
Gruppetime 10

IN1030

Plan for dagen

- Smidig utvikling
 - Scrum
 - Kanban
- Prosjektledelse og risikohåndtering
- Use case- og sekvensdiagram

Litt recap av prosessmodeller

- En prosessmodell er en forenkling, et slags “veikart” som kan brukes som utgangspunkt for å planlegge ulike aspekter ved et prosjekt.
- En forenklet presentasjon av prosessen
- Noe man må lage før man starter en prosess
- Bør inkludere:
 - ◆ hvilke metoder og teknikker man skal bruke
 - ◆ hvordan man skal jobbe sammen
 - ◆ hvilke aktiviteter man skal gjøre
 - ◆ hvilke roller de som er med har

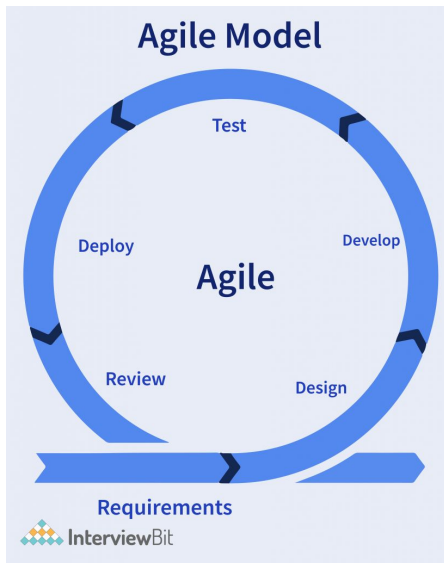
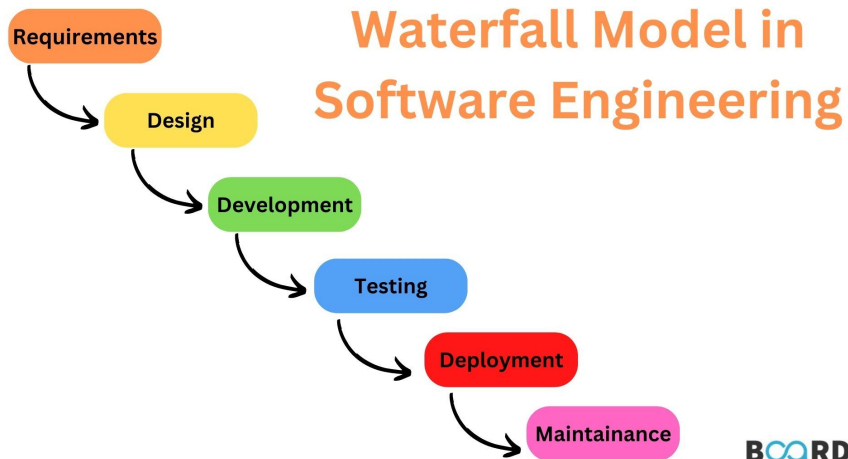
Plandreven vs smidige prosessmodeller

Plandreven:

- Består av separate faser der utviklingen styres av forhåndsdefinerte planer
- En fase skal være ferdig før teamet starter på neste
- Hver fase avhenger mye av leveransen til den forrige fasen

Smidig:

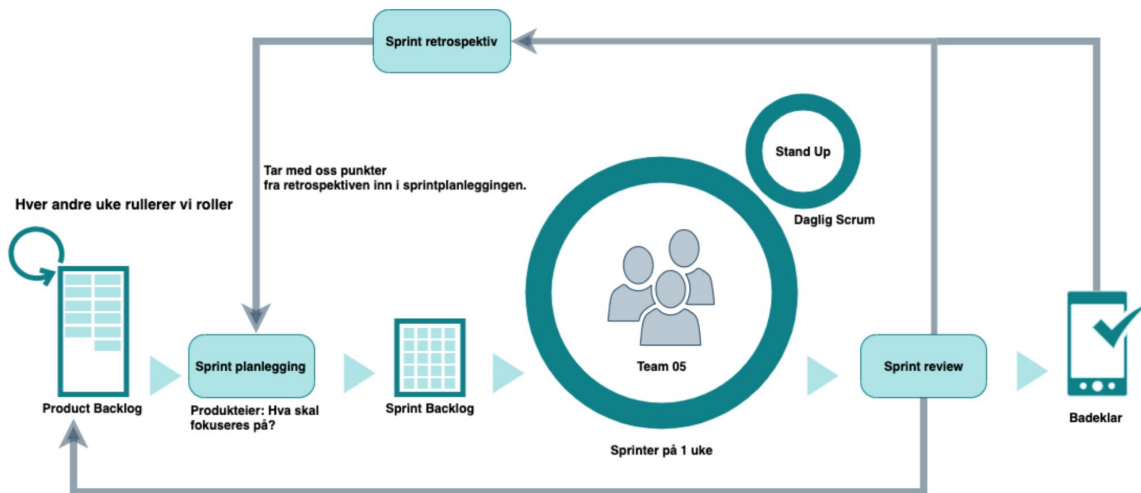
- Handler også om å bryte et prosjekt i ulike faser, men med konstant forbedring i hver fase
- Kan stadig gjennom prosessen iterere og implementere nye/forbedrede funksjonaliteter



