrives og sendes til kursansvarlige, sammen med hva prosjektet har gjort for å løse dette. Skissér også en revidert realistisk plan for det videre arbeidet. På bakgrunn av en slik beskrivelse kan også ufullstendige leveranser godkjennes. Hvis en person er syk i noen dager, så ansees ikke det som en holdbar grunn for at prosjektet eventuelt er blitt forsinket. At noen i prosjektgruppa blir syke er noe man må ta hensyn til i selve prosjektplanleggingen (risiko-faktor).

## 4 Organisering av prosjektarbeidet

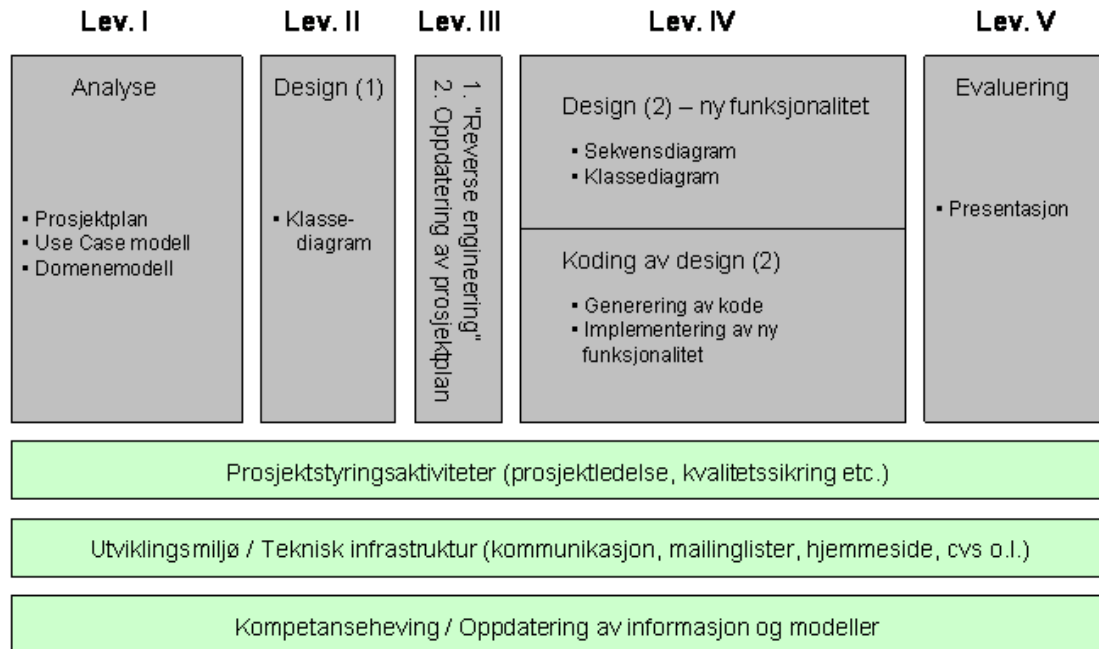
### 4.1 Grupper

Du vil tilhøre en gruppe på 4-5 personer. Gruppetilordning bestemmes av kursansvarlige på bakgrunn av en kartlegging av erfaring til den enkelte. Oppdragsgiver til prosjektet vil være en av gruppelærerne.

