**Versions styring**

Hvad dækker over versions styring? Det kan beskrives så simpelt at have navngivning af filer samt mapper I en eller anden form for arbejdsgang. Det er ofte også beskrevet som en metodisk tilgang og kan ydermere suppleres med noter om ændringer der er tilgået løbende.

Hvorfor benytter man så sig meget af det, og hvorfor er det vigtgit for os at kunne?

Til at elimere fejl i en arbejdsproces hvor der er flere hænder på samme projekt. Derudover kan man gå tilbage til tidligere iterationer hvis det alligevel skulle ske. Det kan også bruges til at holde styre på indtil flere delprojekter og samme element kan også laves I forskellige versioner til flere forskellige kunder på en gang.

Det bruges I alle typer af it-projekter og nærmest alle der arbejder professionelt med it-udvikling. Det kan være fra Store organisationer til open source hobby koder folket. Det kan også bruges til meget andet end kildekoder, som f.eks aftaler, rapporter. Det er et værktøj til alle som arbejder på samme type af projekt.

**GIT**

Git er en standard protolk amn bruger til at have version styring med. Og kom I sin tid til vejebragt af Linus Throvalds (Faren til Linux OS og kernel) for at kunne holde styr på udviklingen af Linux systemet efterhånden som der kom flere og flere med i udviklingen.

Git**Hub** er platformen vi lærte at bruge som server til at samle vores arbejde. Det smarte ved det, er at det ligger i skyen og man samtidig kan have et lokal repository med sine arbejdsfiler i. Derudover er tankegangen bag at det skal fugnere som et kodedelings-site/Social Media. Det har en meget fleksibel og nem at lære tilgang især med desktop programmet. Man kan også gøre det via kommandolinje, men det kræver at man skal huske på flere commands selv.

Udtryk man kommer omkring i fbm Git og på Github

* Repository: Samling af alle filer i dit projekt
* Branch: En forgrening der løber parallet i dit repository men ikke har nogen kontakt til master
* Master branch : Hovedforløbet der gælder. Dette er den man samler alt i når projektet er færdigt
* Commit: En ændring/Revision, det er som at gemme, blot har man mulighed for at gå tilbage senere I processen
* Pull request: Forslag til en ny/Ændret udgave med rettelser
* Merge: Når ændringer lafvet i en branch samles med en anden (Eks. Samler sin arbejdsbranch med master laver du en merge)
* Pull: Når du henter ændringerne og merger dem