Notater til eksamen

* MMI
  + Brukerkvalitet for nettsteder
  + Intro til interaksjonsdesign
  + Interaksjonsdesign som prosess
  + Universell utforming
* UML
* Systemutviklingsmetodikk
* Testing
* Kontinuerlig Integrasjon
* Ytelsestesting
* Etikk
  + IEEE-CS/ACM Joint Task Force on Software Engineering Ethics and Professional Practices
  + NITOs etiske retningslinjer
* Web-ressurser GDPR/Personvern

MMI – Brukskvalitet for nettsteder

UML

Større og større krav settes i dag.

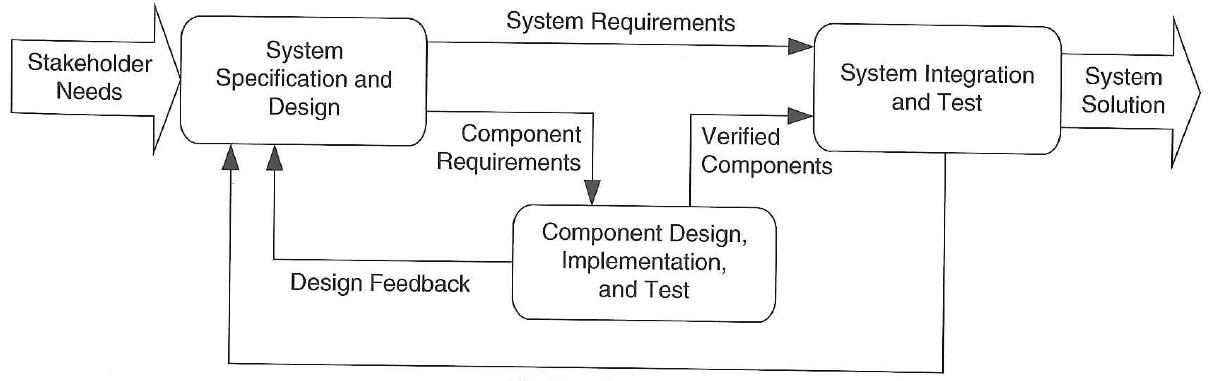


Figure 1: Den iterative prosessen

Gir en god pekepinn, og er lett å gå tilbake til for å endre på eventuelle ting.

Ved at alle interessenter er med å modellere vil et felles språk og forståelse av systemet komme fra.

**Model-Based System Engineering (MBSE)**

* Modellbasert utvikling, alternativ til tradisjonell dokumentbasert utvikling
* UML kan beskrive
  + Krav til komponenter og systemet som helhet
  + Systemets struktur
  + Oppførselen til og samarbeidet mellom komponentene i systemet
* En modell i UML er en abstrakt representasjon av aspekter ved systemet
* Har også en definert grammatikk