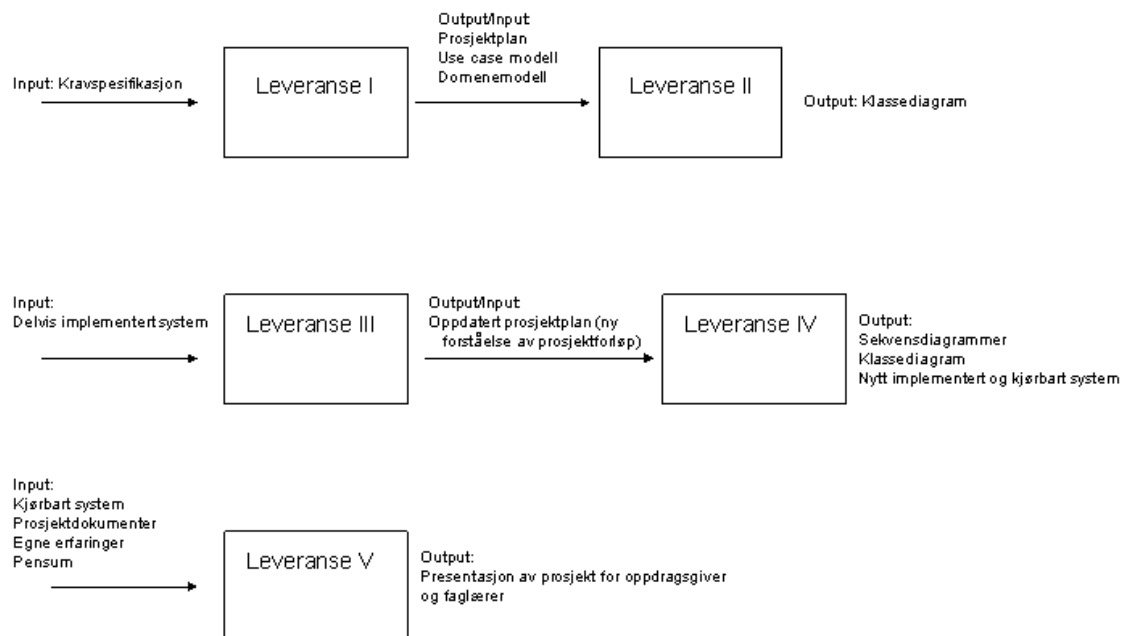
Alle i gruppen bør ha god oversikt over alt som foregår, men alle behøver ikke å utføre alt.

Dersom en leveranse ikke godkjennes må det arbeides mer på denne og det blir tilsvarende dårligere tid på de neste leveransene. Sørg for at det ikke skjer! Foreta gjerne noen kontroller mot gruppelærer underveis for å sikre at leveransen blir god nok.

## 4.2 Viktige aktiviteter og leveranser



Figur 1: Leveranser og prosesselementer i gjennomføringen av INF3120 prosjektet.



Figur 2: Input til og output fra de ulike leveransene.

### **Viktige aktiviteter:**

- 1) Etablering av prosjektgruppene (inndeling i prosjektgrupper, opprettelse av prosjekt-hjemmesider, etablering av roller, etc.)
- 2) Oppdragsgivers kravspesifikasjon legges ut på INF3120-hjemmesiden (Prosjektnavn: Hospital 2004, Systemnavn: HSS)
- 3) Utarbeidelse av prosjektplan
- 4) Eventuelt: Gjennomføring av opplæring i verktøyet "Tau UML" for de som ikke er kjent med verktøyet.
- 5) Utarbeidelse av Use Case Modell og domenemodell
- 6) Utarbeidelse av detaljert og fullstendig UML design for hele systemet.
- 7) Midtrapport: "Reverse engineering": Generering av UML klassesdiagram ut i fra delvis ferdig implementert kode. Oppdatering/forbedring av prosjektplan basert på erfaringene så langt og ny informasjon om gjenstående arbeid (delvis ferdig implementert kode). Rapporten skal også inneholde en kort oppsummering av hvordan dere gjennomførte inspeksjonen ("code review") for å forstå eksisterende kode.
- 8) Design av ny funksjonalitet (sekvensdiagrammer og klassesdiagrammer)
- 9) Generere kode i Tau UML og implementere ny funksjonalitet
- 10) Presentere "nytt" kjørbart system for oppdragsgiver (gruppelærer)
- 11) Evaluering av eget og en annen gruppes prosjektgjennomføring. Presentasjon av evalueringen for oppdragsgiver (på gruppeøvelsen).

Etter hver leveranse vil gruppelærerne gi tilbakemeldinger og bestått/ikke bestått. Tidsfrister for leveransene er beskrevet i kapittel 6 i dette dokumentet.

