# 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 valg at bruge dette. Vi har udover SCRUM også kigget på et andet framework, 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 individuelt. 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 at danne overblik over hele projektet. Vi vil benytte de fire faser Inception, Eleboration, Construction og transition. Grunden til vi vil benytte både SCRUM og UP er, at vi i vores projekt, har brug for at få en bredere forståelse af domænet, samt hvilke krav der er til løsningen af opgaven, det kan business modeling og requriments faserne hjælpe os med. I SCRUM er det dog også meningen, at man planlægger i starten af projektet, vi vil blot benytte disse redskaber fra UP til og hjælpe os med at danne et samlet overblik, da SCRUM ikke har en decideret metode til dette.  
 Vi vil benytte os af udtrykket sprints og ikke iterationer. Disse sprints kan hjælpe os med at danne et tidsoverblik, samt give os nogle krav der skal overholdes, i form af fx en definition of done, der skal overholdes 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 en problemformulering og få opsat nogle mål.

### Eleboration, construction og Transition

Da vi bruger sprints frem for iterationer, giver det god mening at disse tre er forbundet. Vores User stories bliver 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, som en user story bliver færdig, dette gør vi ikke, fordi vi har krav fra en kunde omkring agil udvikling, men for at hjælpe os med at få et overblik over hvor langt vi er kommet i projektet.

## SCRUM

En af de ting der er lagt op til i opgaven og af product owner er, at der er mulighed for at der kan komme ekstra krav til features løbende i projektet. Hvis det skulle være tilfældet, kan vi ligge disse features ind som 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. I starten af vores sprints vil vi vælge de opgaver, som har højest prioritet for at få lavet projektet, i en rækkefølge der kan være med til at skabe en rød tråd igennem hele projektet.

### 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 et gældende sprint, 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 vil desuden bruge planing poker til at tidsangive vores stories, det gør vi 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. Derudover vil vi tage fat i nogle af de store problemstillinger først, fx hvordan vi kan benytte Xenas api. Det er ikke alle vores stories, der tager udgangspunkt i brugeren, da vi har en rapport der skal laves og vi vil gerne have alle vores opgaver på scrumboarded. Vi vil derfor have nogle 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, hvis de ikke er blevet lavet, vil dette blive taget op i vores sprint retrospective møde.  
Work in Progress, er de tasks som er under udvikling af en fra teamet, vi sætter navne på tasks, når de bliver flyttet WIP, for at holde styr på hvem der er i gang med hvad.  
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 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 retrospektive vil, der blive kigget på om det er lykkes at ændre på de pågældende ting.

## Agile Manifesto

I og med SCRUM er et agilt framework følger det værdierne fra Agile Manifesto. Det består af 4 sætninger, med 2 påstande i hver. Disse sætninger er med til at definere agil softwareudvikling, og sikrer at man bruger agil softwareudvikling. Sådan som definitioner af sætninger er skrevet, er der lagt op til følgende udsagn: ”Der er værdi i punkterne til højre, men vi værdsætter punkterne til venstre mere”. Dette vil ikke nødvendigvis passe 100 procent til vores projekt, men dette vil vi komme nærmere ind på under hver værdi.   
De fire sætninger er følgende:

* **Individer og samarbejde** frem for processor og værktøjer.
* **Velfungerende software** frem for omfattende dokumentation.
* **Samarbejde med kunden** frem for kontraktforhandling.
* **Håndtering af forandringer** frem for fastholdelse af en plan

### **Individer og samarbejde** frem for processor og værktøjer.

SCRUM er et holdbaseret framework, hvor samarbejder er en nødvendighed, for at bruge det effektivt. Der er mange møder, der kræver åbenhed og ærlighed fra alle medlemmerne i grupper, og derfor er vi nødt til at finde et fælles ståsted, der kan fungere for alle i gruppen, for at opnå et godt samarbejde. Vi vil dog stadig benytte processor og værktøjer, til at hjælpe med vores planlægning og udførelse. Et værktøj vi bruger, som også er en del af SCRUM, er vores Scrumboard, det vil vi have et aktivt fokus på at benytte, da det er med til og sikre vi når alle vores opsatte krav.