Scrum

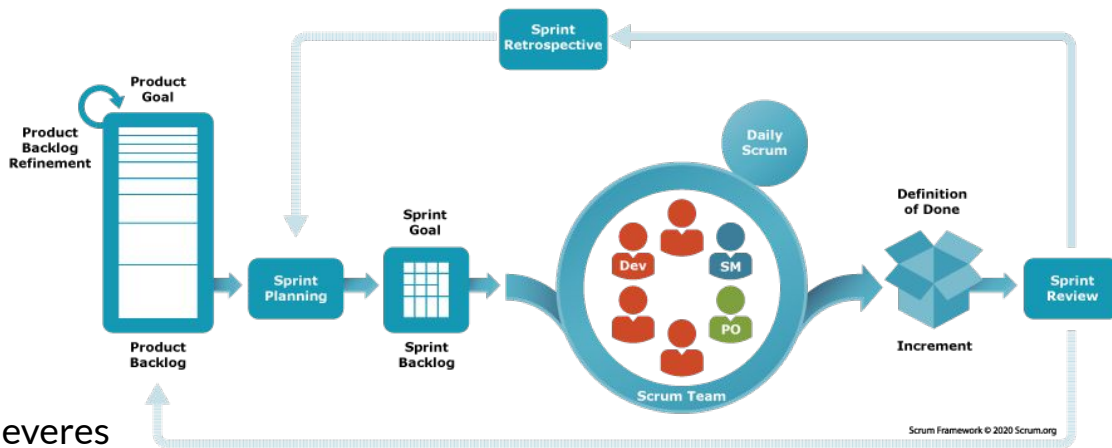
Smidige metoder: Scrum

- Baserer seg på tidsbokser (også kalt sprinter), som ofte har en varighet på 2-4 uker
- Man velger noen prioriterte oppgaver som skal jobbes med i sprinten
- Består av tre faser:
 - ◆ Planleggingsfasen
 - ◆ Gjennomføringsfasen
 - ◆ Avslutningsfasen



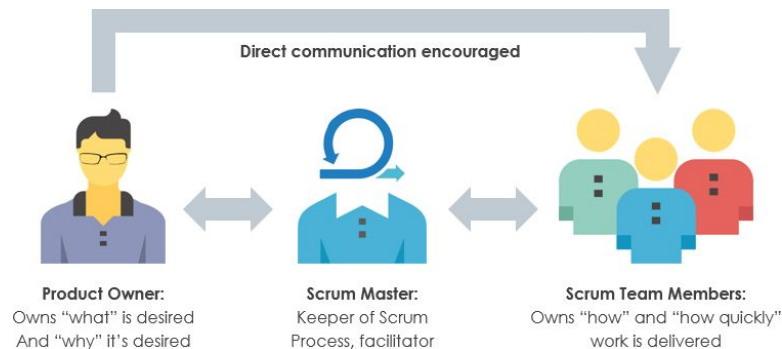
Hvilke aktiviteter har vi i Scrum?

- Sprint planlegging
 - ◆ Product backlog
 - ◆ Sprint backlog
 - ◆ Mål for sprint
- Sprint utførelse
 - ◆ Design, kode og testing
- Daglig scrum
 - ◆ Stå opp møter (stand-up møter)
- Sprint review (Demo)
 - ◆ Inkrement eller ferdig produkt leveres
- Retrospektivmøte
 - ◆ Snakke om hvordan det gikk og hvorfor det gikk som det gikk



Rollene i Scrum

- **Scrum Master**
 - ◆ Ansvarlig for å holde daily stand-ups
- **Product Owner**
 - ◆ Eier av visjon og representant for interessentene
 - ◆ Ansvarlig for Product Backlog og prioritering av oppgaver
 - ◆ Product Owner skal aldri være Scrum Master
- **Scrum team**
 - ◆ Et tverfaglig team som skal utvikle inkrementet i hver sprint



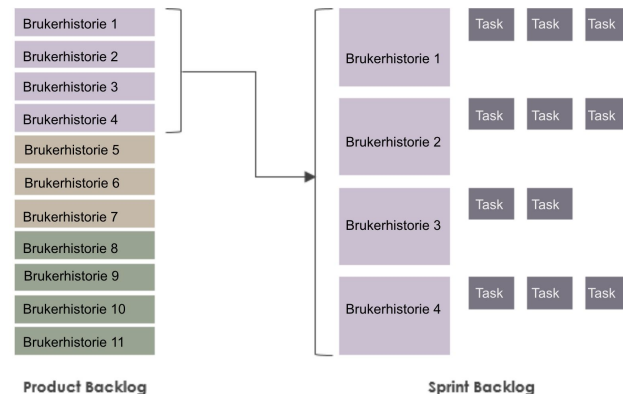
Product Backlog vs. Sprint Backlog

→ Product Backlog:

- ◆ Prioritert oversikt over alle oppgaver som skal til for å få lagd endelig produkt
- ◆ Styres og prioriteres av Product Owner
- ◆ Ofte er dette liste over brukerhistorier

