

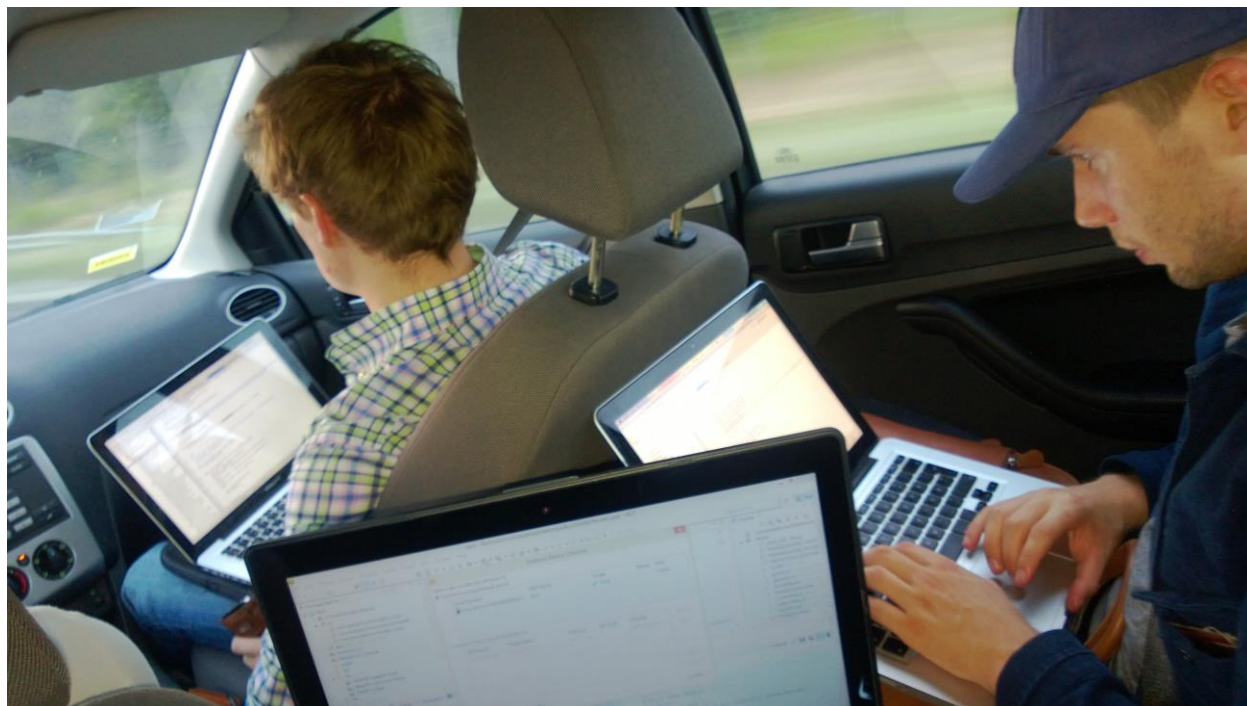
Post mortem Report

Post mortem report by Anton Dahlström (900313-0317) in Software Engineering project (DAT255) – lp4, 2013

Which processes and practices did you use in your project?

Syftet med projektet har inte bara varit att utveckla en färdig mjukvara, utan att också få kunskaper i hur ett lämpligt upplägg för mjukvaruutveckling kan se ut. Vi har på föreläsningar i kursen gått igenom några vanliga tillvägagångssätt för mjukvaruutveckling, där olika upplägg och dessas fördelar och nackdelar har förklarats. Det har inte funnits några riktlinjer om hur arbetet bör läggas upp, utan det har varit upp till var grupp för sig att välja den metod de tycker passar dem själva bäst.

I uppstarten av projektet planerade vi att använda oss av Scrum-metodiken och lägga upp arbetet enligt detta. På grund av att vi hade vår inlämning av vårt kandidatarbete samtidigt och att detta skulle lämnas in relativt tidigt, fanns det dock inte så mycket tid till övers för denna kurs till en början. Vi jobbade då till en början var och en för sig med att lära oss att skapa mjukvara för Android-plattformen, samtidigt som vi diskuterade hur vår applikation skulle kunna utformas. När väl kandidatuppsatsen hade lämnats in planerade vi att avsätta den tid som krävdes för att färdigställa applikationen på den återstående tiden. Det var i detta skede som majoriteten av arbetet koncentrerades till. Vi åkte också upp till en sommarstuga, där vi kunde arbeta ostört med projektet i tre dagar.



