Jesper Petersen: [cph-jp284@cphbusiness.dk](mailto:cph-jp284@cphbusiness.dk)

Claus ……

Link til github: <https://github.com/CK2800/Fog>

Fog carport

19-12-2018

# Process:

## Arbejdsprocessen faktuelt:

**Hvilke sprints der var, og hvilke user stories der blev arbejde med.**

|  |  |
| --- | --- |
|  | Sprint 1: Der blev lagt vægt på PO-møde at Fog (Ronnie) godt kunne tænke sig noget ”*business value*” med at man som kunde vil kun afsend en carport forespørgslen og som Fog-medarbejder vil kunne åbne og se forespørgslen som kunden har opret i systemet. |
| **Sprint 2**: Der blev fik vi taget 5 user story ind hvor vi senere fandt ud af det var alt for mange i forhold til at vi kun opnå at lave dem færdig.  Dermed var vores mål, at giv Fog mulighed for, at beregn stykliste for en carport og dermed vil kun også kun se hvor meget kunden kommer til, at betale for carport hvis man vælgere at oprette forespørgslen. Så kunden vil kun se hvor meget det kommer til, at koste at opbygge den carport.  Hvor man har angivet de mål som man har oplyst.  PO-møde (Ronni) vil som udgangspunkt også kun se tegning af carport som er blevet opret af kunden. |  |
|  | **Sprint 3**: Vi har måtte gør vores Sprint 3 mindre i forhold til Sprint 2 efter som vi stadigvæk ikke er færdig med Sprint 2 på daværende tidspunkt.    Vi vil arbejde med vores SVG tegning så Fog kunne se en tegning af carporte. Vi fik lavet vores carport område færdig med Carport konfiguration og tagberegning. |
| **Sprint 4**: Efter møde med PO d 30 nov. Forgår dialog på, at vi vil gerne ha lavet nogen af de mange små huler som vi har haft igennem 2, 3 og 4 sprint.  Derfor har vi måtte giv det sidste sprint 4 to dage ekstra for, at kun få lavet nogen af de mange fejl som vi har haft undervejs. |  |

**Hvem der var scrum master i hvilke dele af projekt perioden. Giv gerne nogle eksempler på hvad scrum master gjorde i udvalgte sprint.**

Jesper og Claus har begge været scrum master. Dermed har Claus og Jesper været Scrum master i en uge og der efter rundt sådan, at det var en ny scrum master.

Det har været med til, at sikker at begge partner fik lov til, at være scrum master.

Vi har gjort alt for, at nå dvs. ting som skulle være færdig til tiden.

Det kun fx være da vi arbejdede med SVG tegning gik den anden partner ind og hjælp ham som havde problemer med at få løst fejl/problemet.

Men så vi dermed fik fundet en optimal løsning på problemet. Så den ene ikke fik brugt alt for mange timer på ingen ting. Så om morgen forklarede den ene af os at man havde problemer med SVG tegning så gik den anden ind og hjælp med at få løst problemet sådan, at begge partner kunne komme videre med andre opgaver.

**Et eksempel på et af PO-møderne, hvad der var planlagt fra jeres side, og hvordan det gik.**

Hver fredag har vi afholdt PO-møderne med Ronnie.

Det første PO-møde forgik på, at vi havde lavet vores ”User story” og dermed diskuteret vi om hvilket user story som vi skulle arbejde med til vores næste PO-møde.

Så som udgangspunkt har vi været forberedt på dvs. PO-møder med Ronnie.

De gange vi har været bagud med opgaver har vi som udgangspunkt sagt det til vores PO-møde hvis vi har været bagud i forhold til vores opgave.

Det har så gået fra Sprint 2 frem til Sprint 4. Hvor Sprint måtte bliv 2 dage længere. Det er efter som vi ser at der kommer en masse små huler som vi blev nød til, at få luk ned for så vi ikke til sidste skulle vende tilbage de dvs små huler som vi evt ikke fik lukket.

**Hvordan i afholdt jeres daglige standup møder.**

Vi har afholdt vores daglige stand-up med at hver dag klokken 8:30.  der snakker vi om hvad vi har lavet dagen forinden det kunne fx være at man er blevet færdig med SVG tegning eller noget helt andet.

Hvis man havde problemer så tog vi det om morgen fordi så hvis den ene var låst fast på problemet så kun man ligesom ikke komme videre.

Så kun man også sikker sig at alle i gruppen kun komme videre med deres ting. Så tiden blev brugt tid på ingen ting.  
  
Derfor sikker vi også at alle i gruppen kun få lavet så meget som overhovedet muligt.

Det sikker vi også hver morgen og når vi begge havde holdt frokost pause. Så snakkede vi sammen igen og hørt hinanden hvordan det gik med opgaven. Og hvis man havde fået problemer ved frokost tiden så kun vi gå ind og hjælpe hinanden. Hvis der var behov for det.

Problemet blev løst via Teamviewer og snakke sammen igennem telefonen.

**Hvornår i holdt retrospectives.**

Som udgangspunkt holdt vi det hver fredag. Fredag mener vi bestemt, at det vil giv mening i forhold vores tilbage blik med ugen.