**Hvorfor modellerer vi?**

* Utforme et design
* Kommunisere et design
* Dokumentere et design

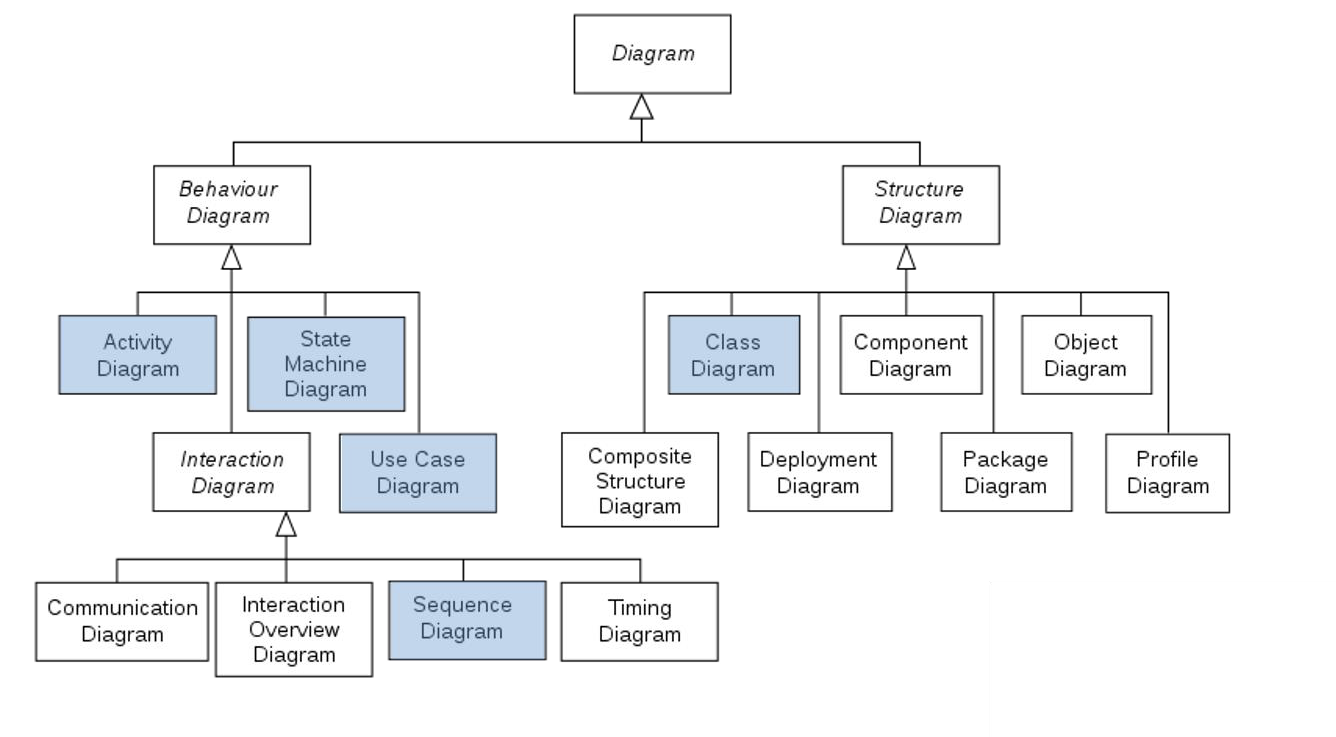
**Model Storming**

* Vanlig å bruke i smidige prosjekter
* Modellers gjerne i grupper
* Gjøres ofte for å løse et problem der og da

**Verktøy**

* LucidChart
* Gliffy

**Oversikt over diagrammene**



* **Use case-diagram**: gir en oversikt over brukstilfellene (use case) i et system. I tillegg beskrives hvert brukstilfelle tekstlig.
* **Aktivitetsdiagram**: beskriver hvilke aktiviteter som er involvert for å få gjennomført et brukstilfelle, hvilken rekkefølge de utføres i og eventuelle betingelser som er involvert.
* **Tilstandsmaskin**: brukes for å beskrive mulige tilstander til objekter i systemet og hvilke hendelser som utløser tilstandsendringer.
* **Sekvensdiagram**: brukes for å beskrive hvilke meldinger som sendes mellom objektene i systemet for å kunne gjennomføre et brukstilfelle, og i hvilken rekkefølge dette gjøres.
* **Klassediagram**: brukes både til å beskrive kode detaljert og til å beskrive de viktigste entitetene i et domene (domenemodell).

Systemutviklingsmetodikk

Noldus er igjen helt tilbakestående, ikke noe nyttig å notere.

Testing

* Tester for å validere at systemet
  + Responderer riktig på alle typer input
  + Yter godt nok, både ved stort trykk og over tid
  + Er brukervennlig nok
  + Møter kravene satt av interessenter
  + Kan installeres i riktig miljø
  + Ikke brekker ved endringer
* Tradisjonelt tester man i gitte faser, men i smidig prosesser prøver man å teste kontinuerlig, erstatter de tradisjonelle «tester» rollen man hadde tidligere
* Kontinuerlig integrasjon (CI)
  + Planlegge
  + Bygge kode
  + Kjøre tester
  + Måle og rapportere
  + Legge bygd versjon i testmiljø, evt. Produksjonsmiljø