 Referat fra vejledermøde #5

**Dato:** 04-03-2016

**Tid:** 12:30 - 13:00

**Sted:** 312E

**Fremmødte:** Nicklas, Anders, Christian, Mette, Rasmus, Troels

**Udeblevet med afbud:**

**Udeblevet uden afbud:**

**Dagsorden:**

**Kommentarer fra mødet:**

**Sprint Review**

De to user stories er implementeret mht. databasen (hvor der lige nu er et konsolvindue til at vise det) men interfacet er ikke blevet implementeret endnu – skulle det gerne blive over weekenden

Når vi viser hvad vi har skal vi sørge for at der ikke står en masse ting som der ikke skal (såsom ting der er implementeret som ikke skal være implementeret endnu…)

Vi skal sørge for at tænke på at der skal tilføjes en vare til en forretning – og ikke til en database

Ligeledes med finde den billigste forretning for en vare

Når der holdes sprint review vil product owner gerne se at de ting der står på taskboardet kan udføres – så han vil ikke se noget i en database men noget for hans forretning

Vi skal huske at få det i review over i done – i review er de stadig pending

I vores næste sprint skal vi huske på de ting vi snakkede om sidst (sprint goal, det med hvordan taskene skal opdeles, osv.)

Vi skal sørge for at der er tekniske opgaver til alle næste sprint – ikke bare til én. Det er hverken sjovt for dem der sidder og ikke laver tekniske opgaver og heller ikke smart at alle ikke har noget at sige de har lavet til eksamen

**Hvad der er lavet**

Se første afsnit i sprint review

**Hvad der skal laves**

Vi skal have færdiggjort det nuværende sprint og planlagt det næste sprint henover weekenden

**Processen**

Andre krav det her semester til at dokumentere processen – det er det vigtigste det her semesterprojekt (men Troels tjekker lige op på det)

**Spørgsmål**

Ikke funktionelle krav:

Svære at formulere så de er testbare og så man kan være enige om at de er overholdt

Bruge en standard fx: Three click / rule of thumb – andre ting til user experience – noget til oppetid og responstid

Vi skal finde nogle guidelines til at støtte os op af

Svært at måle sådan noget som oppetid, for det er svært at måle hvis ting ikke går ned

Sekvensdiagrammer / flowcharts

Troels kan godt lide at en boks i flow chart har ét entry og ét exit point

Flow: Ikke taget stilling til detaljen – fortæller flowet i det der skal ske

Sekvensdiagram: Mere detaljeret – fortæller mere hvilke roller der er i spil og hvilke metoder der kaldes

Struktur beskrives godt i klasse og package

Adfærd beskrives godt i sekvens og tilstand

Troels kan ikke så godt lide flowcharts – sekvensdiagrammer beskriver det bedre med roller og de ting der sker – de udtrykker lidt mere

Omkring klassediagrammer og databasen:

DB context skal kun med på klassediagram hvis det er vigtigt at vise. Evt. kun lave ét diagram hvor den kan ses på og så ikke vise den igen på nogle andre

ERD diagrammer viser databasestrukturen så skal ikke vises på klassediagrammet

Klassen database skal hedde noget andet – det er en der arbejder på databasen – evt. hedde storemanagement til den der arbejder på databasen

Vi skal ikke kun have én database til både storemanager og user

Vi er alt for teknologispecifikke – skal vi stoppe med ☺

**Generelt**

Memory database kan evt. bruges til at teste de forskellige funktioner til databasen

Vi er for afhængige af databasen – skal laves om på en eller anden måde så vi er afhængige af et interface i stedet og derved kan indsætte en fake database til at teste på – bruge mock til at se om tingene er blevet kørt

Kaldes et repository pattern (tror Troels)

Huske at det skal deles ud i to forskellige projekter (eller tre hvor den tredje er common)

Hvad sker der hvis der er en der er ved at opdatere databasen mens en anden er ved at udlæse fra databasen? Bare den der kommer først? Eller skal der gøres noget?

**Næste møde:**

Med vejleder: mandag d. 14. marts 2016 klokken 09.00-10.00

**Kage:**