Final Team Reflection - Team Abacus

DAT257 - Agile Software Project Management

May 2021

Izabell Arvidsson

Moa Berglund

Marcus Randevik

Lisa Samuelsson

Amanda Styff

Noa Tholén

Kerstin Wadman

1 Customer Value and Scope

1.1 The chosen scope of the application under development including the priority of features and for whom you are creating value

1.1.1 Sammanfattning av projektet (A)

Vår kund, Lisa Eskilson, ville byta lägenhet inom SGS, och insåg att det enda sättet att komma i kontakt med andra från SGS var att sätta upp en lapp i byggnaden. Efter att vi pratat med kunden kom vi fram till att det vore smidigt med en app där man kan publicera sin lägenhet som en annons, och därmed hitta en bredare publik. Appen i sig erbjuder inte själva bytet, utan på appen hittas andra som kan tänkas byta lägenhet.

Till en början tog vi fram olika funktioner som skulle skapa värde för kunden. Ett annonsflöde, möjligheten att filtrera på olika stora lägenheter, och att kunna skapa nya annonser var de första funktionerna som bedömdes som viktiga. Dessa funktioner implementerade vi under projektets gång, och presenterades för kunden varje vecka i samband med sprint reviews. Utöver tidigare nämnda funktioner var ett av kundens förslag en like-knapp, där man kan "gilla" en annons, så att den sparas i en specifik lista under en flik. Även möjligheten att ta bort sin annons ansågs viktig enligt kunden, vilket vi höll med om.

Från början tänkte vi även ha med funktioner såsom att logga in och ändra sin profil, samt en chattfunktion. Detta presenterade vi för vår kund och hon tyckte att det lät som bra funktioner. Däremot insåg vi relativt snabbt att det var för tidskrävande för att vi skulle kunna leverera dessa funktioner under projektet, eftersom de sågs som "nice to have" och inte viktiga för en minimal viable product (MVP). Genom att kommunicera och diskutera med kunden kom vi fram till kompromisser, där chattfunktionen kunde ersättas med att visa telefonnummer och email, vilket kunden såg som en mycket bra lösning på problemet. Scopet i sig förändrades därmed inte i stor utsträckning under projektets gång, eftersom vi kunde leverera den funktionalitet och det kundvärde som vi hade kommit överens om från början. Däremot var vi tvungna att välja bort vissa funktioner, då de inte bidrog tillräckligt mycket till kundvärde kontra implementeringstiden jämfört med någon annan funktion.

Den prioritetsordning som vi valde från början var snarlik till hur vi valde att fortgå med projektet, där vi började med det som vi bedömde som viktigast. Därmed utvecklade vi menyn, annonslistan, filtrering till en början. Under projektets gång tillkom en del user stories, vilket innebar att vi var tvungna att omprioritera tillsammans med kunden för att få den funktionalitet i appen som både gruppen och kunden var nöjda med. En diskussion med kunden uppkom då gruppen ville fokusera mer på appens design i ett tidigt skede, och då var vi tvungna att ta ställning till hur det påverkar prioriteringen. Efter att ha diskuterat ämnet med kunden kom gruppen gemensamt fram till att fokuset skulle ligga på funktionalitet, för att appen skulle generera maximalt kundvärde då. När projektet närmade sig slutet, så tog vi

upp ämnet återigen för diskussion med kunden, och då blev det uppenbart att design var viktigt för att appen skulle vara tillräckligt användbar för kunden i fråga, vilket innebar att vi omprioriterade våra user-stories för att möta kundens krav.

Det värde som skapas med appen, är att studenter som bor i SGS lägenheter har möjlighet att hitta en större marknad för byte av lägenheter. Kundvärdet finns därmed inte enbart hos vår kund Lisa, utan hos alla studenter som bor i SGS-bostäder med intresset. Appen har ingen begränsning gällande vilken typ av lägenhet som förmedlas, vilket ger en hel del möjligheter. Detta betyder att appen lätt kan byggas vidare på för att erbjuda lägenheter som tillhör Chalmers Studentbostäder eller liknande. På så sätt skapar appen kundvärde för många.

1.1.2 Hur bör framtida projekt se ut? (B)

En nyttig lärdom att ta med sig till liknande framtida projekt är att börja med ett mindre scope och sedan bygga på det efter hand i mån om tid. Det kan vara svårt att uppskatta tidsåtgången i början av projektet och att börja litet underlättar för leveransen av värde varje sprint och för att kunna göra en lämplig avvägning mellan funktionalitet och design.

Genom projektets gång har vi haft en kontinuerlig och nära kundkontakt för att kunna leverera värde till kunden varje sprint. Utifrån Lisas feedback har vi omprioriterat ordningen på user stories, valt hur vi ska prioritera mellan funktion och design, samt valt bort vissa funktioner såsom chatten. Det är av stor vikt att inkludera kunden under hela projektets gång eftersom värde för kunden kan ändras eller omvärderas jämfört med utgångsläget.

1.1.3 Vad krävs för att nå dit? (A till B)

För att i framtiden kunna ta fram rimliga scope och prioriteringsordningar behövs en del färdigheter och taktiker. Dels är det viktigt att undersöka i förhand hur svåra funktioner är att implementera, och lägga scopets nivå därefter. Utan att utvärdera komplexiteten hos funktionerna är det svårt att inte behöva göra ändringar i scopet under projektets gång, och därför tror gruppen att dessa utvärderingar kan vara värdefulla.

Utöver undersökningar ses även utförlig kommunikation med kunden som en viktig faktor för att bestämma rätt nivå på scopet. Med detta menas att diskussioner förs kring hur en minimal viable product skulle se ut, och i vilken omfattning design och funktionalitet ska utvecklas. Därmed kan scopet vara mer representativt för vad som kan levereras, och det medför också att prioriteringsordningen för user stories kan bli mer effektiv.

Slutligen ses planering och kundkontakt som två viktiga sätt för att uppnå maximalt kundvärde. Med detta menas att inför och under projektet sker planering och kontinuerlig kundkontakt, för att säkerställa att det som utvecklas och fokuseras på genererar kundnytta i slutändan. Det kan vara ett sätt att minimera onödigt arbete, för att fokusera på att skapa en produkt som kunden är nöjd med, som i slutändan kan ge utrymme för ännu bättre funktionalitet eller snyggare design.

1.2 The success criteria for the team in terms of what you want to achieve within the project

1.2.1 Sammanfattning av projektet (A)

Vi har haft två övergripande success criteria i projektet. Den första har varit att kunna leverera en applikation som uppfyllde kundens och våra förväntningar, och den andra har varit att lära oss om, och använda oss av, Scrum och Agile Project Management. Vi ville att samtliga gruppmedlemmar vid projektets slut skulle känna att de skulle kunna och vilja genomföra liknande projekt framöver.

Under planeringen av varje sprint har vi poängsatt de user stories som skulle kunna göras under veckan för att kunna uppskatta hur mycket tid och resurser en uppgift behöver. Målet har varit att prestera runt 100 poäng per vecka. Även om vi för det mesta blev bättre på att sätta passande poäng med tiden var det ibland svårt att förutse hur tidskrävande en user story skulle vara, eftersom det ibland var svårt att hitta bra lösningar till problem som dök upp.

Fler success criteria som vi har haft är att våra KPIer ska se bra ut och utvärderas i slutet av sprinterna. Ett annat kriterium har varit att alla ska känna sig nöjda med sin insats och arbetsbelastning, och att det ska kännas bra i teamet och att alla är involverade. Det har vi åstadkommit genom planerings- och utvärderingsmöten samt parprogrammering. Samtliga i gruppen har varit involverade i alla projektets moment. Utöver detta har vi även lagt in vår Definition of Done som ett success criteria.

1.2.2 Hur bör framtida projekt se ut? (B)

Vi tror att det är bra att ha success criteria som inkluderar punkter som relaterar till koden och produkten som ska levereras. Med det är också bra med punkter som rör gruppens samarbete och gruppmedlemmarnas uppfattning och känslor angående projektet, för att skapa och bibehålla ett hälsosamt arbetssätt.

Vi tror att det är viktigt med utvärdering av de framsteg som gjorts under varje sprint men också utvärdering av samarbetet i gruppen. I vårt projekt har vi haft en väl fungerande grupp med god kommunikation och där alla gick in med liknande förväntningar på projektet. I framtiden kan det hända att individerna i projektgruppen har olika uppfattning om hur projektet ska se ut. Ett tydligt scope med väldefinierade mål underlättar samarbetet och gör att gruppen kan leverera sina mål.

1.2.3 Vad krävs för att nå dit? (A till B)

För att ha en gemensam uppfattning av projektet krävs en god planering från start där gruppen bestämmer success criteria gemensamt. För att se till att alla gruppmedlemmar är överens om vad som ska uppnås är det bra med ett tydligt scope och att gruppen går igenom det i detalj i ett tidigt skede.

Det är också viktigt att gruppen lär känna varandra och varandras förmågor eftersom alla kommer från olika erfarenheter. På så sätt kan gruppen skapa en god arbetsmiljö där alla kan bidra för att nå upp till de mål som sätts och maximera sin learning outcome.

1.3 Your user stories in terms of using a standard pattern, acceptance criteria, task breakdown and effort estimation and how this influenced the way you worked and created value

1.3.1 Sammanfattning av projektet (A)

I projektet använde vi user stories för att bryta ner större epics i mer lätthanterliga mindre uppgifter. Med detta menas att vi hade exempelvis en epic som kallades för *"Skapa annons"* där user stories såsom *"Ta bort annons"* och *"Lägga till bild i annons"* ingick. Våra user stories skulle vara utformade för att kunna bli gjorda på 1-2 dagars arbete, vilket i majoriteten av fallen stämde. Många av de user stories som vi gjorde skapades i början av projektet när vi definierade de funktioner som skulle finnas med, däremot tillkom flera efter vägen, eftersom problem och idéer kom fram under projektets gång. Det framkom även under projektet att några user stories som vi definierat från början var alltför omfattande, vilket medförde att vi delade upp dem i mindre. Detta såg vi som nödvändigt, eftersom det är viktigt att hålla tempot som gruppen bestämt sig för, men också för att det ingår i det agila arbetssättet.

