

POST MORTEM REPORT

1. Which processes and practices did you use in your project?

Vi har träffats minst två gånger per vecka i grupp och diskuterat appens arkitektur, funktioner samt implementation. Vi använde oss av Scrum och hade veckovisa sprintar. På tisdagar avslutade och startade vi varje sprint, delade upp "User stories" i små uppgifter och tilldelade oss själva de uppgifterna.

För att underlätta arbetet och se till att inte folk gör exakt samma sak använde vi oss av verktyget Trello. Där hade vi alla "User stories" som etiketter och kunde även markera dessa med våra namn. Vi kom överens om att tydligt markera vad som skulle göras, vad som gjordes och vad som hade gjorts.

I början av utvecklingsprocessen jobbade alla självständigt på de små uppgifterna och lärde sig Android stegvis. Efter de flesta möten hade vi en egen workshop där vi programmerade tillsammans och tog tag i de uppgifter som diskuterats under tidigare möte.

Efter två veckor var vi färdiga med de större delarna av appen. Uppgifterna som var kvar var mer utmanade, både i storlek och svårighetsgrad, och därför förändrade vi vårt arbetssätt. Vi samarbetade två och två för att kunna klara av uppgifterna på ett bra sätt.

2. 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?

Varje individ hade totalt cirka 4 timmars schemalagda kursmoment per vecka. Våra möten/workshops varade i snitt mellan 2 och 3 timmar och vi träffades på detta formatet 1-2 gånger per vecka. Var och en av deltagarna har sedan jobbat hemma i relativt stor utsträckning. Totalt har varje individ jobbat ungefär 15 timmar per vecka vilket summerar till 300 timmar totalt. Detta är alltså inte tid lagt endast på appen utan vad vi totalt lagt på kursen.

Gruppmedlemmar

Maxim - 4 timmar föreläsningar + 4 timmar möte + 7-8 timmar per vecka. = 15-16 timmar/vecka.

Alexander K - 4 timmars tråkiga föreläsningar + 4 timmar möte + 7 timmar = 15 timmar/vecka

Kevin - 4 timmar föreläsningar + 4 timmar möte + 8 timmar per vecka = 16

David - 4 timmar föreläsningar + 4 timmar möte + 6-7 timmar per vecka = 14-15

Alexander H - 4 timmar föreläsningar + 4 timmar möte + 6 timmar per vecka = 14

3. For each of the techniques and practices used (such as standup meetings, pair programming, TDD, etc.) in your project you should answer all the questions:

- i. What was the advantage of this technique based on your experience in this assignment?**
- ii. What was the disadvantage of this technique based on your experience in this assignment?**
- iii. How efficient was the technique given the time it took to use?**
- iv. In which situations would you use this technique in a future project?**
- v. In which situations would you not use this technique in a future project?**
- vi. 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?**

Traditionella möten

Fördelen med möten enligt oss var att det var lätt att kommunicera med varandra. Vi kunde lätt gestikulera för att förklara något och hade alltid en tavla tillgodo, där alla kunde måla upp viktiga punkter. Det var lättare för oss att fokusera med en bra arbetsmiljö. Alla blev också nöjda med de beslut som togs efter varje möte eftersom alla fick vara delaktiga.

Vissa tillfällen var det jobbigt att ta sig till skolan eftersom vi bor i olika delar av Göteborgsområdet och ibland var det därför svårt att hitta en tid som passade alla. Bokning av lokal var också ett problem eftersom de ofta var upptagna och vissa lokaler var trånga. Möten kunde ibland spåra ur och leda till diskussion om annat, vilket gjorde att det kunde ta lång tid innan man kom fram till viktiga beslut.

Det tog oftast lite längre tid att ha möten i skolan än jämfört med att ha Skypemöten som vi ibland hade. I längden gav det dock mer eftersom alla i gruppen var mer säkra på vad de

skulle göra tills nästa möte. Var någon osäker och hade frågor var det lätt att besvara frågorna genom redovisning på tavla eller ha en öppen diskussion där inga pratade rakt på varandra. Det hjälpte oss också att hålla deadlines och ha en bra struktur på arbetet. Då vi hade Skypemöten upplevde vi att det var svårare att upprätthålla en fokuserad diskussion.

Det finns tillfällen där användning av Skypemöten är mer lämpliga. Det kan handla om att gruppmedlemmar är utspridda över hela världen, vilket hade gjort traditionella möten praktiskt svåra. Skypemöten kan även funka bra som korta standupmöten.