→ Sprint Backlog:

- ◆ Prioritert oversikt over alle oppgaver som skal til for å få lagd planlagt inkrement av endelig produkt
- ◆ Viser estimert tids- og ansvarsfordeling for hver enkelt oppgave
- ◆ Oppgavene er mindre og mer spesifikke, definert ut fra oppgavene fra Product Backlog
- ◆ Opprettes i Sprint Planning-møtet, og gjennomføringen vurderes i Sprint Review-møtet
- ◆ Oppgavene skal gjøres innenfor Sprint-tiden, men vil overføres til neste Sprint om de ikke blir gjort ferdig



Eksempel på scrum board

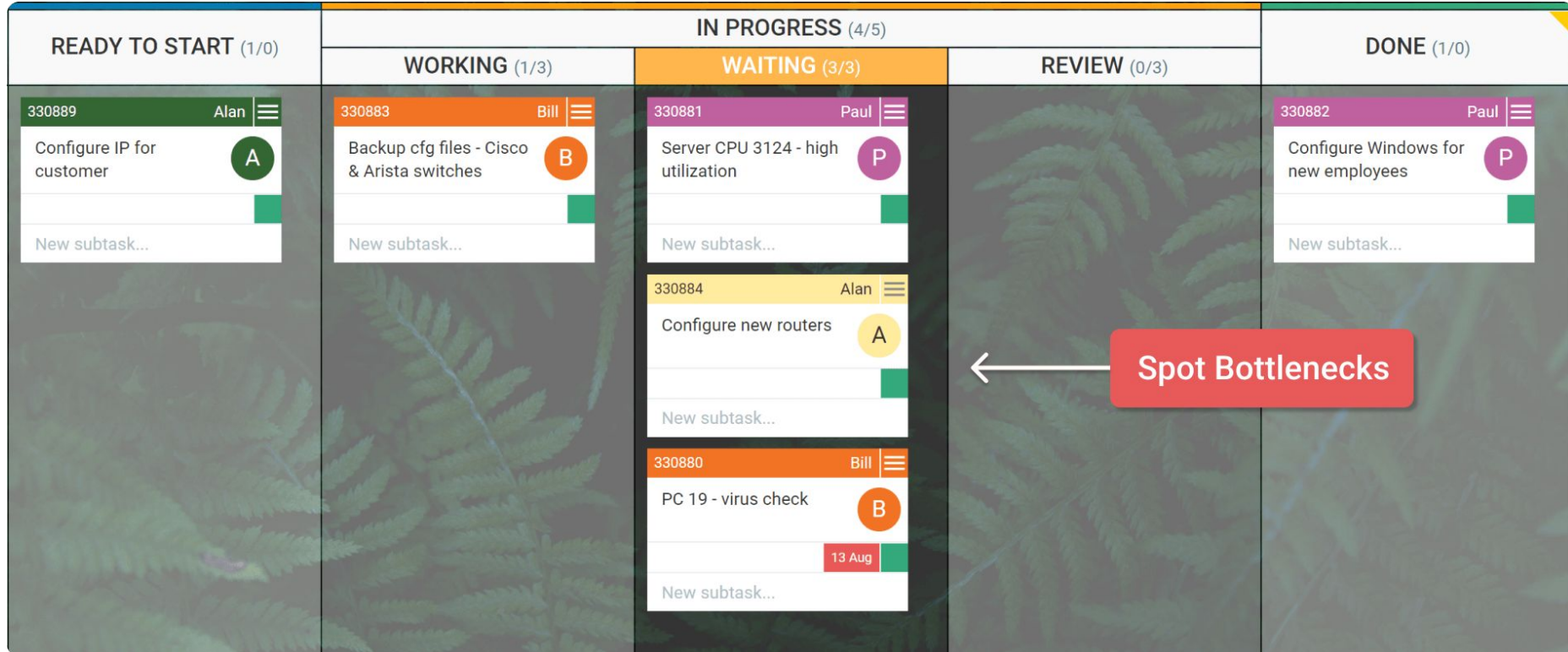


Kanban

Kanban

- Fokuserer å forbedre produksjonsprosessen
- Man gjør en oppgave til den er ferdig (Work in progress - WIP)
- Når en oppgave er utført velger man en ny som er viktigst der og da
- WIP er viktig for å sikre arbeidsflyt og se oversikt over nåværende status
- Unngå flaskehals
 - ◆ Når et kritisk problem oppstår stopper man produksjon og fikser det slik at det blir løst og ikke hopper seg opp

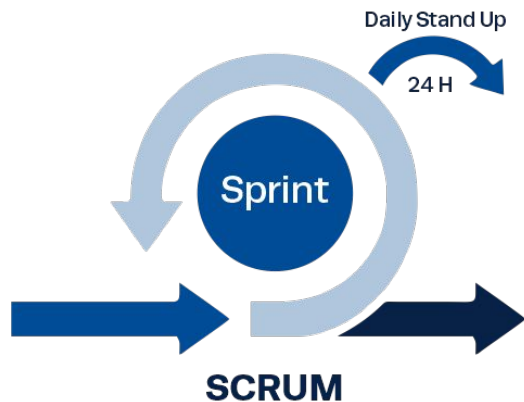
Eksempel på kanban board



Scrum vs kanban

Scrum:

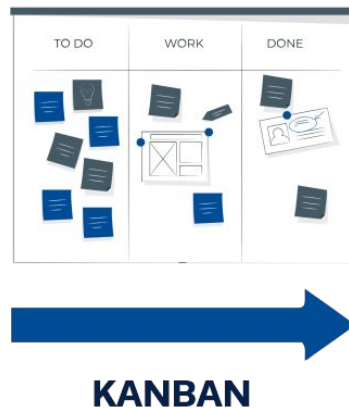
- ◆ Basert på faste tidslinjer kalt sprinter.
- ◆ Arbeidsoppgaver er organisert i sprinter, vanligvis 2-4 uker lange.
- ◆ Sprinter starter med en planleggingsmøte og avsluttes med en gjennomgang.
- ◆ Har roller som scrum master, product owner og teammedlemmer.
- ◆ Mer kommunikasjon



VS

Kanban:

- ◆ Fokuserer på kontinuerlig arbeidsflyt.
- ◆ Ingen faste tidslinjer som sprinter; arbeidet flyter kontinuerlig (WIP).
- ◆ Ingen faste roller, men teamet kan ha fleksible ansvarsområder.
- ◆ Legger vekt på å begrense arbeid i gang (WIP) for å opprettholde jevn flyt.



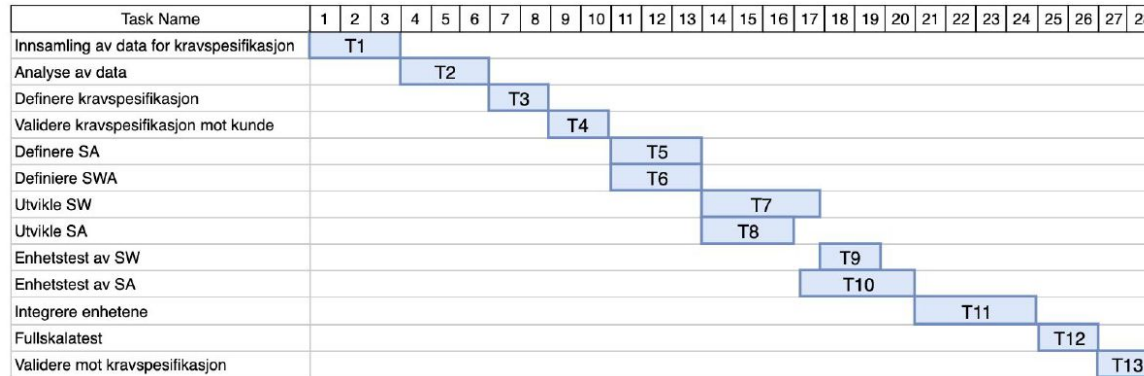
Prosjektledelse

FOKPRE: 6 tema som er viktig innenfor prosjektledelse

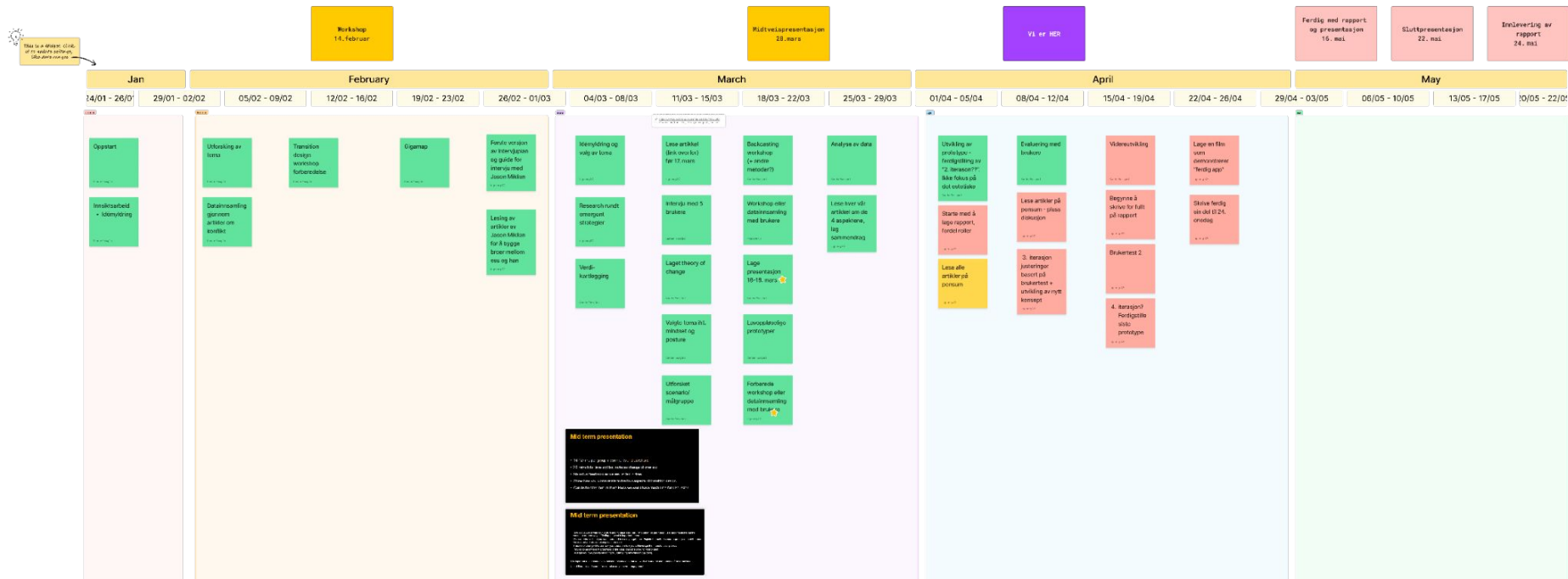
- Forretningsplan
 - ◆ Hvorfor?
- Organisasjon
 - ◆ Hvem?
- Kvalitet
 - ◆ Hva og hvor bra?
- Plan
 - ◆ Hvordan? Hvor mye? Når?
- Risiko
 - ◆ Hva hvis?
- Endring
 - ◆ Hvilke endringer skaper systemet?

