

SnoozeLoze

Sleep smart. Have more fun.

Post Mortem Rapport



Thorleifz

Daniel Chiniquy
Andreas Henriksson
Rebecca Härkönen
Hampus Lück
Jacob Moos
Adam Myrén

Innehållsförteckning

1. Inledning	3
1.1 Bakgrund	3
1.2 Vision	3
1.3 Appen.....	3
2. Metod	4
2.1 Scrummetodik.....	4
2.2 Sprints	4
3. Resultat och diskussion.....	6
3.1 Processer och arbetssätt i projektet	6
3.1.1 Tre olika branches i GitHub	6
3.1.2 Scrum var en stor tillgång för projektet	6
3.1.3 Arbete i mindre grupper inom gruppen	7
3.2 Tid spenderad på projektet.....	7
3.3 Fördelar med vald teknik baserat på erfarenhet från projektet.....	8
3.4 Nackdelar med vald teknik baserat på erfarenhet från projektet	8
3.5 Teknikens effektivitet och tidsåtgång för att använda den	9
3.5 Situationer då tekniken skulle kunna tillämpas i framtiden	9
3.6 Situationer då tekniken inte bör tillämpas i framtiden	10
3.7 Fördelar med arbetssättet	10
3.8 Nackdelar med arbetssättet.....	10
3.9 Reflektioner kring beslut ej relaterade till någon specifik process	11
3.10 Samarbete och interaktion inom gruppen	11
3.11 Lärdomar inför framtida projekt	11
4. Sammanfattning och slutsats.....	12

1. Inledning

1.1 Bakgrund

I januari tidigare i år fick två av Thorleifz gruppmedlemmar den stora möjligheten att tillsammans med en god vän skriva ett kandidatarbete. Tidigt i arbetsprocessen visade det sig dock att gruppmedlemmarna hade väldigt skilda inställningar till tidspassning. Den tredje gruppmedlemmen kom, trots ständiga påpekanden från övriga i gruppen, oftast mellan en till två timmar sent till varje möte. Det gick så pass långt att gruppen började reflektera kring huruvida den så ofta sent anlända gruppmedlemmen hade sömnproblem, missbrukade sitt alarms snoozefunktion eller överhuvudtaget ställde ett alarm. Detta scenario ledde fram till visionen av Android-appen SnoozeLoze.

1.2 Vision

Har du någonsin varit med om att en gruppmedlem kommer försent till ett möte? Har du någonsin oroat dig över om andra gruppmedlemmar minns mötet följande dag? Inget av detta kommer att vara ett problem med SnoozeLoze.

Gruppfunktionen låter dig skapa grupper i vilka du kan ställa in ditt alarm och samtidigt se dina vänners, din familjs eller dina arbetskamraters alarmtid, medan de kan se din. Var medlem i flera grupper samtidigt och ge varje grupp ett formellt eller roligt namn - detta gör SnoozeLoze perfekt för både privat och professionellt bruk.

Vill du veta vem som stigit upp? Vill du få notiser när någon snoozear? SnoozeLoze är den perfekta appen för att säkerställa att dina vänner är i tid till tåget på morgonen eller att dina arbetskamrater är i tid till ert möte. Sov smart och ha mer tid att ha kul tillsammans!

1.3 Appen

SnoozeLoze är en app för så kallad "social alarmering". Appen är tänkt att nyttjas av exempelvis vänner, familjemedlemmar eller arbetskamrater, vilka ska kunna följa varandras alarmfunktion. I appen kan grupper skapas för olika ändamål och inom varje grupp kan respektive medlem ställa ett individuellt alarm som är synligt för övriga gruppmedlemmar. Varje användare kan ingå i flera sådana grupper och i var och en av dessa kan ett alarm ställas. När det är dags att gå upp på morgonen kan övriga gruppmedlemmar uppmärksamma om någon snoozar och skicka meddelanden till denne. Genom att i varje grupp ha översikt över övriga gruppmedlemmars alarmtid och status kan gruppen hålla koll på varandra så att alla kommer upp och är i tid på morgonen.

2. Metod

2.1 Scrummetodik

Projektet valdes att genomföras genom scrummetodik¹ i form av fem kortare sprints med ett tidsintervall på en vecka var. Detta tillvägagångssätt valdes efter att gruppen testat scrummetodiken under en övning under den inledande delen av kursen och kommit fram till att arbetssättet passade gruppen bra och var något alla ville testa under projektet. De första fyra sprintsen bestod i huvudsak av planering och utveckling av appen, medan den femte och sista sprinten till största del bestod av att bugfix, sammanställning av slutgiltig dokumentation, rapportskrivande och presentation av projektet.

