Prosjektnr

Prosjektnavn
Kravdokumentasjon

Versjon <1.0>

I kravdokumentasjonen skal du beskrive funksjonelle krav til systemet som er laget. Begrunnelsene for valgene som er tatt hører generelt hjemme i Hovedrapporten.

Hvis dere bruker UP som metode bør dere bruke UP-mal dere har fått tidligere i studiet, for å beskrive kravene. Da trenger dere ikke dette dokumentet. Hvis dere derimot bruker en mer agil metode kan det være hensiktsmessig å bruke dette dokumentet som veiledning.

Kapittelet om «User Stories» er essensielt hvis dere bruker denne malen. Ellers kan dere ta bort kapitler som ikke er aktuelle for prosjektet. Om du finner det hensiktsmessig kan du eventuelt legge inn andre kapitler.

Dette dokumentet ligger som del av zip-filen http://iie.ntnu.no/fag/hpr/retnLinjerDataingenior.zip
Dokumentet er laget av Nils Tesdal.

Revisjonshistorie

Dato	Versjon	Beskrivelse	Forfatter
<dd/mmm/yy>	<x.x>	<detaljer>	<navn>

Det er meget aktuelt å levere inn dette dokumentet i flere omganger til oppgavestiller/veileder. Det noteres her, og også navnet på den som har gjort de siste endringene.

Innholdsfortegnelse

1. Introduksjon

I hvilken forbindelse er dette dokumentet skrevet? Hva er hensikten med dokumentet? Hva inneholder dokumentet?

2. User Stories

Lag user stories som dekker all brukerfunksjonalitet for systemet. En user story kan skrives på denne formen:

```
Som <rolle>
Ønsker jeg <mål>
Slik at <fordel>
```

Et eksempel kan være:

```
Som administrator
Ønsker jeg å opprette brukerkontoer
Slik at jeg kan gi nye brukere tilgang til systemet
```

For at user story'ene skal fungere godt som kravdokumentasjon bør dere også beskrive testbare akseptansekriterier for hver story. Disse kan beskrives med

- En punktliste med funksjonelle og evt ikke-funksjonelle krav til story'en.
- Scenarier.

Det mest naturlige er å velge å bruke enten scenarier eller en punktliste, men det er fullt mulig å kombinere disse.

En punktliste kan se slik ut:

- Hvis jeg er en administrator, så kan jeg opprette brukerkontoer
- Jeg kan opprette en brukerkonto ved å skrive inn følgende informasjon om brukeren: a. navn, b. epost-adresse, c. telefonnummer d. rettigheter (Power/Basic/None), e. status (Aktiv/Inaktiv), f. rapporterer til (fra en liste med aktive brukere)
- Jeg kan ikke sette «rapporterer til» til en inaktiv bruker.
- Jeg kan ikke sette rapporterer til slik at det oppstår en syklisk avhengighet.
- Systemet forteller meg at det har sendt en epost til den nye brukeren med et system-generert passord som må endres ved første gangs bruk.

Akseptansekriterier kan også beskrives med scenarier. Dette er en teknikk hentet fra testdrevet utvikling. Hvis du bruker TDD bør du derfor velge å bruke slike. Fordelen med dette er at scenariene er direkte anvendbare som akseptansetester, manuelle eller automatiske.

Scenarier kan skrives på denne formen:

```
Scenario: Tittel
Gitt [kontekst]
  Og [mer kontekst]...
Når  [hendelse]
Så   [resultat]
  Og [mer resultat]...
```

Under er et eksempel på en user story med to scenarier:

```
Som administrator
Ønsker jeg å opprette brukerkontoer
Slik at jeg kan gi nye brukere tilgang til systemet

```

```
Scenario: starte opprettelse av brukerkonto
Gitt at jeg er administrator
Når jeg trykker på «Opprett bruker»
Så skal jeg kunne legge en navn, epost, telefon, rettigheter, status og velge
hvilken annen bruker den nye «rapporterer til».
Og listen med brukere i «rapporterer til» skal kun være aktive brukere som
ikke fører til sykliske avhengigheter

Scenario: lagre ny brukerkonto
Gitt at all påkrevd brukerinformasjon er fylt ut
Når jeg trykker på «lagre»
Så skal brukerkontoen opprettes
Og jeg skal få beskjed om at det er sendt en epost til den nye brukeren, med
et systemgenerert passord som må endres
```