Figur 1 - Kodande i bilen på väg till #JavaCamp

Vi valde att lägga upp arbetet på så sätt av vi ritade upp lösningar på en whiteboard, med utgångspunkt i våra User Stories och hur vi hade tänkt oss att en användare lätt skulle kunna använda applikationen. Vi tog redan i detta skede hjälp av utomstående personer, som på ett översiktligt sätt

granskade och validerade våra ursprungliga val. Vi delade därefter upp arbetet i olika ansvarsområden, där varje projektmedlem fick ansvar för en avgränsad del av projektet. Sådana ansvarsområden inkluderade till exempel att sätta upp en databas för att lagra användardata, implementera en funktionalitet för att spela in och skicka video, hur presentationen av information till användaren kunde ske, samt sätta upp inloggningsfunktionalitet med medföljande funktionalitet. Den ursprungliga planeringen anpassades hela tiden allt eftersom projektet framskred, hela tiden med utgångspunkt i vad att utveckla funktionalitet som var så lätt att använda för en användare som möjligt. Mjukvaran förbättrades på så sätt genom en iterativ process i projektgenomförandet.

På grund av vårt pressade tidschema valde vi att sitta tillsammans och utveckla mjukvaran. Vi hade hela tiden möjlighet att ställa frågor i gruppen och det var ofta någon i gruppen som visste hur man kunde gå tillväga för att lösa ett visst problem. Vår arbetsmetod kan sägas innehålla många av de koncept som kan återfinnas i mer agila utvecklingsmetoder, där arbetet hela tiden anpassas efter den rådande situationen. Vi hade en lista med funktionalitet som hela tiden utvärderades och sorterades med avseende på hur de olika användarfunktionernas kritikalitet; målet var hela tiden att i första hand ha en fungerande applikation, för att senare lägga till funktionalitet allt eftersom. Eftersom vi valde att lägga upp arbete så att vi satt i majoriteten av tiden, använde vi oss inte så mycket av t.ex. PivotalTracker som vi kanske kunde ha gjort. Istället använde vi oss av listor och post-it lappar för att spåra vad som skulle göras och prioriterade dessa från hur viktiga de var i förhållande till vår övergripande målsättning. Vi samlade dessa uppgifter till en gemensam pool, där varje gruppmedlem kunde välja bland uppgifter: ibland skedde en direkt tilldelning av uppgifter, på grund av medarbetarnas inbördes specialisering till vissa uppgifter. Vi hade flera kortare möten eller samtal varje arbetspass, där gruppmedlemmarna fick redogöra för de framsteg som skett och problem som uppstått. På så sätt tydliggjordes mjukvarans utveckling steg för steg. Vi hade också oftast ett avslutande möte innan vi gick hem för dagen och innan arbetet startade, då vi följde upp vad som hade gjorts och vad som behövde göras; vid dessa tillfällen skedde ofta tilldelning av arbetsuppgifter.

Vi hade som mål att hela tiden ha en applikation som fungerade att använda vid slutet av varje arbetsdag. På så sätt har det varit enkelt att se förändringar som skett under dagen. På grund av applikationens koppling till databasen på internet testades mycket av funktionaliteten löpande genom att praktiskt använda applikationen och dessa funktioner och jämföra utfallet med det förväntade. Detta var också en anledning till varför det var viktigt att ha en fungerande release varje dag, som kunde fungera som en bas för fortsatt utveckling av andra delar. Genom att testa applikationen som användare, fick vi också uppslag till nya idéer för förbättring av gränssnitt och funktioner. Målet av att ha en fungerande applikation var något som, av naturliga skäl, inte kunde uppfyllas under projektets tidigare skeden. Efter att grundfunktionaliteten hade implementerats kunde vi dock oftast uppfylla detta mål. Detta var också något som vi hade stor användning av Git för att klara av. Git och GitHub har gjort versionshantering och skilda utvecklingsvägar betydligt enklare att använda. Gruppen hade innan inte mycket erfarenhet av att använda detta stöd innan projektet startade, vilket gjorde att vissa problem uppstod efter vägen. Efter vi fick mer erfarenhet av att använda Git, flöt detta dock på allt bättre.

