[simula . research laboratory]

Kravspesifikasjon med UML use case modellering

Erik Arisholm

25.02.2009

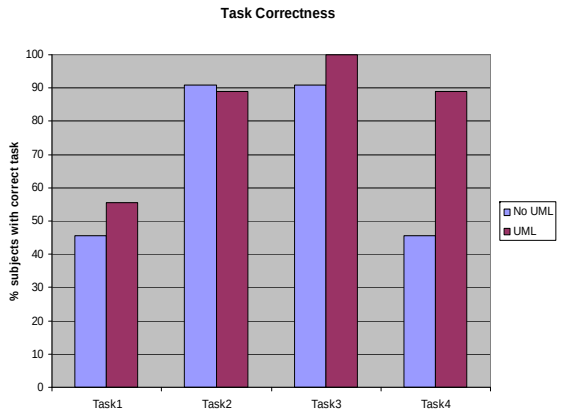
Unified Modeling Language (UML)

- Notasjon som støtter opp under modellbasert systemutvikling
  - objektorientert analyse (”hva systemet skal gjøre”)
  - objektorientert design (”hvordan systemet skal gjøre det”)
  - simulering, prototyping, kodegenerering og testing
- Dokumentasjon av systemets krav, arkitektur, design og implementasjon

UML diagrammer

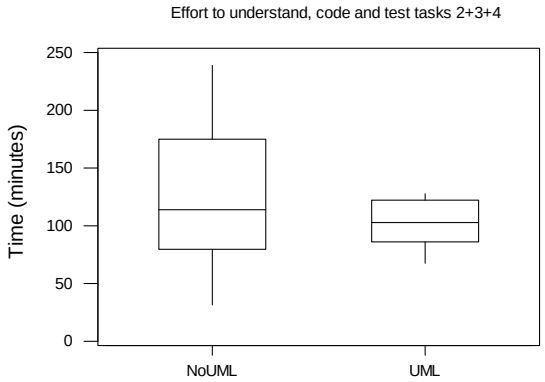
	Use-Case view	Logi-cal view	Compo-nent view	Concur-rency view	Deploy-ment view
Use Case diagram	■				
Class/object diagram		■			
Sequence diagram		■		■	
Collaboration diagram		■		■	
State diagram		■		■	
Activity diagram		■		■	
Component diagram			■	■	
Deployment diagram				■	■

Nytteverdien av UML som dokumentasjon – resultater fra et kontrollert eksperiment* (m/Java endringsoppgaver og Tau UML)



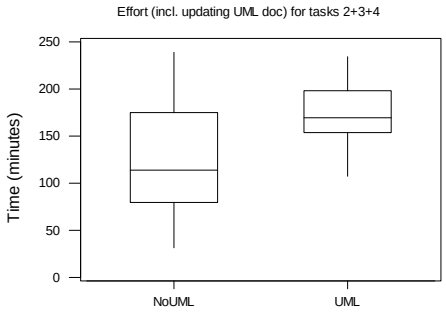
* E. Arisholm, L. C. Briand, S. E. Hove and Y. Labiche. The Impact of UML Documentation on Software Maintenance: An Experimental Evaluation, *IEEE Transactions on Software Engineering* 32(6):365-381, 2006.

Nytteverdi av UML – tidsforbruk på koding og testing*



•* E. Arisholm, L. C. Briand, S. E. Hove and Y. Labiche. The Impact of UML Documentation on Software Maintenance: An Experimental Evaluation, *IEEE Transactions on Software Engineering* 32(6):365-381, 2006.

Nytteverdi av UML – ekstra tidsforbruk for å oppdatere UML-modellene med Tau UML*



•* E. Arisholm, L. C. Briand, S. E. Hove and Y. Labiche. The Impact of UML Documentation on Software Maintenance: An Experimental Evaluation, *IEEE Transactions on Software Engineering* 32(6):365-381, 2006.

Beskrive krav med UML use cases (norsk: brukstilfeller eller bruksmønstre)

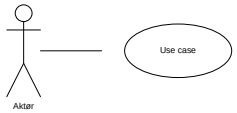
- Identifikasjon og beskrivelse av krav er viktige aktiviteter i et systemutviklingsprosjekt (jmf. forelesningen 4/2). Kravene bør være
  - forståelige (alle interessenter/stakeholders må kunne forstå kravspesifikasjonen),
  - testbare (vi må kunne avgjøre om det ferdige systemet gjør det det skal), og
  - sporbare (vi må vite hvilken del av koden som skal endres når det kommer nye krav).
- De funksjonelle kravene til et system kan beskrives v.h.a. **use cases**.
  - Use cases har blitt en "de facto" standard for systematisk analyse og modellering av krav i systemutviklingsprosjekter.
- Use cases beskriver de funksjonelle kravene til systemet, dvs.
  - hvordan systemet skal respondere på eksterne hendelser, og
  - hvem skal bruke systemet.
- Use cases beskriver de funksjonelle kravene på et **standard format**.