## **4.3 Støtte til prosjektarbeidet**

Prosjektene skal i størst mulig grad klare seg selv og bruke medstudenter. Dersom problemer ikke kan løses på denne måten er neste steg å ta kontakt med gruppelærer som er oppdragsgiver. Dersom det likevel er behov for oppklaringer eller støtte er ansvaret for dette hos:

<b>Verktøy</b>	<b>UK (Ummear Ahmad Khan)</b>
<b>Kravspesifikasjon/design</b>	<b>HG (Hans Gallis)</b>
<b>Prosjektplan/styring</b>	<b>MJ (Magne Jørgensen)</b>
<b>Annet</b>	<b>HG/MJ/DS (Dag Sjøberg)</b>

## 4.4 Prosjektgruppene

Prosjektgruppene vil bli delt inn, så langt det er mulig, på bakgrunn av kartleggingen fra første forelesning. Informasjon om hvem som er på hvilken prosjektgruppe vil bli lagt ut på hjemmesiden til kurset så snart inndelingen er ferdig.

# 5 Kravbeskrivelse

Ikke forvent at kravbeskrivelsen er 100% klar mhp hvordan kravene skal forstås. Mindre viktige avklaringer tar dere innad i prosjektet. Viktige avklaringer tas med oppdragsgiver (gruppelærer).

## 5.1 Fullstendig kravspesifikasjon

Finner dere på kurset hjemmeside (ProjectHospital2004.pdf).

# 6 Retningslinjer for leveransene

Tidsfrister for leveransene er beskrevet på kursets hjemmeside og i parentes ved forklaringen til hver leveranse i dette kapitlet.

## 6.1 Generelt

### 6.1.1 Tittelside

Alle dokumentleveranser SKAL ha en tittelside som angir:

- Nummer på din prosjektgruppe (+ evt. firmanavn)
- Identifikasjon av leveranse (f eks: Design – Leveranse II)
- En liste over medlemmene i gruppen/firmaet
- Angivelse av hvem som har bidratt med hva (f eks: Prosjektledelse: Magne, Kvalitetssikring: Lise)
- Dato for ferdigstillelse (+ ev. oppdatering) av dokument

### 6.1.2 Lagringsformat

Alle dokumenter SKAL legges på prosjektets hjemmeside på formatet html, postscript, pdf eller som ASCII-tekst. Ikke generér html fra Word.

### 6.1.3 Endringer

Det SKAL beskrives i leveransene hva som eventuelt er endret siden forrige leveranse (forrige iterasjon). Dette skal beskrives i begynnelsen av dokumentet eller i begynnelsen av det aktuelle kapittelet slik at oppdragsgiver lett kan se hva som er endret.

## 6.2 Leveranse I (innleveringsfrist: 24.09)

### 6.2.1 Prosjekthjemmeside

Alle prosjekter SKAL ha en prosjekthjemmeside, der alle relevante prosjektdokumenter legges på en oversiktlig måte. Prosjektdokumentene skal være lette å finne for gruppelærere og forelesere. Ikke bruk mye krefter på "fancy" layout og funksjonalitet. Prosjekthjemmesiden vil kun evalueres ut fra hvor lett det er å gjenfinne prosjektdokumenter.

Brukernavn og passord til prosjektgruppens hjemmeområde vil kunne ta noe tid å få generert (f eks vil prosjektgruppe 12 ha sitt område på ~i3120g12). Start derfor gjerne med å utvikle på et prosjektmedlems eget hjemmeområde. Bruk gjerne andre enkle teknologier istedenfor å lage hele web-siden fra scratch. Se for eksempel WIKI og Blog (se under "Relevante web-sider/dokumenter" på kursets hjemmeside for mer informasjon om disse teknologiene).

### 6.2.2 Prosjektplan

Prosjektgruppen bør tilstrebe å være realistisk i rollen som leverandør. Dette innebærer at gruppen opptrer rimelig koordinert overfor oppdragsgiver (gruppelærer). Vurder om det er hensiktsmessig å tildele roller innad i gruppen (f eks at en utpekes som prosjektleder, en som er ansvarlig for hjemmesiden, en med hovedansvar for design, test, konfigurasjonsstyring, kvalitetssikring osv.).

