# Jenkins

Jenkins er i det følgende beskrevet, da på trods af at det ikke kom til at virke under dette projekt, er brugt mange timer på det og der er kommet en masse erfaringer ud af det.

Jenkins har haft til formål at sikre en objektiv og løbende håndtering af de tests, der er skrevet i projektet. Det kobles sammen med et git-repository og har derigennem adgang til den nyeste version af projektet. Jenkins builder selv projeket og kører de test-suites, der er lavet igennem NUnit og NSubstitute NSubstitute er vel bare et værktøj brugt sammen med NUnit, og derfor ikke decideret, hvad testene er lavet igennem. Det kan også sættes op til at udføre statisk analyse og coverage, hvis det ønskes.

I dette projekt er der blevet brugt lang tid på at få Jenkins til at fungere. Dette er dog ikke lykkes efter hensigten. Det git-repository, der er brugt i projektet kører på GitHub, hvilket viste sig at være et problem. Under opsætningen af Jenkins projektet skal der specificeres, hvornår Jenkins skal bygge den VisualStudio solution, der henvises til. Til omfanget af dette projekt, vil det være optimalt at bygge projektet hver gang der bliver lavet et nyt push. For at dette kan lade sig gøre med et git-repository, der kører på GitHub, skal der installeres et plugin på Jenkins-serveren. Det har ikke været muligt at finde en måde at gøre dette på, uden at prøve sig frem med opsætningen af et af de plug-ins der findes til nettop dette. Da projektgruppen ikke har direkte adgang til Jenkins serveren kunne det derfor ikke lade sig gøre.

Der blev senere udleveret et git-repository af skolen, som kører på Git-Swat og derfor ikke havde ovenstående problem. I forbindelse med udviklingen af projektets database tilgang, blev der brugt et modelling-projekt i Visual Studio. Denne type af projekt kræver at der er nogle bestemte filer i installationen af Visual Studio. Disse filer var ikke tilgængelige på Jenkins-serveren og Jenkins kunne derfor ikke bygge solution’en. Det blev forsøgt at omgå dette ved at ligge de manglende filer i en mappe i projektet, og importere dem i app.config (STUFF VAR DET HER DU FORSØGTE AT IMPORTERE DEM ?). Resultatet heraf blev at Jenkins kunne bygge projektet, men at det ikke kunne bygges lokalt i Visual Studio.

Da underviseren, der står for Jenkins lagde den manglende fil der, hvor Jenkins søgte efter den, kunne projektet bygges på Jenkins og lokalt. På dette tidspunkt var størstedelen af de ønskede test-suites skrevet og kørt, og grundlaget for at bruge Jenkins var ikke længere tilstrækkeligt.

Det ovenstående beskriver processen med Jenkins i forhold til WPF applikationen. Jenkins skulle efter planen også have været brugt til at teste web applikationen. Her manglede Jenkins-serveren ligesom ved modelling-projektet en fil, for at kunne bygge web-applikationer. Tidligere erfaringer viste, at det ikke kunne betale sig at forsøge at løse dette problem, uden selv at have adgang til serveren, og Jenkins blev derfor droppet.