

Post Mortem Report – Simon Wessberg

Jag är med i gruppen *Androidsquad* som utvecklat appen *Coloristance*. I början av arbetet använde vi oss delvis av en definierad process (*defined process*), då vi ägnade en del tid åt att planera arbetet och se vilka krav vi hade med applikationen och hur vi skulle kunna genomföra de planer vi hade. Däremot så upptäckte vi relativt tidigt att ett empiriskt angreppssätt (*empirical process*), med kortare iterationscykler, lämpade sig betydligt bättre för vårt projekt. Just en iterativ arbetsmetod var det som passade vårt projekt bäst, och det slutade med att vi använde oss av en *agile* eller iterativ arbetsprocess. Denna iterativa arbetsprocess innebar för oss att applikationen kontinuerligt under arbetets gång byggdes på med ny funktionalitet. Det var alltså en iterativ arbetsprocess i den mån att när en viss funktionalitet adderats så gick vi tillbaka, analyserade applikationen i dess helhet, och beslutade om vilken extra funktionalitet som skulle tillkomma. Denna iterativa arbetsprocess var mycket viktigt för vårt arbete då den lät oss hela tiden bygga vidare på existerande funktionalitet, och på så sätt vara en flexibel och dynamisk arbetsprocess.

För att implementera denna iterativa arbetsprocess valde vi inledningsvis att använda oss av *Scrum*. Vi hade ett inledande möte där vi tog fram *User stories* som förklarade de tänkta kraven från användarna för vår applikation. Dessa *User stories* placerades i vår *Product backlog* för vår applikation. Sedan hade vi ett inledande *Sprint planning* möte där vi bestämde vad som skulle åtas under första sprinten, det vill säga den första *Sprint backlog*. Under veckan försökte vi ha dagliga Scrum-möten, men ett problem vid denna tid var att vi alla samtidigt skrev vårt kandidatarbete, vilket gjorde att det var svårt att lägga fokus på applikationens utveckling. Detta gjorde alltså att vi knappt hann ha några Scrum möten alls under veckan. Vid *Sprint review* den kommande veckan insåg vi dock direkt att vi inte hade uppskattat tiden det krävdes för att göra dessa åtaganden. Vi hade inte hunnit klart de uppgifter vi åtog oss och de uppgifter som hade delats ut var inte jättetydligt isolerade mellan de olika gruppmedlemmarna. Efter två veckor in i projektet valde gruppen att överge Scrum. Detta berodde dels på det faktum att vi inte var erfarna med Scrum och att det därför var svårt att göra realistiska antaganden över hur lång tid de olika uppgifterna skulle ta. Framförallt berodde dock detta beslut på det faktum att vi inom gruppen inte var på den erfarenhets- och kunskapsmässiga nivå inom programmering som skulle krävas för att varje medlem skulle kunna ta på sig tydliga uppgifter varje Sprint och lösa dem till nästa veckas Sprint review. För oss var det många problem som för oss föreföll svåra, som för andra mer erfarna programmerare eventuellt skulle framstå som mer basala programmeringsbekymmer. Vi kände helt enkelt att vi inte hade tillräcklig tidigare erfarenhet inom programmering som helhet för att effektivt kunna implementera den arbetsprocess som Scrum innebär. Eftersom vi inte hade den tidigare kompetensen inom programmering märkte vi att det var betydligt mer effektivt för oss att arbeta som en grupp, och istället dela den kunskap som vi var och en hade, för att på så sätt öka den totala kunskapen inom gruppen.