Vid projektets start definierades en vision och en product backlog i form av ett antal themes för vilken funktionalitet som appen skulle innehålla, som sedan bröts ner till ett antal epics, i enlighet med tillvägagångssättet Thomas Luvö beskrev under sin föreläsning "Scaling Agile".

Då ingen egentlig kund fanns för projektet valde projektgruppen att se handledaren som kund, även om gruppmedlemmarna själva delvis stod för denna roll. Inte heller någon product owner fanns för projektet utan projektgruppen var gemensam ägare. Gruppen kom tillsammans överens om att inte tilldela varandra roller eller utse någon scrum master i gruppen då det inte ansågs nödvändigt med tanke på gruppens begränsade storlek. Istället ansågs det mest effektivt att tillsammans ansvara för beslut för att alla medlemmar skulle kunna delta aktivt i alla sprints.

2.2 Sprints

Vid starten av varje sprint gjordes en sprint planning där gruppmedlemmarna tillsammans planerade sprinten. Då valdes ett antal epics ut från backloggen som bröts ner till user stories som sedan skulle slutföras under sprinten. Dessa valdes ut efter hur kritiska de var för appens funktionalitet indelat i en tregradig skala (bestående av "viktig", "semi-viktig" och "kul"), men även efter hur omfattande de uppskattades vara att utveckla. Det användes dock ingen systematisk metod för att uppskatta våra user stories omfattning utan besluten baserades på erfarenhet. Detta innebar att uppskattningen var svår att göra till en början men underlättades under projektets gång.

En annan aspekt som vägdes in vid valet av user stories inför varje sprint var att försöka tillföra synligt värde för kunden, genom att utveckla back-end och front-end funktionalitet parallellt och på så vis efter varje sprint kunna visa upp ett resultat som kunden kunde interagera med. Under varje sprintplanering sammanställdes både to do-listor baserade på de user stories som valts ut och how to-listor innehållande hur arbetet planerades att genomföras.

Under projektets gång kom backloggen att ändras genom att bland annat epics omprioriterades och antalet user stories utökades. Att arbetet genomfördes i scrum var något som underlättade dessa förändringar. Genom att dela upp arbetet i sprints kunde projektet riktas om i önskad riktning efterhand vilket gjorde arbetsprocessen flexibel.

¹ Scrummetodiken är baserad på föreläsningsmaterial, övningar samt wikipedias beskrivning (http://en.wikipedia.org/wiki/Scrum_%28software_development%29)

Under varje sprint delades arbetet in i mindre grupper inom gruppen. Detta gjordes eftersom gruppmedlemmarna hade olika erfarenhet inom Android-programmering och att det ansågs effektivare att samarbeta än att arbeta enskilt. Varje sådan grupp fick ansvar för ett antal relaterade user stories och den funktionalitet som var kopplad till denna. Under vissa sprints användes pair programmering vid en dator där den ena personen kodade och den andra sökte information på en annan dator och hjälpte till att kontrollera koden. Under resterande sprints skedde kodandet i grupper om tre där en person fortfarande skrev koden medan de övriga två exempelvis sökte information, testade koden och arbetade i databasen.

Under programmeringen användes Git branches. En branch skapades för varje feature där koden relaterad till denna utvecklades, testades och bugfixades. Varje sprint avslutades i samband med ett handledningsmöte. Inför detta mergades alla branches i Git in till en developer-branch. Detta gjordes för att få en ny version av appen från varje sprint. Ursprunglig plan var att använda en master-branch, men denna kom att bli ganska onyttjad. Handledningen fungerade som en sprint review där det diskuterades vilka user stories som levererats under varje sprint och vilka user stories som var lämpliga att prioritera under nästa sprint. Den fungerade även som en sprint retrospective där det diskuterades kring arbetssättet under sprinten och vilka lärdomar gruppen kunde ta med sig till nästa sprint.

Under arbetets gång har det eftersträövats att följa scrum så gott det går, men det har dock inte alltid varit möjligt. Exempelvis var det svårt att samordna alla gruppmedlemmar för daily scrums vilket ledde till att en gemensam dagbok istället skapades där de mindre gruppindelningarna kunde rapportera vad de arbetat med och hur det gått.

