

# UNIVERSITETET I OSLO

## Det matematisk-naturvitenskapelige fakultet

Eksamen i :	IN219 — Store programsystemer
Eksamensdag :	Torsdag 13. desember 2001
Tid for eksamen :	09.00 – 13.00
Oppgavesettet er på :	3 sider
Vedlegg :	Ingen
Tillatte hjelpemidler :	Alle trykte og skrevne

*Oppgave 1 vektlegges 50%.*

*Oppgave 2 vektlegges 50%.*

*Kontroller at oppgavesettet er komplett før du begynner å besvare det.*

*Det anbefales å lese gjennom hele oppgavesettet før du begynner med besvarelsen.*

*Legg vekt på god struktur og relevans til det du skriver. Les oppgavene nøye!*

*Innfør og begrunn dine egne antakelser der du finner det nødvendig.*

*Legg mer vekt på å illustrere prinsipper fremfor å presentere store mengder detaljer (gjelder særlig oppgave 1).*

### Oppgave 1 (50 %)

Et meglerfirma ønsker et Web-basert system for å vise alle eiendommene som er til salgs og for å håndtere budrunder på eiendommene. I dag foregår budrundene per fax og telefon. Systemet skal ha funksjonalitet for å opprette en hjemmeside med beskrivelser av alle eiendommene som er til salgs. Hver eiendom skal registreres med adresse, type bolig (enebolig, tomannsbolig, rekkehus eller leilighet), antall kvadratmeter, antall rom, verditakst, lånetakst og prisantydning samt en tekst som beskriver eiendommen i detalj. Meglerfirmaet må kunne legge inn beskrivelser av nye eiendommer etter hvert som oppdragene kommer inn, og fjerne beskrivelsene etter at eiendommene er solgt. Videre skal det være mulig for meglerfirmaet å endre opplysninger om eiendommene.

Personer som ønsker å gi bud må registrere seg i systemet med personinformasjon og en finansieringsplan som viser egenkapital, lån som skal innfris, salg ev eksisterende bolig etc. Personinformasjonen utgjøres av navn, adresse, telefonnummer, epost-adresse og fødselsnummer. Når budgiver er registrert, vil han eller hun få et brukernavn og et passord. En registrert budgiver skal kunne legge inn bud på en gitt eiendom. Et bud består av et pengebeløp og et tidspunkt som angir når budet utløper. En budgiver som har gitt bud på en eiendom skal kunne se informasjon (anonymisert) om andre bud som er kommet inn på samme eiendom. En budgiver skal bare kunne by på én eiendom av gangen. Når en budgiver har bud inne på en bolig, skal vedkommende få epost hver gang

et nytt bud på boligen kommer inn. En registrert budgiver skal også kunne endre sin personinformasjon og finansieringsplan.

- Oppg. 1A** Lag en domenemodell og et use case diagram for systemet. Use case diagrammet skal vise hvilke aktører som skal kommunisere med systemet og hvilken funksjonalitet systemet skal tilby.
- Oppg. 1B** Lag detaljerte use case beskrivelser av use casene fra oppgave 1A som har med registrering og håndtering av budgiver og bud på eiendommer å gjøre.
- Oppg. 1C** Lag et komplett klassediagram ut fra domenemodellen fra oppgave 1A.
- Oppg. 1D** Lag sekvensdiagrammer som viser hvilke klasser og metoder som er nødvendige for å realisere *to* (velg selv hvilke) av use casene som du laget detaljerte beskrivelser av i oppgave 1B.

## Oppgave 2 (50 %)

### Oppg. 2A PROSESSMODELL

Angi situasjoner der en inkrementell prosessmodell (det vil si den dere brukte i den obligatoriske prosjektoppgaven) sannsynligvis er mer hensiktsmessig enn en fossefallsmodell. Trekk inn egne erfaringer fra arbeidet med den obligatoriske prosjektoppgaven der det er relevant.

#### Situasjoner/momenter som bør være med:

Inkrementell prosessmodell er sannsynligvis mer hensiktsmessig når:

- det er ekstra viktig å få levert kjernefunksjonalitet (funksjonalitet med høy prioritet) tidlig (f eks for at kunden skal kunne teste eller at man kommer tidlig på markedet med et produkt)
- det er viktig å få erfaringer med deler av teknologien (teknologisk risiko er høy, kunnskap er lav, opplæring trengs) så tidlig som mulig (f eks for å få redusert risikoen for resten av inkrementene)
- når oppdragsgivers krav er uklare og vil kunne endre seg underveis (inkrementell prosess muliggjør at de deler som er mest stabile utvikles først, og muliggjør økt forståelse for resten av leveransen)
- det er viktig å redusere risiko, f eks når et nytt verktøy introduseres eller det er usikkert om kunden vet hva han vil ha. Tidlige inkremitter gjør at ev. problemer (f eks med test-plattform) blir oppdaget tidligere og at planlegging for resten av utviklingen er basert på mer kunnskap.
- (utvidbarhet er viktig – dette momentet kan diskuteres. Er det slik at inkrementell utvikling fører til mer utvidbare systemer. Det er argumenter for og mot.)
- (leveransen er inkrementell/oppdelbar)

(Jeg har stort sett gjort det slik at dersom kun ett av momentene er med får kandidaten 3,0-4.0 avhengig av hvor viktig dette momentet er. Dersom to-fire er med får kandidaten 2.0-3.0 og dersom alle/mer enn fire er med og ”modent” beskrevet gir jeg 1.5.