Som kan uttydas från ovanstående text, har vi använt oss av agil programmering, där vi mixat vissa koncept från eXtreme Programming med Scrum. Vårt arbetssätt liknar dock till störst del eXtreme Programming, dock med vissa avsteg från de grundläggande delarna av denna metod. Vi har strävat av att ha en fungerande applikation tillgänglig vid varje arbetsdags slut. Vi har också utvecklat vår arbetsprocess

allt eftersom i projektet: de delar som vi ansåg fungerade smidigt och bra behöll vi, medan vi valde bort andra som vi upplevde inte tillförde så mycket värde. Något som vi prioriterade bort på denna basis var, till exempel, användandet av PivotalTracker som ett projektstruktureringsverktyg. Vi har använt detta verktyg som en hjälp för att samla våra User Stories; den faktiska uppdelningen av arbetet var dock något som skedde med hjälp av mer rustika metoder, där ibland ovan nämnda post-it-lappar. Vi har ansett att denna arbetsmetod har lämpat sig väl i förhållande till de förutsättningar som fanns.

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?

Som nämnt ovan arbetade vi med vårt kandidatarbete de första veckorna av denna läsperiod. Vi försökte att väva in även denna kurs så ofta det gick, men med tanke på att gruppmedlemmarna var med i olika kandidatgrupper, var det svårt att få kontinuitet i arbete till en början. Från början av maj fram till idag, den 31 maj 2013, har alla i gruppen arbetat heltid med detta projekt. Med heltid avses i stort sett all tillgänglig vaken tid för alla gruppmedlemmar. Detta är något som har varit genomförbart, men vi inser att det inte är en optimal situation att försätta sig själv i. På grund av upplägget av vår utbildning på programnivå, finns dock svårigheter att lägga upp arbetet på ett annorlunda sätt. Ett alternativt hade varit att skapa en enklare applikation och på så sätt minska den totala mängd arbete som krävs, men detta var ingenting vi ville göra, eftersom vi aspirerar mot att få ett bra betyg i denna kurs.

Eftersom vi satt i grupp den absoluta majoriteten av tiden, har vi inte haft någon större differens i total arbetsmängd internt inom gruppen. Vi hade också en relativt jämn och hög nivå av teknisk kunskap i gruppen, vilket gjorde att det sällan blev problem på grund av det, samt att vi klarade att lösa de problem som uppstod relativt snabbt. Vi har inte fört någon direkt arbetslogg där vi registrerat antalet timmar, men min uppskattning är att vi lagt i genomsnitt runt 220 timmar per person på detta projekt, vilket kan summeras till uppskattningsvis 840-920 timmar för hela projektgruppen.

Vi valde delade upp arbetet i skilda delar med en ansvarig gruppmedlem, vilket har gjort att vissa gruppmedlemmar har blivit gruppens expert inom detta område. Denna expert har implicit blivit ansvarig för dokumentation och slutförande av dessa enskilda områden, vilket faller sig naturligt och blir relativt tidseffektivt om det genomförs väl.

Techniques used: Git/GitHub

Vi fick, som ett led i kursen, uppgift att använda oss av Git, kopplat till GitHub. Detta skedde enligt instruktion, för att underlätta för handledare och andra kursansvariga att hålla koll på status i projekten. Detta verktyg var ingenting som vi hade haft någon erfarenhet av att använda tidigare, vilket gjorde att det fanns en viss startsträcka innan vi kom igång med detta ordentligt. Det fanns också i början vissa fall, där vi på grund av vår oerfarenhet fick problem vid sammanslagningen av kod skriven av olika gruppmedlemmar, vilket orsakade en hel del arbetstid som gick åt intet. Detta uppkom till viss del eftersom var och en i gruppen ursprungligen skapat sina egna projekt, som utanför vår gemensamma Git-repo, för att kunna testa androidutveckling. När vi skulle inkorporera den koden i vår gemensamma repo, uppkom en hel del problem. Efter den initiala startsträckan, tycker jag dock att vi ha fått upp farten lite med att använda oss av verktyget i fråga. Vi har fått betydligt större säkerhet i hanteringen av Git och har lärt oss använda ett verktyg som kan vara nyttigt att ha i framtiden.