Det er ingen krav om at gruppen velger en prosjektleder/gruppeleder, men erfaringsmessig (erfaringer fra tidligere prosjektoppgaver i INF 3120 (IN 219) og andre universiteter) er dette svært nyttig for prosjektet. Regelmessige prosjektmøter er viktig både for læringen og for prosjektsuksessen.

Det kan også være lurt å la en eller to personer være såkalte kvalitetssikrere. Det vil si at en eller to personer er ansvarlig for å sikre at alle SKAL-krav (som er beskrevet i dette dokumentet) er oppfylt og at rapporten oppfyller de generelle kriteriene før man leverer til gruppelærer. På den måten kan man unngå at leveransen blir underkjent p.g.a. unødvendige slurvefeil.

Noen tips for å få størst mulig kvalitet i prosjektplan og i videre arbeid:

- Start på prosjektorganiseringen og planleggingen umiddelbart.

- Ha fokus på neste leveranse, men jobb hele tiden i parallell med å forberede andre aktiviteter. Det er f.eks viktig at dere tenker igjennom hva som trengs av kunnskap for å gjennomføre de ulike aktivitetene.
- Når flere jobber på samme dokument/program bør konfigurasjonsstyring innføres eller i det minste regler for oppdatering og format på dokumenter og programmer avtales (filstruktur og navngiving av filer).
- Forbered dere på at estimering av ressursforbruk er vanskelig og må oppdateres jevnlig som følge av ny innsikt. Ikke forvent svært nøyaktige estimer.

Kravene til hva prosjektplanen SKAL inneholde, samt forslag til struktur og estimeringsstøtte, er beskrevet i maler (se mal\_prosjektplan\_2004.rtf) og estimeringsark (se mal\_aktivitetsplan\_2004.xls) som finnes på INF3120-sidene. Disse malene SKAL brukes. Tips og råd fra Sommerville-boka bør også tas hensyn til.

**Viktig:** Husk at dere SKAL registrere timeverk dere har brukt på aktivitetene/leveransene. Bruk samme aktivitetsstruktur som i estimeringsarket for denne registreringen. Sluttrapporten (presentasjonen) fra prosjektet (leveranse V) vil IKKE bli godkjent dersom sammenligning av estimert og virkelig forbruk av arbeidsmengde per aktivitet mangler! Hver deltaker i prosjektet skal altså registrere timer underveis.

### 6.2.3 Use Case Modell

Use Case Modell-leveransen SKAL inneholde følgende:

- Vurdering av oppdragsgivers krav og tekniske konsekvenser som følger av disse.
- Use case modell med:
  - Use case diagram som viser aktører og use cases. Dette SKAL lages i Tau UML.
  - Beskrivelse av hver aktør i henhold til malen nedenfor.
  - Beskrivelse av hvert enkelt use case (minimum 5 stk.) i henhold til malen nedenfor. Disse beskrivelsene kan lages direkte i Tau UML, eller de kan lages i Word, Latex etc.
  - Hendelsesflyten i use casene skal beskrives som angitt nedenfor.
- Domenemodell
- Beskrivelse av ikke-funksjonelle krav knyttet til de enkelte use case eller til systemet som helhet
- Se også tabellen i figur 6.17 i Sommerville for en oversikt over hva en kravspesifikasjon bør inneholde.