Sjekk at studenten har skjønnet forskjell på evolusjonær og inkrementell prosessmodell, dvs at det ikke er en ren evolusjonær prosessmodell uten del-leveranser som beskrives. Studentene er ikke bedt om å beskrive de ulike modellene, mao det gis ikke ”ekstrapoeng” for disse beskrivelsene (men det kan være momenter skjult i denne teksten som skal telle positivt).

Det bør trekkes noe (men ikke så mye) for at beskrivelsene ikke er rettet mot oppgaveformuleringen (”angi situasjoner”), mao, vi bør være litt ”snille” her. Mange beskriver fordeler og ulemper som kun indirekte beskriver situasjoner.

Mange har ikke fått med seg den generelle strukturen på inkrementell utvikling, men tenker kun på varianter som den de brukte i oblig. Det er vanskelig å trekke mye for dette, men for å få en svært god besvarelse bør denne modenheten være tilstede. (særlig siden oppgaveformuleringen kanskje indikerer at vi spør om ”den modellen de brukte i oblig).

Les Sommerville’s momenter (som jeg har forsøkt å nyansere noe i forelesningen, f eks mhp Sommerville’s påstand om at store prosjekter ikke bør bruke inkrementell utvikling!?!). Vi kan vanskelig trekke for at de gjengir pensum her, selv om noen ikke helt har fått med seg stoffet fra forelesningene.)

## **Oppg. 2B KODEGENERERING**

Angi fordeler og ulemper med kodegenerering fra design mot det å skrive koden selv. Trekk inn egne erfaringer fra arbeidet med den obligatoriske prosjektoppgaven der det er relevant

Fordeler:

- Koden vil automatisk bli designet i samsvar med designmodellen, dvs. tettere integrasjon mellom designmodell og kode (sporbarhet)
- Endringer i struktur vil kunne utføres på designnivå (før ny kodegenerering)
- Sparer tid på koding
- Redusere sannsynligheten for feil, avanserte verktøy vil også kunne sjekke for en del typer feil i designet før kodegenereringen utføres
- Koden vil følge standarder/konvensjoner/prinsipper - bestemt av verktøyet, evt. spesifisert av bruker
- Dokumentasjon på designnivå som også vil kunne være dokumentasjon av koden (for eksempel metoder og attributter)
- Enklere å gjenbruke (deler av) design i ulike systemer
- Vil kunne bidra til en mer standardisert utviklingsprosess

Ulemper:

- o Feil i designet vil forplante seg (og evt. forstørres), dvs. store krav til godt og korrekt design
- o Koden vil kunne følge uønskede standarder/konvensjoner/prinsipper
- o Koden vil kunne bli uhensiktsmessig omfattende/uoversiktlig
- o Koden vil kunne bli ineffektiv
- o Koden vil ofte måtte gjennomgå manuelt for tilpasninger – kan være vanskeligere enn om all kode var designet fra scratch av bruker med oversikt over spesielle behov
- o Kan være tungvint å håndtere manuelt skrevet spesialkode når endringer er skjedd i designet og koden må re-genereres
- o Noen verktøy vil kreve proprietære klassebiblioteker, dvs. gjør den genererte koden mindre portabel
- o Som ved mange andre verktøy: opplæringskostnader (både verktøy og prosess), kan være dyrt i innkjøp/lisensleie

### Oppg. 2C AKTIVITETSDIAGRAM

- I. Lag et aktivitetsdiagram (activity network) basert på listen av aktiviteter, varigheter og avhengigheter i figur 1 s. 3.
- II. Angi aktivitetene i aktivitetsdiagrammet som ligger på ”kritisk sti”.
- III. Hvor egnet er aktivitetsdiagrammer for planlegging av prosjekter som følger *evolusjonær* prosessmodell?

- I) Trekk ca. 0,5 i karakteren dersom i stedet Gantt-diagram er beskrevet.
- II) Alle aktiviteter untatt A5 og A9
- III) Denne oppgaven er ikke helt lett og det er lite av pensum som går direkte på dette. Sjekk at studenten i det hele tatt besvarer spørsmålet og ikke skriver om noe helt annet (f eks om evolusjonære prosessmodeller). Hovedmomentet er at i evolusjonær prosessmodell er styrken at man gjør flere iterasjoner innen hver fase og på hver aktivitet. For at alle prosjektaktivitetene skal kunne overføres til et aktivitetsdiagram med avhengigheter mellom aktivitetene må dermed hver ”logiske” aktivitet deles opp i flere del-aktiviteter – en per iterasjon. Dette blir lett et nokså komplisert aktivitetsdiagram, særlig dersom man ved planleggingens start ikke engang vet hvor mange iterasjoner som skal gjennomføres.

Dette henger for øvrig sammen med at man gjerne velger evolusjonær prosessmodell når det er uklart hva man skal lage. Når det er uklart hva man skal lage er det vanskelig å bruke en notasjon som antar at du vet alle aktiviteter og avhengighetene mellom disse.

Andre momenter: Avhengighetene mellom aktiviteter i evolusjonær modell er vanskelig å beskrive. Ofte vil det være slik at man gjør alt litt om hverandre. Det kan for øvrig være store variasjoner i hvordan evolusjonær modell brukes. I en ”strukturert” (der f eks hver iterasjon følger fossefallsmodellen) bruk av

evolusjonær modell vil f eks aktivitetsdiagrammer kunne være hensiktsmessig for planlegging av neste iterasjon.