Den effort estimation som gjordes för våra user stories blev med hjälp av poängsättning (scrum poker). Poängsättningen grundades i att ungefär 100 poäng skulle hinna göras under en sprint. Först diskuterades den user story som skulle poängsättas, där alla fick säga sitt om hur de bedömde att svårighetsgraden var, och därefter skickade alla sina poäng samtidigt i en chatt. Därefter analyserades poängen, och om det var skilda uppfattningar om vilka poäng som user storyn krävde så fick alla förklara varför de poängsatte som de gjorde. Detta ledde ofta till att gruppen fattade beslut om att ändra poängsättningen, men i andra fall så behövdes inte detta. Gruppen bedömde det som både effektivt och rättvist, och genom denna typ av poängsättning kunde vi därmed säkerställa att alla bedömde sina uppgifter som rättvisa.

Våra user stories är skrivna med ett relativt enkelt språk, för att minimera eventuella missförstånd inom teamet, men också mellan gruppen och kunden. Kunden kunde därmed få en god uppfattning för vad varje user story innebar, och kunde därmed prioritera dem rättvist.

De acceptance criteria som använts för våra user stories har varit tydliga, där vi diskuterat tillsammans som grupp exakt vad som ska göras. Gruppen har också varit noga med att göra exakt det som står i acceptance criteria, och inte mer. Detta såg gruppen som oerhört viktigt, då det är en princip inom agilt arbete som underlättar mycket för att inte få merge-konflikter när man är flera utvecklare. Vi blev med tiden bättre på att tydligt formulera det som skulle bli gjort. Ett exempel på detta är att i projektets start så definierades inte om en användare skulle vara hårdkodad eller inte, vilket vi tydliggjorde efter att vi insett problematiken.

Arbetssättet som gruppen har jobbat utefter under projektet har påverkats i stor utsträckning av hur våra user stories utformades. Dessutom har värdeskapandet påverkats i sin tur av detta. Genom att strukturera våra user stories som presenterats så kunde gruppen arbeta efter ett jämnt tempo, vilket var önskvärt och effektivt. Utöver själva arbetssättet så blev värdeskapandet tydligt, eftersom kunden var involverad i väldigt hög grad. Med detta menas att kunden var delaktig i möten, där hon kunde direkt uppfatta vad våra user stories innebar, och därefter prioritera dem för vad hon ansåg vara mest värdeskapande. Detta ledde till att gruppen alltid jobbade med aktiviteter och funktioner som gav värde för kunden. Att våra user stories var designade för att bli klara på 1-2 dagar gjorde att gruppen ofta kunde hinna med fler user stories än vad vi räknat med, vilket var mer önskvärt än stora komplicerade uppgifter som sträcker sig över flera sprinter. Undantaget var sprint 5, där två funktioner var svårare att implementera än vad gruppen hade uppskattat, vilket medförde att de blev klara nästkommande sprint. Detta sågs däremot inte som ett problem från varken gruppen eller vår kund, då gruppen redan kommit upp i mer än 100 avklarade poäng, vilket innebar att vi ändå var i fas. Det sättet som vi strukturerade arbetet innebar även att gruppen inte behövde arbeta mer än cirka 15 timmar per vecka, vilket bedömdes som rimligt med tanke på kursens omfattning. Detta kunde hållas under hela projektet, där arbete disponerades ut på vardagar, för att säkerställa att ingen gjorde mer arbete än någon annan och ge ett hälsosamt arbetssätt.

1.3.2 Hur bör framtida projekt se ut? (B)

I framtida projekt vill gruppen kunna dela upp user stories bättre från första början. Eftersom merparten av gruppen inte arbetat i liknande projekt tidigare så är det inte särskilt förvånande att gruppen behövde dela upp våra user stories under projektets gång, men i nästa projekt kan detta troligtvis förbättras. Det är viktigt att ha välplanerade user stories från projektstart, med tydliga acceptance criteria som diskuterats med kunden. Det förenklar arbetet med effort estimation och hur arbetsfördelningen inom gruppen läggs upp, och bidrar till att värde kan skapas för kunden. Självklart kan det fortfarande tillkomma user stories eller göras ändringar i befintliga user stories under projektets gång. Tydliga acceptance criteria från start underlättar såklart och är något vi förutspår i nästa projekt nu när vi insett vikten av detta.

I ett framtida projekt är det fortfarande viktigt med en hälsosam och jämn arbetsbelastning där ingen överskrider de förväntade arbetstimmarna avsevärt.

1.3.3 Vad krävs för att nå dit? (A till B)

För att nå mer stabila och definitiva user stories och acceptance criteria så behöver gruppen försöka tänka mer proaktivt i början av projektet i planeringsfasen. Med detta menas att utförliga diskussioner behöver ske redan från första början, och att samtliga funktioner försöker definieras och uppskattas. Detta är ett arbetssätt som är ganska minutiöst och krävande, och det kräver därför en grupp med mycket kunskap inom området, där många perspektiv kan tas upp.

Scrum poker var ett bra sätt för att få koll på arbetsbelastningen, och är ett koncept att fortsätta med.

1.4 Your acceptance tests, such as how they were performed, with whom, and which value they provided for you and the other stakeholders

1.4.1 Sammanfattning av projektet (A)

I slutet av varje sprint har vi haft ett möte tillsammans med kunden där vi har kört applikationen för att visa framstegen som gjorts under veckan. Det har fyllt funktionen som acceptance test och kunden har under dessa möten kunnat ge feedback och framföra önskemål. I sprint 2 valde vi även att hålla kunden uppdaterad via Messenger under veckan för att hon skulle vara involverad i processen för att kunna ge mer feedback på fredagsmötet. Vi hade en tanke om att ha en mer interaktiv upplevelse genom att ge kunden möjlighet att själv köra applikationen hemma. Kunden upplevde däremot inte att tiden som behövdes läggas ner på denna funktion var värt det, vilket medförde att funktionen bortprioriterades.

Varje user story skapades i en enskild branch. När en user story's acceptance criterias hade uppfyllts gjordes en pull request, som någon i gruppen som inte varit med och skrivit koden gick igenom. Därefter kunde den mergas till main-branchen om inga frågor eller konflikter uppstod. Vi hade som mål att ha all kod på main innan mötet med kunden för att kunna köra demon på en enhet för att undvika att det blev rörigt. Om vi inte hann klart med en user story fick den ligga kvar på sin branch. Genom att programmera i grupper om 2-3 personer har gruppen kunnat producera effektiv kod och det har också fungerat som acceptance test.

1.4.2 Hur bör framtida projekt se ut? (B)

Med en mer ingående planering från start hade vi kunnat formulera bättre acceptance criteria från början. Vi har gjort ändringar efter hand i projektet men det är något vi tror hade kunnat bli bättre i framtiden eftersom att vi har fått erfarenhet från vårt projekt. En ytterligare förbättringspunkt skulle kunna vara att i större utsträckning inkludera kunden när acceptance criteria formuleras och på så sätt fånga upp kundens förväntningar på olika funktioner. En bra och kontinuerlig kommunikation med kunden har varit mycket värdefullt för vårt projekt och i framtiden tror vi att det är viktigt att fortsätta med det.

1.4.3 Vad krävs för att nå dit? (A till B)

Projektgruppen bör i samråd med kunden komma överens om hur acceptance tests ska utföras under projektet och i vilken omfattning. Testerna är ett sätt att säkerställa att krav från kunden uppfylls och att koden håller den kvalitet som utlovats. I vårt projekt var sccopet relativt litet och applikationen kunde testas genom att köra koden allteftersom nya funktioner lades till. I

större projekt kan egna tester vara väsentligt för att undvika buggar och se till att koden är effektiv, vilket är värdefullt för både kunden och projektgruppen.

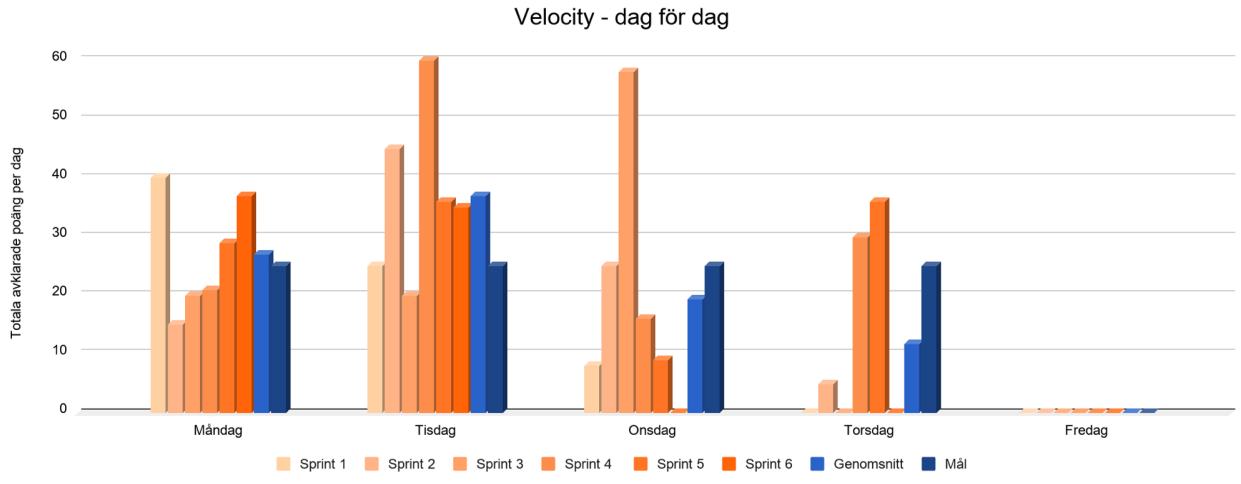
1.5 The three KPIs you use for monitoring and how you use them to improve your process

1.5.1 Sammanfattning av projektet (A)

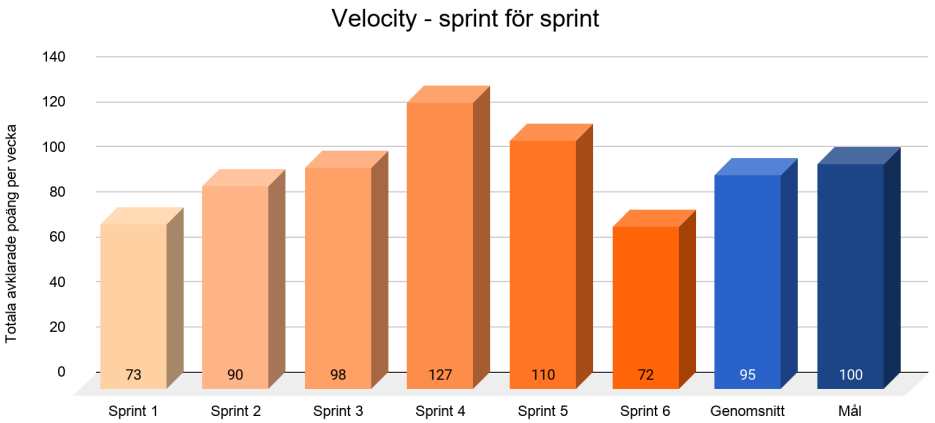
Under projektet användes KPIerna velocity, burn down chart och Niko-Niko kalender, vilka presenteras nedan.

