



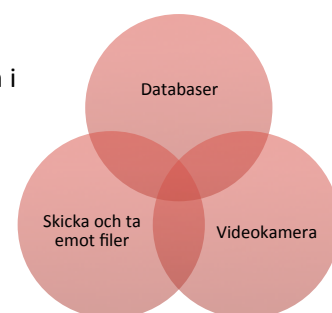
Post-Mortem Report

Which processes and practices did you use in your project?

Inledningsvis av projektet beslutade vi oss för att genomföra projektet enligt Scrum-metodik. Mötena som genomfördes handlade primärt *hur* och *vad* vi skulle lära oss Android-programmering, då detta var nytt för samtliga i gruppen. Dessutom bestod mötena av återkoppling om vad vi hade lärt oss mellan de olika mötena. Dock fungerade detta inte speciellt bra, då samtliga i gruppen ägnade sin fulla tid åt kandidatarbetet som skrevs parallellt med denna kurs, vilket innebar att vi oftast inte hunnit göra något mellan mötena, vilket gav att utfallen av mötena inte var så positiva. När kandidatarbetet lämnats in lades full fokus på att få en fungerande Android-applikation. Samtliga i gruppen var överens om att det förekom en relativt stor tidspress och att planeringsarbetet bakom applikationen var tvungen att ske på ett effektivt och strukturerat sätt. Detta innebar att vi skrev user stories i PivotalTracker samtidigt som vi ritade upp applikationens alla tänkta vyer på en stor whiteboard, för att tydligt kunna visualisera och konkretisera olika user stories. På så vis lyckades vi på ett effektivt sätt få en gemensam bild över hur applikationen skulle se ut, vilka funktioner den skulle ha och hur utvecklingsprocessen skulle ske.

När user stories var etablerade fortskred arbetet genom att dela upp dem i olika kategorier för att varje gruppmedlem skulle få varsitt ansvarsområde. Exempel på var hur en användarbas kan sättas upp, hur videokameran fungerar samt hur man kan skicka och ta emot filer mellan olika enheter. Eftersom samtliga av gruppmedlemmarna inte tidigare hade någon erfarenhet av Android-programmering var det svårt att uppskatta hur lång tid de olika delarna skulle ta, vilket innebar att indelning till olika områden och funktionaliteter justerades efterhand i takt med att kunskap ökades. All programmering skedde med syfte till att användaren skulle återfå en god funktionalitet och en bra användarvänlighet.

Vi valde att sitta tillsammans i gruppen under i princip hela utvecklingsarbetet. På så vis uppstod diverse fördelar. Exempelvis kunde vi hjälpas åt när problem uppstod, vilket innebar om någon i gruppen hade erfarenhet liknande problem kunde detta lösas på ett effektivt sätt. Dessutom fick samtliga i gruppen direktfeedback vad de olika gruppmedlemmarna arbetade på för funktioner, vilket innebar att dubbelarbetet blev minimalt samt att det gav en möjlighet att tänka på hur de olika delarna skulle integreras. Dessutom ansåg vi att många av de olika ansvarområden gick in i varandra till olika grad, vilket innebar att vi skulle få diverse synergier genom att sitta tillsammans.



Vårt arbetssätt efterliknar till stor del de som beskrivs som ett *Agile* tillvägagångssätt. Detta till stor del på grund av att arbetet hela tiden anpassas efter den rådande situationen. Vårt mål var att varje dag ha en fungerande app, och successivt tillägga funktionalitet. Vi sorterade alltid funktioner efter hur kritiska de var att implementera utifrån ett användarperspektiv. PivotalTracker användes till viss del för att dela upp arbetet och hålla koll på vem som gör vad samt hur lång tid det uppskattas ta att implementera, men jag tycker att vi inte använde dess fulla potential, utan mer såg det som ett stöd. När



applikationen kommit till det stadiet att majoriteten av dess funktionalitet var implementerat användes post-it lappar om buggar uppstod eller att en ny funktion skulle implementeras. I praktiken innebar detta att om en bugg hittades så skrevs denna ned på en post-it lapp och delgavs till den individ som skrivit koden så denna snabbt kunde åtgärda buggen. Om en funktionalitet efterfrågades för att visst område skrev detta också ner på en post-it lapp och gavs till den gruppmedlem som var mest lämpad att implementera funktionen. Detta ansågs vara ett väldigt smidigt sätt att behålla ett strukturerat arbetssätt, samtidigt som det var någon form av felsäkring i och med att alla buggar återrapporterades i realtid. Varje individ fick själv prioritera sina post-it lappar med avseende på hur kritiska de var för att få en fungerande applikation samt att få en välfungerande app ur ett användarperspektiv.

