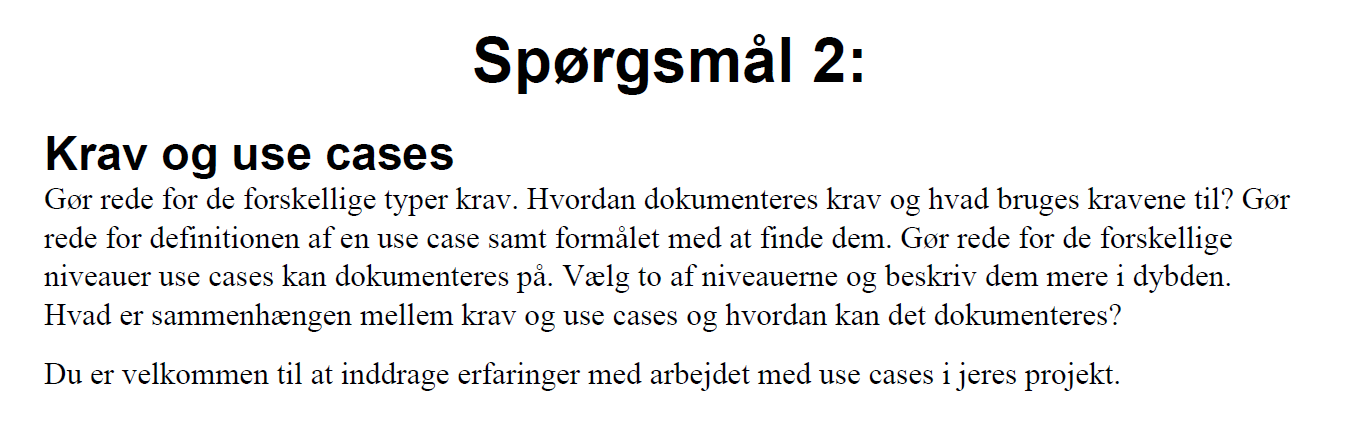
Spørgsmål 2



Så hvad bruges krav til? Krav er at føre udviklingen mod det “Rigtige” system, det system som vores case gerne vil have.

Hvordan laves krav? Vi tager ud og analyser en virksomhed, som har et system de vil lave, vi ser på hvad de gerne vil have, laver en case og ud fra dette laver vi en kravliste der kort kontant beskriver hvad systemet skal kunne.

Hvordan afdækker vi krav? Vi finder funktionelle og ikke funktionelle krav, og derefter prioriterer vi dem via en system vi har lært der kaldes MOSCOW

Moscow bruger vi 4 ord til at definerer hvor vigtig hvert et krav er.

I henhold to Moscow har vi:

* M - Must have.
* S - Should have.
* C - Could have.
* W - Would like to have.

De funktionelle krav er det man skal kunne gøre med systemet. Et eksempel på dette kan være: “Systemet skal kunne håndtere flere betalingsmetoder.”

De ikke funktionelle krav er de ting der kan skabe kvalitet eller egenskab ved systemet. Et eksempel på dette kan være: “Systemet skal være brugervenlig og en erfaren bruger må maks tage x minutter om at udføre et arbejde.”

Når man så har lavet en kravliste begynder man at lave et use cases baseret på dette, dette oftest laves via en use case diagram og senere en use case beskrivelse.

En use case er en måde at beskrive funktionaliteten i et system.

Et use case diagram, har du aktører, som er forbundet til bestemte use cases, disse use cases er beskrevet via et nummer for lettere identifikation, og er skrevet efter en bestemt måde der forklarer kort hvad de er. De skrives med verbum som er stærk og et navneord for det område det omhandler. Et eksempel på dette kan være: “Register sales.”. Ofte er der sammenhænge mellem use cases, og det ville beskrives med en extend eller include. Hvis en use case er included så skal den use case gøres hver gang den use case den indeholder gør noget. Ved extend så kan den use case kun gøres hvis en bestemt betingelse er opfyldt fra den use case extend pilen kommer fra.

Det næste punkt ville være use case beskrivelser. Her tager du nogle af de sværere use cases og beskriver den process der er mellem aktøren og systemet helt ned til detaljer. Dette gør dette lettere for programmøren af forstå hvordan interaktionen mellem bruger og system skal foregå.

Her bruger vi en standard skabelon, Der har use case navn, hvordan use case bliver starter, en kort beskrivelse, de aktører der er, relaterede use cases, interessenterne, de precondition og postcondition der skal til, samt flowet af interaktionen.

Oftest til sidst laver man en krydstabel for at sammenligne krav med use cases for at se om der mangler nogen. Eller flere krav bliver opfyldt af samme use case.