Velocity

KPIn velocity är till för att uppnå ett jämnt arbetstempo, där poängen som avklaras på varje user story beskriver tempot. Som grupp sattes ett mål på att klara av 100 poäng varje sprint, vilket innebar 25 poäng varje dag från måndag till torsdag. Anledningen till varför inte fredagen räknades in är för att den dagen var ämnad för arbete med vår team reflection, sprint review och demo för vår kund utan att stressa. Under projektets gång varierade mängden poäng som klarades av varje dag och varje sprint. I projektets början lades mer fokus på att vänja sig vid Android Studio och dess ramverk, därför kunde inte 100 poäng uppnås under denna tid. Efter några sprinter kunde gruppen klara av fler än 100 poäng på en sprint, vilket kunde motiveras med att kunskapsnivån hos gruppen var högre, men också eftersom gruppen programmerade i mindre grupper så var kunskapen diversifierad inom gruppen. Sista sprinten lyckades gruppen inte uppnå 100 poäng, men detta kunde förklaras med att mycket fokus lades på presentationen av projektet, vilket inte kunde kopplas till våra user stories.



Diagrammet ovan visar vilken velocity gruppen uppnått varje dag, tillsammans med ett genomsnitt för dagen, men också det mål som sattes upp. Från vad som kan utläsa i diagrammet gjordes merparten av arbetet på måndagar och tisdagar, där genomsnittet för dagen överstiger målet. Onsdagar och torsdagar var inte lika produktiva i jämförelse, och fredagar ägnas åt andra moment i projektet än user stories.



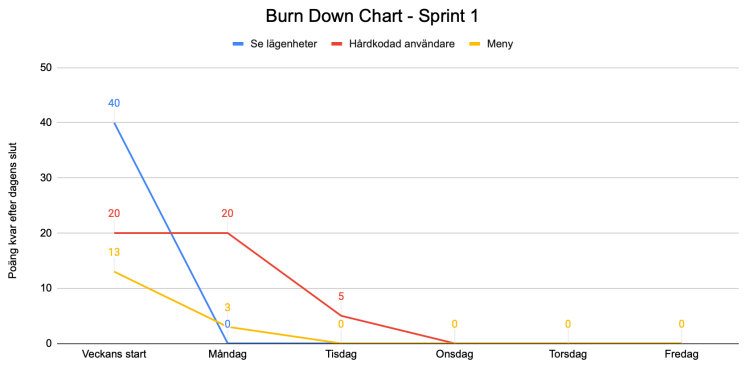
Stapeldiagrammet ovan beskriver velocity för varje sprint, tillsammans med alla sprinters genomsnitt och målet. Genomsnittet på 95 poäng för varje sprint anser gruppen vara tillräckligt, då vi tydligt kunde se en förbättring i antal avklarade poäng varje sprint.

Velocity var väldigt användbar eftersom gruppen använde sig av poängsättning på user stories inför varje sprint, och alla hade en idé om vilken velocity som skulle uppnås. Tillsammans med kontinuerlig kommunikation blev velocity en uppmuntrande faktor i processen.

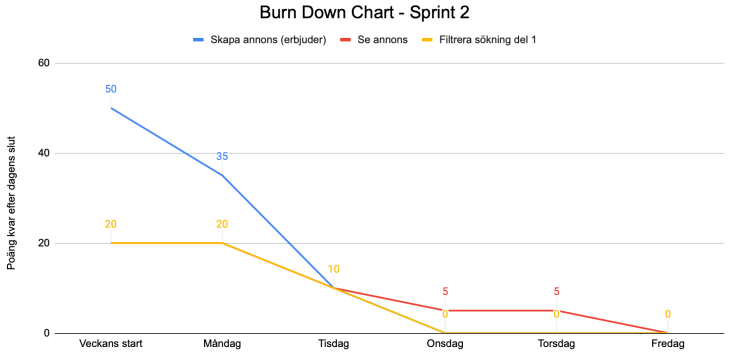
Burn Down Chart

Denna KPI används för att bedöma hur mycket som görs under sprinterna, och är ett bra sätt för att veta hur mycket som finns kvar att göra för varje sprint. Till skillnad från velocity och Niko-Niko som användes kontinuerligt under projektet användes inte denna KPI i lika stor utsträckning. Anledningen till detta har varit att gruppen konstaterat att velocity har varit ett väldigt effektivt sätt att uppskatta resterande arbetsbelastning, tillsammans med den ständigt uppdaterade scrum boarden i Trello. Däremot användes den mer och mer under projektets gång, och den är praktisk för att utvärdera sprinter och projektet i stort.

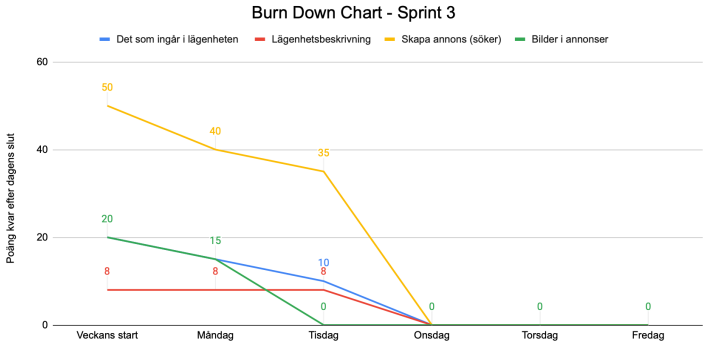
Nedan följer förklaring till graferna, vilket summeras av en sammanställd graf där sprinterna jämförs med varandra, och slutligen hur hela projektet fortgått.



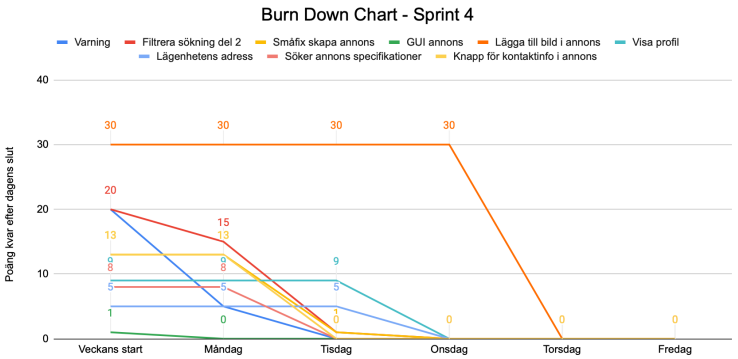
Sprint 1 fokuserade gruppen på inlärning, vilket resulterade i mindre än 100 avklarade poäng denna sprinten. (“*Se lägenheter*” uppgick alltså till 40 poäng, och alla poängen gjordes på måndagen, därmed är det 0 poäng kvar vid måndagens slut)



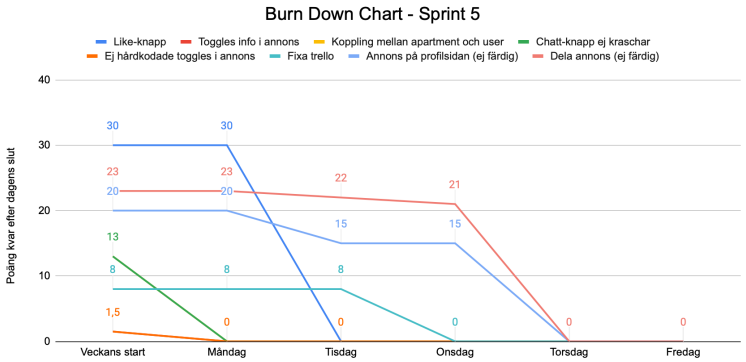
Sprint 2 lyckades gruppen komma upp i 90 avklarade poäng, med ett jämnt arbetstempo.



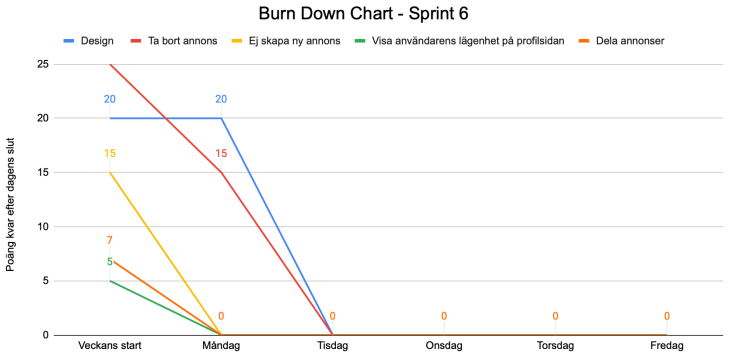
Sprint 3 resulterade i 98 avklarade poäng, där arbetstempot återigen var jämnt.