Git i sig är väl lämpat för den typ av utveckling som detta projekt inneburit. Vi har vid ett flertal tillfällen skapat egna branches där vi kunna driva utvecklingen av funktionalitet utan att påverka den ursprungliga applikationen. Att använda Git istället för till exempel Dropbox eller andra tjänster, har också inneburit att vi fått större möjlighet att arbeta parallellt, eftersom flera personer kan jobba med samma filer på samma gång, utan att för den skull orsaka oåterkalleliga problem.

Det som varit negativt med att använda Git har varit att det ibland uppkommit problem vid tillfällen då kod från olika personer ska sammansättas. Det finns också risk att, när kodbasen växer, att fel smyger sig in som från början inte är uppenbara. Det kan då ibland orsaka en viss osäkerhet om att den kod som finns på master-branch faktiskt är korrekt; det är lätt att man blir lite paranoid efter ett tag.

Efter att ha fått prova på detta verktyg i denna situation, inser jag nu de styrkor som finns med det. Jag kommer nog att fortsätta att använda mig av Git och GitHub även i framtiden, speciellt i projekt som involverar flera deltagare. Jag kommer nog också att använda mig av verktyget om jag jobbar i egna projekt, dock i mindre omfattning och utsträckning än vid fallet med större grupper.

Techniques used: PivotalTracker

En ytterligare del i kursen var användandet av PivotalTracker för att planera projektet och de uppgifter som fanns att göra. I vårt fall insåg vi dock ganska snabbt att detta verktyg inte riktigt passade vårt arbetssätt, utan vi valde istället att ha en mer analog metod, där vi använde oss av papper/penna och whiteboards för att strukturera upp arbetsgången.

Fördelarna med PivotalTracker är att det är lättare för handledare i kursen att följa upp hur olika grupper ligger till i planering och genomförande. Det kan också vara ett kraftfullt verktyg att strukturera arbetet omkring om det görs löpande och väl. Att allt sker på nätet och på så sätt synkroniseras och arkiveras efter att det är färdigt, gör också att det finns möjlighet att gå tillbaka och återskapa processen steg för steg, vilket kan vara bra i uppföljningssyfte. Kort sagt: det finns helt klart flera fördelar med att använda sig av detta verktyg.

Som nämnt flera gånger innan passade inte PivotalTracker vårt sätt att jobba speciellt bra. Jag tror att vi skulle haft större potential att använda oss av detta verktyg om vi programmerat i par och/eller suttit på olika platser. När vi istället satt alla tillsammans och jobbade mer eller mindre enskilt var för sig, blev dock detta verktyg mer ett onödigt hinder som bara medförde mer administration. Att istället använda oss av papper-och-penna-metoden innebar att vi, direkt när vi hittade något som behövde åtgärdas eller kom fram till en förändring av den övergripande strukturen, kunde skapa en "task" genom att helt enkelt skriva upp den på ett papper eller på en tavla.

Jag skulle nog kunna tänka mig att använda PivotalTracker i framtiden, men då i större utsträckning än vi gjort i detta projekt. Jag är övertygad om att det finns fördelar med att strukturera upp problemet på det sätt som detta verktyg innebär. Det är dock lämpligare för projekt som löper över längre tid än vårt projekt och som inte är lika intensiva i form av tidsåtgång och möten.

Project evaluation

Användandet av eXtreme Programing har varit en essentiell del för att klara av att genomföra projektet. Genom den mycket agila arbetsgången, klarade vi att utveckla en applikation med ett tillfredställande slutresultat på den begränsade tidsramen som vi hade att tillgå. Genom att i så stor utsträckning som

möjligt försöka hålla sig till de riktlinjer som ingick i detta tillvägagångssätt underlättades arbetsgången. Alla projektmedlemmar har, genom användandet av dessa tekniker, även fått en bra översikt över hur projektet framskridit.