Istället för Scrum valde vi därför att använda oss av *eXtreme Programming (XP)* som teknik. Vår implementation av XP innebar just det XP gör, nämligen att vi utnyttjade det som fungerade för oss och använde det till sin extrem. Vårt sätt att arbeta med XP innebar att vi satt och programmerade och arbetade i helgrupp under större delen av tiden i arbetet. Detta har gjort att vi har fått en oerhört bra kommunikation inom gruppen, samt en väldigt

homogen och tydlig bild och förståelse av alla inom gruppen för projektets situation, problem och utvecklingsområden. Vi har suttit mest tillsammans som helgrupp, men även givetvis inom gruppen delat upp arbetsuppgifter. Till skillnad från så tydliga arbetsuppdelningar som inom Scrum har vi dock använt mycket *parprogrammering*. Denna parprogrammering har skett väldigt dynamiskt, då olika konstellationer har uppstått och splittrats allteftersom. Alla inom gruppen har under någon gång i arbetet parprogrammerat med i stort sett alla de övriga gruppmedlemmarna. Viss lite tydligare arbetsuppdelning förekom dock: den mest programmeringserfarne vägledde de rent programmeringsmässiga idéerna, jag och en annan i gruppen ansvarade för huvuddelen av dokumentationen och de sista två var huvudansvariga för testningen av applikationen.

Som tidigare nämnt arbetade vi under större delen av projektet tillsammans som en hel grupp, till den extrem att vi befann oss i ett och samma rum tillsammans under större delen av tiden. Detta medförde, också som nämnt, kommunikationsmässiga fördelar. Det innebar också att vi inte behövde schemalägga några bestämda tidpunkter för dagliga möten, utan vi hade kontinuerlig kontakt under hela dagen. Om ett problem uppstod så togs det omhand direkt, och frågor och strategiska beslut togs upp direkt i helgrupp. Vi hade alltså inga specifika möten, förutom varje morgon då vi stämde av vad som skett sedan gårdagens arbete, utan kommunicerade kontinuerligt. Man kan inte direkt påstå att vi har använt oss av *test-driven development*, då testningar inte har styrt all kod som skrivits. Dock vill jag påpeka att testning av applikationens funktionalitet har skett kontinuerligt, framförallt genom iakttagelse på fysiska Android-enheter, med skärmar av olika storlekar. Vi tog dessutom in testning via JUnit relativt tidigt i arbetsprocessen, något som har gynnat utvecklingen också.

I enlighet med XP har vi använt oss av kontinuerlig *design improvement*, då vi strävat efter att ständigt utveckla och förbättra designen av koden. Detta har återigen varit en iterativ arbetsprocess då vi har återgått till tidigare kod som skrivits och sedan strukturerat upp denna för att göra den mer lättförståelig och mer effektiv. På detta sätt har vi kontinuerligt använt oss av design improvement enligt en iterativ arbetsprocess. Vidare i enlighet med XP och även den iterativa arbetsprocess som jag nämnt tidigare så hade vi en kontinuerlig påbyggnad av applikationens funktionalitet. På grund av att vi arbetade så pass tätt, i stort sett i samma rum hela tiden, så lanserades en fungerande version av applikationen så fort en ny funktionalitet var klar. Möjligheten att arbeta på detta sätt byggde på att vi använde oss av *collective code ownership*, då koden ägs av alla medlemmar samtidigt. Detta åstadkom vi genom att använda oss av Git och GitHub.

Git var ett verktyg som inledningsvis var väldigt nytt för oss; vi hade inte använt det innan och förstod därför inte riktigt hur det skulle användas. Allteftersom vi tog till oss hur detta hjälpmedel skulle användas så insåg vi att det var en väldigt effektiv och hjälpsam teknik, och vi skrev därför tillsammans i gruppen ihop ett dokument "Git for Dummies" som är tänkt att i framtiden hjälpa studenter eller programmerare som är i samma sits som vi var, och förtydliga detta effektiva redskap för att utveckla mjukvara. Vi använde oss alltså av Git-kommandon via terminalen (eller Windows Powershell på Windows) för att ladda upp kod till GitHub så att alla i gruppen kunde ta del av denna kod. På så sätt ägde alla koden och alla kunde ta del av koden. Dock, på grund av vår brist på tidigare erfarenhet av Git, så utnyttjades inte Git fullt ut under projektets gång. Det tog ett tag för oss att komma in i det

och faktiskt använda olika grenar för olika funktionalitet, och inte bara använda oss av en huvudgren dit vi sparar all kod. Vi kunde alltså kommit igång tidigare med att använda detta verktyg, men vi har nu lärt oss att använda det, och det har varit oerhört värdefullt. Ytterligare ett redskap som användes för att strukturera upp och prioritetsordna våra User stories var Pivotal tracker, där vi kunde hålla koll på våra User stories, se om vi hade lyckats genomföra dem eller, hur viktiga de var och hur lång tid ett genomförande hade tagit.

