

# UNIVERSITETET I OSLO

## Det matematisk-naturvitenskapelige fakultet

Eksamen i :	IN219 — Store programsystemer
Eksamensdag :	Torsdag 14. desember 2000
Tid for eksamen :	09.00 – 13.00
Oppgavesettet er på :	2 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.*

*Les gjerne gjennom hele settet før du begynner med besvarelsen.*

*Forsøk å legge vekt på god struktur i besvarelsen.*

### **Oppgave 1 (50 %)**

*Som student på universitetet må du registrere deg til de kurs du vil ta (emnepåmelding). Du må angi kurs og hvilke øvingsgrupper du vil følge. Visse kurs krever bestått eksamen i andre kurs. For å kunne ta eksamen er det også et krav om godkjent obligatorisk oppgave (fra samme semester som eksamen eller fra tidligere semestre). Et annet krav for å kunne ta eksamen er at du har gyldig studiekort og har betalt semesteravgift. Før eksamen får du melding om når og hvor eksamen skal foregå. Etter at sensuren er falt, kan du ringe universitetet for å få opplyst ditt resultat.*

*I denne oppgaven skal du anta at du er ansatt i et konsulentfirma som står for utviklingen av et elektronisk kursadministrasjonssystem kalt KURSADM. KURSADM skal støtte universitetets arbeid med å gjennomføre og sikre kvalitet i de aktivitetene som er beskrevet ovenfor.*

*I alle underoppgavene nedenfor skal du selv gjøre ekstra antakelser dersom du finner det nødvendig.*

**Oppg. 1A** *Lag en UML use case-modell (både use case-diagram og use case-beskrivelse) for KURSADM. Begrunn kort ditt valg av aktører.*

#### **Aktører**

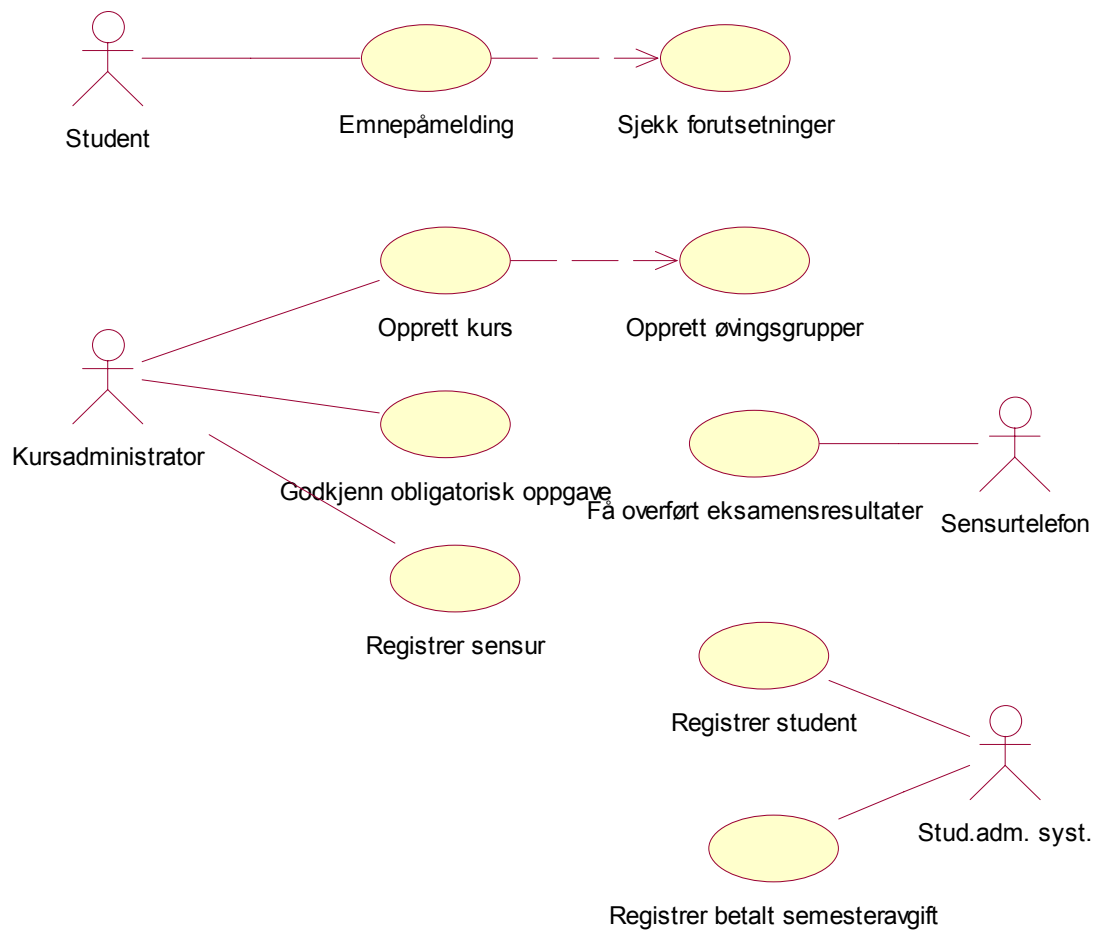
Student - Registrerer seg til kurs ved å angi kurs og øvingsgruppe.