Dermed har gjort at hver fordi har vi gennem gået vores tænker og hvad vi kan gør i forhold til næste uge. Det er som udgangspunkt når sprint har været færdig så har vi taget møde.

Det er for vi har ugens opgave i friskhed. Så vi kan huske tilbage på hvad vi har gjort af overvejelser i forhold til dvs ting.

# Arbejdsprocessen reflekteret

**Om scrummaster rollen fungerede, hvilke problemer I så i den, og hvad I gjorde for at rette op på det.**

**Hvad der var de væsentligste emner på jeres retrospektiv møder**

Vi har løst de problemer med det samme.

Nye taske havde ikke tid til, at tildel dem point.

**Om I havde problemer med at nedbryde user stories i tasks**

I starten havde vi lavet et godt forarbejde i forhold til at nedbryde dvs. user stories i taske. Dermed vidste vi også godt hvad vi skulle i de tasks områder.

Men undervejs fandt vi ud af der kom flere og flere taske frem som vi slet ikke havde set.

Derfor til sidste måtte vi bare tag dem som var super vigtig i forhold til opgaven fordi ellers kun vi bliv ved med at kig frem og tilbage på hvad vi kun tilføj af ting til tasks området i forhold til den user stories vi skulle lave.

Så hvis vi havde kig på om det var vigtig fra starten af. Så er jeg også sikker på, at vi kun nå langt mere end hvad vi havde kun.

Dermed finder vi første ud af det meget sidste i sprint i forhold til hvad vi havde af tid til at kunne nedbryde tasks til user stories.

**Om I var spot-on med jeres estimeringer**

I starten fik vi sat dvs. point på de områder. Dermed mener vi bestemt selv, at vi fik sat dem på en fair måde og dermed brugt vi også godt med tid på, at få gjort det.

Vi fandt dog senere ud af vi manglede nogen ting. Det medført i, at vi ikke fik sat tid af til, at angiv de point til dvs. områder.

Det medført at vi første meget sent henne i sprint 4 måtte bruge et par timer på, at tildele dvs. point på de vi manglede.

**Om der var problemer med vejledningen og PO møderne**

De problemer som frem kom det var at hvis vi fx havde Ronnie om onsdag og fredag. Og Det samme med Thomas.

Der hvor vi kun se problemet det var ved at om Fredagen ugen inden. Der havde vi snakket med Ronnie om nogen ting som vi vil ret op på til næste uge og så finder vi ud af Ronnie fx ikke er på skole og så skal vi ha Thomas.

Hvilket gjord at det er svært at fortælle Thomas hvad Ronnie fik sagt til og dermed gør det svært at vores PO ikke er der fx ugen efter.

**Hvor langt inde i processen I fandt en rytme der var produktiv**

Der hvor vi fandt de gode rytmer i Fog-opgaven var bestemt i sprint 2 og 4.

Sprint 2:

Sprint 4: Der var vi begge to indstillet på, at få lavet nogen af de mange huler færdig og dem gjort opgaven så godt som færdig. Dermed mener vi bestemt at Sprint 4 var klar den bedste af dem alle sammen.

Der fik vi virkelig arbejdet hårdt for, at opnå at få lavet tingene færdig til tiden.

**Andre elementer der har at gøre med at forsøge at arbejde i et scrum team**

**Hvilke informationer gemmes i session**

Vi gemmer Email, Navn, Rank, id og postnr.

Dermed er vores tænker omkring det er at hvis man vælgere at updater ens email, navn eller post så vil man skulle log ind igen.

Det er for at udengå, at skulle fjerne session fra det gammel ”data” i session og så der efter opret en ny session med de nye værdier.

Derfor mener vi bestemt, at den bestemme måde er at send folk over til log ud og derefter skulle log ind igen.

Vi mener bestemt ikke at det er nogen **god ide**, at gemme password i session. Og Vi ser heller ikke nogen grund til, at skulle gemme password i session.

At indholde plaintext som adgangskode overalt i enhver form er normalt en dårlig ide. Vil aldrig håndtere ”kundens” hemmelighed.

**Hvordan håndterer man exceptions.**

**Hvordan håndtere man bruger rank?**

Vi har valgt at håndtere vores bruger rank på en måde som skal sikker at der altid vil være 1 admin tilbage. Man vil ikke selv kun gå ned i rank men dermed vil man selv kun tilføj admin rank til andre bruger.

**Overvejelser**:

De overvejelser vi har gjort også det er at vi sikker også imod, at man ikke kan gå ned i rank hvis man nu er admin selv. Det er taget udgangspunkt i at Martin ikke vil kun gør sig til alm bruger ved en rank med mindre, at en anden medarbejder gør ham til alm bruger.

Dermed så vil den anden medarbejder giv tildel admin rank til de bruger som han mener skal ha admin rank i systemet.

**Sikkerhed i det**:

Det er som udgangspunkt, at der hele tiden skal være EN og man vil ikke vil kun undgå, at der slet ikke er nogen form for admin i systemet.