Mal for beskrivelse av aktør:

<b>Aktør</b> < navn >	
<b>Beskrivelse</b> < En kort tekst som beskriver aktøren >	
<b>Eksempler</b>	



Mal for beskrivelse av use case:

<b>Use Case</b> < navn >	
<b>Aktør</b> < navn >	
<b>Trigger</b> < Hendelse som starter use caset >	
<b>Pre-betingelser</b> < Betingelser som skal være oppfylt for at use caset skal kunne utføres >	
<b>Post-betingelser</b> < Betingelser som er oppfylt når use caset avsluttes >	
<b>Normal hendelsesflyt</b> < En liste av handlinger som utføres i en normal gjennomgang av use caset >	
<b>Variasjoner</b> < En liste med beskrivelser av mulige variasjoner til den normale flyten i use caset >	
<b>Relatert informasjon</b>	

Beskrivelse av hendelsesflyt:

- Hendeseflyten skal beskrives som en liste av hendelser på formen < nr >< hendelsesbeskrivelse >.
- Hendelsesbeskrivelsene skal være på formen < agent >< verb >< direkte objekt >< preposisjonssetning >
- Hver hendelsesbeskrivelse skal være en eller flere av de følgende:
  - En aktør sender en forespørsel eller data til systemet
  - Systemet validerer inkomne data
  - Systemet endrer sin interne tilstand
  - Systemet svarer til aktøren med resultatet

Fokuser på at dere har nok informasjon til å lage et godt design og senere for å teste at systemet tilfredsstillende funksjonelle kravene.

Kravspesifikasjonen (Use Case modellen m.m.) SKAL kvalitetssikres. Følgende sjekklister støtter denne kvalitetssikringen:

1. Aktører:

Se på aktørene i use case modellen.

- Er alle aktørene som deltar i use casene klart definert?
- Er det klart hvilke aktører som er involvert i hvilke use case?
- Er beskrivelsen av hvordan systemet interagerer med en aktør konsistent med beskrivelsen av aktøren?

- Er forventet input og output definert for hver aktør?
2. Use case modellen som helhet:  
Se på use case diagrammet, de tekstlige beskrivelsene av use casene og domenemodellen.
    - Er det samsvar mellom use cases og domenemodell, dvs. gjenfinner dere konseptene som beskrives i use case modellen i domenemodellen og vice versa?
  3. Hvert enkelt use case:  
Gå gjennom hver enkelt use case beskrivelse.
    - Er beskrivelsene av hvert use case slik at det er forståelig hvilke aktiviteter som er nødvendig for at aktøren skal nå sitt mål?
    - For hvert use case, er systemets output på alle aktørens input for både gyldig og ugyldig hendelsesflyt definert?
    - Er det ikke-verifiserbare ord eller setninger i use casene, som f.eks. "fungerer bra", "rask", "god ytelse", "skjer vanligvis"?
    - Er hendelsesflyten i use casene beskrevet i konkrete termer og med målbare kvantiteter?
    - Er alle pre-betingelser og post-betingelser beskrevet på en slik måte at de kan testes?
  4. Sammenhengen mellom use casene:  
Se på hvordan use casene tilsammen beskriver ønsket funksjonalitet.
    - Brukes include-relasjonen til å faktore ut felles oppførsel?
    - Matcher pre- og post-betingelsene for de enkelte use casene?

## 6.3 Leveranse II (innleveringsfrist: 01.10)

### 6.3.1 UML Design v.h.a. klassediagram

Designmodellen skal bestå av klassediagrammer (og ikke sekvensdiagrammer som vil bli påkrevd i leveranse IV). Designmodellen skal dekke hele systemet.

For klassediagrammene gjelder følgende:

- Alle attributter og metoder skal ha en type (public, private, protected).
- Alle assosiasjoner skal ha angivelse av roller, multiplisitet og navigerbarhet (retning).

Ved behov vil det være hensiktsmessig å beskrive designet tekstlig (i selve UML-modellen eller som et vedlegg til selve modellen). Beskrivelsen bør være et svar på "hvorfors er det hensiktsmessig å gjøre det slik?".

Prosjektene vil få tilbakemelding fra gruppelærere (og evt. foreleser) på styrker og svakheter ved deres design.

## **6.4 Leveranse III (innleveringsfrist: 15.10)**

### **6.4.1 "Reverse engineering"**

Det skal genereres et UML klassediagram ut ifra den ferdig implementerte koden (som dere får utlevert). Prosessen med å gjennomføre "reverse engineering" (genereringen av UML klassediagram) vil bli beskrevet steg-for-steg i et eget notat som legges på kursets hjemmeside. Klassediagrammet vil forhåpentligvis også kunne gi dere bedre innsikt i den implementerte delen av systemet.