Möte varje vecka är något som vi skulle fortsätta använda i de flesta grupparbeten i framtiden eftersom det är ett bra sätt för att diskutera frågor och dela med sig av idéer. Det gäller dock att det inte är för många i gruppen eftersom det kan bli svårt att hålla en gemensam konversation med alla medlemmar.

Bor gruppmedlemmarna nära varandra är tekniken ett väldigt bra sätt att jobba på eftersom det gick fortare framåt, möten hjälpte alla medlemmar att jobba mot samma mål, istället för att låta medlemmar glida iväg eftersom alla har olika uppfattningar om saker. Dvs med hjälp av möten kunde vi sätta en standard.

Workshop

Vi anser att en fördel med workshop var att det är relativt effektivt och vi fick gjort mycket under dem. Vi kunde par-programmera, ställa frågor i gruppen och diskutera. Om en enskild person fastnade på ett problem kunde vi lätt lösa det tillsammans. Det var en god arbetsmiljö under workshoppen eftersom vi alltid satt i ett grupprum vilket hjälpte med motivationen. Alla arbetade mycket tillsammans, vilket gjorde att vi lärde känna varandra bättre samt bli van vid varandras tankesätt. Detta gjorde att vi producerade bättre kod som inte behövde refaktoreras.

Nackdelen med workshoppen var att mötena lätt kunde gå off-topic och kunde spåra ur när vi lärde känna varandra bättre. Det gjorde att det ibland var mer snack än kod.

Vi rekommenderar denna teknik när man behöver programmera, det är perfekt att ha tid utdelad för att programmera och gör det enkelt att komma igång med koden.

Parprogrammering

Vi tyckte parprogrammering var ett bra sätt att lösa lite svårare koduppgifter. Ibland hade vi olika kunskaper om olika områden och man kompletterade varandra när man programmerade ihop. Vi kunde diskutera mer komplexa logiska problem och komma fram till lösningar snabbare än om vi hade kodat en och en.

Har man mindre eller lättare uppgifter kan det vara mindre effektivt då den ena förmodligen hade kunnat lösa uppgiften precis lika bra själv medan den andra hade kunnat jobba på något

annat. Det finns en risk att en person blir sittande sysslolös om man parprogrammerar exempelvis getters och setters.

Man kan applicera detta på många olika sorters projekt där det för vissa uppgifter krävs att man är två som tänker. Tekniken är också användbar i projekt där man jobbar länge, då kan t.ex. en "senior developer" parprogrammera med en "junior".

Om vi inte använt den här tekniken skulle det i vissa fall tagit längre tid att koda för att vi hade varit tvungna att söka information, undersöka om det funkar, refaktorera tills det stämde överens med standarder och följde alla konventioner.

Scrum

Fördelen med Scrum är att fokuset alltid ligger på en särskild del av projektet i taget, vilket gav oss snabba resultat. Istället för att göra större delar som sedan skulle sättas ihop till en helhet hade vi hela tiden mindre delar som fungerade, vilket minskade riskerna att det inte skulle funka i slutet av projektet. Det var också enkelt att se vilka uppgifter som var lediga att jobba på, vilket minskade chanserna att två personer tog samma uppgift.

Om vi överskattade vår förmåga klarade vi inte av en sprint och fick inte något levererat. Om vi underskattade vår förmåga blev vi klara för snabbt och hade inget att göra tills nästa sprint. Båda fallen var problematiska, även om den första var lite värre. Det var därför väldigt svårt, speciellt i början, att hamna på rätt nivå. Allt eftersom blev det lättare att uppskatta sin hastighet och sprintarna blev bättre avvägda.

Med hjälp av Scrum metodens snabba resultat blev gruppen mer motiverad att fortsätta och samarbetet blev bättre.

Är projektgruppen väldigt stor kan det vara svårt att uppskatta tiden. Om detta är ett problem kan man alltid dela upp gruppen mindre lag och låta dem ansvara för oberoende delar av projektet. Ett annat tillfälle där vi skulle välja att undvika Scrum är om projektet är av liten i storlek eftersom projektet inte behövs delas upp i olika delar.