Kursadministrasjon - Legger inn hvilke kurs og øvingsgrupper som tilbys samt godkjente obliger og eksamensresultater for hver student.

Student administrativt system - Eksternt system som holder oversikt over studenter med gyldig studiekort og betalt semesteravgift.

Sensurtelefon - Eksternt system som eksamensresultater overføres til.

### Use case diagram:



**Tekstlige beskrivelser av use casene**

En optimal løsning bør ha en kortfattet beskrivelse (ett par linjer) av hvert use case samt detaljert beskrivelse av ett par av use casene.

Bare en kort beskrivelse av hvert use case er akseptabelt når diagrammet inneholder svært mange use case.

*Til dem som spurte opplyste Dag på eksamen at de ikke trengte å beskrive alle use casene hvis tiden ikke strakk til; det var bedre å beskrive noen færre, men heller grundig.*

Eksempel på kortfattet beskrivelse:

Use case Opprett kurs

Kursadministrator registrerer nødvendig informasjon om et nytt kurs. Informasjonen vil omfatte navn på kurset, emnekode, når og hvor det holdes samt navn på forelesere. Hvis kurset inneholder øvingsgrupper så inkluderes et use case for å opprette disse.

Eksempel på detaljert beskrivelse:

Use case EmnepåmeldingAktør Student

Trigger En student ønsker å melde seg på ett eller flere kurs

Precondition Studenten må være registrert i systemet

Normal hendelsesflyt

1. Studenten logger seg inn med navn og passord
2. Systemet sjekker at studenten har betalt semesteravgift.
3. Systemet henter frem registrerings skjema for dette semesteret
4. Studenten fyller inn informasjon om kurs, øvingsgrupper samt adresseinformasjon
5. Systemet sjekker at studenten er kvalifisert til å ta de valgte kursene v.h.a. use case "Sjekk forutsetninger".
6. Studenten bekrefter informasjonen og avslutter
7. Systemet lagrer informasjonen

Alternativ hendelsesflyt

1a. Studenten bruker navn og/eller passord som ikke er kjent av systemet.

1a.1. Systemet ber studenten om å forsøke igjen. Maks. 3 forsøk.

2a. Systemet oppdager at studenten ikke har betalt semesteravgift.