Use case modellen

- Beskriver
  - systemet sett utenfra,
  - interaksjonen mellom brukeren og systemet,
  - 'hva' som skjer, ikke 'hvordan' det skjer.
- dokumenterer omfanget til systemet,
- visualiserer kravene til systemet,
- Use casene driver hele utviklingsprosjektet fra kravanalyse, estimering, prosjektplanlegging, design til test, og er nyttige i de fleste prosjekter.

Use case modellen forts.

- Use case model =
  Use case diagram +
  Use case beskrivelser
- Use case diagrammet består av aktører og use case og gir en enkel oversikt.
- Use case beskrivelser kan f.eks. være "vanlig" tekst, strukturert tekst eller andre diagrammer.



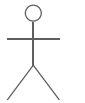
Prebetingelse:
X er nødvendig for at use caset skal starte
Hovedflyt:
1. Use caset starter
2. Use caset fortsetter
3. Use case avslutter

Alternativ flyt:
Use caset fortsetter på en annen måte
Postbetingelse:
Systemet er i tilstand Y

Aktører

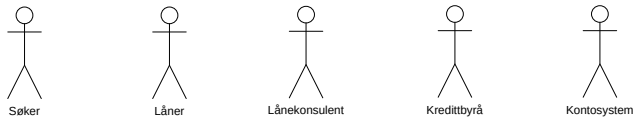
- tegnes som fyrstikkmennesker med et navn

En aktør
- kommuniserer med systemet via ett eller flere use cases,
- er en rolle som et menneske, et annet system eller en hardwarekomponent har når det kommuniserer med dette systemet,
- har mål med bruk av systemet,
- kan delta i mer enn et use case, og
- kan, men må ikke, være representert som en klasse når systemet implementeres.

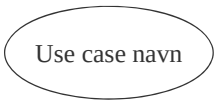


Aktører, et eksempel

Aktører for et system for behandling av lånesøknader



Use Case



- tegnes som ellipser med et navn i eller under ellipsen,
- representerer en komplett funksjonell enhet,
- beskriver en sekvens av handlinger som systemet utfører for å oppnå et observerbart resultat av verdi for en aktør,
  - hvert **use case steg** beskriver en enkelthandling mellom aktør og system
  - en **operasjon** er ett eller flere steg som må utføres samlet
  - et komplett use case består av flere ulike hendelsesforløp (flyt)
- beskriver testbar systemoppførsel,
- Navnet er vanligvis et aktivt verb og en substantivsetning som sier hva use caset skal gjøre.
- Et use case *må* ikke interagere direkte med aktører (men ofte gjør de det).

Use case forts.

Noen use cases for lånesystemet:

Stegene i use case modellering

1. Identifiser **primære aktører** (aktørene som initierer hendelser i systemet)
2. Identifiser use cases fra de primære aktørenes mål med systemet.
3. Identifiser **sekundære aktører**, dvs. aktører som ikke har egne mål, men som er nødvendige for å gjennomføre use casene.
4. Tegn use case diagram. Dette gir et overblikk over aktører og use cases og dermed over funksjonaliteten til systemet.
5. Lag tekstlige beskrivelser av use casene. Disse viser hvordan aktører oppnår resultater ved bruk av systemet.

Identifisere aktører og use case

- Spørsmål for å finne aktører:
  - Hvem eller hva skal starte hendelser i systemet?
  - Hvem eller hva skal kommunisere med systemet for å la systemet svare på en hendelse?
  - Hvem eller hva skal informeres om hendelser i systemet?
  - Skal det være grensesnitt for rapportering eller administrasjon?
  - Skal systemet ha grensesnitt mot eksisterende systemer?
- Spørsmål for å finne use case:
  - Hvilke resultater vil aktørene oppnå med bruk av systemet?
  - Et use case -

    ’One person - one place - one time’

(Use case kan også brukes på et høyere nivå for å gi en overordnet beskrivelse av et systems funksjonalitet, for eksempel ”Håndter lånesøknader”)

Hvilke av disse setningene tror du representerer et use case?