I och med att vi satt i grupp i princip hela utvecklingstiden ställdes inte speciellt stora krav på återrapportering och strikta möten, utan dessa fördes kontinuerligt under utvecklingsprocessen. Dock fördes alltid en diskussion innan vi avslutade ett arbetspass för att summera vad som har gjorts och vad som ska göras, samt att olika gruppmedlemmar får möjlighet att delegera uppgifter till andra gruppmedlemmar om en balansering av arbetsbelastning skulle vara nödvändig. Dessutom användes Github som en källa för informationsdelning och versionshantering. Genom att hämta från Github kunde man se vilka förändringar som gjorts, vem som hade gjort dem och när de hade gjorts. På så vis fick återkoppling om vilka framsteg som gjorts och vilka problem som potentiellt finns kvar att åtgärda. Vi hade som mål att alltid ha en exekverbar version av applikationen på Github servern. Detta ställde vissa krav på testning av sin kod innan man laddade upp den till Github. På så vis fördes kontinuerliga test för att säkerställa att applikationen fungerade utifrån ett användarperspektiv. I ett tidigt skede av projektet uppfattades Github som tidskrävande och ineffektivt, men i takt med att vi lärde oss Git och med att antalet filer och överlappande arbete ökade sågs en tydlig nytta. Eftersom vi hade olika erfarenhet av Git användes detta inledningsvis olika mycket av olika gruppmedlemmar, vilket medförde diverse problem. Exempelvis kunde det ta väldigt lång tid att åtgärda *merge-conflicts* i ett tidigt skede, men som i slutet av projektet inte var något större problem.

Arbetsprocessen efterliknar alltså *Agile* programming, med till stor del *eXtreme* programming och till viss del även *scrum*. Vi har alltid strävat efter att ha en fungerande applikation vid slutet av varje arbetspass. Vi har använt oss av PivotalTracker för hantering av user stories, och även post-it lappar för felrapportering, utbyggnad av funktionalitet och övrig delegering av uppgifter. Dessutom har whiteboards använts när dessa har funnits tillgängliga för att visualisera relevant information, såsom gruppens agenda samt hur olika skärmar skall vara sammankopplade. I slutet av arbetsprocessen har arbetsindelningen även övergått mer från olika områden inom applikationen till områden utöver applikationen, såsom testning och dokumentation. Eftersom vi suttit tillsammans har en effektiv kommunikation återfunnits för att tätt koppla applikationen, testningen och dokumentationen. Jag anser att vårt arbetssätt fungerat väldigt bra utifrån de förutsättningar vi haft, alltså tidsbrist och inledningsvis begränsad kunskap i ämnet. Vi har kunnat ta snabba beslut som grupp och successivt kunnat säkerställa att vi har ett gemensamt mål och en gemensamuppfattning om applikationens slutresultat.



Approximately, how much time was spent (in total and by each group member) on the steps/activities involved as well as for the project as a whole?

Projektet påbörjades ordentligt från och med den 12 maj, då gruppmedlemmarnas olika kandidatarbeten var inlämnade. Innan detta datum förekom stora svårigheter att spendera tid på kursen, på grund av kandidatarbetets intensitet. Projektet avslutades 31 maj. Mellan den 12 maj till den 31 maj har vi arbetat nästan samtliga dagar i princip så mycket vi orkat. En vanlig arbetsdag spenderas på Chalmers mellan klockan 08.00-22.00, alltså cirka 14 timmar per dag. Men några dagar har uteblivit helt på grund av diverse möten och så vidare. Jag uppskattar att vi lagt cirka 240 timmar per person till projektet. Självklart hade det varit bättre att sprida ut arbetstimmarna över samtliga studieveckor avsedda för projektet, men detta har varit svårt då vi varit tvungna att prioritera vårt kandidatarbete.

Jag anser att arbetsfördelningen varit jämn, då vi till väldigt stor del suttit tillsammans i grupp och arbetat under samma tider. Även produktiviteten har varit relativt jämn, då samtliga i gruppen har relativt jämlika kunskaper inom ämnet. Uppdelningen av projektet har gjorts funktionsvis, alltså att varje gruppmedlem har ansvarat för varsin funktionalitet av applikationen. Jag har varit ansvarig för login och registreringssystemet samt till viss del kopplingen mellan användargränssnittet och data genererad från en databas. Anton har varit huvudansvarig för databasen och dess lokala representation i form av modellklasser, samt dess kommunikation sinsemellan. Alexander har ansvarat för användargränssnittet och även metoder för att presentera data på ett bra vis. Slutligen har Adam ansvarat för all funktionalitet kopplad till inspelning, uppspelning samt skicka och lagra video. Utöver denna funktionsindelning har projektet även inneburit dokumentation och testning, där jag har haft huvudansvaret för testning. De tre övriga har till viss del bidragit till testning, men framförallt skrivit dokumentation och åtgärdat diverse problem med applikationen. Uppdelningen mellan applikationsutveckling, dokumentation och testning har ungefär varit 80 % applikation, 15 % testning och 5 % dokumentation.