2.a.1. Systemet gir melding til studenten

2.a.2. Use case avsluttes

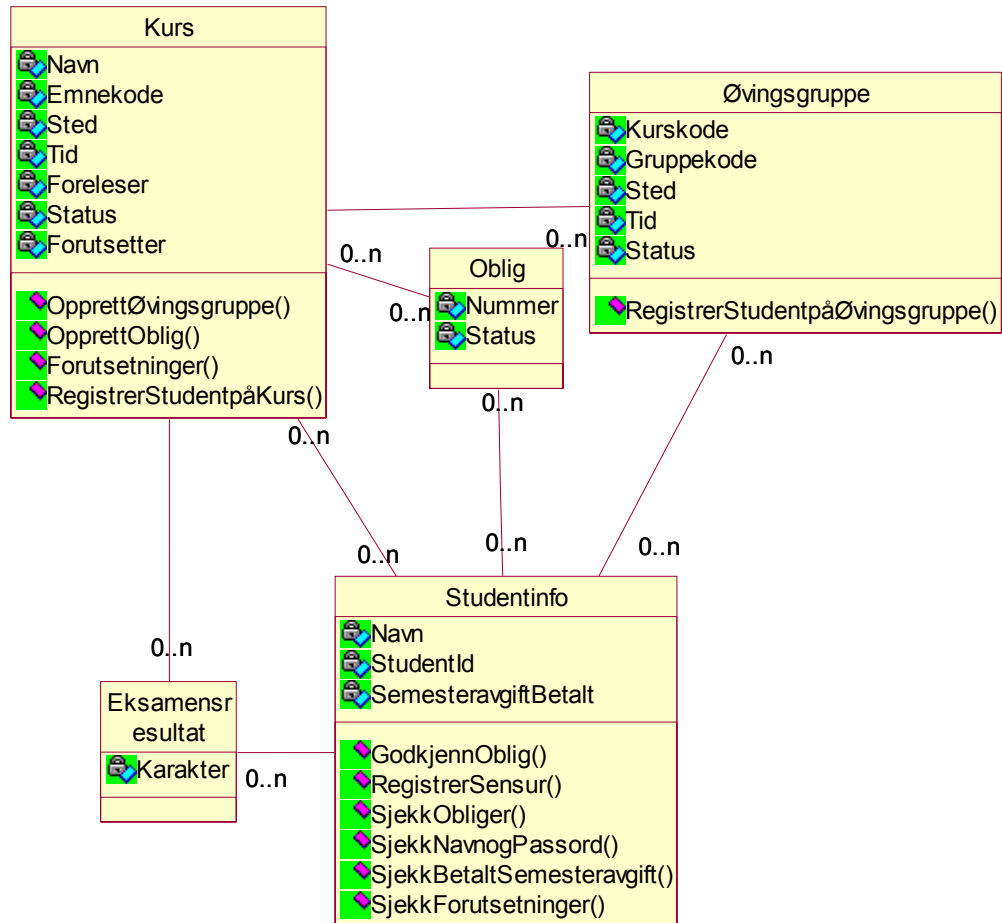
5a. Systemet oppdager at studenten ikke er kvalifisert til å ta et av de valgte kursene

5.a.1. Systemet gir melding til studenten

Postcondition

Systemet er oppdatert med de kursene som studenten skal ta inneværende semester.

**Oppg. 1B** Lag et objektorientert design for KURSADM der du anvender UML klasse-diagrammer. (Du må selv vurdere omfanget av ditt design i forhold til tiden du har til rådighet.)

**Klassediagram**

Forslaget ovenfor 1 B er noe "slapp i fisken". Der bør vel klassene for kurs, gruppe og obligatorisk oppgave splittes i to, en for beskrivelser av "fenomenet" og en for instanser knyttet til påmeldingen: F.eks. en for kursdefinisjon (eksempel på instans: IN219 i høstsemesteret 2000) og en for kurspåmelding for en gitt student. Resultatet blir da at noen av relasjonene <0:n, 0:n> forsvinner og de andre endres til <1, 0:n>. I en "normalisert" design vil det da være nok med én relasjon ut fra studentinfo, nemlig til kurspåmeldingsinstansen.

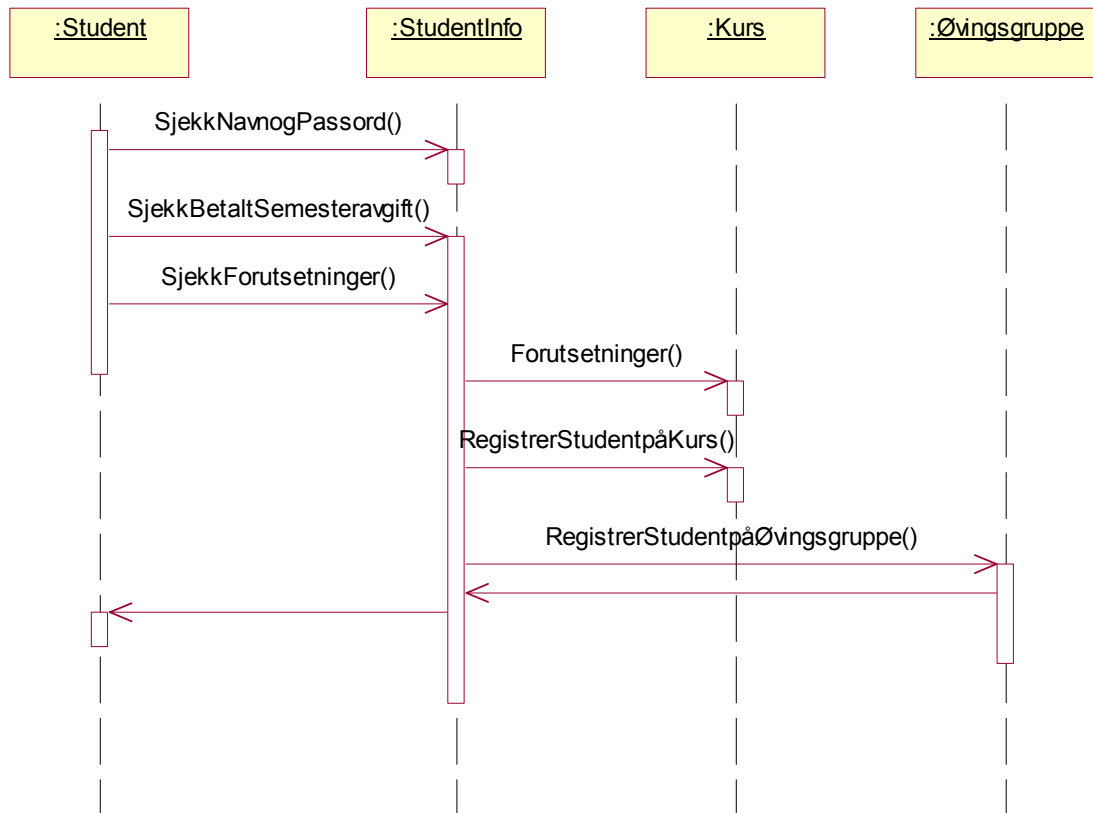
Det er heller ikke lett å se hvordan mønsterløsningen håndterer regler knyttet til bruk av godkjente øvelser fra tidligere kurs.