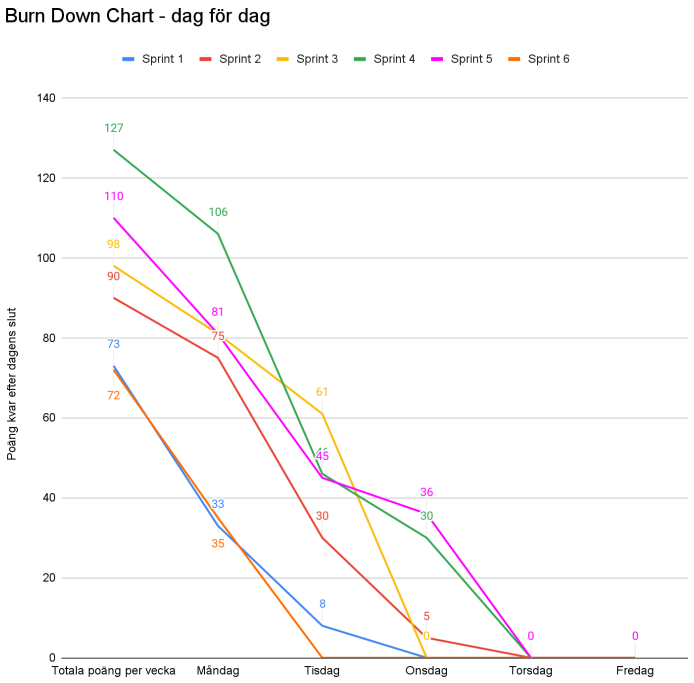
Sprint 4 klarade gruppen av 127 poäng, med ett bra tempo.



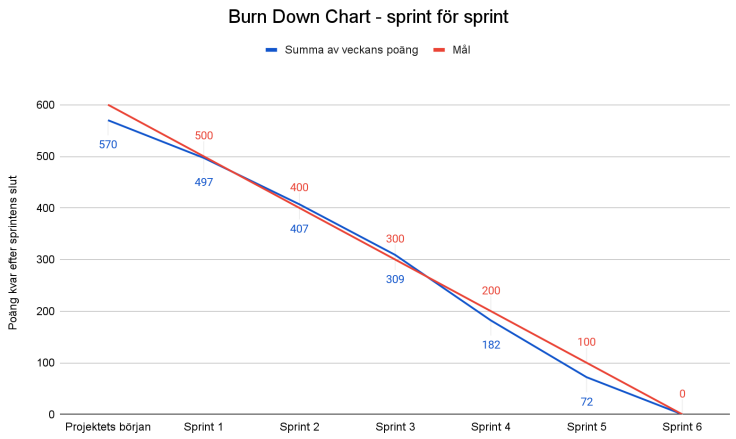
Sprint 5 klarade gruppen 110 poäng, med jämnt tempo.



Sprint 6 bestod av 72 avklarade poäng, där mycket fokus lades på presentation av appen, och andra moment i kursen, därmed inte lika många poäng som tidigare sprinter.



Utifrån grafen är det tydligt att gruppen arbetat med ett jämnt tempo, oavsett vilken mängd poäng som klarats av.



Slutligen visas projektets sammanslagna graf, som visar att vi varit oerhört nära vårt mål på 100 poäng per sprint under hela projektets gång.

Burn down chart påverkade vår process positivt, då det blev uppenbart hur många poäng som var kvar vid varje tidpunkt under sprinten. Därmed kunde gruppen hålla ett jämnt tempo.

Niko-Niko kalender

Denna KPI består av en tabell för varje sprint, där varje arbetsdag får en kolumn, och varje medlem i gruppen får varsin rad. Varje dag fyllde gruppmedlemmar i en emoji som beskrev hur man kände sig relaterat till projektet. Varje vecka, i samband med vår sprint review, så gick gruppen igenom tabellen, där var och en fick förklara sina emojis. Under projektets gång ville vi kunna snappa upp oroväckande emojis snabbt, men insåg under tiden att eftersom gruppen jobbade så tätt med varandra behövdes det inte. Nedan visas exempel på tre olika sprinter, vecka 1 som var spännande, sprint 5 som var komplicerad, och sprint 6 som var bra.

Hur vi mår i projektet sprint för sprint

Sprint 1

	Måndag	Tisdag	Onsdag	Torsdag	Fredag
Lisa	😊	😄	😄	😊	😀
Marcus	😊	😃	😄	😊	😊
Kerstin	🧐	😅	😊	😊	😀
Izabell	😊	🤓	😊	😊	🥳
Moa	🤩	😀	😄	😅	😀
Amanda	😊	😄	😊	🧐	😊
Noa	😎	🤔	🤗	🤗	😎

Sprint 5

	Måndag	Tisdag	Onsdag	Torsdag	Fredag
Lisa	😊	😕	😘	😻	😊
Marcus	😐	😎	😀	🤗	😎
Kerstin	😊	😤	🥴	🧐	🤠
Izabell	🥳	🤬	😤	😔	😊
Moa	😀	🧐	😊	😊	😊
Amanda	🤠	🤯	🤯	🧐	🥳
Noa	🤠	😕	🤗	😉	😊

Sprint 6

	Måndag	Tisdag	Onsdag	Torsdag	Fredag
Lisa	🤠	😄	🤩	🤩	🥳
Marcus	🥳	🥳	🥳	😊	😊
Kerstin	😎	😊	😓	😎	😎
Izabell	😅	🤩	🤩	😎	🥳
Moa	😊	🤩	😊	😊	😊
Amanda	😄	😓	🤩	😃	🥳
Noa	🤠	😎	😎	🤗	🥳

Niko-Niko var användbar i bemärkelse att den var kopplad till hur var och en upplevde sin insats och rent generellt hur projektet fortgick. Detta skiljer från de andra KPIerna, och gruppen fick en möjlighet att reflektera över veckan tillsammans på ett nyttigt sätt.

1.5.2 Hur bör framtida projekt se ut? (B)

I framtida projekt skulle gruppen gärna vilja se mer utökad användning av velocity och burn down chart. Anledningen är att det har uppskattats som oerhört hjälpsamt och positivt när det väl användes, vilket var fallet med velocity i merparten av projektet. Däremot användes inte burn down chart i lika stor utsträckning förrän mot slutet, vilket är synd, eftersom den också är användbar. Samtidigt vill även gruppen försöka uppnå en jämnare velocity, då det är fördelaktigt både ur ett agilt och hälsosamt perspektiv. Velocity kan dock vara av större vikt vid mer omfattande projekt som pågår under en längre tid. Det samlade intrycket är att

samtliga KPIer kompletterar varandra och hjälper projekt att fungera bättre genom att förtydliga olika punkter och öppna för reflektion och diskussion.

Efter närmare eftertanke vore det av värde att ha något tydligt mått på hur nöjd kunden är. Eftersom vi hade fördelen att kunna prata med vår kund flera gånger i veckan upplevde vi ändå att vi hade god förståelse för henne, men det kan vara ett bra sätt med en delad KPI mellan kund och grupp för att mäta "nöjdheten" hos kunden i framtida projekt med kunder som har svårare att delta ofta.

1.5.3 Vad krävs för att nå dit? (A till B)

Jämnare velocity kan uppnås genom att försöka planera user stories mer noggrant, och att utgå utifrån att 25 poäng ska avklaras varje dag mellan måndag och torsdag. Detta har inte varit möjligt under projektet, eftersom kursen inte var på helfart och många scheman inte alltid gick ihop, men det kan vara värt att försöka åtgärda med att bestämma mötestider mer utifrån vilken velocity som ska uppnås. Att öka användningen av burn down chart och velocity är inte komplicerat att göra, utan det är framförallt viktigt att stipulera från början i vilken utsträckning det ska användas. Efter vår positiva erfarenhet tror vi på en utökad användning av velocity och burn down.

Ett förslag på en KPI som hade fungerat som mått för hur nöjd kunden är att ha en delad Niko Niko-kalender med kunden, där hen kanske också kan lägga kommentarer om det förekommer någon avvikande emoji. Detta hade fungerat som en god indikation på när det behövs föras mer diskussioner för att förbättra något eller när arbetet kan fortsätta som det är.

2 Social Contract and Effort

2.1 Your social contract, i.e. the rules that define how you work together as a team, how it influenced your work, and how it evolved during the project

2.1.1 Sammanfattning av projektet (A)

Vid uppstarten av projektet samlades hela gruppen för att utforma det sociala kontraktet efter allas önskemål och åsikter. Gruppen utgick från ett kontrakt från tidigare kurs och valde ut de delar som upplevdes relevanta för detta projekt. Kontraktet innehöll förhållningsregler gällande följande delar:

- **Möten**

Kommunikation om möten sker i gruppchatt, vid mer än 10 minuter sen ankomst bjuder personen på fika. Stående möte måndagar 8.00.

- **Arbetsfördelning**

Jämn fördelning i gruppen, öppen dialog om vad alla ska åstadkomma varje vecka. Vid problem är det viktigt att be om hjälp och vara tydlig mot resten av gruppen om man inte hinner klart och varför.

- **Beslutstagande**

Målet är att genom diskussion nå gemensam konsensus, annars används röstning. Vid ytterligare diskussioner kan handledare kopplas in för rådgivning.

- **Dokument- och filhantering**

All dokumentation sker i Google Drive och Github.

- **KPI:s**

Gruppen kom överens om att använda Niko-Niko, Velocity och Burn down-chart.

- **Ansvarsområden**

Scrum master byts varje vecka. Övriga roller har fördelats inom gruppen så att en person är product owner, en har ansvar för github/utvecklingsmiljö, två för kod och två för final reflection. Detta innebär inte att det är dem som hanterar respektive område utan endast att de ser till att allt blir gjort och hjälper till extra med planering och fördelning av arbete.

Någon vecka senare då första sprinten skulle genomföras så insåg gruppen att även DoD, Definitions of Done, borde vara med i kontraktet. Även de mötestider som gruppen satt upp hade uppdaterats och därefter lades följande punkter till:

- **DoD**

Alla acceptance criteria måste vara uppfyllda

Koden måste ligga på main-branchen

Koden ska ha lästs igenom av minst en person som inte kodade den

Koden ska ha testats med kopplad testkod (ej set o get)

Koden ska ha dokumenterats i javadoc (ej set o get)

Kunden ska vara nöjd med resultatet i slutet av varje sprint

- **Kontinuerliga mötestider**

Måndagar, tisdagar och fredagar 8.00. Samtliga ska ha kameran på under mötena för att kunna kommunicera lättare och mer effektivt.

