# Sikkerhed

Af: Anders

## GitHub – versionskontrol LÆSES

Vi har valgt på gruppen at tage GitHub i brug som vores versionskontrol værktøj. Det et værktøj vi har arbejdet meget med det seneste semester og er et meget mere fleksibelt værktøj end hvad vi lærte om på første semester ”Team Foundation”. Ved brug af TFS[[1]](#footnote-1) lå al vores data på en lokal server, som Klaus havde sat op, som krævede vi tilgik den via. EADanias vpn[[2]](#footnote-2) forbindelse, hvis vi skulle have mulighed for at tilgå den fra andre steder end hvis vi var på samme IP-adresse som serveren. Hvilket som sådan ikke ville være noget problem medmindre IP’en som serveren kørte på var nede, risikoen ved at det kunne ske ved en lokal server er større end hvis vi ligger det på en af GitHubs serveren.  
Derfor er valget taget ud fra vores egen erfaring med begge versionskontrol værktøjer, samt at Github har været en del af vores læringsmål på dette semester at lære om cloud services, som netop github og TFS er. Vi har bl.a. haft besøg af Johan Abildskov, som arbejder i ’Praqma’ der tilbyder forskelling kurser, som ville lære os på klassen alt basis om github og hvordan vi skulle bruge det.

Det er en mulighed for andre folk at clone hele vores GitHub, dog vil de ikke kunne push til vores GitHub og lave ændringer. Det vil kræve at de bliver tilføjet som collaborators til vores repository. Muligheden for at gøre vores repository privat vil altid være der, hvilket gør at andre folk end os der skal arbejde på det, ikke har nogen form for adgang. På den måde kan vi sikre at vores source code ikke bliver brugt af andre hvis det, er det vi ønsker. GitHub er et meget oplagt valg som vores versionskontrol grundet sikkerheden i at bruge det, samt det er Github vi har størst kendskab til, så vi slet ikke skal sætte os ind i det som vi ville være nød til med TFS. Da det var Klaus der ordnede alt for os ang. TFS vi sku bare bruge den til at dele vores source code.

## Test

### Whitebox

En af vores test består af ”whitebox” test. Ved at køre whitebox test giver vi source code med til personen som skal teste softwaren. Det en lidt dyr måde at teste på, da det kræver at en anden sætter sit eget arbejdet på pause for at kunne teste. Fordele ved at lave whitebox test er at der måske vil blive opfanget ”dead code” eller andre fejl, da det en person som forstår koden, men ikke har arbejdet med det specifikke stykke kode.

### Blackbox

Ved at køre blackbox test sikre vi at en person uden viden indenfor vores kode tester app’en[[3]](#footnote-3). Det gør at det ikke er en mulighed at finde eventuelle crash ved hjælp af koden. Vores primære blackbox tester vil være vores PO[[4]](#footnote-4), da han er en erfaren tester, og vi på samme tid kan få respons på mulige ønsker angående tilføjelser til app’en eller hvis der noget vi har misforstået ud fra forklaring.

### Destructive

Vores destructive test vil blive udført på samme tid som whitebox og blackbox test, formålet ved denne test form er at prøve alle tænkelig metode at få vores app til at crashe, så vi kan finde ud af hvor der skal sættes tid af for at sikre at når app’en bliver deployed at kunderne ikke vil kunne få programmet til at crashe

### Usability

Denne test vil primært blive udført som blackbox test for at vi kan sikre at vores interface er let at forstå for andre end os der har arbejdet på det i en længere periode.

## Database – valg af data

Vi har valgt i vores database, at vi ikke vil gemme alle folks oplysninger fra Xena når de vil sammenligne deres budgetmanager med Xena’s budget. Dog har vi valgt at det vil være en mulighed at kunne lave en identisk kopi af deres finanskonto navne, finansgruppe navne og kontonumre, da det vil være nødvendigt at de er helt identiske for at kunne sammenligne de 2 budgetter. Dette vil gøre det lettere for den budgetansvarlige at inddatere al nødvendige data, idet alle finaskontoer, finansgrupper og kontonumre er tilføjet, så der kun mangler diverse omkostninger og indtægter.   
Da det ikke er personfølsomme data måtte vi i teorien godt gemme alt deres data så vi kunne implementere det hele i deres nye budget som de kunne arbejde videre i. Men vi mener selv på gruppen det ikke vil være en nødvendighed. På den måde sikre vi også de ikke bare har 2 ens kopier af et budget, men at de får 2 forskellige budgetter de kan arbejde ud fra, og på den måde kan de få hjælp til at finde ud af hvor de skal optimere eller skære ned.

Ved at vi vælge ikke at gemme kontaktoplysninger eller lignende, sikre vi også at vi på ingen måde vil kunne miste oplysninger vi mener andre folk ikke bør få fat i, i tilfælde af hacking.

1. Team Foundation [↑](#footnote-ref-1)
2. Virtuel private network [↑](#footnote-ref-2)
3. Applikation [↑](#footnote-ref-3)
4. Product owner [↑](#footnote-ref-4)