– ”Logg inn på systemet”

– ”Lei en bil”

– ”Behandle retur av leiebil”

– ”Øk produksjonen med 20%”

– ”Varesalg”

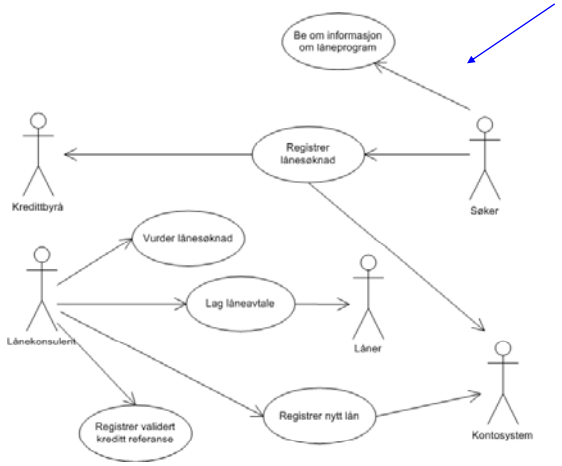
Hint:

– Hvor går systemgrensen, hvem er primær aktør og hvilke(t) mål oppfyller use caset?

Tekstlige krav som utgangspunkt for å identifisere aktører og use cases

#	Krav	Aktør	Use case
1	*Systemet skal på oppfordring vise informasjon om alle tilgjengelige låneprogrammer.*	*Søker*	*Be om informasjon om låneprogram*
2	*En lånesøker skal kunne søke om lån online ved å registrer all nødvendig informasjon i et skjema. Informasjonen sjekkes automatisk og gis en kreditt score basert på informasjon fra kredittbyrå samt kundens forhold til banken.*	*Søker, Kredittbyrå, Kontosystem*	*Registrer lånesøknad*
3	*Søkeren skal til enhver tid kunne se status for lånesøknaden.*	*Søker*	*Se status*
4	*Registrerte lånesøknader skal vurderes for godkjenning. Hvis søknaden godkjennes skal dette markeres i systemet slik at et nytt lån kan opprettes.*	*Lånekonsulent*	*Vurder lånesøknad*
5	*Lånekonsulenten skal kunne registrere kredittinformasjon for lånekunder som valideres manuelt som for eksempel eksterne bankkonti og inntektskilder.*	*Lånekonsulent*	*Registrer validert kredittreferanse*
6	*Låneavtaler skal kunne opprettes og sendes til kunden for elektronisk godkjenning via internett.*	*Lånekonsulent, Låner*	*Lag låneavtale*
7	*Nye lån skal registreres i bankens system for kontooversikter.*	*Lånekonsulent Kontosystem*	*Registrer nytt lån*

Use case diagram



Assosiasjon

- Viser kommunikasjon mellom en aktør og et use case.
- Pil kan brukes til å vise hvor kommunikasjonen er startet (men NB! Må ikke forveksles med retningen på ”dataflyt”)

Detaljering i iterasjoner

1. En høynivå use case modell består av diagram og en kort beskrivelse av alle use cases
2. Deretter beskrives hovedflyt for (de viktigste) use casene
3. Use casene kan oppnå sitt resultat på flere måter, og kan feile (også på flere måter), så en detaljert use case modell har use cases med både hovedflyt og alternative flyt.

(Men det er ofte ikke nødvendig å detaljere ut alle use casene)

Høynivå use case beskrivelse

Use case	**Registrer lånesøknad**
Kort beskrivelse	Use caset lar en søker registrere en søknad om lån i systemet.
Aktører	Søker, kredittbyrå, kontosystem.
Hovedflyt	Lånesøker fyller ut en lånesøknad og sender den til banken. Systemet validerer informasjonen i lånesøknaden og beregner søkerens kreditt score basert på kredittrapporter og søkerens kundeforhold til banken. Systemet gir en initiell anbefaling om godkjenning.