Under projektets gång har gruppen varit lyhörd på om något skulle behöva bytas ut, läggas till eller ändras i kontraktet. Senare under projektet diskuterades det även hur gruppen känt gällande kontraktet och samtliga höll då med varandra om att de inte tänkt så mycket på kontraktet under projektets gång men när de läst det i efterhand så tycker de att det följs bra.

Under projektets gång så har kommunikation varit en nyckelfaktor inom gruppen. Gruppchatt och Zoom har använts då olika problem och frågor dykt upp. I och med att alla i gruppen var med och utformade kontraktet och gemensamt kom fram till punkterna så var alla med på vad som gällde från början och därför har kontraktet kunnat följas utan svårigheter av alla i gruppen.

Den tydliga kommunikationen i gruppen har även hjälpt till under veckor där någon i gruppen inte kunnat närvara lika mycket som övriga. Det har berott på anledningar såsom tung arbetsbelastning i parallell kurs, bortrest under röda vardagar eller sjukdom. Det har inte uppstått någon situation där en gruppmedlem ej meddelat om sin frånvaro utan alla har varit tydliga med hur mycket tid de kan lägga och när de kan arbeta under veckan.

2.1.2 Hur bör framtida projekt se ut? (B)

Något som inte togs upp i kontraktet men som präglat den absolut största delen av app-utvecklingen är att gruppen har programmerat i grupper. För det mesta har två till tre personer tagit sig an en user story tillsammans och arbetat med den ihop tills den är klar. Parprogrammering var något som gruppen diskuterade i början av projektet och det fanns en stor efterfrågan av att få arbeta tillsammans då denna typ av projekt var mycket mer komplext än vad många av gruppmedlemmarna gjort innan. I och med att gruppen arbetat tätt tillsammans har det aldrig uppstått oklarheter om vem som har bidragit till arbetet eller inte, men i ett framtida projekt där man kanske inte får kommunikationen att fungera lika bra kan det vara viktigt att ha med parprogrammering och dess riktlinjer i ett gruppkontrakt, för att undvika eventuella konflikter och att olika personer får göra väldigt olika mängd arbete.

En annan aspekt som inte heller skrevs ner i kontraktet är den diskussion om arbetsbelastning som gruppen har haft kontinuerligt under projektets gång. Samtliga har varit överens om att det är viktigt att minska den stress som ofta upplevs i stora projekt, exempelvis inför deadline, och istället ha en jämn arbetsbelastning med fasta tider och mål. Gruppen bestämde att man bör lägga ungefär tre timmar om dagen, måndag till fredag, i snitt och att i de första sprinterna snarare ta på sig färre user stories än vad som är möjligt för att komma in i arbetssättet och inte skapa en dålig vana där man pressar gruppen för mycket. I linje med detta använde gruppen även Niko-Niko som en av sina KPI:s, för att fånga upp om någon kände sig stressad eller mådde dåligt på annat sätt relaterat till projektet, för att kunna förbättra detta till nästa sprint. Under hela projektet har arbetsbelastningen diskuterats och på Velocity KPI:et går det att se en ökning av mängden arbete som avklarades varje vecka. Det möjliggjordes genom att gruppen var lyhörd på hur samtliga upplevde arbetsbelastningen och därmed kunde mängden poäng som genomfördes varje vecka öka under de senare sprinterna.

I ett framtida projekt där man kanske arbetar med en mjukvara i flera månader är det ännu viktigare att arbetsbelastningen är hållbar. Det kan också finnas mer press på att leverera till kund än vad som finns i ett skolprojekt och därmed ökar risken för att gruppen pressar sig för hårt så att personer börjar må dåligt. Därmed är det viktigt att få med vad som förväntas av varje person i kontraktet och att det är tydligt vad som ska levereras varje vecka. Utan ett kontrakt där det står hur mycket arbete som förväntas kan det uppstå konflikter och en ohållbar och ohälsosam arbetsmiljö kan utvecklas då pressen är stor på att leverera allt som kunden önskar.

Något som skulle kunna bli ett problem i ett framtida projekt är att alla diskussioner skett med hela gruppen. Därmed skapas inte alltid möjligheten att ta upp obekväma ämnen såsom konflikter eller om personer känner sig överbelastade eller tycker att det är en ojämn fördelning av arbete. Om alla får möjlighet att uttrycka sin åsikt om huruvida gruppkontraktet följs eller inte i ett mindre utsatt forum än framför hela gruppen skulle det kanske få fram problematik i gruppen snabbare, innan möjliga problem hinner växa.

2.1.3 Vad krävs för att nå dit? (A till B)

Gällande parprogrammering så är det viktigt att sätta upp regler och riktlinjer för hur detta ska genomföras. Parprogrammering kan vara ett effektivt arbetssätt då man direkt kan tackla problem tillsammans men det kan också skapa en miljö där personer endast sitter bredvid medan en annan gör allt arbete. Därför bör man bytas av vem som skriver koden och sitta lika många timmar vid tangentbordet som bredvid i snitt. Det kan vara viktigt att föra in det i kontraktet och sedan ha en inrapportering där alla skriver hur många timmar man själv, och personen man programmerat med, skrivit kod.

Vad gäller arbetsbelastningen är det även där viktigt att det finns med i kontraktet hur många timmar man förväntas lägga och vad som händer om man under dem timmarna inte hinner klart med sin uppgift. Olika personer kan ha väldigt olika åsikt om det bör innebära att man ska arbeta över helgen eller ta vid nästa vecka. Det måste vara tydligt och nedskrivet så att alla i gruppen arbetar utefter samma regler.

När det kommer till att upptäcka problematik och konflikter inom gruppen så är det viktigt att det finns en person i gruppen som är ansvarig för att prata med samtliga och säkerställa att allas åsikter kommer fram. Genom att i kontraktet även skriva med vilka olika typer av feedback-kommunikation som sker i gruppen kan personerna känna sig sedda och därmed använda det kommunikationssätt som passar dem bäst för att föra fram sin åsikt.

2.2 The time you have spent on the course and how it relates to what you delivered

2.2.1 Sammanfattning av projektet (A)

Under hela projektet har gruppen haft en pågående diskussion gällande arbetsbelastningen och hur samtliga upplevt varje sprint. Ett exempel på detta är KPI:n Niko-Niko där alla får uttrycka hur de mått varje dag relaterat till projektet. Innan första sprinten hade gruppen en diskussion där samtliga var överens om att man inte tyckte att kursen ska kräva att man arbetar på helger utan att dessa är till för återhämtning. Man sa även att om man under arbetstimmarna inte hinner klart med en arbetsuppgift lämnas den till nästa vecka.

Gruppen har även under de senare sprinterna varit flexibla angående mängden user stories som slutförts per vecka. Genom att ha en tydlig prioriteringsordning i vår scrumboard så har grupper som avklarat sina user stories tidigt under veckan kunnat gå på en ny som ligger högt upp på prioriteringen, förutsatt att de kommunicerat tydligt till gruppen att de är klara och därmed påbörjar ytterligare en user story.

Utefter de mål gällande låg stressnivå och ett hållbart arbetssätt som gruppen satte ut i början är gruppen väldigt nöjda med vad som åstadkommits. Genom att vara flexibla och tydliga i

kommunikationen har mer arbete än planerat kunnat genomföras många veckor och gruppen har i slutet av samtliga sprinter varit nöjd med sin prestation!

I början av projektet skrev gruppen samtliga user stories som behövde genomföras för att klara av alla funktioner som var tänkta med appen. Under ett tidigt möte sattes även prioriteringsordningen bland dessa tillsammans med kunden och under projektets gång har justeringar av ordningen alltid skett tillsammans med kunden. I slutet av projektet var näst intill alla user stories avklarade och endast de med lägst prioritering fanns kvar. Dessa stories behövdes inte slutföras då de flesta relaterade till att lansera appen, något som inte varit tanken från början. Därmed var även kunden nöjd med slutprodukten och gruppen hade utefter projektets tidsram gjort en bra uppskattning av mängden arbete som skulle hinnas.

2.2.2 Hur bör framtida projekt se ut? (B)

Under ett framtida projekt kan det vara en fördel att vara konkret i början med vilka kunskaper man har och vilka förutsättningar för att producera kod som finns inom gruppen. Med en ökad förståelse av varandras kunskaper kan man effektivisera arbetet genom att de med mindre kunskap sätts ihop med någon som har bra förståelse. Den med bättre förståelse får då som uppgift att under första veckan introducera utvecklingsmiljön och se till att den andra personen har fått göra sig bekant med miljön och alla redskap runtom. Det finns dock en risk för att den med mest kunskap gör allt arbete och att den andra mest tittar på. För att undvika det är det viktigt att man ser till att båda sitter lika många timmar framför tangentbordet och gör arbetet tillsammans. Det kan då förkorta inlärningsfasen och gruppen kan snabbare komma upp på en högre genomsnittlig nivå.

Det kan även vara en god idé att utvärdera om det satta Velocity-målet fortfarande är relevant en bit in i arbetet. Om man lär sig mer och blir snabbare på att lösa user stories kan det vara bra att öka målet. I många fall finns det en viss inlärningströskel och när man kommit över den så går arbetet snabbare, samt att fler personer kan känna sig bekväma med att programmera ensamma och då få ännu fler stories gjorda.

Något som gruppen gärna vill ta med sig till framtida projekt är den öppna diskussionen om arbetsbelastning och stressnivåer. Niko-Niko har upplevts som en väldigt bra KPI att ha med då alla får lyfta sitt mående. Ska man i framtiden arbeta i ett längre projekt blir det ännu viktigare med en hållbar arbetsmiljö över tid.

2.2.3 Vad krävs för att nå dit? (A till B)

För att nå dessa mål krävs det att gruppen har ett öppet klimat, både gällande ens kunskaper och ens brister. Genom att i ett inledande skede vara tydliga med vad man känner att man har mycket kunskap inom och vad man känner sig osäker på så kan man fånga upp individer som behöver lite extra hjälp i starten. Det är även en god idé att ha fler utvärderingssamtal om huruvida KPI måttet Velocity fortfarande känns väl anpassat till gruppens förutsättningar. Det kan även vara så att det är åt andra hållet än vad gruppen upplevde, att man klarar av mycket

mindre varje vecka än vad man tänkte från början. Isåfall är det också viktigt att vara flexibel och då sänka ribban, det kan annars bli väldigt jobbigt att känna att gruppen underpresterar varje vecka. Om det satta målet är orealistiskt är det bättre att diskutera varför det är det och om man kan justera värdet så det passar gruppen bättre.