**Oppg. 1C** Lag et sekvensdiagram over operasjoner som vil inngå i to av use case-diagrammene beskrevet i oppgave 1A.

### Sekvensdiagram for use case "Emnepåmelding"

Studentene har lært å angi hvilken aktør det er som initierer interaksjonen. *Student* vil derfor bli vist som en aktør i sekvensdiagrammene i besvarelsene fordi det er slik det gjøres i Tau UML. I Rational Rose (som er benyttet i dette løsningsforslaget), vises bare objekter i sekvensdiagrammet.

Merk at studentene ikke har lært om syntaks og semantikk for piler i sekvensdiagrammer. Husk at de er bedt om å lage sekvensdiagram for *to* use cases.



**Oppg. 1D** Beskriv noen metoder/teknikker for identifisering og/eller spesifisering av kravene til et programvaresystem. Gjør rede for deres svake og sterke sider.

**Noen teknikker:**

**Fra foil av Ray Welland (dataflyt og datamodellering er lite vektlagt i IN219):**

De skal *både* beskrive teknikkene *og* fordeler/ulemper

- *Data flow modelling (describing an existing system)*

Lite vektlagt i kurset

- *Semantic Data Modelling (describing complex data structures; databases)*

Lite vektlagt i kurset

- *Object modelling, class diagrams and use cases - Unified Modeling Language (UML)*

*Mye vektlagt.*

*Fordeler:*

Enkelt å forstå (?) pga. intuitive diagrammer og naturlig språk

Enkelt å lære (??)

*Ulemper:*

Flertydig - ulike tolkninger hos ulike aktører (stakeholders)

“Verbos” notasjon

- *Viewpoint-oriented methods*

*Vord er den metoden som omtales her*

*Fordeler:*

(Fra Wellands foiler:)

- good for interactive systems (services)
- external viewpoints are natural, easy to identify
- valid viewpoints must have an interaction with the system (can eliminate invalid viewpoints)
- services have associated non-functional requirements.

- *Formal specification*

*Z er forelest her.*

*Fordeler:*

- Automatisk feilfinning, reduserer risikoen for feil, inkonsistens etc.
- Konsis og presis notasjon

*Ulemper:*

(Fra Wellands foiler:)

- The effort involved (mostly by hand) and skills required
- Lack of tool support, although some are becoming available
- Lack of necessary background and poor training of existing staff, together with the use of unfamiliar notations

- Lack of knowledge among project managers
  - Validation problems
  - hard to communicate ideas to users - might build perfect, but invalid, system
  - again, tools required - animation, alternative representations
  - Problems of scale
  - formal specification techniques not suited for very large projects - lack of modularity, information hiding in some traditional f.s. techniques
- *Prototyping (customer interaction, validation)*