Detaljerte tekstlige use case

- Beskriv hva som gjøres, ikke hvordan det gjøres

Beskriv:

- Involverte **aktører**
- **Prebetingelser** (**preconditions**), disse beskriver forutsetningene for å gjennomføre use caset.
- **Detaljert beskrivelse** av hendelsesflyt med **hovedflyt og alternative flyt**
  - Skriv hendelsesflyten som en nummerert liste på formen
    1. <Lånekonsulent> <registrerer> <kredittinformasjon>
    2.<Systemet><viser><ubehandlede lånesøknader>
  - Finn riktig detaljeringsnivå
    - Beskriv kun 1 hendelse per steg
    - Vanligvis beskrives ikke detaljer om brukergrensesnitt.
      Eksempel: Ikke 'Aktør trykker på 'Send'-knappen'
- **Postbetingelser (postconditions),** disse beskriver endringer i tilstanden til systemet

Detaljert beskrivelse av hovedflyt

1. Søkeren fyller ut en online lånesøknad
2. Søkeren sender søknaden til banken via internet
3. Systemet validerer informasjonen i lånesøknaden ved å sjekke at den er så korrekt og komplett som mulig.
4. Systemet innhenter kredittrapport for søkeren fra et eksternt kredittbyrå for kredittrapport.
5. Systemet henter søkerens kontohistorie med banken fra kontosystemet.
6. Systemet beregner søkerens kredittscore basert på kredittrapport og kontohistorie.
7. Systemet informerer søkeren via e-mail om at søknaden er mottatt og blir vurdert.
8. Systemet setter status på lånesøknaden til "Initiell kredittsjekk ferdig".
9. Systemet allokerer lånesøknaden til en lånekonsulent for videre behandling.

Pre- og postbetingelser

Er viktige for funksjonell test av systemet (black-box testing)

Use case "Vurder lånesøknad":

Prebetingelse:

Lånesøknaden har status "Initiell kredittsjekk ferdig"

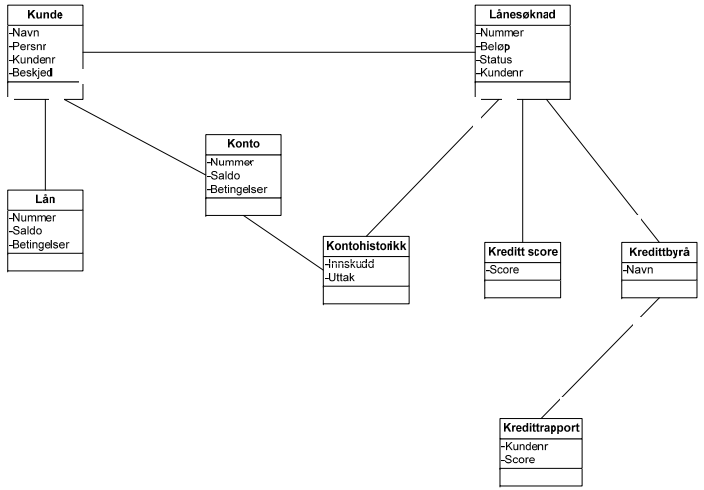
Postbetingelse:

Lånesøknaden har status "Godkjent",
Lånesøknaden har status "Informasjon mangler", eller
Lånesøknaden har status "Avslått" og søker har fått beskjed om at søknaden er avslått.

Domenemodell (Domeneklassediagram)

- Domenemodellen er nyttig i forbindelse med use case modellering fordi
  1. Domenemodellen fanger opp informasjonen om objekter i use casene og er et viktig verktøy for å sjekke at use casene er beskrevet med riktig detaljeringsnivå.
  2. Klassene i domenemodellen kan brukes i utforming av mer presise pre- og postbetingelser.
- Domenemodellen viser objekter i problemdomenet, dvs. objekter som skal håndteres av systemet,
- den beskrives med UML klassediagrammer uten metoder, og
- hensikten med domenemodellen er å forstå objektene og få en oversikt over terminologi.

Domenemodell



Mer presise pre- og postbetingelser

- **Prebetingelse:**

  Lånesøknad.status = ”Initiell kredittsjekk ferdig”

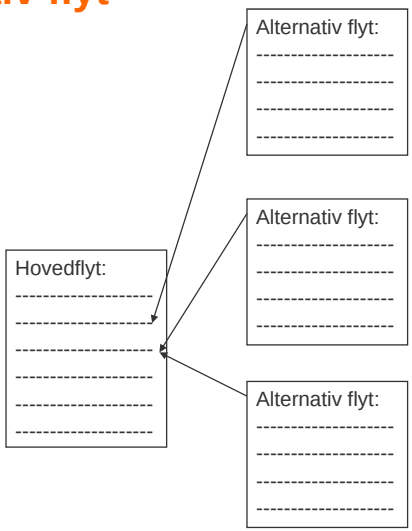
- **Postbetingelser:**

  Lånesøknad.status = ”Godkjent” or
  Lånesøknad.status = ”Informasjon mangler” or
  (Lånesøknad.status = ”Avslått” and Kunde.beskjed = ”Sendt”)

  Mer presise pre- og postbetingelser er nyttige i funksjonell test.