Prosjektplanlegging

- Oversikt over alle oppgaver, hvor lang tid de tar, og avhengigheter til hverandre
- Kan lage visualisering i form av søylediagram



Prosjektplanlegging



Risikoanalyse og håndtering

- Risiko er en sannsynlighet for at uønskede omstendigheter skjer
- Deler det gjerne inn i en risikotabell (neste slide)
- Tre hovedtyper risiko:
 - ◆ **Prosjektrisiko**
 - Tidsplan og/eller ressurser
 - ◆ **Produktrisiko**
 - Kvaliteten eller programvaren som utvikles
 - ◆ **Forretningsrisiko**
 - Organisasjonen som utvikler/eier programvaren

Vurderinger for risikoanalyse:

Sannsynlighet: *svært lav - lav - moderat - høy - svært høy*

Konsekvens: *ubetydelig - mindre alvorlig - alvorlig - katastrofal*

Risikotabell med eksempel

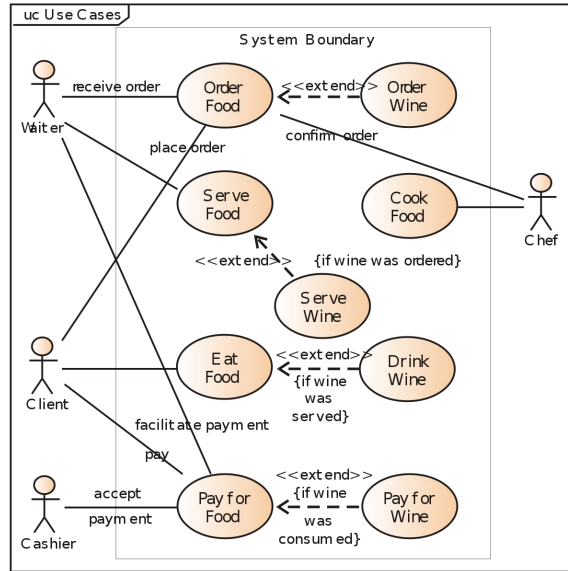
Få med:

- Risikotype
- Sannsynlighet for risiko
- Konsekvens av risiko
- Tiltak for å unngå risiko
- Hvem som er ansvarlig

Risikotype	Sannsynlighet for risiko	Konsekvens av risiko	Tiltak for å unngå risiko	Hvem som er ansvarlig
Ansatte blir syke	Høy	Middels	Ha flere backup ansatte	Ledelsen som ansetter folk
Intern konflikt 	Middels	Lav (kommer an på hvor i hierarkiet konflikten er)	Team-building møter en gang i blant	Ansatte og ledelsen som står for arbeids - oppsettet
Systemet blir hacket	Middels	Alvorlig	Forebyggende sikkerhetstiltak i systemutvikling	Systemutvikler
Systemsvikt	Middels	Alvorlig	Backup - program i tilfelle svikt	Systemutvikler
Masseskyting på kino (USA-edition)	Lav	Katastrofal	Høyere sikkerhet	Ledelsen som ansetter sikkerhets - personell
For dårlig markedsføring	Høy	Alvorlig	Hente inn et PR-byrå/ konsulentfirma	Ledelsen som ansetter eller de som står for den dårlige markedsføringen

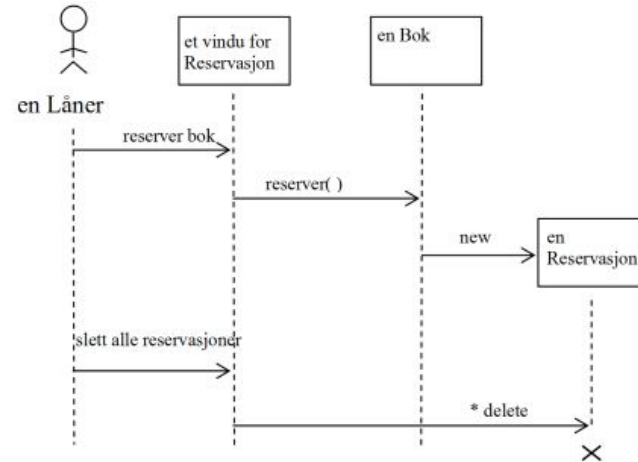
Use case- og sekvensdiagram

Use case diagram



Viser interaksjon mellom et system og omgivelsene (aktører). Tar utgangspunkt i primæraktørs mål og hvordan sekundæraktører assisterer dette målet gjennom systemet.