Practices and Techniques used

Här kommer en förklaring av vilka *practices* och *techniques* som använts under utvecklingsprocessen. Avsnittet är uppdelat efter en rad frågor, och under varje fråga beskrivs eXtreme programming, git/github och PivotalTracker.

What was the advantage of this technique based on your experience in this assignment?

Extreme programming ansågs vara den mest lämpade tekniken att använda genom utvecklingsprocessen av applikationen. Detta på grund av dess agila karaktär, alltså möjligheten att förändra arbetsgången baserat på ny information och ny kunskap. Ytterligare fördel är att vi suttit tillsammans under hela processen, vilket inneburit en viss direktfeedback från alla deltagare i gruppen, vilket ständigt givit ny information och nya beslut för att färdigställa applikationen.

Fördelarna som uppstod med Github var flertaliga. Jag anser att den främsta fördelen är dess effektiva versionshanteringsystem. På så vis kan man inte skriva över andras filer och man kan enkelt se vilken fil som är den mest aktuella och vilka ändringar som gjorts. Github ger även en bra övergripande bild om i vilket stadie utvecklingsprocessen befinner sig, genom att kunna smidigt kunna undersöka olika *branches* för att se hur olika gruppmedlemmars kod utvecklas till att bli en del av huvudapplikationen. Ytterligare källa till kontroll över processen är PivotalTracker. Detta är också dess främsta fördel, alltså att man smidigt kan få



en tydlig bild om hur arbetet fortskrider och även se vad olika utvecklare arbetar på för tillfället.

What was the disadvantage of this technique based on your experience in this assignment?

Det som var en nackdel med den utvecklingsprocess vi använde baseras i att vi arbetade under stor tidspress. Vi ansåg att vi var tvungna att ansvara och utveckla olika funktioner individuellt för att på ett så effektivt sätt som möjligt färdigställas applikationen, men insåg samtidigt fördelarna med *Pair Programming*, som tyvärr inte kunde genomföras. Ytterligare nackdel är att vi inte drog några tydliga gränser på *när* en user story är färdigställd (kopplat till *DoD - definition of done*). Detta innebär att någon i gruppen kunde via PivotalTracker leverera och acceptera en user story under annorlunda preferenser gentemot de övriga, vilket kan bidra till problem i form av att vissa arbetar under antagandet att en user story är fullständigt testad när den är accepterad, medan andra arbetar under antagandet att den skall testas senare etc. Den största nackdelen med Git och Github anses vara att det ofta uppstod tekniska problem, i synnerhet under ett tidigt stadiet i utvecklingsprocessen. Dock löste sig detta i takt med ökad kunskap.

How efficient was the technique given the time it took to use?

Jag anser att tillvägagångssättet som liknar eXtreme programming var väldigt effektivt. Jag tror att detta var en nödvändighet för att färdigställa applikationen under de tidsförhållandena vi arbetade under. Eftersom vi saknade erfarenhet av Android-programmering helt var det en nödvändighet att kunna vara flexibla och kunna svara mot ny kunskap och nya förutsättningar. PivotalTracker användes relativt sparsamt i vissa avseenden. Vi skrev välformulerade och genomtänkta user stories, men däremot formulerade vi inte *hur* dessa skulle genomföras. Detta var upp till var och en av utvecklarna att komma fram till när de valt en user story att arbeta på. Med bakgrund i detta så var alltså PivotalTracker ett väldigt effektivt sätt att få en överblick av user stories och arbetsprocessen, men kanske inte lika effektivt för att strukturera upp tillvägagångssätt och dylikt. Git/github var inledningsvis väldigt ineffektivt. Detta på grund av bristande kunskap i hur det skulle användas. Dock anses det väldigt effektivt när man väl blivit införstådd i hur det fungerar och vad dess huvudsakliga syften är.

In which situations would you use this technique in a future project?

Jag skulle absolut välja ett likartat tillvägagångssätt under samma omständigheter som detta projekt, alltså vid en arbetsuppgift där man är relativt oerfaren samt har en påtaglig tidsbegränsning. Jag tycker även generellt att det är ett bra sätt att inte i förväg definiera olika roller, utan successivt låta gruppens medlemmar ta olika ansvarområden och tillsammans ta beslut och strukturera arbetsgången. Extreme programming lämpar sig även väl när det finns möjligheter att arbeta tillsammans i gruppen under större delen av tiden, detta är också en parameter som behöver beaktas för att använda vårt tillvägagångssätt. Angående PivotalTracker och Github anser jag att det är två effektiva verktyg att använda sig av oavsett vilka förutsättningar man arbetar under. Jag har svårt att se en situation där PivotalTracker och Github *inte* lämpar sig för ett software engineering-projekt. Detta gäller i synnerhet Github, då PivotalTracker lättare går att byta ut mot andra lösningar såsom exempelvis en whiteboard eller ett gemensamt och välstrukturerat word-dokument.