Alternativ flyt

- Et use case har typisk en eller flere alternative flyt i tillegg til hovedflyt for å beskrive variasjoner og feilsituasjoner.
- Alternativ flyt (inkl. feilsituasjoner) er viktige da det ofte er mer ”uenighet” blant prosjektets interessenter om hva som skjer i de tilfellene enn for hovedflyt.
- Ethvert steg i hovedflyt kan være utgangspunkt for en alternativ flyt.



Eksempel - Registrer lånesøknad

Hovedflyt:

1. Søker fyller ut en online lånesøknad og sender den via web til banken.
2. Systemet sjekker så langt som mulig at informasjonen i lånesøknaden er korrekt og komplett.
3. Systemet innhenter kredittrapport for søkeren fra et eksternt kredittbyrå.
4. Systemet henter søkerens kontohistorikk fra kontosystemet.
5. Systemet beregner søkerens kredittscore basert på kredittrapport og kontohistorikk.
6. Søkeren informeres per e-mail om at lån er mottatt og behandles.
7. Systemet setter lånesøknadens status til ”Initiell kredittsjekk komplett”.
8. Systemet sender lånesøknaden til en lånekonsulent for videre behandling.

Alternativ flyt 1, steg 2: Informasjonen i lånesøknaden er ikke komplett og korrekt:

A1.1. Systemet returnerer lånesøknaden til søker for ytterligere utfylling.

A1.2. Systemet setter lånesøknadens status til ”Avventer”.

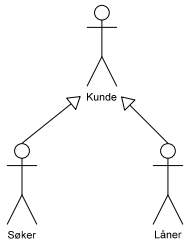
Alternativ flyt 2, steg 3: Det eksisterer ikke noe kredittinformasjon om søkeren eller kredittrapporten er for dårlig til å innvilge lån:

A2.1. Systemet sender beskjed til søker om at søknaden er avslått

A2.2. Systemet setter lånesøknadens status til ”Avslått”.

Relasjoner mellom aktører

- Kan definere en generell aktør hvis to eller flere mer spesialiserte aktører har en del felles oppførsel



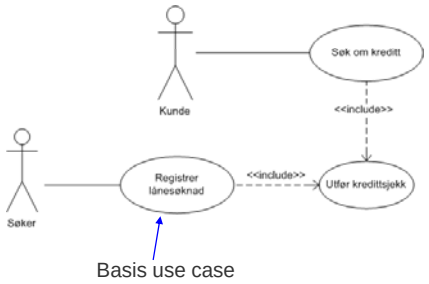
Søker og låner kan gjøre alt det som Kunde kan + mer

Use case relasjoner

- **Include-relasjonen:**
  Et use case kan være en del av ett flere andre use case.
- **Extend-relasjonen:**
  Et use case som beskriver tilleggsoppførsel som utføres under gitte omstendigheter

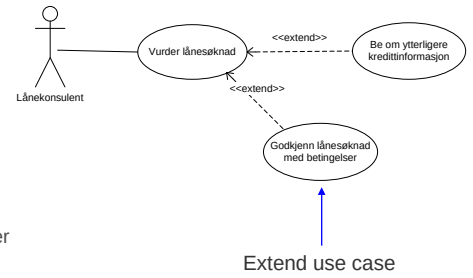
Include relasjonen

- To eller flere use cases kan ha en felles del (noen like steg). Denne delen kan da legges ut i et eget use case som disse use casene kan inkludere.
  - Include kan også brukes for å forenkle store use case med mange steg
  - Include kan også brukes for å håndtere steg som kan forekomme når som helst i utførelsen av use caset
- Basis use caset vet hvilke use case det inkluderer