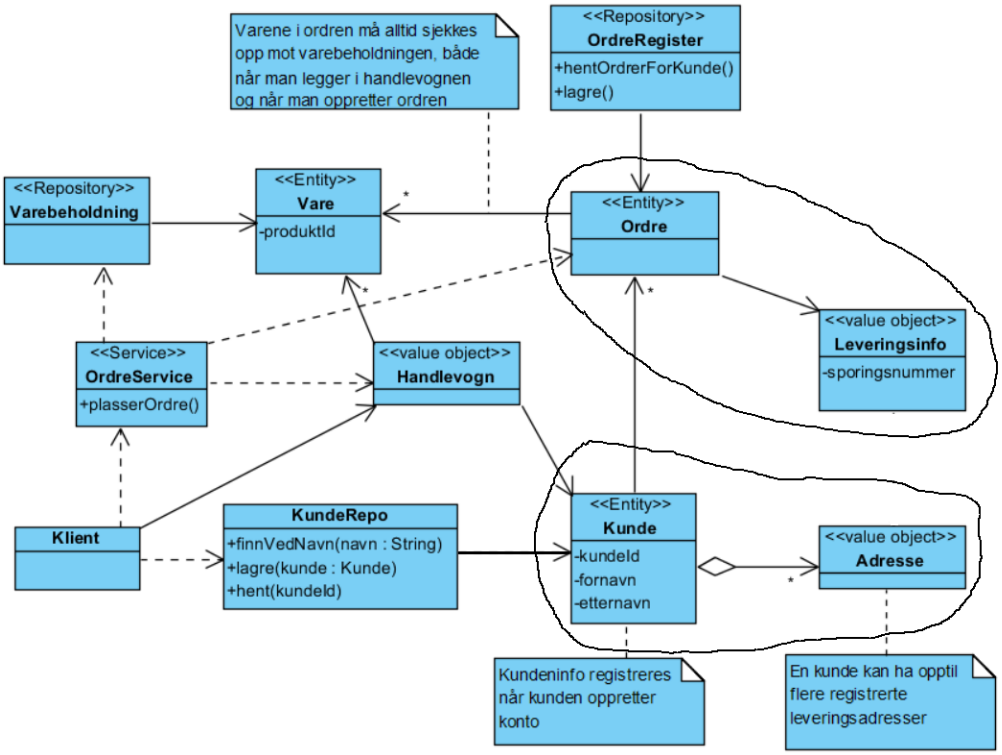
3. Domenemodell

Domenemodellen er en viktig del av kravdokumentet i UP. Det er også den mest sentrale komponenten i Domain-Driven Design. Lag gjerne en problem-domenemodell slik den er beskrevet i UP, eller velg en DDD-stil domenemodell hvis dere kjenner til dette.

Domenemodellens funksjon er å få utviklere og domeneeksperter til å enes om en felles forståelse av problemdomenet. Utviklerne må forstå domenet de skal jobbe med og alle involverte må snakke om det samme når man bruker begreper fra domenet. Dette er viktig for å unngå misforståelser underveis.

Bruk domenemodell hvis prosjektet opererer i et ikke-trivielt problemdomene og/eller for å oppnå best mulig kommunikasjon med oppdragsgiver.

Under er et eksempel på en domenemodell (i DDD-stil):



3.1 Sekvensdiagrammer

I Domain-Driven Design prøver man gjerne også å teste domenemodellen ved å ta for seg et scenario og iscenesette domeneobjektene i et sekvensdiagram for å se om klassene i domenemodellen er hensiktsmessige. Deretter kan man gjøre endringer i domenemodellen og teste på nytt etterpå. Hvis man gjør dette kan man ganske enkelt vise progresjon ved å ta bilde av sekvensdiagrammene underveis. Disse må ikke være en del av kravdokumentet men kan eventuelt være vedlegg til rapporten.

4. Prototyper

Hvis du utvikler en applikasjon hvor grensesnittet er en viktig del bør du vurdere å bruke en eller annen form for prototyping for å beskrive krav. Tre mulige måter å gjøre dette på er:

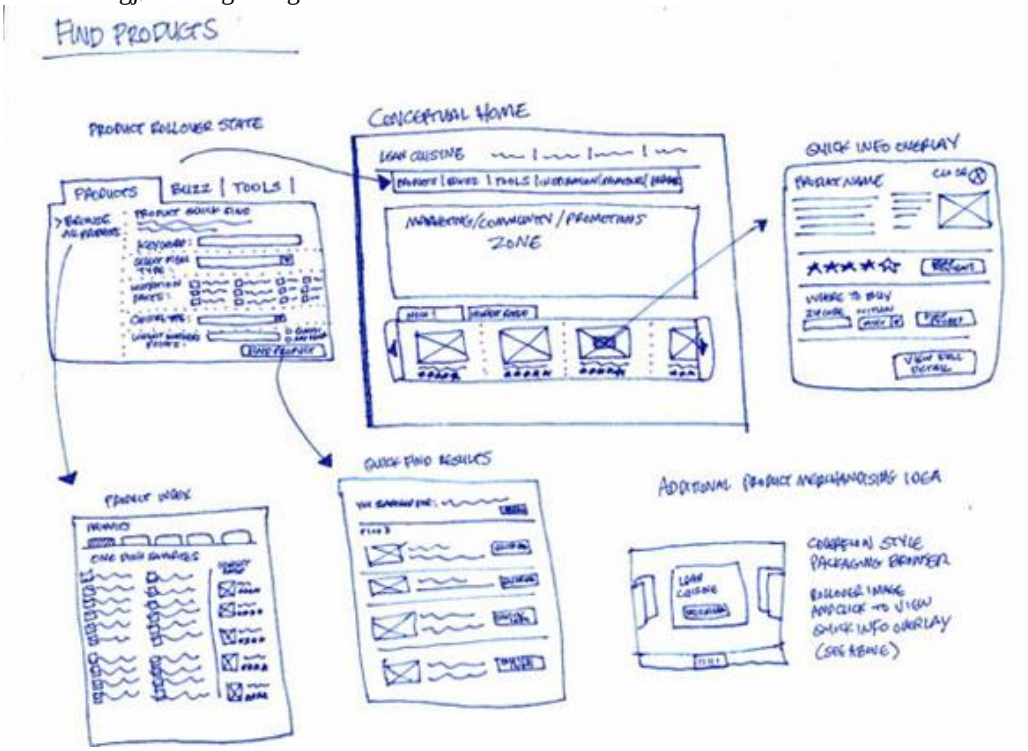
- HTML-prototyper
- Wireframes
- Storyboards