Sekvensdiagram



Viser interaksjon og informasjonsflyten mellom aktørene og systemet og systemkomponentene i form av objektklasser.

Interessent vs aktør

→ Interessent

- ◆ En som generelt blir påvirket av systemets implementasjon.

→ Aktør

- ◆ En som har en aktiv rolle med systemet og faktisk anvender det.

→ Vil en aktør alltid være en interessent?

→ Vil en interessent alltid være en aktør?



Primær vs sekundær aktør

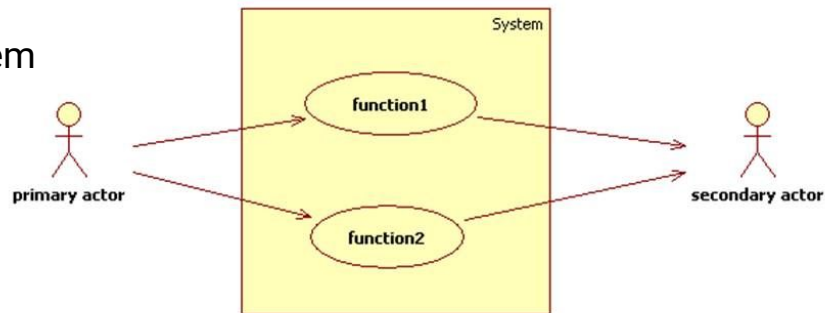
→ Primær aktør:

- ◆ Har et eget mål i kommunikasjon med systemet
 - Eksempel: bruker av banksystem

→ Sekundær aktør

- ◆ Trengs for at primæraktøren skal kunne nå målet sitt - kommuniserer også med systemet
 - Eksempel: banktjenesten sin kundeservice

- ❖ Aktører trenger ikke være en person - kan være et system



Use case diagram

- **Brukes for å se hvordan systemet og aktører forholder seg til hverandre.**
- Lages gjerne i tidlig fase av systemutvikling - typ kravspec.

Hvilke mål har primær aktøren? → Boble

Hvem hjelper til med å nå primær aktørs mål? → Sekundær aktør

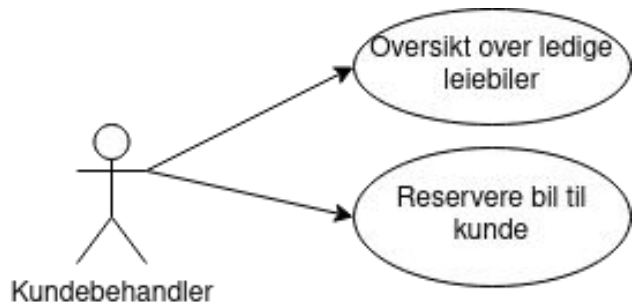
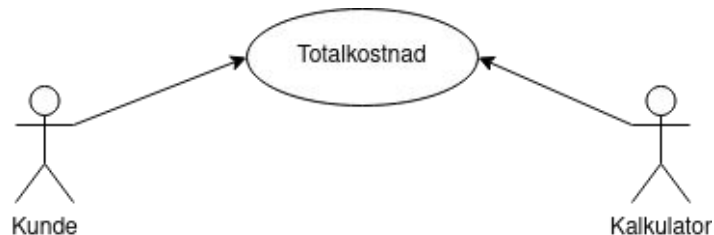
Fra brukerhistorie til Use Case diagram:

Som kunde ønsker jeg å vite totalkostnaden på å leie bilen slik at jeg vet om jeg har råd.