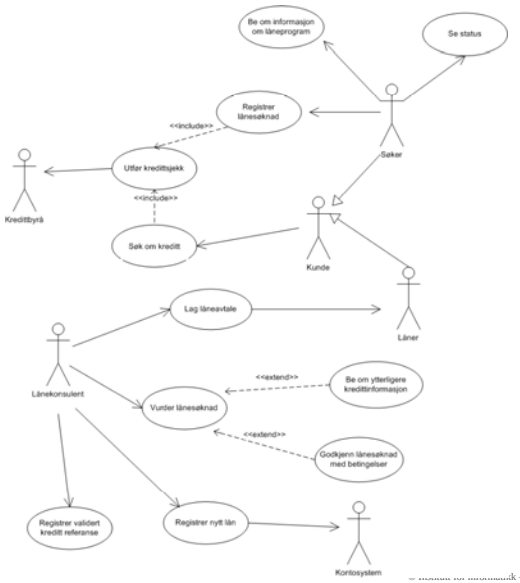


Extend relasjonen

- Alternativ oppførsel som utføres i noen tilfeller kan skrives som eget use case som utvider (extender) et annet.
- Extend use case beskriver hvordan oppnå et tilleggsresultat.
- Basis use caset er fullstendig definert uten extensions, disse utvider funksjonaliteten.
- Basis use caset kjenner sine extended use case
- Bruk av alternativ flyt vs. bruk av extend use case:
  - Alternativ flyt beskriver hva som skjer ved avvik i normal flyt, mens
  - extend use case beskriver hvordan oppnå tilleggsresultat.



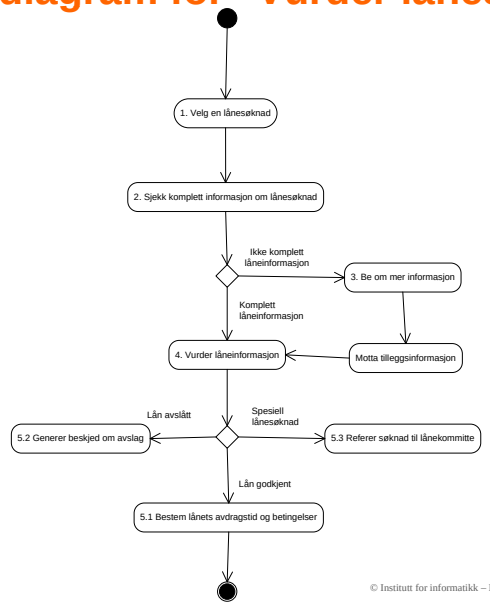
Detaljert use case diagram



Aktivitetsmodellering

- Et **aktivitetsdiagram** kan grafisk representere hendelsesflyten i et use case, og består av
  - Startsymbol ●
  - Aksjoner, dvs. det som skal gjøres
  - Piler som viser flyt av kontroll fra en aksjon til den neste
  - Beslutningspunkter ◇
  - Spredning og samling, når to eller flere jobber skal gjøres samtidig og så samles til en aktivitet igjen
  - Sluttsymbol ◉
- Kan representere et steg mot programkode
- Kan også brukes:
  - For å forstå en forretningsprosess på et høyt nivå før use case produseres
  - På et lavere abstraksjonsnivå for å designe komplekse algoritmer eller applikasjoner med flere parallelle tråder

Aktivitetsdiagram for "Vurder lånesøknad"



Bruk av use case modellen

Kunde	Validering av krav
	Prosjektplan og estimering
	Utgjør basis for akseptansetest (testen som gjøres av kunden for godkjenning av systemet)
Bruker	Validering av krav
	Viser brukerens interaksjon med systemet
Prosjektleder	Risikoanalyse
	Prosjektplan og estimering
	Fremdriftsoversikt (hvilke use case er implementert)
	Sporbarhet av krav (hvilken kode må endres når krav endres)
Systemarkitekt	Hjelp til valg av arkitektur og teknologi
	Sporbarhet av arkitekturkrav
	Vurdering av kompletthet, konsistens og koherens for arkitekturen
Systemutvikler	Modell av kravene som kan brukes i design av systemet
	Systemdokumentasjon
Vedlikeholder	Gir retningslinjer for hvordan gjennomføre endringer

Use cases i prosjektplanlegging

- Planlegg hvilke use case som skal implementeres i hvilke iterasjoner av prosjektet:
  - Implementer use casene i henhold til hvor viktige de er og/eller hvor vanskelige de antas å være å implementere.
  - Normal hendelsesflyt implementeres først, deretter variasjonene.
  - Estimer hvor mange use cases (eller hendelsesflyt og alternative flyt) som kan implementeres i en iterasjon.

Oppsummering

- Use case modellering er en systematisk metode for å identifisere og beskrive funksjonelle krav til et system.
- Use case modellen består av
  - aktører og use cases
  - diagram og use case beskrivelser
    - De enkelte use casene kan beskrives på forskjellige måter avhengig av prosjektet, det vanligste er strukturert tekst, men aktivitetsdiagram kan også brukes.
    - Domenemodellen støtter use case beskrivelsene.
- Use casene er sentrale i de fleste aktiviteter og for de fleste interessenter i et systemutviklingsprosjekt.