

# A3B

## Erfarenhetsrapport

version 1.0

Andersson, Adam, adaan690  
Andersson, Conny, conan414  
Andersson, Martin, maran703  
Bolin, Andreas, andbo467

2012-12-12

## **Martin Andersson**

Nyttan av OOA och OOD märktes när vi satte igång att programmera. Klasserna som låg bakom GUI:t hade vi bra koll på då dessa i stort sett var "vanliga" C++ klasser vilket gjorde att vi kunde skriva bra OOA/OOD för dessa och de fick en bra struktur när vi implementerade dem. Klasserna som hanterade GUI:t hade vi dock inte skrivit någon vidare specificerad OOA/OOD vilket märktes i resultatet som blev mindre strukturerat och krävde betydligt mer felsökning. Lärdomen av detta är att vi borde satt oss mer in i hur GUI:t fungerade redan under analys- och designfasen för att kunna strukturera programmet bättre.

Till en början delade vi upp arbetet så att två av oss implementerade två olika delar av koden bakom GUI:t och de andra två satte sig in i GUI:t. Mot slutet var det dock bara GUI:t kvar att jobba med och då blev det en del dubbelarbete beroende på bristande kommunikation inom gruppen. Vi delade upp GUI:t så alla kunde sitta i olika delar dock borde vi haft mer avstämning kring generella funktioner då dessa skapades efterhand som vi insåg att de behövdes. Där hände det ibland att två stycken implementerat olika lösningar på samma problem. Detta problem hade också kunnat undvikas om vi planerat implementationen av GUI:t bättre.

Något som tog onödigt lång tid var för alla att få igång en fungerande miljö med Qt. När det var gjort gick dock arbetet med Qt smidigt då dels Qt Creator samt deras hemsida har bra dokumentation.

## **Adam Andersson**

Det har varit ett bra projekt där man har kunnat strukturera upp sitt arbete väldigt mycket med hjälp av OOA och OOD. OOA och OOD har man dock inte kunnat hålla helt då man under tiden som man programmerat märkt att flera funktioner har behövts.

Från början så delade vi upp arbetet emellan oss men sedan allt eftersom personer blev klara så hjälptes vi åt med återstående delar. Det har blivit en del krockar där personer har jobbat på samma dokument så att vi har fått olika lösningar på vissa problem, detta på grund av dålig kommunikation. Men en bit in i projektet så började vi använda oss utav ett kalkylark (workspace) på google där vi kunde kryssa i vilka dokument man arbetade i så de andra kunde se vilka dokument de inte skulle röra, detta har fungerat ganska bra tycker jag.

Det var svårt i början att få igång Qt på windows men sedan så gick det bättre då flera utav oss gick över till ubuntu där man fick igång Qt i stort sett direkt. Vi har även använt oss utav Github för att ladda upp våra filer, detta har inte fungerat bra för mig alls när jag satt i ubuntu. Vi löste det så att jag fick ladda upp filerna jag jobbat i till våran dropbox så fick någon annan pusha dom åt mig.

## **Conny Andersson**

Att vi gjorde en tydlig och bra OOA och OOD var till stor fördel i projektet. Dock hade det i efterhand varit smart att göra GUI delen mer detaljerad och informativ, det hade underlättat bygget av GUI:t.

Vi delade upp projektet i början, vilket fungerade bra för de bakomliggande klasserna och databasen. Eftersom Qt inte vill köras på windows uppstod det en del problem och tidskonsumtion innan alla hade fått igång det. Så grafikdelen av projektet hade längre startsträcka och krävde mer tid än resten av projektet vilket innebar att alla jobbade tillsammans i det.

För att hålla koll på versionshantering och dylikt använde vi GitHub, vilket inte har fungerat så bra. Dels för att det blivit krockar där vi inte genom Git har lyckats lösa konflikterna. Och dels för att programmet på windows valde att krascha. Så lösningen vi gjorde var att skapa ett excelark online, där alla fick fylla i vad de jobbar med. Och på så sätt undvika konflikter, och lägga upp dokument i Git som de var tänkt från början.

Dokumentet fyllde fler funktioner då man kunde läsa kommentarer och skriva vad som behövde fixas och vem som gjorde detta under resans gång. Vi har jobbat kontinuerligt med projektet och har gjort klart delstegen i godtid, vilket innebär att det inte blivit stressigt innan deadline.

## **Andreas Bolin**

Det var lite trögt i början att få igång och förstå sig på hur QT fungerade. Efter mycket googlande och testande med MinGW så gick jag över helt till Ubuntu och efter ett par guider var jag snart igång med den senaste versionen av QT. Vi ville egentligen få igång en senare version på Windows eftersom två i gruppen satt med Windows och en tredje med mac. Själv satt jag med både Ubuntu och Windows. Dock så hjälpte jag de andra att installera Ubuntu parallellt med Windows och då kom vi igång på riktigt. Vi delade upp projektet i tre delar. En del innefattade databasen SQLite, som var nytt för oss alla, En annan del innebar övrig backend och den sista delen var gui:t, vilket också var också var nytt för oss alla, som Adam och jag arbetade med. Eftersom det var en lite större del med gui:t så kändes det bra att vara två på den delen. Trots att vi var två på gui:t så slutade det i att alla gruppmedlemmar arbetade med bakgrundfunktioner i gui:t för att implementera backend och databasen. Dock så komplicerade det arbetet en hel del eftersom det är i princip en enda stor klass, vilket gör det svårt att arbeta med den samtidigt. Martin kom dock på den smarta idén att dela upp funktionerna till gui-klassen i sex olika filer. Detta underlättade arbetet betydligt. Det har varit mycket lärorikt att lära sig grafik med QT, det har dock tagit mycket tid.

Kravspezifikationen och OOD har varit bra eftersom man då kan gå tillbaka och titta på vad de olika klasserna ska göra, hur de ska kommunicera och vart det ska göras eller vad vi har kommit överens om hur programmet skulle utformas. Vi hade till en början en Carpark klass som skulle fungera som en övergripande klass som skulle agera kommunikationen med main. Vi kom dock fram till att den var överflödig och tog bort den.

Till en början, under dokumentationsfarsen använde vi oss av Dropbox, men så fort vi kom igång med kodningen så visade det sig att det inte var rätt redskap och vi gick snabbt över till git i form av GitHub som har versionshantering. Det har varit mycket bra eftersom alla får de senaste ändringarna och man kan jobba parallellt samt se varandras framsteg och förändringar. Dock så har vi haft en del strul med git, främst då vi mergar filer som blir korrupta och inte kan kompileras innan man går in och ordnar till dom manuellt. Vi fann då en lösning att lägga upp ett excel dokument på Google Drive som alla hade tillgång till och ändrades, för alla, i realtid. På detta vis så hade vi koll på att ingen annan satt och arbetade med samma fil vilket undanröjde en del problem med git. Speciellt på slutet då alla gruppmedlemmar arbetade med gui biten. Dropbox har dock fungerat som en backup när det har strulat till sig rejält med git.

Tidsmässigt så har detta projekt personligen tagit alldeles för mycket tid, långt över vad som poängen motsvarar. Det beror bland annat på att vi aldrig arbetat med QT eller något gui i c++ innan. Det kan också bero på att projektet var större än vad vi först trodde. Vi skulle behövt vara en gruppmedlem till. Vidare så kan vissa delar av dokumentationen reduceras, så att man kommer igång att koda snabbare. Även responsen från tidigare labbar hade kunnat varit snabbare så att man inte behövt lägga tid kompletteringar när projektet dragit igång.