### **6.4.2 Oppdatert prosjektplan (midtrapport)**

I denne leveransen skal dere stoppe opp i en kort periode for å se om prosjektet styres i riktig retning til riktig tid.

Prosjektplanen (fra leveranse I) SKAL oppdateres ihht til ny kunnskap (delvis ferdig implementert system som dere får utdelt) og økt forståelse (erfaringer fra prosjektarbeidet som allerede er gjennomført). Hele prosjektplanen SKAL evalueres i fellesskap innad i gruppa. De delene av prosjektplanen som krever tiltak skal endres og endringene SKAL merkes slik at gruppelærer lett kan se hva som er endret fra leveranse I. Prosjektgruppa bestemmer selv hva som krever oppdatering (hva har fungert dårlig og hvorfor, og hva har fungert bra og hvorfor?), men følgende SKAL evalueres og oppdateres:

- Aktivitetsplan med tilhørende oppgaver og estimater.
- Risikoplan (topp 10 risikoliste)

Hvordan man gjennomfører denne evalueringsprosessen er opp til prosjektgruppa å bestemme. Et tips er å benytte Post Mortem Analyse (PMA). Dokumentasjon angående PMA vil legges ut på kursets hjemmeside. Hvordan man setter seg inn i den eksisterende (ferdig implementerte) koden er også opp til prosjektgruppa å bestemme. Et par forslag kan likevel nevnes: For eksempel kan dere lese koden enkeltvis og deretter ta en felles diskusjon for å oppklare uklarheter, eller dere kan diskutere koden parvis.

## **6.5 Leveranse IV (innleveringsfrist: 19.11)**

### **6.5.1 Design av ny funksjonalitet**

Dere skal benytte klassediagrammet fra "reverse engineering" som utgangspunkt (og ikke klassediagrammet over hele systemet fra leveranse II). Dette klassediagrammet skal utvides slik at det dekker ny funksjonalitet (dere får vite hva slags funksjonalitet det er snakk om når dere får utdelt den implementerte koden i leveranse III).

For sekvensdiagrammene gjelder følgende:

- Et sekvensdiagram skal modellere hendelsesflyten i et og bare et use case.
- Alle use case som er relatert til den nye funksjonaliteten skal være modellert v.h.a. sekvensdiagrammer.

Og for klassediagrammene gjelder følgende (samme retningslinjer (SKAL-krav) som for leveranse II):

- Alle attributter og metoder skal ha en type (public, private, protected).
- Alle assosiasjoner skal ha angivelse av roller, multiplisitet og navigerbarhet (retning).

**NB:** Det SKAL (se del-leveranse nedenfor) genereres Java-kode vha Tau UML på bakgrunn av designet (se kapittel 6.5.2). Ta hensyn til dette!

## 6.5.2 Kodegenerering og implementering av ny funksjonalitet

På bakgrunn av den ”nye” designmodellen (UML klassediagram) skal dere generere kode. Den genererte koden vil inneholde alt som tidligere har blitt implementert (fullt implementerte klasser og metoder) samt et kodeskjelett (”tomme” klasser og metoder) for den nye funksjonaliteten.

Dere vil få en steg-for-steg beskrivelse av hvordan kodegenereringen skal foregå. Det kan være hensiktsmessig å finne ut hva Tau UML ”krever” av klassediagrammet for å generere kode før dere designer den nye funksjonaliteten (hva som MÅ være med av relasjoner mellom klasser, attributter, parametere etc. for at dere skal få generert kode).

Implementeringen SKAL følge SUN’s Java kodestandard (se egen link under ”Annet” på kursets hjemmeside). Dere bestemmer selv hvordan dere skal teste systemet (det nye implementerte systemet).