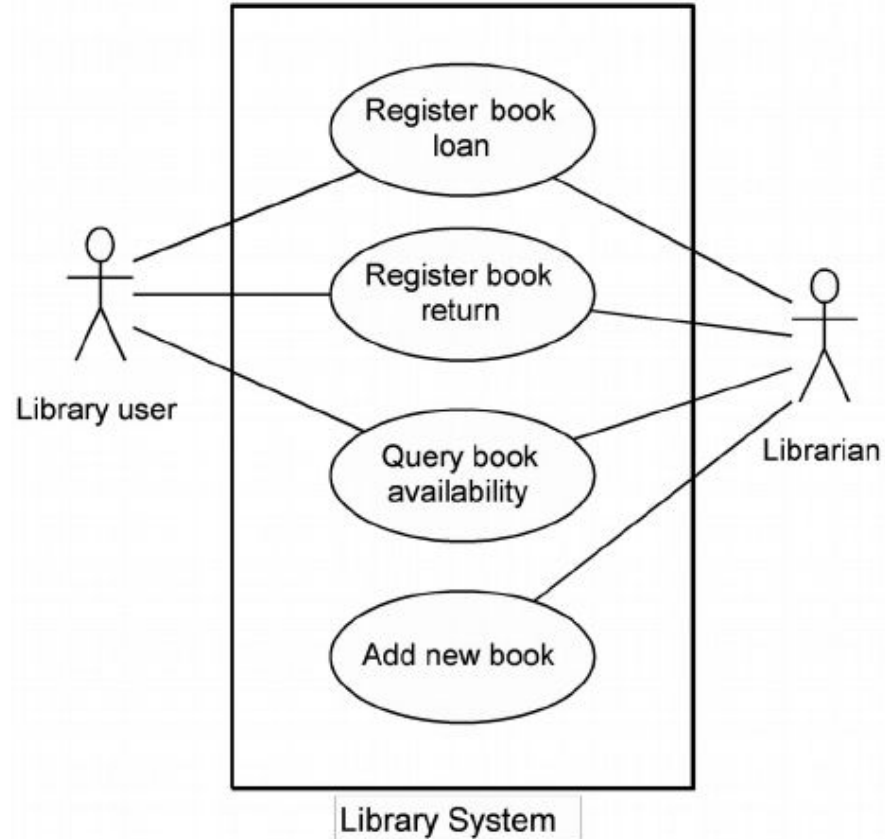
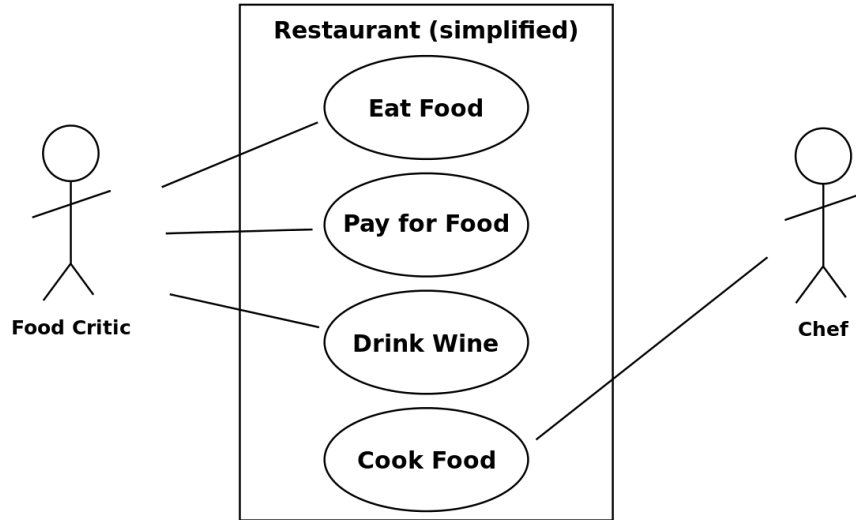
Som kundebehandler ønsker jeg å se hvilke biler jeg kan leie ut til kunden slik at jeg ikke dobbeltbooker.

Som kundebehandler ønsker jeg å reservere bil til kunden slik at jeg kan yte god kundeservice

Bilreserveringssystem



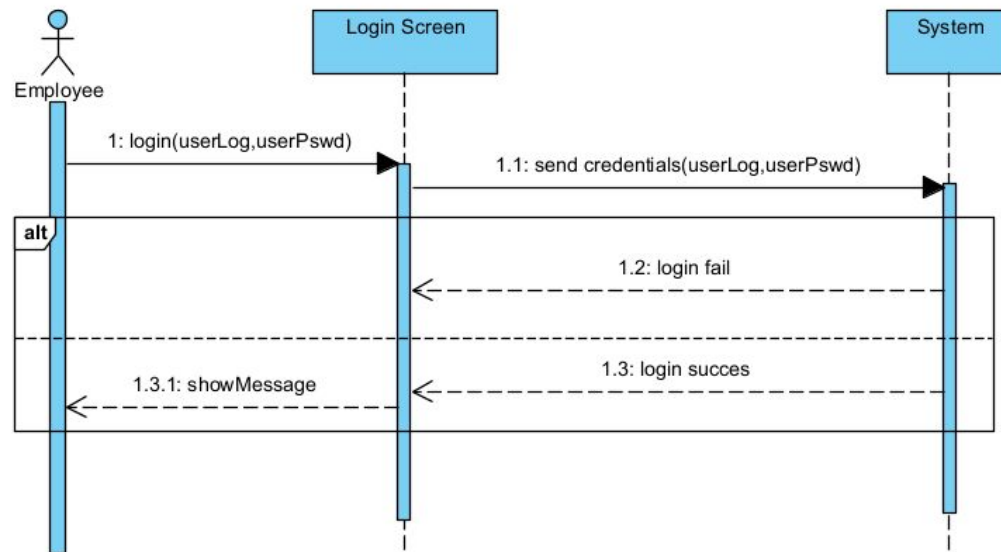
Use case diagram - enkle eksempler



Sekvensdiagram

- Brukes for å se hvordan objekter og komponenter interagerer med hverandre over tid
- Lages gjerne når systemet skal designes - etter kravspec
- Teknisk avbildning av interaksjonen

Aktøren interagerer med systemet og vi ser hvilke metodekall som gjøres i de ulike klassene.