Gruppen vill fortsätta använda Niko-Niko i framtiden för att fånga upp hur alla mår och diskutera om arbetsbelastningen är på en bra nivå. För att ingen ska känna sig konstant stressad, eller understimulerad, är det viktigt att kolla på hur alla mår och anpassa sprinterna efter det. Att även från första början lyfta ämnet psykisk ohälsa och hur viktigt det är med en hållbar arbetsmiljö kan man även få med hela gruppen i ett hållbart tänk kring vad som förväntas av alla.

3 Design Decisions and Product Structure

3.1 How your design decisions (e.g. choice of APIs, architectural patterns, behaviour) support customer value

3.1.1 Sammanfattning av projektet (A)

Redan tidigt i projektet togs besultet att följa de riktlinjer och etablerade metoder som beskrivs av Google själva och andra med auktoritet inom utveckling av Android projekt. Detta ansågs viktigt eftersom att utveckling i Android miljö var nytt för majoriteten av gruppen. Att då kunna förlita sig på andras riktlinjer minskade risken för att vi skulle implementera saker på “felaktigt” sätt som senare skulle minska vår utvecklingstakt.

Något som uppdagades rätt tidigt var den stora skillnaden i hur mycket kod som förlitar sig på plattformsspecifika ramverk i jämförelse med tidigare kurser där koden till stor del varit antingen skriven av en själv eller från Javas standardbibliotek. Resultatet blev att en stor del av tiden dem första veckorna spenderades med att läsa dokumentation för att förstå vilka verktyg som fanns tillgängliga och vilka problem de löste. Även om inläsningen tog mycket tid försökte vi också samtidigt att implementera väldigt små uppgifter så att det var enklare att sätta det som lästes i ett sammanhang och på så sätt minska tiden till mer produktivitet.

Som nämnts tidigare var det under hela projektets gång ett stort fokus på att arbeta i mindre grupper och hela tiden byta ansvarsområden så att det inte bara var en enda person som visste hur en specifik del av appen fungerade. Dock var detta något vi inte uppnådde i samma utsträckning när det kom till arkitekturen. Istället var denna en mer implicit kunskap som inte tydligt överfördes regelbundet utan snarare endast när problem uppstod på grund av att man just implementerat något som i en komponent skulle ha ett annat ansvar.

3.1.2 Hur bör framtida projekt se ut? (B)

För att systemets arkitektur ska bidra till att man utvecklar värde för kunden krävs att man tar hänsyn till de yttre förutsättningar som systemet ska verka inom och på så sätt designa komponenter som har så få beroenden som möjligt. Detta möjliggör att förändringar inte påverkar stora delar och därmed tar mindre tid att implementera. För att detta ska vara möjligt är det viktigt att alla har en gemensam förståelse om hur den övergripande arkitekturen ser ut, och kanske ännu viktigare, varför den är som den är. Missar man att förmedla det underliggande skälet till varför man har utformat arkitekturen på ett visst sätt är det senare svårt att följa den.

3.1.3 Vad krävs för att nå dit? (A till B)

Något som borde gjorts redan tidigt var att ha en gemensam genomgång av den tänkta arkitekturen. I detta stadie är inte tanken att allt ska vara planerat i detalj (i så fall motverkar det hela iden med agil utveckling) utan snarare så att man beskriver dem delar som är tänkta att finnas och vilka som är dem större flödena i systemet. Arkitekturella besult är alltid jobbigare att ändra, oavsett om man jobbar agilt eller inte, men de blir enklare att följa om alla har en gemensam förståelse.

3.2 Which technical documentation you use and why (e.g. use cases, interaction diagrams, class diagrams, domain models or component diagrams, text documents)

3.2.1 Sammanfattning av projektet (A)

Till en början tillämpade vi bara en typ av teknisk dokumentation; att skriva Javadoc enligt standarder på allt vi implementerar. Detta stod med i vår Definition of Done. Vi utökade med mer teknisk dokumentation i sista sprinten. Anledningen till att vi inte implementerade det tidigare var för att vi inte fann något behov av det i vår arbetsprocess. Vi jobbade tätt inom gruppen med flera möten i veckan, roterande välfungerande parprogrammering och relativt smalt scope vilket gjorde att alla i gruppen hade en god insyn i hur projektet var uppbyggt.

Ett problem som uppstod i mitten av projektet var att vi insåg att vi hade missat att skriva javadoc så pass utförligt som vi bestämt. Vi avsatte då tid där hela gruppen skrev javadoc för att komma ikapp. Trots att detta inte var ett önskat scenario lärde vi oss mycket av det då var och en inte skrev javadoc enbart för kod som hen skrivit, vilket gjorde att vi fick en mer detaljerad förståelse för en del funktionalitet. Trots att detta stannade upp vår implementering av värde för kunden visar diagrammen ovan att vi levererade som mest poäng denna veckan (sprint 4). Niko-Niko sammanställningen visar också att gruppen inte mådde märkbart sämre på grund av denna ökade arbetsbelastning, så det var trots allt ett fungerande tillfälle att avsätta tid från det förväntade då vi i övrigt hade en mycket välfungerande sprint här.

I sprint 6 började vi att skapa UML diagram och ett dokument som förklarar arkitekturen för projektet, för att kunna tydliggöra för utomstående vad vi åstadkommit. Dessa finns tillgängliga i readme-filen.

3.2.2 Hur bör framtida projekt se ut? (B)

Det är bra att ligga i fas med arkitekturbeskrivning av projektet under hela utvecklingen för att tydliggöra och förenkla för att göra eller tänka ut vidare förändringar. Det är också bra för att möjliggöra att utomstående förstår strukturen under hela processen. I ett projekt som detta där de utomstående, kund och kursledare, inte velat se strukturen förrän vid slutet och där

gruppen jobbat mycket tätt är det dock inte av lika stort intresse, eftersom den troligtvis kostar mycket tid då mycket förändras med tiden.

3.2.3 Vad krävs för att nå dit? (A till B)

För att ha en beskrivning av projektet i fas hela tiden, krävs det att man startar upp för detta från början. Det underlättar med framtagna rutiner för när dokumenten/diagrammen ska uppdateras. Förslagsvis skulle det kunna vara en del av DoD för varje user story, som vi hade med javadoc, eller att det ska ske vid varje sprints slut. För att hålla en tydlig struktur i dokumenten/diagrammen kan det vara bra att tillsammans i gruppen ta fram en mall som var och en kan följa när hen uppdaterar dokumentet. För att förebygga att punkter i DoD glöms av kan denna lista placeras tydligt någonstans i projektet där deltagarna blir upprepande påminda av den, förslagsvis i scrum boarden.

3.3 How you use and update your documentation throughout the sprints

3.3.1 Sammanfattning av projektet (A)

Vid projektets start skapade vi ett gruppkontrakt för att lägga grunden för vårt grupparbete, som vi har förhållit oss till under tidens gång, och även uppdaterat vid behov. Vi har alltid haft den uppdaterade versionen i vårt repository på github för att göra den lättillgänglig. Vi förde också mötesprotokoll för att säkerställa att inga slutsatser eller frågor till handledaren glöms av, och för att möjliggöra att repetera vad som hänt.

Vi skapade tidigt en scrum board som vi har jobbat tätt med under hela projektet. Vid varje sprints start (alltså varje måndag) markerade vi vem som skulle göra vad i scrum boaden för att tydligt ge en överblick över sprinten, vilket gav värde både för oss och kunden som kunde se vad som förväntades denna veckan. Vissa veckor var det svårare att uppskatta hur mycket som skulle hinnas med, och då höll vi oss till de mindre alternativet för att inte skapa stress, men bestämde att det var fritt fram att ta sig an nästa på prioriteringslistan i scrum boarden om man hade tid. Det adderades många user stories till scrum boarden under projektets gång.

För att skapa sammanhållning koden med flera utvecklare lämnade vi kommenterar i koden när något behövde förtydligas. Vi använde oss också av annotationen TODO för att enkelt kunna hitta ställen där något behöver göras.

Ett problem som uppstod under sprint 3 var att två par-programmeringspar implementerade lösningar till samma user story. Som tur var så var detta en liten user story. Detta gjorde att vi insåg att vi behövde förbättra våra rutiner kring hur gruppen dokumenterar uppdelningen. Vi införde då en påminnande regel om att scrum borden ska vara så realtidsbaserad som möjligt. Kontraktet uppdaterades även med tillägg om att medlemmar måste berätta i vår gruppchatt om man tar sig an en user story som inte tilldelades på sprint planeringen.

Sista sprinten utvecklade vi dokumentationen gällande två av våra KPIer. Tidigare hade vi samlat all information om velocity och burn down i ett textdokument där vi har fyllt i varje vecka hur poängen har fördelats och hanterats. I slutet sammanställde vi dokumentet i grafer.

3.3.2 Hur bör framtida projekt se ut? (B)

Att ha ett välarbetat gruppkontrakt är bra då det direkt skapar normer för gruppen och ger indikationer på projektets önskade arbetssätt. Det är också givande att låta detta vara dynamiskt för att uppdatera vid behov eftersom en process sällan fungerar felfritt från början. Vetskapen om att det är ett dynamiskt kontrakt öppnar också för en löpande diskussion om arbetssättet, vilket är mycket bra för att inte fastna i dåliga rutiner.

I ett framtida projekt uppstår förhoppningsvis ingen överlappning av vem som implementerar vilken user story. Det är också viktigt att koden är lättläst för alla i gruppen. KPI;erna behöver inte ta så mycket tid i slutet i ett framtida projekt för att bana vägen för ifall det skulle vara stressigt mot slutet.

3.3.3 Vad krävs för att nå dit? (A till B)

Då vår gruppdynamik och sätt att hantera arbetsprocessen har fungerat väl är vårt nuvarande gruppkontrakt en bra mall att följa även i vidare arbeten. Detta tar upp relevanta punkter för att förbereda ett bra samarbete. Dock varierar såklart punkternas relevans mellan olika grupper och projekt. Ett sätt för att lättare ta upp diskussionen om något inte skulle fungera bra, skulle vara att införa utvärdering av sprinten från kontraktets vy under sprint review. Där skulle det då passa att uppdatera kontraktet vid behov. Under detta projekt har vi dock inte känt att veckovis utvärdering av kontraktet har behövts då arbetssättet fungerat så bra.