Som tidigare nämnt så skrev alla gruppmedlemmar inledningsvis parallellt på sitt kandidatarbete. Detta innebar att fullt fokus inte kunde läggas på applikationen, och vi var istället tvungna att ta igen denna tid i slutet. Detta innebar att vi inte följde den aspekt av *sustainable pace* som finns inom XP. Vi arbetade dock alla inom gruppen lika mycket. Vissa dagar var det några som arbetade på andra jobb, men detta tycker inte jag påverkade arbetet överhuvudtaget, då dessa personer i så fall tog igen detta med råge senare på kvällar exempelvis. Uppskattningsvis arbetade var och en, under de första sex veckorna, 10 timmar per vecka. Under de sista tre veckorna arbetade var och en 70 timmar per vecka, vilket egentligen är alldeles för mycket. Dock innebar detta att vi landade på totalt 270 timmar per person och ett snitt på 30 timmar per vecka för hela perioden, vilket är bättre. Som jag sagt tidigare utgjordes majoriteten av tiden av parprogrammering. Ca 2/3 av den totala tiden som lades ner på att skriva kod gjordes med hjälp av parprogrammering, medan den sista tredjedelen gjordes enskilt, mestadels på kvällar. Totalt har vi dock alla arbetat lika mycket, och alla har visat ett oerhört engagemang och intresse för arbetet!

Jag kan tänka mig att Scrum kan passa väldigt bra som arbetssätt, men den ställer höga krav på den grupp som ska använda det. Fördelen med Scrum är att man får en väldigt tydlig arbetsuppdelning och en tydlig struktur på vad som ska göras och när det ska göras. Om de dagliga mötena hålls på rätt sätt och uppskattningen av tiden uppgifterna tar görs realistiskt så kan nog ett effektivt arbetssätt erhållas. Vi såg tidigt i vår process att vi, genom att använda Scrum, fick en tydlig bild av vad som skulle göras och till nära, vilket bidrog till en bra struktur. Däremot innebär Scrum höga krav på de som ska använda denna teknik. I vårt fall passade det inte alls bra. Vi märkte att det var för oss, men ingen tidigare erfarenhet av denna teknik, svårt att uppskatta den tid och arbete som skulle krävas för varje arbetsuppgift. Dessutom märkte vi att de uppgifter som delat upp inom gruppen var för svåra för oss som enskilda programmerare att lösa, eftersom vi inte innan kursens början befann oss på den kunskapsnivå som hade varit nödvändig för att kunna implementera ett effektivt Scrum arbetssätt. Vi insåg helt enkelt att Scrum gjorde att vi blev för isolerade från varandra, vilket ställde höga krav på oss som gruppmedlemmar att ha en god kommunikation, vilket försvårades ytterligare av den höga arbetsbelastning vi hade i vårt kandidatarbete samtidigt som inledningen av detta projekt.

Baserat på de erfarenheter vi har av Scrum skulle jag därför inte rekommendera att projektgrupper som befinner sig i liknande situation som vi gjort i vårt projekt att använda sig av Scrum. Ett arbetssätt enligt Scrum ställer högre krav på medlemmarna i en sådan grupp, och jag anser att det krävs en högre kunskapsnivå och erfarenhet av programmering än den vår grupp besatt i inledningen av arbetet. Jag skulle dock vilja påpeka att det finns en risk att implementera Scrum, även om gruppmedlemmarna är mer erfarna, och det är risken att de olika medlemmarna blir isolerade och det ökar risken för dålig kommunikation, kodkonflikter och dubbelarbete. Vi har lyckats ha en väldigt bra kommunikation och hög gemensam förståelse för projektet inom gruppen, tack vare att vi har arbetat med just XP.