**Hvorfor ikke blive admin ved oprettelse?:**

Vi mener bestemt ikke at det skal være muligt. Fordi så vil alle kun blive admin ved starten når man opretter som bruger på siden.

Vi skal sikker også at Fog’s medarbejder kun har adgang til den data som bliver oprettet ved forespørgelse til carport.

**Hvordan man på har valgt at lave brugerindput validering**

**Forespørgelse**:

Der sikker vores også, at carport skal være min 300 længde/bredde og max må være 900. Dermed må Hældning kun være max 40. Det vil sige at den kan gå fra 0 til 40.  
Dermed sikker vi også at skur længde/bredde skal være min 150 og max 900.

Kommentar har vi valgt at lad være efter som hvis kunden ikke ønsker at skrive kommentar til forespørgelsen.

**Opret bruger:**

Der sikker vi også hele vejen at man ikke kan opret bruger før man har udfyldt hele området. Det er lige fra e-mail, adgangskode, navn og telefon og post er valgt.

Alle de information skal være opfyldt før at man vil kun opret sig som bruger på siden.

Vi sikker også at bruger kan opfylde de fleste af de værdier.

**Log ind / Glemt adgangskode**

Der sikker vi også kun e-mail som udgangspunkt fordi hvis man vælgere at sige at man har glemt adgangskode så skal den ikke kræve at man også har password.

Så derfor har vi valgt at sikker også at man skal ha tilføjet e-mail som udgangspunkt.

**Hvilke brugertyper der er valgt i databasen, og hvodan de er brugt i jdbc**

# Status på Implementation:

**Hvor langt er vi nået med implementationen:**

* Opret bruger/konto
* Log ind & log ud
* Afsend/Updater/opret/åbn carport forespørgelse
* Administrer varer
* Stykliste beregner
* Nulstill brugerkonto
* Administrer brugerkonti
* Validering af brugerinput
* Hent/vis tegning
* Se/udskriv tegning

**Hvilke implemenatation fik vi ikke gjort?:**

* Updater/opret af materialer
* Tilføj den enkelt forespørgelse til kunden.
* Tilføj forespørgelse som order til kunden.
* Leverprisen for carport
* Send beskeder til kunden og svar på kunde spørgsmål.
* Administrer kunde
* Besvare forspørgelse
* Status på order

# Test

**Hvilke klasser er testet**

**Data:** userDAO, RooftypeDAO, MaterialDAO og CarportRequestDAO.

**Logic**: CalculatorIntegration, CalculatorUnitTest og DTOunitTest.

**Calculators**: CaluculatorIntegrationTest.

**Hvilke metoder der er testet**

Hvert område vi tester der åbner vi for database få forbindelse og når test filen er færdig. Så bliver der lukket ned for database forbindelse.

**Data**:

**UserDAO**:

* (testCreateUser) - Vi tester at man vil kun opret bruger i databasen og dermed når man opretter den enkelt bruger i databasen. Så skal den sende et bruger id tilbage som skal være med til, at kunne fortælle at man gerne vil ha slettet bruger med dette her id som er blevet tildelt ved oprettelsen.

**RooftypeDAO**:

* (testGetRooftypes) - Vi tester at vore tagtype ikke indhold ingen ting og der er mere end 0 i databasen.

**MaterialDAO:**

* (testCreateMaterial) - Vi tester om det er muligt at opret en material i database med nogen bestemme værdier som har skrevet i testen.
* (testGetSingleMaterial) Vi vil gerne hente en bestemt materiale i database.
* (testGetMaterials) Vi henter alle vores material ud i en liste.
* (testGetMaterialsFails) Vi lukker for forbindelse før den skal hente alle material i databasen.

**CarportRequestDAO**:

* (GetCarports) - Skal hente alle forspørgelser i databasen.
* (getCarportsFails) - Skal fejl ved at hente alle forespørgelser i databasen.
* (getCarport) - Skal hente en bestemt carport foresprørgelse i database ude fra hvilken id der kommer.
* (updateRequest) - ????
* (createCarportRequestFails) - Her skal vi opret en forespørgelse men det kommer til, at fejle pga der bliver lukket for forbindelsen.
* (testOpretForespoergselMedSkur) - Der bliver oprettet en forespørgelse med skur.
* (testOpretForespoergselUdenSkur) - Der bliver oprettet en forespørgelse uden skur.
* (testHentAlleForespoergsler) - Der bliver oprettet en forspørgelse og der efter

**Dækningsgrad af dine tests for de valgte metoder og klasser**

# Navigationsdiagram

Oversigts diagrammet. Hvis det bliver for stort må man dele det op. Men det er vigtigt at der er et overordnet diagram.

Hvis man har benyttet sig af en “fælles navigations bar” i toppen af alle sider skal man forklare det.

Hvis nogle sider kun kan nåes af nogle brugere (dem der har konto, dem der er logget ind, dem der arbejder i butikken,…), så skal det fremgå.

Navne på jsp sider skal fremgå, og hvilke servlet der bringer en fra den ene side til den næste.