Lite vektlagt i kurset

## **Oppgave 2 (50 %)**

*I denne oppgaven skal du kombinere teori fra pensum og praktiske erfaringer fra den obligatoriske prosjektoppgaven. Oppgaven skal kunne besvares også dersom du gjennomførte prosjektoppgaven i 1999 eller tidligere. Angi år for gjennomføring av obligatorisk prosjektoppgave i begynnelsen på besvarelsen!*

**Oppg. 2A** *Angi hvilken prosessmodell (systemutviklingsmodell) din prosjektgruppe anvendte i det obligatoriske prosjektet og gjør rede for hovedprinsippene i denne prosessmodellen.*

I 2000 var det angitt at studentene skulle bruke en inkrementell utviklingsmodell. I 1999 og 1998 var prosessen nærmere en vannfallsmodell, men det kan hende at noen jobbet mer evolusjonært. Angitt prosessmodell må sjekkes opp mot resten av besvarelsene i oppgaven mhp konsistens. Hovedprinsippene i prosessmodellen beskrives gjennom fasers sekvens og iterasjoner, samt hvordan leveranser er delt opp. Stoff om dette står i foilene om prosjektplanlegging, litt spredt i Sommerville-boka kap. 3 + at det er gitt en beskrivelse av inkrementell prosessmodell i oblig-retningslinjene – se in219-hjemmesidene. Sommerville definerer prosessmodell som: "Abstract representation of a software process (from a particular viewpoint)" og bruker en tradisjonell inndeling i vannfalls, evolusjonær, ..... modell.

For å gjennomføre en svært god besvarelse bør studentene i tillegg ha tenkt på mer grunnleggende prinsipper rundt prosessmodellen, f eks anvendt noe av stoffet fra gruppedynamikken:

Alle systemutviklingsmodeller gir strukturer for rekkefølge på aktiviteter. Modeller er ment å sikre at en hensiktsmessig rekkefølge på systemutviklingen gjennomføres.

Systemutviklingsprosessen blir mer oversiktlig gjennom oppdeling i delproblemer (faser, aktiviteter, prosesssteg), mao støtte av en nokså universell problemløsningsstrategi.

Noen systemutviklingsmodeller (inkrementelle) gir strukturer for leveranseoppdeling.

Dette er gunstig for problemløsningen dersom delproblemene har svært ulik kompleksitet eller at leveransetidspunktet bør være ulike. Mao, matcher en naturlig problemløsningsstrategi for visse type problemer.

Noen systemutviklingsmodeller gir strukturer for problemforståelse gjennom utprøving (f eks evolusjonær systemutvikling). Dette er naturlig problemløsningsstrategi for

problemer der forståelsen mhp når problemet er løst (behovet er dekket) må antas å være ufullstendig.

**NB:** Vi bør se an litt hva studentene har vektlagt før vi sier noe endelig om hvor mye vi skal belønne/trekke for at slike mer grunnleggende prinsipper er tatt med/ikke tatt med.

**Oppg. 2B** *Skisser kort de viktigste sammenhengene mellom prosessmodellen og prosjektplanen i ditt prosjekt, dvs hvordan prosjektplanen konkretiserte hovedprinsippene i prosessmodellen beskrevet i oppgave 2A.*

Her forventes det at studentene skal, relatert til **sitt prosjekt**, skisserer og konkretiserer sammenhenger relatert til at (tatt fra undervisningsfoilene):

Faser i prosjektmodellen blir til ”overordnede aktiviteter og milepæler” i prosjektplan (men prosjektplan vil typisk inneholde flere og mer detaljerte aktiviteter).

Relasjon mellom faser i prosjektmodellen blir til Gantt-diagram eller lignende i prosjektplan.

Innhold i faser i prosjektmodellen blir til leveranser, dokumentasjon, verifikasjon og validering, risikostyringsaktiviteter m.m. i prosjektplan.

Prosjektplanen er prosjektleders virkemiddel for å sikre leveransene under rammebetingelsene som settes av prosessmodellen. Konflikter oppstår dersom disse to målene ikke samsvarer (ref. ”skunk projects” og ”byråkrati”).

En god besvarelse vil være kjennetegnet ved at studenten både evner å konkretisere viktige sammenhenger og å se dette i en større sammenheng. Spesielt viktig blir det (for de som har brukt inkrementell metode – dvs flesteparten) å angi hvordan inkrementene ble implementert i prosjektplanen (“læreinkrement” – som de ble anbefalt å ha, se oblig-retningslinjer, “prioritert inkrement” – som de skulle ha, og “rest-inkrement” - som vel de færreste rakk).