Jag skulle därför rekommendera att även om Scrum används, så att använda kanske vissa inslag av XP, exempelvis mer frekventa möten, eller gemensam programmering några dagar i veckan, för att förbättra den interna kommunikationen.

Vidare skulle jag alltså rekommendera att använda XP i liknande fall som vårt. Fördelen med att använda XP har i våra fall varit många. Genom att programmera tillsammans kunde vi åstadkomma ett betydligt bättre resultat än om vi hade programmerat mer individuellt. Denna gemensamma programmering innebar att kunskap kunde delas mellan gruppens medlemmar, vilket bidrog till att den totala kunskapsnivån inom gruppen kunde höjas, och vi kunde tillsammans prestera bättre. Denna delning av kunskap ledde även till att alla gruppens medlemmar tog del av varandras kunskap och utvecklades under kursen. Jag personligen känner att jag har utvecklats något enormt i och med denna kurs, och jag är övertygad om att jag inte hade utvecklats på samma sätt om vi som grupp inte hade använt oss av XP. Som tidigare nämnt bidrog också XP till att vi fick en väldigt homogen och enhetlig förståelse för projektet och arbetet, då vi alla hela tiden befann oss på samma ställe och jobbade; frågor kunde redas ut med en gång, viktiga beslut kunde tas direkt och den interna kommunikationen blev väldigt bra, vilket ledde till en välstrukturerad och effektiv arbetsprocess. Dessa har varit de huvudsakliga positiva aspekterna med att använda XP i vårt fall. Jag skulle därför rekommendera XP, inte bara till grupper som befinner sig i liknande fall som oss själva, utan även till andra mer erfarna grupper av programmerare, då jag är övertygad om att detta arbetssätt förenklar den interna kommunikationen och möjliggör en effektivare beslutsprocess och bättre problemlösningsmöjligheter.

För oss är det inte mycket som har fungerat dåligt med XP. Vi har dock märkt att, genom vårt nära samarbete, att vi har blivit något bortskämda gällande den goda interna kommunikationen, då det var väldigt tydligt att problem och kodkonflikter uppstod betydligt enklare då någon medlem arbetade individuellt under någon dag. Ytterligare ett problem har väl varit att uppdelningen av arbetsuppgifter inte alltid varit så tydligt isolerade som kanske hade varit fördelaktigt; vissa av uppgifterna har ibland berott av andra uppgifters resultat, vilket har gjort att vissa i gruppen kanske inte kunde jobba fullt ut under en kort period. Det skulle även kunna vara så att genom att enbart använda parprogrammering så skulle några personer i gruppen kunna klara sig igenom ett projekt utan att tillföra så mycket själva. Detta upplever jag dock inte har varit något problem överhuvudtaget, då vi har bytt konstellationer i parprogrammeringen väldigt ofta vilket har gjort att alla har varit tvungna att bidra. Framförallt skulle jag dock vilja påstå att detta inte har varit något problem då jag upplever att alla i gruppen har visat ett oerhört engagemang och intresse för arbetet.

Sammanfattningsvis är jag väldigt nöjd med gruppens arbete. Vi har haft en grupp som har haft väldigt god stämning och som har presterat väldigt bra utefter våra premisser på förhand. Vi har lärt oss väldigt mycket och tagit till oss olika arbetssätt för hur man löser komplexa projekt av denna kaliber vid utveckling av programvara. Jag tycker att vi till slut landade i den för vår grupp bästa arbetsmetoden, nämligen XP. Jag tycker att vi har jobbat väldigt bra tillsammans som grupp; vi har haft väldigt god kommunikation och haft en effektiv struktur där viktiga beslut har tagits snabbt och effektivt. Vi har även haft en välfungerande arbetsuppdelning, och en väldigt bra lösning med parprogrammering för att både bättre kunna utveckla vår applikation, men samtidigt effektivare kunna sprida kunskap inom gruppen. Vårt samarbete har fungerat väldigt bra, och jag tycker inte det är någon som har tillfört mindre eller inte fungerat att arbeta med, utan vi har haft en genomgående god

