# Metode

Af: Lasse

Vi havde i gruppen en diskussion om hvilket framework der ville passe bedst, til den opgivet projektopgave. Vi har på vores 3. semester arbejdet meget med SCRUM, og det ville være et naturligt valgt at bruge dette. Vi har udover SCRUM også kigget på et andet framework som extreme programming, dette fandt vi dog ikke attraktivt, da det store fokus her besidder på pair-programming, og det er et krav i opgaven at alle skal deltage aktivt, og der skal navn på hvad hver person har bidraget med. Ved XP ville der komme to navne på alt hvad vi har lavet, og det vil derfor ikke give et indblik i hvad hver person har bidraget med til gruppen. Vi har derfor valgt at benytte SCRUM, da det giver os en masse brugbare værktøjer.

Udover SCRUM vil vi også benytte os af Unified Process metoden, til at hjælpe med at skabe en rød tråd igennem hele arbejdsprocessen. Vi vil også benytte UML notations diagrammer i form af fx Domæne model, SSD, SD, m.v.

## Unified Process and SCRUM

Vi vil som tidligere nævnt benytte SCRUM og Unified Process sammen, for at hjælpe med overblikket at hele projektet. Vi vil benytte de fire faser Inception, Eleboration, Construction og transition. Grunden til vi vil benytte både SCRUM og UP er, at SCRUM ikke har nogen definition for de discipliner der er i UP. Vi har i vores projekt brug for at få en bredere forståelse af domænet, samt hvilke krav der er til løsningen af opgaven, det kan business modeling og requriments hjælpe os med. Vi vil ikke benyttet os at iterationer, da det giver mere mening for os, at bruge sprints. På denne måde for vi et tidsinterval samt nogle krav der skal overholdes, i form af fx definition of done, før en given tasks kan defineres som færdig lavet.

## Unified Process

### Inception

Vi vil starte ud med inception fasen for at få en bedre forståelse af Domænet, samt finde frem til alle de krav der er opsat i opgaven. Vi vil derfor starte med at fokusere på de User stories der har fokus på disse opgaver, og andre relevante opgaver der måtte være, som fx at få skrevet problemformulering og mål.

### Eleboration, construction og Transition

Da vi bruger sprints frem for iterationer, giver det god mening at disse tre er forbundet. Vores User stories er lavet på den måde, at analyse og design er forarbejdet på en user story, der skal laves inden implementation kan finde sted. Så i et sprint vil der både være analyse og design samt implementation før en user story, kan blive betragtet som udført. Derudover skal vores definition of done overholdes, som indeholder krav til testning. Deploy af software vil foregå løbende.

## SCRUM

En af de ting der er lagt op til i opgaven er, at der er mulighed for at der kan komme ekstra krav til features løbende. Hvis det skulle være tilfældet, kan vi ligge disse features ind som en user stories i vores backlog, og derefter finde frem til, hvilket sprint de skal med i, alt efter hvor stor en prioritet de har.

### Sprints

SCRUM er som tidligere nævnt delt op i sprints, vores projekt er på 3 uger, og vi har derfor delt det op i tre sprints af en uge af gangen. Vores sprints er tidfast og har en klar definition på hvad der skal være færdig ved udgangen af sprintet.

### Roller

Vi har tre roller opfyldt i vores scrum team, vi har en Product owner som er Klaus Nørregaard, han har stillet opgaven med de krav der skal opfyldes, og det er ham vi kan gå til ved eventuelle spørgsmål til projektet. Vi har en Scrum master som er Lasse Meldgaard, han står for indkaldelse af møder, samt hjælpe med at udbedre eventuelle impediments der måtte være for Development teamet. Development teamet er den sidste rolle der skal udfyldes, og denne består af alle fire medlemmer i gruppen. Dette team står for at færdiggøre de enkelte sprints, indenfor de fastsatte rammer.