**Oppg. 2C** *Vurder fordeler og ulemper med den prosessmodellen dere anvendte i gjennomføring av den obligatoriske prosjektoppgaven i forhold til en alternativ prosessmodell. Angi og begrunn om du ville ha valgt den alternative prosessmodellen eller en annen prosessmodell dersom du skulle gjennomføre et lignende prosjekt på nytt.*

Her bør studenten komme med vurderinger om sammenhengen mellom valgt prosessmodell og prosjektets rammebetingelser, dvs størrelse, teknologisk risiko, usikkerhet i estimate, prosjektprioriteter (tid, kvalitet, kostnad), behov for opplæring m.m. Det meste av dette kan neppe leses direkte ut av Sommerville, og reflekterte erfaringer fra prosjektet (som viser at de har lagt innsats i og lært fra prosjektet) må vektlegges.

**Oppg. 2D** *Angi hvilke systemutviklingsverktøy dere benyttet i prosjektet. Klassifiser disse verktøyene. Beskriv fordeler og ulemper med bruk av disse verktøyene i forhold til oppgavene som ble løst. (Tips: Angi kriterier du har brukt i evalueringen og hva du sammenligner med når du angir fordeler og ulemper.)*

**Hvilke verktøy:**

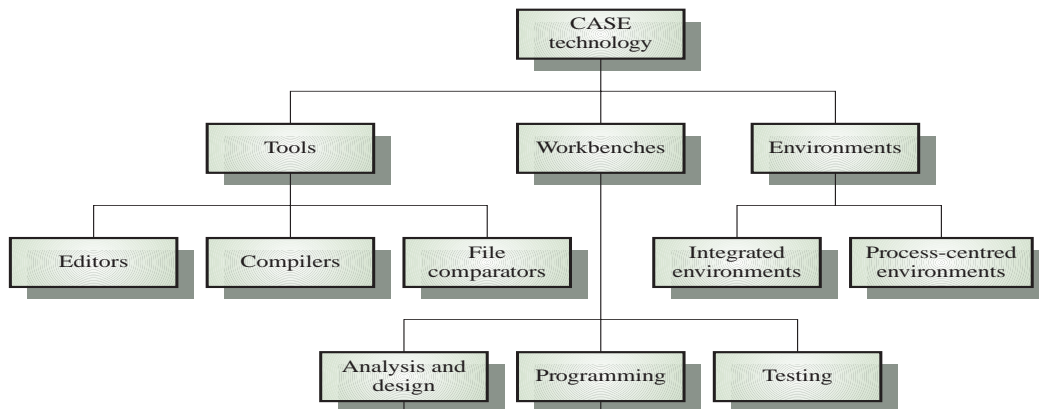
Årets studenter bør nevne Tau UML, helst også CVS



Tidligere års studenter bør i hvert fall nevne Make og CVS.

Vi godtar her alle typer verktøy (Sommerville er vid i sin def., jfr. figur nedenfor). Så det å nevne editor, kompilator er bedre enn ingenting.

**Klassifisering:**



Tool type	Examples		
	Multi-method workbenches	Single-method workbenches	General-purpose workbenches
Management tools	PERT tools, Estimation tools		Language-specific workbenches
Editing tools	Text editors, diagram editors, word processors		
Configuration management tools	Version management systems, Change management systems		
Prototyping tools	Very high-level languages, user interface generators		
Method-support tools	Design editors, data dictionaries, code generators		
Language-processing tools	Compilers, interpreters		
Program analysis tools	Cross reference generators, static analysers, dynamic analysers		
Testing tools	Test data generators, file comparators		
Debugging tools	Interactive debugging systems		
Documentation tools	Page layout programs, image editors		
Re-engineering tools	Cross-reference systems, program structuring systems		

**Eks. på produkter**

- MS project
- Emacs, MS word,
- Tau UML, Rose
- RCS, CVS, Make, Tau
- Generas Genova
- Tau UML, RosE

Jbuilder, VisualAge  
Fuse

Se også fig. 3.15 i sommerville som viser ulike typer verktøy i de ulike fasene:  
Spesifikasjon, design, implementasjon, Verifisering & validering

**Fordeler og ulemper**

Her kan det sies mye, det viktigste er at de har en god begrunnelse og at de har definert kriteriene sine, som kan være:

Funksjonalitet, og kvalitet og bredde på støtten.