Dere trenger ikke å levere koden skriftlig. I dokumentet til denne leveransen skal dere beskrive hvor filene til det ”nye” kjørbare systemet ligger slik at oppdragsgiver (gruppelærer) kan teste det og se over koden. Koden skal kunne lastes ned fra hjemmesiden til prosjektgruppa.

Det anbefales å benytte et konfigurasjonsstyringsverktøy til implementeringen (for eksempel cvs eller svn).

### 6.5.3 Oppdatert klassediagram

Dokumentet fra leveranse IV SKAL inneholde de opprinnelige UML modellene dere laget i begynnelsen av leveranse IV og oppdaterte UML modeller som beskriver det faktiske implementerte systemet. Om dere gjør denne oppdateringen manuelt (endrer UML modellene fortløpende) eller lar Tau UML ta seg av det ("reverse engineering" til slutt) er opp til prosjektgruppa. Gruppelærerne vil også kreve å se den opprinnelige designmodellen man benytter til kodegenereringen før man begynner selve implementeringen. Hvis det er betydelige endringer fra det opprinnelige klassediagrammet og det faktiske implementerte systemet SKAL årsakene til dette beskrives kort (hvorfor ble det slik?).

## 6.6 Leveranse V (innleveringsfrist: Prosjektet presenteres i perioden 22.11 - 26.11)

### 6.6.1 Presentasjon av prosjektet

Prosjektgruppen skal avholde en presentasjon for oppdragsgiver (gruppelærer), en av faglærerne og resten av øvingsgruppa, der resultater fra prosjektgjennomføringen, designet, kodegenereringen og implementeringen av den nye funksjonaliteten inngår. Oppdragsgiver vil, sammen med faglærer, gi bestått/ikke bestått for dette på bakgrunn av om hun/han tror at dette vil kunne føre til et tilfredsstillende system. Det kan derfor være nyttig å gå noen runder med oppdragsgiver i forkant av innleveringsfristen for å avdekke om det er mangler (sett fra oppdragsgiver) som må rettes opp.

Det er opp til hver prosjektgruppe hvordan man organiserer og gjennomfører presentasjonen, samt hva man legger vekt på, men presentasjonen SKAL inneholde følgende:

- En kort demonstrasjon av systemet hvis mulig
- En sammenligning av virkelig og estimert ressursforbruk per aktivitet. Estimert ressursforbruk tas fra regnearket/prosjektplanen og inkluderer også minimum og maksimumsestimater.
- Erfaringer fra prosjektgjennomføringen (dette kan gjøres ved hjelp av bl.a. Post-Mortem Analyse – PMA). Mer detaljert informasjon om denne teknikken vil legges ut på kursets hjemmeside. Erfaringene skal også diskuteres opp mot pensum (trekke inn relevante aspekter).
- Evalueringer av kvaliteten til leveransene og mulighetene til å implementere et godt system basert på det som er gjort så langt i utviklingen av systemet. Her skal dere ta utgangspunkt i deres eget prosjekt og sammenlikne med en annen prosjektgruppes gjennomføring. Det vil si at dere skal evaluere deres eget og se det opp mot en av de andre prosjektgruppene prosjekt (basert på dokumentasjonen som finnes).

- Ris, ros og forbedringsforslag til gjennomføringen av obligatorisk prosjekt i INF3120/INF4120.

Presentasjonen skal være på **maksimalt 20 minutter**.

**Tips:** Diskuter erfaringene med å jobbe i gruppe, relasjonen til oppdragsgiver og hva prosjektet burde gjort annerledes. Alle i gruppen bør delta aktivt i dette arbeidet, da dette er svært eksamensrelevant.

I denne prosjektoppgaven har det ikke vært lagt vekt på følgende (som normalt ville vært viktige aspekter i et systemutviklingsprosjekt):

- Realistisk kunde–leverandør forhold
- User interface design (usability engineering)
- Testing
- Utrulling av prosjektet (systemet) i bedrift (installering, brukeropplæring og lignende)

Disse aspektene og flere andre kan også trekkes inn i sluttrapporten hvis det er hensiktsmessig.