### Scrumboard

Vi bruger [www.trello.com](http://www.trello.com), til håndtering af vores Scrumboard, hvor vi har listet alle vores User stories op.  
En af de store fordele ved et scrumboard er at det hjælper os med at holde styr på vores user stories samt task, så vi ved hvad der skal laves i sprintet, og hvem der laver hvad. Det sikrer også at vi hele tiden holder alle i gruppen i gang, da de kan påbegynde en ny opgave, ved blot at se hvilke tasks der er tilbage på boardet.

Vi har derefter brugt planing poker til at tidsangive vores stories, det har vi gjort for at få et så præcist estimat som muligt, og så alle kan klare en opgave inden for den planlagte estimering. Vores scrumboard består af 6 kolonner, som er Backlog, To Do, Work in progress, Test af kode, dokumentation korrektur og en Done.

Backloggen indeholder alle de User stories vi har fundet frem til, de ligger i en prioriteret rækkefølge efter product owners ønske. I vores tilfælde skal vi aflevere et færdigt projekt, og ikke med løbende afleveringer efter hvert sprint, til en kunde. Vi har derfor valgt at prioritere dem efter hvordan rapporten skal opbygges, samt hvordan applikationen skal bygges op. Det er ikke alle vores stories, der tager udgangspunkt i brugeren, da vi har en rapport der skal laves. Vi vil gerne have alle vores opgaver på scrumboarded, derfor har vi to User stories der tager udgangspunkt i os som udvikler, der sikrer vi får dokumenteret alt i rapporten.  
To Do er vores sprint backlog, hvor vi har opdelt vores user stories i tasks, som vi igen har tidsangivet ved hjælp af planning poker. De tasks vi har i denne kolonne skal være færdige efter sprintet er ovre.  
Work in Progress, er de tasks som er under udvikling af en fra teamet.  
Test af kode, er hvor vi flytter en task hen når den er klar til at blive testet, vi har et krav om der som minimum skal være whitebox testing af alle tasks, før de kan flyttet over i Done. Det samme gør sig gældende på dokumentation, hvor der i kolonnen ”dokumentation korrektur”, hvor der skal læses korrektur, på den skrevne dokumentation. På den måde får vi spottet eventuelle fejl og mangler inden, vores arbejde bliver erklæret Done, som er den kolonne der det bliver flyttet i når alle krav er overholdt. De krav der skal overholdes har i defineret i vores Definition of done. De fire krav er:

1. Dokumentation er skrevet

2. Kode er testet (White box)

3. Dokumentation læst igennem

4. Alt er i versionskontrol

#### Møder

Vi kommer i vores team til at holde 3 forskellige slags møder daily scrum meeting, sprint planning meeting og sprint retrospective. Vi holder hver morgen vores daily scrum meeting, det foregår udenfor arbejdsrummet, hvor computere kan være en distraktion. Til mødet er det scrum master og development teamet der deltager, der er sat 10 minutter af til mødet. Der vil hver morgen blive stillet tre spørgsmål til alle deltagere til mødet:

* Hvad lavede du i går?
* Hvad skal du lave i dag?
* Er der nogen hindringer i vejen for arbejdet?

På denne måde får alle et indblik i hvor langt man er kommet i det nuværende sprint. Skulle der være nogle hindringer for arbejdet, skal disse løses.  
Vi holder vores sprint planning meeting hver mandag morgen klokken 8.15. Her finder vi de User stories der skal i vores sprint backlog for det kommende sprint. Vi laver ikke et sprint goal, da vores product owner, ikke vil deltage på disse møder. Det sidste møde, sprint retrospective, holder vi hver fredag klokken 13.30, her vil vi igen fokusere på 3 spørgsmål, disse vil blive stille i åbent forum og ikke til hver enkelt deltager.

* Hvad skal vi begynde at gøre?
* Hvad skal vi stoppe med at gøre?
* Hvad skal vi fortsætte med at gøre?

Efter der er kommet ideer/svar til disse spørgsmål, vil vi i samarbejde finde til enighed om, hvilket af disse ting der skal ændres eller bibeholdes. På næste retrospective vil, der blive kigget på om det er lykkes at ændre på de pågældende ting.