Scrum var lite svårt att komma igång med, orsaken var att man vill se snabba resultat men samtidigt designa appen så att det var lätt att bygga på ny funktionalitet. En av anledningarna att vi inte klarade av vår första sprint var att vi fokuserade på att göra en bra grund för applikationen, vilket tog mer tid än förväntat. När vi väl har kommit igång med Scrum gick det väldigt snabbt att integrera nya funktioner i appen. Möten och Scrum berodde väldigt mycket på varandra, för att Scrum skulle gå bra var vi tvungna att komma överens varje möte vad vi skulle göra i projektet. Samtidigt väger varje lagmedlems insats väldigt mycket. Alla använde Trello vilket gjorde det enkelt för hela gruppen att veta vad som pågick i utvecklingen.

Om man jämför med en annan projektkurs, TDA367 Objektorienterad programmeringsprojekt, där vi inte använde Scrum tog det ungefär fem veckor innan vi fick se något färdigt. Problemet var att vi började på allt samtidigt och fokuserade väldigt mycket på backend och modellen.

4. What worked well in how you worked in this project? (No need to state what you already wrote for 3 again.)

N/a

5. What did not work well in how you worked in this project? (No need to state what you already wrote for 3 again.)

Vi tyckte att det var svårt att utse en gruppmedlem till produktägare när vi använde Scrum eftersom det var vi själva som stod för idén av appen, inte någon utanför gruppen. Detta gjorde att det blev väldigt svårt att kunna skilja på rollerna som utvecklare och produktägare, och rollen som produktägare blev då helt onödig.

6. Reflections over non-process specific decisions (such as non-mandatory tools, APIs etc. but also aspects like training and 'fika')

Trello

Vi använde oss av en tjänst som heter Trello. Det är ett verktyg för projekthantering som är väldigt flexibelt. Det var väldigt enkelt att använda, det räcker att man skriver in text på olika etiketter och sedan "drag and drop":ar där man vill ha det. Trello är väldigt användbart, men endast om alla i gruppen använder det. Det är mycket lättare att ha koll på vad som görs i projektet, men då är alla ansvariga att fylla in etiketter och flytta runt dem. I vår grupp funkade det som det skulle, då vi sedan dag ett kom överens att alltid använda och uppdatera uppgifter på Trello. Motiveringen för detta var att om man inte följer den regeln så skulle man bjuda på fika nästa gång.

Slack

För att hålla kommunikationen på topp använde vi oss av Slack. Tjänsten kan enkelt beskrivas som en live-chatt som arkiverar dina meddelanden. Slack funkade bra då den hade notifikationer som skickades till våra telefoner när någons namn blev nämnd i chatten. Ifall någon jobbade vidare på någon annans kod och undrade något så var det bara att fråga på

Slack och man fick alltid snabb respons tillbaka. Vi kom överens om att installera det på våra telefoner för att få notifikationer, eftersom alla av oss har alltid telefon med sig.

Google drive

Användes som lagringsutrymme för våra dokument, t.ex Konceptbeskrivning, filmmanus. Fördelen med det är att alla kan jobba där samtidigt och lätt se historiska ändringar som har hänt.

Fika

För att inte fastna på samma problem allt för länge är det bra att ta en liten paus i form av fika. Fika hjälpte oss att lära känna varandra bättre, gav oss tid att diskutera men gav oss också näring så vi kunde arbeta mer.

Facebook API

Facebook API:t var förvånansvärt bra; de har ett verktyg som heter "Graph Explorer" där du kan testa dina anrop och se resultat, samt generera koden för det anropet. Facebook har väldigt mycket dokumentation och kodexempel vilket gjorde att det inte tog mer än en timme att komma igång med det.

Google Maps API

Google Maps API var enkelt att arbeta med och var väl integrerat med Android. Vissa grundläggande funktioner som att flytta en markör programmatiskt saknades dock förvånansvärt nog men gick att arbeta runt tämligen smärtfritt (I detta fall genom att ta bort den gamla markören och sätta ut en ny).

Träning

Vi lärde oss Android bl.a genom "Android Developer API Guide", den kändes dock ibland inkomplett med lite få exempel vilket ledde till att det tog mer tid för oss. Samtidigt var inte all dokumentation uppdaterad med den nya versionen av Android, vilket gav oss en del problem, t.ex med rättigheter i Android. I den nya versionen av Android frågas det efter rättigheter efter behov och inte direkt vid nedladdningen. Det var ett problem som tog lång tid att upptäcka och som var problematiskt att lösa på grund av den bristfälliga dokumentationen.

7. How did you work together as a group in the project?

What worked and not in your interaction(s)?

