Inden den første sprint

Lav en Projekt Backlog (for hver enkelt projekt) med alle specifikationer for projektet
Projekt Backlog skal som minimum indeholde:

i. ID
ii. Prioritet (Prioriteret af Kunde/Kontaktperson, kunden kan i dette tilfælde også være en anden gruppe med viden inden for domænet)
iii. Estimat (ikke i tidsformat)
iv. How to demo (Sådan skal featuren bruges)
v. Evt: Note (Kort beskrivelse af implenterings metode)
vi. Evt: Done felt (Check af om det er færdigt eller ej)

Backlog Skabelon
HUSK!! Lav en kopi i stedet for at ændrer i orginalen (Filer -> Lav Kopi)

Overordnede regler

i. Alle **backlogs** (både project og sprint backlogs) er **tilgængelige** for alle, enten på SVN eller som dokument/excel ark på google docs.
ii. Vil man **tilføje features** til **andres backlogs**, skal man have prioriteret de features man vil tilføje efter egne prioriteter, så det er overskueligt for modtageren

Asynkrone vs Synkrone Sprints

Askynkrone Sprints:

i. Fordele
    a. Man kan lave **features hurtigt** så andre grupper kan bruge featuren ii deres næste sprint
    b. Man kan **hurtigt komme videre** med næste sprint da planning, review og retrospective **kun er internt** i gruppen
ii. Ulemper
    a. Man kan **ikke** lærer af hinandens erfaringer da erfarings dragning og planning ligger forskudt.
    b. Man kan **ikke have direkte** indflydelse på planlægning af andre scrums sprints ud over at man sætter en feature på andre gruppers backlogs.
    c. Man **bliver ikke informeret** om andre gruppers features, via planning, review og retrospective (+ demo).

Synkrone Sprints

i. Fordele
    a. Man kan have **erfarings deling** grupperne imellem, da der altid vil være et **mellemrum imellem sprints** hvor der er planlagt review, retrospective og planning møder.
    b. Man kan have mere **indflydelse på andres planlægning** af scrums da

gruppernes planer bliver fremlagt på planning mødet.

c. Det er **nemmere** for den enkelte gruppe at **involverer** sig i de andre gruppers arbejde da alt bliver fremlagt imellem sprints på enten planning, review eller retrospective møderne.

ii. Ulemper
    a. Alle grupper er meget **låst** på at skulle have **sprints samtidig**.
    b. Man skal bruge **ekstra tid** på at hører på andres præsentationer og man skal bruge **tid på at lave en præsentation** som andre kan forstå af det man har lavet i sin sprint.
    c. Man er afhængig af at **alle holder tidsplaner** i forhold til at færdiggøre features andre grupper skal bruge i efterfølgende sprints.

Sprints generelt

- Planning
- Sprint
- Evaluation

(Fælles møder er kun for synkroniserer sprints)
Planning (individual) [1 hour]
-> **Plan meeting** (present the plan for the other groups) [1 hour]
-> **Sprint**
-> **Evaluation** (Review/Retrospect - individual) [1 hour]
-> **Evaluation meeting** (present the evaluation for the other groups) [1 hour]
-> Planning (individual) … etc.

Planning og evaluation tager ca en halv dag, en kort fredags eftermiddag eller en kort mandags morgen
Sprint længde kan ændres på evaluerings møder!

Alle har fået **Scrumboards uddelt,**
Når grupperne har holdt stand-up-meeting, udfyld Burndown chartet og print to kopier, hvor den ene sættes på fælles opslagstavle.

Burndown Chart Template
HUSK!! Lav en kopi i stedet for at ændrer i orginalen (Filer -> Lav Kopi)

Sprint Planning (Individuel)

Alle grupper har gjort sig klart hvad de vil **lave i den kommende sprint**. Opgaver til den kommende sprint er planlagt.
Har nogle grupper brug for, at en **bestemt gruppe skal have en funktionalitet færdig** for at kunne fortsætte, skal det være defineret og beskrevet så den modtagne gruppe kan sætte det ind i deres backlog under dette møde.

Plan Møde (Alle)

Dette møde bruges til at **fremlægge** sin **plan** for den **kommende sprint** for alle grupper.
Dette betyder også at det er vigtigt at alle har gjort sig klart hvilke funktionaliteter, der har **prioritet** når der skal planlægges sprints.