stämning, och framförallt upplever jag att alla i gruppen har haft samma iver och engagemang att producera en bra applikation, vilket har varit väldigt inspirerande för mig själv. Jag är personligen också väldigt nöjd med att vi använde XP, då jag har lärt mig och utvecklats något oerhört i denna kurs, vilket inte hade varit möjligt om vi inte hade använt oss av XP. Mer konkret skulle jag vilja påstå att det faktum att vi relativt tidigt i arbetet inkorporerade testningar av applikationens funktionalitet, samt inledde arbetet med dokumentationen, var gynnsamt för vårt senare arbete.

Vad man kan säga som gjordes mindre bra i vårt fall och som borde kunna utvecklas är det faktum att vi inte kom igång med arbetet riktigt förrän efter några veckor. I vårt fall berodde detta väldigt tydligt på det faktum att vi alla lade mycket fokus på vårt kandidatarbete under den inledande fasen med utvecklingen av applikationen. Detta var inte optimalt, och arbetet hade kommit igång bättre, och vi hade undvikit att behöva arbeta så mycket i slutet, om vi hade kommit igång med arbetet tidigare. Att tidigare fastställa vilken arbetsprocess som ska användas, istället för att byta efter några veckor, och att tidigare fastställa vilket designmönster som ska användas, borde vara gynnsamt för projektet. Dock vill jag påstå att vi har lärt oss mycket på att inledningsvis försöka oss på en process som vi sedan inte använde; på så sätt lärde vi oss när denna process skulle kunna vara fördelaktig att använda och när den inte borde vara det, vilket också är en lärdom att ta med sig från denna kurs. En mer konkret aspekt som kunde gjorts bättre är det faktum att vi inledningsvis inte kommenterade tillräckligt i vår skrivna kod. Ibland kunde det vara väldigt svårt att sätta sig in i kod som skrivits av någon annan i gruppen. Detta underlättades av det faktum att vi ofta arbetade i samma rum under dagarna, men under kvällar kunde det vara ett problem. Detta förbättrades dock efterhand. I ett framtida projekt skulle jag därför sträva efter att komma igång tidigare, och mer kontinuerligt använda kodkommentering. Vidare skulle jag dock vilja implementera de aspekter som vi lyckats med i vårt projekt, nämligen ett effektivt arbetssätt enligt XP, vilket lett till ökad delning av kunskap inom gruppen, samt en effektiv arbetsstruktur och god intern kommunikation.

Rörande kursen som helhet tycker jag att kursen har varit en av de mest givande för mig personligen. Jag har tagit med mig väldigt mycket lärdomar, både rent programmeringsmässiga, men också hur man strukturerar och genomför ett mjukvaruutvecklingsprojekt, vilket har varit väldigt lärorikt och kommer troligen vara ovärderligt inför framtiden. Jag tycker att vår handledare Max har gjort ett jättebra jobb med vår handledning. Han har hela tiden kunnat svara väl på frågor vi har lagt fram till honom, och samtidigt på ett konstruktivt sätt bolla idéer med oss i gruppen, samtidigt som han också har skrivit kandidatarbete. Det han har gjort extra bra är att han tagit hänsyn till vår kunskapsnivå inom området sedan tidigare, och det har varit väldigt hjälpsamt. Dock skulle jag vilja påpeka att den första föreläsningen med Git kunde innehållit ännu mer grundläggande beskrivningar av Git och hur det används. Det kan verka banalt, men vi i vår grupp hade inte använt detta verktyg alls tidigare, och därför blev den föreläsningen svårförstådd. En tydligare introduktion till vad Git faktiskt är och hur man kommer igång med det hade hjälpt avsevärt, och förkortat tiden det tog för oss att komma igång och använda det. Sammanfattningsvis skulle jag dock vilja påstå att det har varit en oerhört lärorik och givande kurs som har gett mig lärdomar som kommer vara användbara i framtiden. Framförallt har det gjort mig fortsatt intresserad av ett område som jag tidigare inte haft mycket erfarenhet inom.