Att istället valt att använda sig av vattenfallsmetoden, där omfattande förberedande arbete skulle krävs innan arbetets start, skulle varit mindre lämpat för ett projekt av den typ som vi nu genomfört. Vi skulle då ha behövt spendera mer tid att i detalj planera och strukturera upp projektet, innan vi börjat med att implementera funktionalitet som användaren har nytta av. Detta skulle sannolikt inte medfört att den slutgiltiga applikationen hade blivit bättre ur en användares synpunkt. Att istället, som vi gjort nu, skapa en enklare ram för arbetet och sedan iterativt utveckla mjukvaran lämpade sig mycket bra för de förutsättningar som fanns. Det fanns då en större möjlighet att anpassa planen allt eftersom och på så sätt hela tiden flytta målsättningen så att det följer den vision om den slutgiltiga lösningen som finns, detta passar speciellt bra om målet är ett *moving target*, vilket ofta är fallet i mjukvaruutveckling mot kund, då kundönskemål ofta ändras allt eftersom.

Något som jag personligen så som lite av ett problem med denna metodik, är att det just inte finns någon tydlig struktur innan arbetet startar igång. Detta kände jag kunna orsaka en viss osäkerhet när man skulle börja implementera avskilda delar av projektet. Efter att vi konstruerat en ursprunglig version av mjukvaran efter några veckor, blev det dock lättare att realisera delar av användarupplevelsen. Att vi arbetar i en relativt stor projektgrupp, innebär också att det finns en utmaning i att koordinera arbetet, där alla projektmedlemmar ska arbeta mot samma mål. Detta skulle också underlättats om man tidigare i projektet hade specificerat de delar som skulle ingå i det slutgiltiga avslutet.

En av de saker som jag dock tycker har fungerat bäst är samarbetet i gruppen. Vi har varit en grupp individer som jobbat mot samma mål och det har aldrig kommit upp problem med arbetsmoral eller liknande. Alla gick in i projektet med inställningen att vi skulle göra bra ifrån oss och var villiga att lägga ner den tid som krävdes för att realisera detta även i praktiken. Jag tycker också att vårt upplägg på arbetet har fungerat relativt bra, där alla medlemmar i gruppen fick bidra med sina talanger på ett bra sätt. Även metoderna för arbetsuppdelning tycker jag har fungerat på ett tillfredställande sätt. Eftersom vi i stora delar av projektet har suttit i helgrupp och arbetat var och en för sig på enskilda uppgifter, har det alltid funnits möjlighet att snabbt fråga de andra om hjälp vid tekniska eller strukturella problem. Vi har också fört en löpande dialog om i vilken riktning utvecklingen av mjukvaran bör fortsätta och på så sätt tagit fram nya uppslag på User Stories allt eftersom.

Om man ska vara självkritisk, så finns det självklart saker som skulle kunna förbättras om man fick göra om samma projekt en gång till. Om jag fick önska, skulle denna kurs ligga på ett sådant sätt att det finns möjlighet att helhjärtat gå in för projektet redan från uppstarten och inte, som nu, tvingas skjuta på stora delar av arbetet till långt in i perioden. Detta beror på hur kandidatarbetet ligger för oss på Industriell ekonomi, vilket betydligt försvårar arbetsprocessen i denna kurs. Detta faktum har gjort att vi har haft en ganska extrem arbetsbelastning i denna läsperiod. Att arbeta runt 12 timmar och uppåt per dag har mer varit regel än undantag. Jag skulle dock, om jag fått göra om projektet, försökt få in fler timmar i början för att antingen strukturera upp och planera lite mer, alternativt läst på om Android-utveckling redan innan projektet startats. Jag tror dock inte att vi hade hunnit så mycket längre med utvecklingen än vad vi skulle ha gjort annars, men jag tror att det kunde ha ett lite lugnare avslut på perioden.