Vi kom bra överens om hur vi skulle fördela arbetet varje vecka tills nästa sprint men det var svårt att uppskatta varje veckas arbetsinsats. Den första veckan överskattade vi vår förmåga

och blev inte klara med det som var planerat, vi lärde oss dock snabbt hur vi skulle hantera detta och fick en mer rimlig bild av hur mycket vi egentligen hann med. Det var även lätt att samarbeta med varandra och det var inga bekymmer att fråga om saker man inte förstod. Alla hjälpte till om någon fastnade på något och problemen löstes tillsammans via parprogrammering eller öppen diskussion.

Värt att nämna var att vi hade sämre koll på hur vissa delar av appen fungerade eftersom olika personer arbetade med olika delar av appen. Detta hade kunnat skapa problem om någon av oss hade varit tvungen att lämna projektet. Som tur var behövde vi inte uppleva detta.

8. What would you do differently in a future but similar project?

Vi skulle försöka komma igång tidigare med inläringen av Android och inte väntat så länge med konceptidén. Det var bland annat orsaken till att vi inte klarade vår första sprint.

Vi skulle gjort fler undersökningar i liknande framtida projekt, för att ha bättre underlag för behovet av vår app. Vår idé var resultatet av en brainstorm och var något vi själva trodde skulle vara populärt, men vi gjorde inga konkreta undersökningar för att bekräfta om det vore fallet. Vi insåg detta under presentationen för tävlingsjuryn, som ställde frågor om våra obefintliga undersökningar.

Vi borde ha testat mer i en riktig miljö, då det finns faktorer vi inte tog hällsyn till när vi simulerade. Ett sådant fall var om två bussar stod nära varandra fast skulle åka åt olika håll. Vi upptäckte även kritiska buggar väldigt sent in i processen, som vi borde ha löst betydligt tidigare.

Vi hade även försökt arbeta mer iterativt i Git-repot och använda olika branches som ett verktyg i vår Scrum-process. Vi gjorde det mesta arbetet i en branch trots att vi efter varje sprint alltid hade något körbart, vi borde oftare ha flyttat koden till en Master-branch så vi alltid hade en version av appen som kunde köras.

Vi hade svårt att uppskatta hur mycket tid vi lagt ner på projektet både under och efter projektets slut. Detta försvårade Scrum-processen ytterligare, hade vi gjort om projektet hade vi fört någon form av personliga loggar för att bättre kunna uppskatta hur mycket tid var och en lagt ner och därefter använda informationen för att planera framtida sprintar.

9. How did you handle the workshops organised as part of the innovation challenge? Describe how you selected team representative(s) and the motivation behind the selection procedure, how you shared stakeholder feedback within the team, what you would do differently next time and if you would have preferred another way of interacting with stakeholders and support personnel (such as student supervisors, teachers as customers/supervisors etc).

Vi valde lagrepresentanter för workshopen genom att komma överens på ett möte som vi hade den första veckan av kursen och valde att ha samma två personer för varje workshop, vilket gjorde det lättare att dela upp arbetet eftersom resten av gruppen kunde fokusera mer på att utveckla appen. Vi kände dock att vi klarade oss utan den första workshopen eftersom vi redan hade bestämt konceptet för appen och även den tredje workshopen som var mest allmän information vilket vi inte kände gav oss något, utan vi spenderade hellre mer tid på att koda dessa dagar.

Den andra workshopen hade kunnat vara väldigt bra eftersom man kunde ställa frågor om själva plattformen, frågorna som vi hade var dock redan besvarade eftersom workshopen förekom något sent i kursen. Det vi kände som gav mest feedback var den sista workshopen, där tre personer från gruppen redovisade demon, eftersom företagare gick runt och ställde bra frågor om själva utvecklingen, samt om hur vi hade tänkt marknadsföra appen. Feedbacken skrev vi ner direkt på ett gemensamt Drive dokument efter varje företags person hade gett sina åsikter och tankar. Dagen efter hade vi ett gemensamt möte där vi gick igenom feedbacken och diskuterade hur vi skulle bemöta den i vår presentation inför juryn, samt vad vi behövde tänka på inför framtida utveckling.

I helhet var workshopen ett bra sätt för att få kontakt med företag och få tips om hur appen skulle vidareutvecklas till exempel vilka funktioner appen skulle ha. De gav även viktiga tips om marknadsföring, till exempel vilka ambassadörer man skulle välja sig till för att sprida appen. Vi kände dock att workshops generellt förekom något sent i kursen och hade hellre velat ha dem tidigare för att ha kunnat utnyttjat dem bättre. Annars var det ett bra och enkelt sätt att kommunicera med företagen.