4.1 HTML-prototyper

Virkelighetsnære HTML-prototyper er en god måte å beskrive krav på. Denne formen for prototyping tar tid, men resultatet kan kanskje brukes i det ferdige produktet og det er lett å kjøre brukertester på dem. En ulempe kan være at det blir tyngre å gjøre endringer etter kjøring av brukertester. Du trenger ikke vise prototypen i dette dokumentet. Vis heller en lenke til den.

4.2 Wireframes

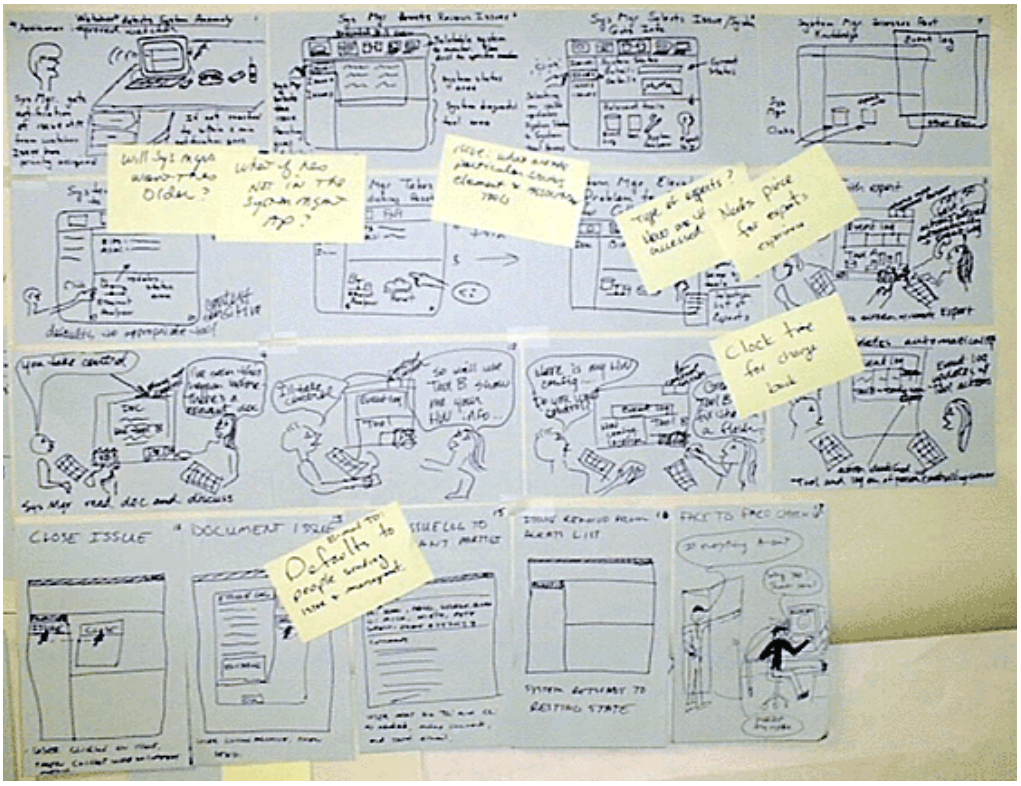
Wireframes er enten frihåndstegninger eller skisser laget av et av mange wireframe-verktøy som finnes på nett. Med nettbaserte wireframe-verktøy kan man lage klikkbare wire-frames som gjør at man også kan navigere mellom skjermbildene. Dette gjør dem godt egnet til å brukes i brukertester.



Hvis du har laget wireframes for hånd kan du lime de inn her. Hvis du har brukt online-verktøy kan du eventuelt vise lenker til dem.

4.3 Storyboards

For å lage storyboards tar man utgangspunkt i et user story og visualiserer den ved å bruke wireframes, snakkebobler og tekstlige forklaringer på brukeraktivitet i et slags tegneserieformat. Dette kan vise mye informasjon på ett sted.



5. Referanser