Sprint

Under sprinten arbejdes der **kun** med de opgaver de individuelle grupper har påtaget sig under sprint planning.
Har man påtaget sig en opgave under sprint planning er denne opgave færdig når sprinten er ovre.
Det betyder at opgaven er lavet **færdig** og **testet** i slutningen af sprinten.

Sprint Evaluation (review/retrospective - Individuel)

Dette møde bruges til at **evaluerer** sprint'en, spørgsmål man kan stille sig selv under evalueringen er.
- Hvad blev der lavet?
- Hvad mangler vi?
- Hvad skal vi have lavet i næste sprint?
- Hvad har vi lært i den her sprint?
- Hvad gør vi anderledes i næste sprint?

Presentation for kontaktperson eller hvem der kunne have interesse for produktet.
Review og retrospective kan også bruges som forberedelse til næste plannings møde.

Evaluation Meeting (Alle)

Alle grupper fremlægger det de snakkede om under deres sprint evaluation.
Erfarings deling er et nøgleord
Begyndelsesdato, afslutningsdato og længde af næste sprint aftales.

Opdelt projekt

Stikord:

i. 4 perioder
ii. Første sprint til at skrive fælles indeledning
iii. Ca 1 måned til at programmerer (inklusiv testing)
iv. Ca 1 måned til at skrive rapport

Skiftende projekt

Stikord:

i. Skiftevis sprint med programmering og med rapport
ii. Testing hører ind under hver programmerings sprint

iii. Rapport skrivning planlægges som en almindelig sprint

Denne metode kan ændres til at man har sprints til at programmerer i og efter hver sprint er der indlagt et mellemrum på et antal dage hvor man kan skrive rapport.

Selvstyrende projekt

Stikord:

i. Hver gruppe bestemmer selv hvad de laver i hver sprint
ii. Det er vigtigt at alle er godt forberedte til "sprint planning" møderne, for at vi kan arbejde sammen.

Man kunne modificerer metoden til at man ved hvert sprint review kan bestemme om man vil have en periode med projekt-skrivning.

Fordele/ulemper ved de faste modeller

Fordel:

i. Man kan altid være **sikker på** at de andre grupper udvikler, når man selv er midt i en udviklings sprint.
    a. Hvis man fx. finder problemer i den funktionalitet man arbejder i, som bunder i en anden gruppes arbejde.
ii. Man **gennemgår sin kode tre gange**
    a. Når man koder
    b. Når man tester
    c. Når man skriver om koden

Ulempe:

iii. Det er **ikke** alle grupper der skal være **færdige med at programmerer samtidig**.
    a. Eksempel: Admin gruppen vil højest sandsynligt forsøge at være færdige så tidligt som muligt, Timer gruppen vil gerne kunne bruge noget af Admin gruppens funktionalitet, derfor kan de ikke lave bindingerne før efter Admin delen er færdig.
iv. Alle grupper går **ikke i gang samtidig**.
    a. Timer app'en kræver en masse forberedelse i form af design i samarbejde med kontaktpersonen og skal derfor højest sandsynligt bruge første del af projektet på at beskrive og lave design.
    b. Admin kan godt gå i gang med det samme og vil senere få tilføjelser når de tre frontend grupper har specificeret deres designs og får nogle kriterier.
v. Man arbejder med det der er **bestemt** ud fra **projekt modellen**.
    a. Det **minder mere om en modificeret traditionel metode** (med iterationer) frem for en agil metode.

Fordele/ulemper ved selvstyrende

Fordel:

i. Man arbejder **udelukkende** ud fra sin **prioritering i backloggen.**
ii. Man kan **selv styrer** hvordan man vil arbejde med i sine sprints.
iii. Man er **ikke tvunget** til at **programmerer** på et tidspunkt hvor man ikke er helt sikker på sin systemdefinition eller **ikke er klar**.
iv. Man er **ikke tvunget** til at programmerer på et tidspunkt hvor man syntes man **er færdig** med sit projekt.

Ulempe:

i. Det kræver at grupperne imellem er rigtigt **gode til at snakke sammen** op til et sprint planlægnings møde for at **undgå disskusions** klubber på **sprint planlægningsmøderne.**
ii. Alle grupperne har **selv ansvart** for at der bliver **programmeret færdigt til tiden** og at der bliver **skrevet rapport til tiden**.