### **Velfungerende software** frem for omfattende dokumentation

SCRUM handler om af få afleveret et velfungerende stykke software ved hvert sprint. I mange tilfælde vil der, derfor være stor fokus på at få lavet netop denne software færdig, da der er en kunde på den anden side, som forventer at det er klar til aflevering efter sprintet. I vores tilfælde er det ikke sagen, vores produkt, rapport og software, skal afleveres samlet. Det er også en systemudviklingseksamen, vi vil derfor have lige så stor fokus på dokumentationen såvel som en velfungerende software, da vi bliver vurderet på vores rapport.

### **Samarbejde med kunden** frem for kontraktforhandling

Vi har fået listet en række krav til vores projekt, hvilket fungere som vores kontrakt. Der er som tidligere nævnt mulighed for, at der vil komme nogle ekstra opgaver ind over projekt fra product owner. Skulle dette være tilfældet, er der nogle nye krav vi skal tage stilling med i samarbejde med kunden/product owneren, for at komme frem til det bedst mulige resultat, på trods af det ikke var nedskrevet i opgaven fra projektopstarten. Vi er derfor også nødsaget til at samarbejde med kunde frem for at kontraktforhandle.

### **Håndtering af forandringer** frem for fastholdelse af en plan

Dette udsagn passer godt med SCRUM, vi har mulighed for i hver sprint opstart at håndtere og prioritere diverse ændringer, der måtte opstå i vores projekt, i og med vi ikke har planlagt de næste tre uger, men kun en uge frem. Ud fra vores oprindelig tidsangivelse på vores user stories, har vi ekstra tid i projektperioden, hvis der skulle opstå nogle uforudsete problemer, der gør vi bruger længere tid på en task, end først angivet, eller hvis der måtte komme nogle ekstra krav fra product owner.

## Cynefin framework

Vi kan bruge cynefin til at hjælpe med at finde ud af oplysninger omkring vores domæne, samt hvordan dette skal anskues. Om vi har brug for hjælp fra eksperter, eller om opgaverne er nogle vi selv kan håndtere. Dette kan vi gøre efter vi har placeret vores domæne i en af de 5 segmenter i modellen, simple, complicated, complex, chaotic og disorder.

Efter gennemlæsning af det opstillede projekt, havde vi alle en god forståelse af domænet og de krav der er til opgaven. Vi har samlet i gruppen et fælles kendskab til budgetter og regnskaber, da det er noget vi tidligere har arbejdet, med under andre forhold. Derudover var der en god beskrivelse af regnskaber og budgetter i opgaveoplægget. Ud fra disse oplysninger befinder vi os i det simple segment, i det ordnede univers. Her er et ”cause- and-effect” forhold, hvilket passer godt med vores viden omkring domænet, vi kan se hvilke resultater vi får, ud fra de handlinger vi laver, inden vi foretager os dem. Dette gør også at vi basere de fleste af vores beslutninger på facts, og den viden vi allerede har omkring domænet, eller en viden der vil være os lettilgængelig.

Man kan argumentere for, at vi kunne bevæge os over i det kompliceret segment, hvis vi tog Xena med i vores vurderinger omkring domænet. Vi skal benytte Xenas API’er, og vi har ikke meget viden omkring dette. Her er vi afhængig af den dokumentation de levere omkring, hvilke data vi skal sende med, samt hvilke data, vi kan forvente at modtage. Vi skal her bruge ekstra tid på analyse af hvilke API vi skal bruge, derudover skal vi have oprettet nogle test data, på deres applikation, som vi kan bruge til at teste vores applikation. På dette område kan det også blive aktuelt, at vi møder udfordringer, der kræver at vi tager kontakt til Xena, for at kunne udbedre dem.