De förbättringar vi införde i samband med att flera omedvetet arbetade med samma user story hjälpte, och vi hade inga mer problem med överlappande/dubbelt arbete. Så detta tycks vara ett vinnande koncept att fortsätta med i framtida projekt.

Gällande förståelse för koden har kommentarer och TODOs i koden fungerat bra för att skriva påminnelser och förklaringar, så det är ett koncept att bibehålla för sammanhållningen. För att undvika att KPI;erna tar mycket tid att sammanställa i slutet kan de som inte krävs att projektet ska vara färdigt för göras i slutet av varje sprint.

3.4 How you ensure code quality and enforce coding standards

3.4.1 Sammanfattning av projektet (A)

Som tidigare nämnts utgick vi ifrån de riktlinjer som fanns från Google gällande hur man bäst designar en apps arkitektur. En god arkitektur möjliggör en att skriva kod som inte bara säkerställer att funktionalitet finns, och därmed ger värde till kund, utan att man över tid kan fortsätta att leverera värde till kund i samma takt. Saker som små klasser med få beroenden

gör det enklare att testa och något som i stor mån styrs av den arkitektur vi från början bestämde oss att använda.

Vi hade tidigt ambitionen att varje ny user story som implementerades skulle ha ett tillhörande test. Detta hindrades något av framförallt två orsaker: majoriteten av allt som gjordes var att hämta värden utan någon logik, samt att den koden som väl utförde någon logik var implementerad med androids ramverk eller hade beroenden till dem. I slutändan förlitade vi alltså oss mer på manuella tester, något som var möjligt då appens scope fortfarande var väldigt litet.

Gällande kodstandarder hade vi aldrig någon explicit mall, dock använde vi alla samma IDE under utvecklingen och följde därmed den standard som IDEn uppmanar till. Utöver det arbetade gruppen nästintill hela tiden med pair/mob programming, något som gjorde att tydliga avsteg från den implicita kodstandarder snabbt kunde fångas upp och åtgärdas.

3.4.2 Hur bör framtida projekt se ut? (B)

Något som definitivt hade behövt förbättras för framtida projekt är graden av automatisering när det kommer till både testning och statisk analys av projektet. Fördelen med att ha detta automatiserat är att man alltid vid att det utförs på samma sätt och att det kan utföras vid varje commit. Skälet till att det fungerade att ha manuell testning är just för att projektet var så pass litet och att man kunde testa all funktionalitet på mindre än 10 minuter.

Automatisering av tester är såklart något som också kräver att det finns en miljö där hela bygget sker, från kompilering till testning, vid varje push till repot. Detta är en investering som till en början kräver en del tid och resurser för att få upp men som säkerställer att testning sker kontinuerligt och inte bara när man som utvecklare kommer ihåg att göra det.

I grunden kräver detta också att det skrivs tester som täcker alla delar av koden. Även om code coverage på 100% egentligen inte betyder att man testat allt vore det säkerligen en bra ambition att testerna ska täcka åtminstone 80% av all kod.

3.4.3 Vad krävs för att nå dit? (A till B)

Som nämndes i 3.4.1 så hade vi två faktorer som begränsade hur mycket tester vi kunde skriva, nämligen att vi inte hade mycket logik och samt att många klasser som berodde på androids ramverk. Inledningsvis gjorde vi försök att sätta upp tester som kunde mocka dessa ramverk men då det tog för mycket tid var det något som fick senareläggas för att gruppen skulle kunna leverera värde till kunden. Även om det i vår slutgiltliga version inte fanns med sådana tester fanns det ändå under tidens gång flera prototyper som nästan hann bli färdiga. I framtida projekt hade mycket av den kunskapen vi fått under detta projekt kunnat att tas med för att snabbare kunna bygga upp en svit av tester som kunnat täcka en större del av projektet.

4 Application of Scrum

4.1 The roles you have used within the team and their impact on your work

4.1.1 Sammanfattning av projektet (A)

Gruppen har valt att jobba med roller som redan finns definierade inom Scrum; scrum master, product owner och scrum team. Under detta projekt har scrum master rollen bestått av att hålla kolla så att de planerade mötena äger rum, men också att de inlämningar som gjorts skickas in i tid. Utöver detta har scrum master haft uppgifterna att se till att alla i gruppen är aktiva och att det som ska behandlas under mötena tas upp. Denna roll har gruppen valt att rotera mellan gruppmedlemmarna så att alla kan få testa på. Eftersom gruppen bestått av 7 medlemmar och projektet pågick i 6 veckor kunde inte en person vara scrum master, vilket inte ses som ett problem enligt gruppen.

I projektet har scrum team bestått av hela gruppen, inklusive product owner och scrum master. Product owner rollen har stannat hos en och samma gruppmedlem under hela projektet. Gruppen valde att låta denna roll stanna hos en medlem dels för att gruppens kund och product owner känner varandra sedan tidigare, dels för att inte göra det förvirrande för kunden. Product owner har haft som uppgift att hålla kunden uppdaterad under projektets gång men även att påminna scrum team om att utföra de user stories som kunden tyckt varit viktiga. Kommunikationen mellan product owner och kunden blev tätare och tätare ju längre projektet pågick, för att gruppen insåg hur viktigt det är att hålla kunden uppdaterad under hela sprinten i stället för att bara stämma av i slutet av varje sprint.

Gruppen har även valt att använda sig av egna roller under projektets gång. Rollerna har varit Rapport, Kod och GitHub/Utvecklingsmiljö frågor. Dessa roller innebar extra ansvar inom något av områdena, inte att en medlem endast skulle ha med det specifika område att göra. Dessa roller skapades så att allt ansvar inte läggs på en enda person utan de skapades så att alla kan dra sitt strå till stacken.

4.1.2 Hur bör framtida projekt se ut? (B)

I ett framtida projekt vill gruppen fortsätta som under nuvarande projekt. Gruppen tycker det har fungerat bra med att byta scrum master varje vecka, då alla har lärt sig mer om vad just scrum master rollen handlar om. Gruppen inser dock att det är annorlunda i arbetslivet och att det då inte skulle fungera att byta scrum master varje sprint.

Rollen product owner har även den fungerat bra under projektet och det är något som, i ett framtida projekt, tas med och användas i samma sätt som det har gjort. Att ha kontakt med kunden under projektet har varit väldigt värdefullt för gruppen, då kundens åsikter har hjälpt

leda oss på rätt väg. Gruppen ser även att detta skulle fortsättas med om ett projekt skulle dyka upp i framtiden.

4.1.3 Vad krävs för att nå dit? (A till B)

För att uppnå liknande resultat i framtida projekt bör gruppen tidigt skriva upp ett socialt kontrakt där förbestämda roller finns. Detsamma gäller för product owner och scrum master. Genom att bestämma dessa roller tidigt och förmedla vad de innebär gör det tydligt vart man ska vända sig om man har frågor till exempel angående kundens åsikter eller hur en speciell user story ska implementeras.

4.2 The agile practices you have used and their impact on your work

4.2.1 Sammanfattning av projektet (A)

Under projektets gång har gruppen jobbat med Trello, en webbapplikation i kanban-stil. Trello har använts för att skapa en scrum board som innehåller en product backlog tillsammans med epics och user stories. De epics och user stories som hör ihop har blivit givna samma färg, så att det enkelt ska gå att se vilka user stories som hör till vilken epic. Gruppen har använt denna scrum board och planerat varje sprint utefter den prioriteringen av de user stories som finns. Då gruppen har valt att jobba med parprogrammering så är detta även ett bra sätt för alla att hålla koll på hur alla grupper ligger till. KPIerna har bidragit till att få en god uppfattning av projektet under tidens gång.

Gruppen har även haft tidsbestämda möten där projektet diskuteras. I början av varje sprint hålls ett möte så att gruppen kan planera kommande sprint. Där bestäms vilka user stories som ska utföras under sprinten och vilka som ska jobba med vad. Även om det fanns dagar som inte hade något möte inplanerat så kunde medlemmarna fortfarande höra av sig till gruppen via Messenger.

Gruppen har även haft sprint reviews i slutet av varje sprint där även kunden närvarat en stund. Därefter diskuteras de KPI:er som gruppen använt.

4.2.2 Hur bör framtida projekt se ut? (B)

Gruppen tycker att det fungerat bra med de agila arbetssätt som använts och kommer att ha de i åtanke inför ett framtida projekt. Om ett framtida projekt skulle uppstå så skulle schemalagda möten vara något av det första som planeras. Via daglig kommunikation så kan man som grupp säkerställa att alla håller sig till tidsplanen. Det hjälper också att hålla alla informerade om problem som uppkommit men även problem som har blivit lösta. Detsamma gäller angående kontakt med kunden.

Det gäller även att hålla koll på att ens scrum team mår bra. Genom att implementera KPI:er, som Niko-niko, kan gruppmedlemmarna enkelt beskriva hur de mår med hänsyn till

projektet, och det blir lättare att veta hur ens team har det. Och om ens scrum team mår bra, presterar de bättre.

4.2.3 Vad krävs för att nå dit? (A till B)

Det som kommer att krävas i ett framtida projekt är att sätta dagar och tider för möten. Detta görs i början av projektet och är något som måste hållas på tills projektet är slut. Det kan även vara bra att ha något typ av ”straff” om man antingen missar eller kommer för sent till ett möte, som till exempel att man måste köpa fika till nästa möte. Att skriva in vad det som gäller för de planerade mötena i projektets sociala kontrakt kan vara ett bra sätt att få alla att närvara.

Angående KPI:er så är det liknande där, bestäm tidigt vilka av de många KPI:er som finns projektet ska följa. Det är viktigt att välja de KPI:er som passar ens projekt bäst för att få ett så bra resultat som möjligt.

Det gäller att tidigt sätta upp ett bra chatt-forum, så att man även kan nå ens team utanför mötena. Detta gör att grupper blir mer flexibla då de inte behöver vänta tills nästa möte för att ställa en fråga och få svar.