In which situations would you not use this technique in a future project?

Extreme programming skulle jag inte föredra i situationer där projektet sträcker sig över en längre tid och där gruppen inte har möjligheter att sitta tillsammans under stora delar av projektet. Istället skulle jag då föredra att arbeta enligt Scrum-metodiken, med mer strukturerade möten och avstämningar. När möjlighet ges anser jag också att *Pair programming* bör användas, vilket alltså inte varit fallet i detta projekt.



Som tidigare nämnt anses både Github och PivotalTracker som effektiva verktyg för ett mjukvaruprojekt. Dock finns alltid substitut i olika former som också kan användas. Vikten av versionshantering och kontroll ökar i takt med att fler individer arbetar på projektet och med projektets komplexitet. Vid väldigt enkla projekt och där få individer arbetar kan alltså andra metoder också användas utan att utfallet påverkas till stor del.

If you had the practice/technique in a part of the project and not the entire project, how was using it compared to not using it?

Jag anser att alla tekniker och tillvägagångssätt användes ungefär lika mycket genomgående under projektets effektiva del. Projektets effektiva del är alltså ungefär från 12 maj till 31 maj. Innan den 12 maj användes till viss del Scrum-metodik, men förkastades när vi ordentligt påbörjade projektet. Scrum-metodiken anses inte som ett lämpligt tillvägagångssätt gentemot eXtreme programming, vilket alltså innebär att det var mer fördelaktigt när eXtreme programming användes gentemot när Scrum användes. Github skulle kunnat användas på ett bättre sätt inledningsvis under arbetet, dock begränsades detta av bristande kunskaper.

Project Evaluation

Jag anser att arbetsgången bakom min del av arbetet och de funktionaliteter som jag bidragit med fungerat bra. Stundtals har det uppkommit diverse tekniska svårigheter primärt kring Eclipse och olika plug-ins, men inget som inte jag inte lyckades lösa. Gällande programmeringen så anser jag att jag lyckats leverera välstrukturerad, välkommenterad och testad kod som var redo att integreras med övriga delar av projektet. Jag försökte till stor del tänka hur min arbetsuppgift var en del av en helhet och därmed göra min kod utbytbar och öppen för tillägg. Jag anser också att jag bidragit med att strukturera upp och planera arbetsprocessen samt hjälpt till för att projektet i sin helhet hållit en hög nivå och god kvalitet.

De arbetsuppgifter som jag tagit ansvar för och utfört som fungerade mindre bra handlar primärt om hur jag själv valt att planera och utföra dem. Många gånger har jag varit för snabb med att försöka implementera kod, utan att göra en ordentlig förundersökning för att identifiera det mest effektiva sättet att lösa uppgiften på. Detta har resulterat i ett visst dubbelarbete, då jag i vissa fall hittat ett bättre sätt att lösa problemet i efterhand. Ett exempel är att vi inledningsvis använde en php-koppling till en webbserver med en MySQL-databas, vilket vi senare ersatte med webbtjänsten parse.com, som anses vara en väldigt smidig och effektiv lösning för att lagra data i "molnet" och med ett bra java-api.

Vi arbetade i princip alltid tillsammans som grupp, vilket fungerade väldigt bra. Jag anser att gruppdynamiken varit bra under hela arbetets gång. Vi har ständigt strävat mot samma mål och haft en gemensam uppfattning om hur slutresultatet skall se ut och fungera. Jag tror att detta beror till väldigt stor del att vi använt whiteboards för att kommunicera våra idéer och de problem som var aktuella. Vi konkretiserade genom att måla upp olika vyer av appen och hur de förhöll sig till varandra. I framtiden skulle jag dock inte sätta mig i samma situation att behöva arbeta så många timmar per dag. Det var både fysiskt och psykisk påfrestande samt gav implikationer i form av att vi potentiellt skulle gjort ett bättre jobb om vi hade slagit ut arbetstimmarna på en längre tidsperiod. Då skulle vi även kunna öka arbetstiden i slutet av projektet om det skulle krävas, vilket var omöjligt i vårt fall då vi redan arbetade ohållbart antal timmar per dygn i slutet. I ett nästa projekt skulle jag även lägga mer tid på testning och dokumentation, då vi i detta projekt av tidsmässiga skäl var tvungna att fokusera väldigt mycket på att producera en fungerande applikation.