Navn: Kjøp billett som eksisterende kunde

Primæraktør: Kunde

Sekundæraktør: -

Prebetingelse: Kunden har internett

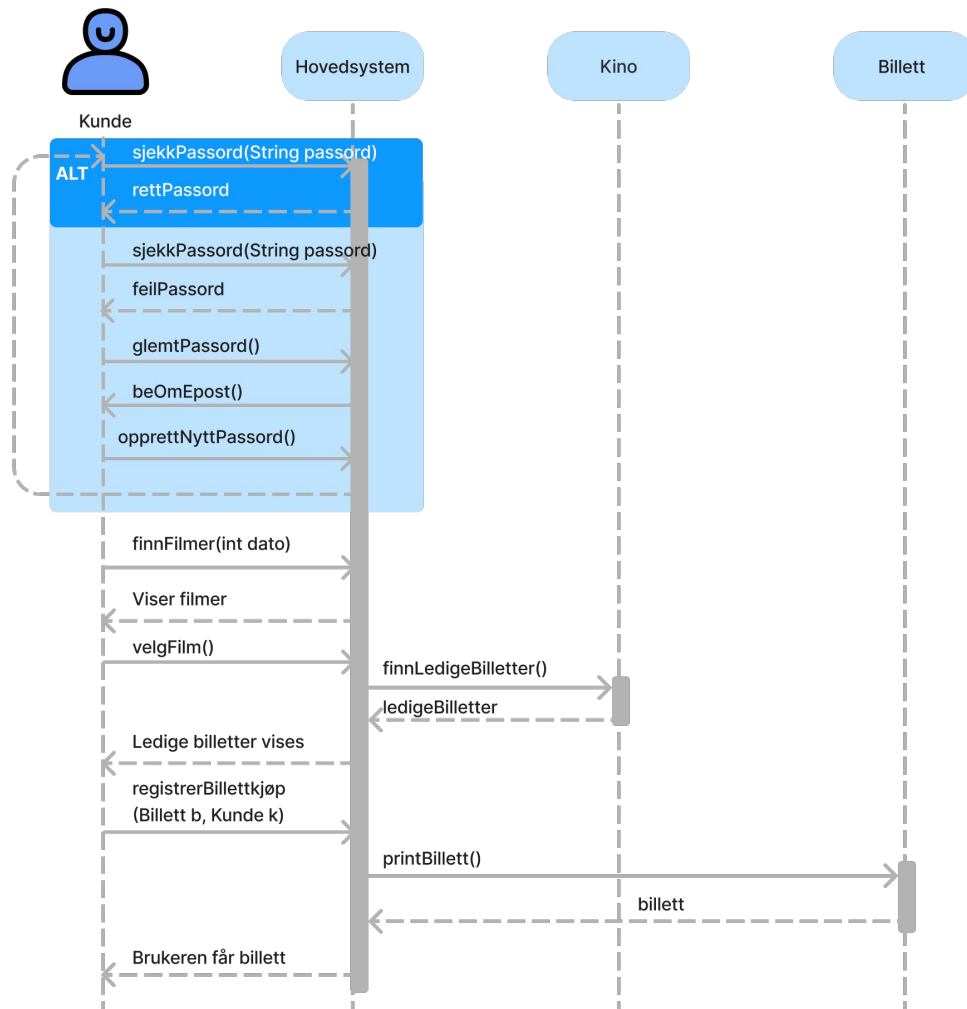
Postbetingelse: Kunde får billett

Hovedflyt:

1. Kunde åpner nettsiden og kommer til innloggingssiden
2. Kunde skriver inn riktig brukernavn og passord og trykker "Logg inn"
3. Systemet slipper inn kunde på sin kundeprofil
4. Kunde går inn på filmoversikt
5. Systemet returnerer en liste over tilgjengelige filmer
6. Kunde trykker på ønsket film
7. Systemet returnerer en liste over tilgjengelige billetter
8. Kunde velger ønsket billett og kjøper med forhåndsinnlagt betalingsinformasjon på sin konto
9. Billetten blir lagt til i kundes liste over billetter og tatt ut av kinoens og hovedsystemets liste over ledige billetter
10. Billett blir lagt inn som PDF i kundes billettoversikt

Alternativ flyt punkt 2:

- 2.1: Kunde husker ikke passord
- 2.2: Kunde trykker på "Glemt passord"
- 2.3: Systemet ber om e-post fra kunde
- 2.4: Kunde får tilsendt link for å opprette nytt passord
- 2.5: Kunde lager nytt passord og logger inn, og returnerer til [steg 3](#)



Spørsmål?

Ikke nøl med å sende oss en mail:)

erikaesk@uio.no

sandefo@uio.no