4.3 The sprint review and how it relates to your scope and customer value

4.3.1 Sammanfattning av projektet (A)

Vi har haft våra sprint reviews på fredagar. Vår kund har varit med varje vecka, och vi har visat henne vad vi har jobbat på under veckan, och berättat hur veckans arbete gått. Vi har visat henne appen i Android Studio, genom att någon av oss delat vår skärm och visat upp det vi har jobbat med på emulatorn. I början hade vi inte rutinen att säkerställa att allt låg på main-branchen innan demot, men vi insåg snabbt att det var mycket bättre för att demot skulle bli sammanhängande och tydligt för kunden. Det var endast under en sprint som vi inte hann med alla user stories, och då förklarade vi för kunden hur vi planerade att lösa det i nästkommande sprint. Vi var även tydliga med att berätta om de delarna av user storyn som vi faktiskt blev klara med under veckan. Det gjorde så att hon förstod att vi gjort framsteg även om de inte syntes på ett demo. Efter demot av appen har vi visat vår scrum board för kunden så att hon kan se hur prioriteringen av user stories ser ut inför nästa sprint. Då har hon även fått möjligheten att komma med åsikter och ändra prioriteringen.

Vår product owner Kerstin har pratat med kunden under sprinten för att ge henne insyn i vad vi jobbar med. I gruppen har vi ibland kommit med förslag om förändringar, till exempel kring placering av en knapp, och då har Kerstin skickat meddelanden till kunden för att bekräfta att kunden tycker att dessa ändringar tillför värde för henne.

Det var först andra veckan vi lade till punkten “att kunden ska vara nöjd” i vår definition of done. Eftersom vi inte har tidigare erfarenhet av att arbeta agilt tog det nog helt enkelt lite tid för oss att helt få grepp om hur man jobbar med kunden i centrum, men vi kände redan efter vår första sprint review att det var självklart att inkludera denna punkt i vår DoD.

Vi insåg också att våra demon inför kund ledde till att vi såg produkten utifrån på ett sätt som vi i början inte gjorde under själva sprinten. Till exempel upptäckte vi saker som inte var så användarvänliga eller snygga rent designmässigt först när vi höll i ett demo för kunden, vilket vi reflekterade över efteråt. Det gjorde att vi efter det försökte hålla kunden ännu mer i tankarna då vi själva testade appen, för att få en känsla för hur en utomstående upplever den.

4.3.2 Hur bör framtida projekt se ut? (B)

Vi tycker att vi i det stora hela lyckats bra i vår kontakt med vår kund. I nästa projekt kommer vi därför att fortsätta med mycket av det vi lärt oss i denna kursen, men självklart finns det delar som vi känner att vi kan förbättra.

En sak som vi diskuterat i team reflections och på möten är att kunden i nuläget inte haft möjlighet att själv köra appen, varken under veckan eller under vår sprint review. Kunden uttryckte i början av projektet inte själv önskemål om att kunna göra det, vilket gjorde att vi inte prioriterade det. Vi började alltså hålla demos under sprint reviews, och fortsatte så under projektets gång, och vi kände att vi hellre lade tid på att programmera och på så vis skapa värde för kunden. Trots att det fungerat bra tror vi att det hade gett kunden en ännu bättre möjlighet att upptäcka appen, och på så sätt komma med mer och bättre feedback till oss.

Om vi i framtida projekt ändå skulle jobba med att visa appen genom demos, bör dessa vara mer strukturerade. Det skulle göra det lättare för kunden att förstå vad som har tillkommit under veckan.

I framtida projekt vill vi redan från början fokusera ännu mer på kunden, och se till så att utveckling alltid sker med kundens behov i bakhuvudet.

4.3.3 Vad krävs för att nå dit? (A till B)

Att kunden kan köra programmet själv kan lösas på flera sätt, och det viktigaste för att åstadkomma det är nog att sätta det som ett mål redan i början av projektet. Då kan man tillsammans med kunden komma fram till det sättet som fungerar bäst, och avsätta tid för att lösa det.

För att göra demot bättre och tydligare för kunden kan det förberedas mer i förväg, så att gruppen har tänkt ut en tydlig struktur för allt som ska visas. Att öva på demot i förväg skulle även säkerställa att det inte finns några oupptäckta buggar eller problem. För att göra demot mer sammanhängande så är det viktigt att allt man jobbat med ligger på samma branch. Att se

till att allt ligger på main-branchen får man etablera som rutin, och detta kan skrivas in det sociala kontrakt så att alla är medvetna om kravet.

Ett sätt att skapa mer värde för kunden är att se produkten hur dennes ögon. Vi tror att det kan göras genom att försöka se produkten ur kundens ögon, och försöka tänka sig hur man hade upplevt den om man såg den för första gången. Vi i gruppen är så vana vid att använda appen, men genom att kontinuerligt gå igenom appen och tänka från användarens perspektiv så tror vi att det är möjligt att upptäcka saker som kan förbättras för att skapa mer värde för kunden.

För att lägga mer fokus på kunden så är det givetvis bra att ha med en punkt i definition of done som gör att inget kan klassas som färdigt förrän kunden har uttryckt att den är nöjd.

4.4 Best practices for learning and using new tools and technologies (IDEs, version control, scrum boards etc.; do not only describe which tools you used but focus on how you developed the expertise to use them)

4.4.1 Sammanfattning av projektet (A)

I detta projekt har vi använt Android Studio för att utveckla vår app, och Github för att lagra vår versionshistorik. För versionshantering har vi använt Git på olika sätt, dels GitKraken och Git Bash, samt direkt i Android studio. Vi har haft vår scrum board på Trello.

Parprogrammering har gjort att vi kunnat lära oss mycket från varandra, givetvis om programmering i sig men även om sätt att söka information och om versionshantering. Vi har insett att vi har olika sätt att använda versionshantering, och det har varit lärorikt för alla att växla mellan dessa sätt.

Vi har sett till att dela upp uppgifterna så att alla får testa på så mycket som möjligt. Detta kanske inte varit mest tidseffektivt, men det har gjort så att alla lärt sig väldigt mycket. Vi har även sett till att vi inte jobbar i samma programmeringsgrupper eftersom vi känner att vi lär oss mer genom att jobba med olika personer från vecka till vecka.

När vi jobbat i dessa mindre grupper finns ofta de andra grupperna i samma Zoom-rum, men i ett annat breakout-rum. Det har gjort att vi kunnat ta hjälp av andra gruppmedlemmar för att lösa problem.

Vi kom in i projektet med olika kunskaper, och vi har gjort vårt bästa för att lära varandra. I början spelade till exempel Marcus in en video om utvecklingsmiljön Android Studio där han berättade om grunderna, eftersom han var den enda som använt det tidigare. Vi har även läst mycket av Android Studios dokumentation på egen hand och allt eftersom delat med oss av relevanta länkar till varandra. Även videos från Youtube och källor som Stack Overflow har varit hjälpsamma för att lösa problem som kommit upp längs vägen.

Redan från början var vi noggranna med att planera för att undvika tidspress. Vi har försökt hålla arbetstakten på en lagom nivå, vilket har gjort att vi har haft tid att läsa på och lära oss.

4.4.2 Hur bör framtida projekt se ut? (B)

Vi är överlag nöjda med hur vårt projekt har fortskridit och känner att vi har lärt oss väldigt mycket. Därför skulle vi i framtida projekt ta med oss mycket av det vi har gjort denna kurs. Att programmera i mindre grupper har fungerat väldigt bra, och om det är möjligt i framtida projekt ser vi det som en framgångsfaktor.

Vi är även nöjda med att vi hjälpt varandra så mycket på de områden där vi har haft olika kunskaper. Tack vare den goda stämningen i gruppen har alla känt sig bekväma med att ställa frågor. I framtida projekt är det alltså värdefullt att hålla en bra stämning inom gruppen.

4.4.3 Vad krävs för att nå dit? (A till B)

För att få en bra stämning i gruppen är det bra om man lär känna varandra i början i projektet så att alla känner sig bekväma. Enligt vårt social contract har även vi haft kameror på under hela projektet, och även det är något som bidrar till en bättre stämning då det känns som att man faktiskt pratar med någon “på riktigt”.

Vi tror även att det är bra att stämma av nivån på alla i gruppen i början, och att då våga vara ärlig om det är något man inte känner att man riktigt behärskar. På så vis kan man hjälpa varandra så gott man kan.

En ytterligare viktig princip är nog att inte överskatta hur mycket man ska hinna med, särskilt i början av ett projekt. Om man räknar med att saker tar lite längre tid kan man undvika stress och hinner då lära sig betydligt mer.

4.5 Relation to literature and guest lectures (how do your reflections relate to what others have to say?)

4.5.1 Sammanfattning av projektet (A)

Vi har inte haft några gästföreläsningar under denna kursen, men vi har använt materialet från föreläsningarna som vi hade i början av kursen. Innan vi var vana vid att arbeta med scrum var kursmaterialet en användbar överblick över arbetssättet.

Vi hade ingen specifik kurslitteratur, men som nämnt i tidigare fråga läste relevant dokumentation. Vi har särskilt använt Android Studios dokumentation: https://developer.android.com/docs

4.5.2 Hur bör framtida projekt se ut? (B)

Även i framtida projekt siktar vi på att i stor utsträckning använda oss av “officiell” dokumentation, då det är en säker källa som ofta är skriven på ett pedagogiskt sätt med hjälpsamma exempel.

För att alla ska ha tillgång till länkar som varit hjälpsamma är det en god idé att samla alla källor som varit till hjälp på en plats. Det hade gjort det lättare att hitta tillbaka till information man tidigare använt. Genom att göra informationssökandet mer strukturerat i början kan man antagligen spara tid senare under projektets gång.

4.5.3 Vad krävs för att nå dit? (A till B)

Ett enkelt sätt att samla information är att skapa ett dokument som samtliga gruppmedlemmar har tillgång till, och sedan under projektets gång samla hjälpsam information där.

I projektets början skulle det vara bra att dela upp informationssökande mellan gruppmedlemmarna, och sedan dela med sig av informationen sinsemellan. Det skulle effektivisera informationssökandet, och samtidigt ge gruppen en stabil grund att stå på.