3. Resultat och diskussion

Under projektets gång har gruppen givits möjlighet att testa olika tekniker och arbetssätt. Gruppen har utefter egen planering och förutsättningar gjort flera val, vilka vid tidpunkten ansågs vara de mest effektiva för projektet. Dessa val har bidragit till att gruppen exponerats för nya tekniker och kunskap, från vilka nya erfarenheter kunnat tillgodoräknas. Det blir således av intresse att föra en diskussion kring hur gjorda val påverkat projektet, gruppen och projektets slutgiltiga resultat. Vidare följer därför i detta kapitel en genomgående diskussion vars struktur kretsar kring de rekommenderade frågeställningar som anslagits på kursens hemsida.

3.1 Processer och arbetssätt i projektet

Samtliga medlemmar i gruppen hade olika erfarenheter och färdigheter när vi inledde projektet. Efter att tillsammans ha kommit fram till en idé som samtliga medlemmar var överens om började vi först att resonera kring vad vi skulle behöva för att kunna genomföra projektet. Redan innan projektets start hade vi fått testa att arbeta i scrum med gruppen, vilket gav samtliga medlemmar goda indikationer att det först och främst var på det sättet vi skulle arbeta projektet igenom. Under projektets gång kom vi att använda flera olika typer av programspråk, såsom SQL, Javascript och Java, för olika delar av appens funktionalitet.

3.1.1 Tre olika branches i GitHub

Initialt bestämde vi oss för att ha tre olika typer av branches: en master för varje veckas slutgiltiga version, en developer där alla merges av features skulle göras och där mindre bugfixar skulle göras, samt en featuretyp där varje ny feature skulle få en egen branch. Idén med detta var att tydligt möjliggöra för separation av arbetsuppgifter för uppnå effektivitet. Detta medförde också att varje ny klass fick en egen featurebranch, vilket dock gjorde att det var svårt att få koden enhetlig och fungera väl i det stora hela. I och med att vi dessutom jobbade i grupper blev det tidigt tydligt att alla dessa featurebranches inte gjorde det enklare för oss eftersom vi ofta hade bra koll på vad resterande gruppmedlemmar skrev i för klasser. Detta gjorde att vi efterhand kom att använda developerbranchen mer och mer för ändringar. Masterbranchen kom också att bli onyttjad.

3.1.2 Scrum var en stor tillgång för projektet

Att projektet genomfördes med scrummetodik medförde flera fördelar. Genom att projektet delades in i mindre delar i form av sprints blev arbetet mer flexibelt. Istället för att ha en enda projektplan som följs från dag ett för att i slutet utvärderas kunde denna process upprepas ett antal gånger under projektets gång och projektets riktning kunde således styras om med tiden. Detta var en fördel då visionen till en början var väldigt övergripande och det var först en bit in i arbetets gång som gruppen kunde se vilka idéer som var bra och vilka som var mindre bra. Genom att kontinuerligt planera, utföra och utvärdera kunde troligtvis ett bättre resultat uppnås än vad som hade varit möjligt med en vattenfalls-metod.

Den kontinuerliga reflektionen över vad som gjorts, vad som behövde göras och vad som fungerade eller inte gav även upphov till en större möjlighet för inläring hos alla gruppmedlemmar. Sprint meetings, reviews och retrospectives gav möjlighet att både utvärdera sin egen och övriga gruppmedlemmars insats, något som samtliga i gruppen är överens om har varit lärorikt.

3.1.3 Arbete i mindre grupper inom gruppen

Redan i den första sprinten bestämde gruppen gemensamt att vi skulle satsa på att dela upp oss i två till tre olika team som under sprinten fokuserade på olika uppgifter. Detta gav oss goda möjligheter att variera uppgifter och färdigheter i gruppen, samt genom att problemlösa tillsammans ge varandra möjligheten att lära oss från varandra.

Inledningsvis planerade vi projektet tillsammans och försökte i bästa möjliga mån att kartlägga hela projektet och processen. För att komma igång upplevde vi att det var en bra idé att dela på oss och uträtta skilda user stories för att på så sätt få så mycket gjort som möjligt. Eftersom det under den första sprinten fortfarande var oklart hur de olika delarna skulle utvecklas delade vi in oss i två grupper om tre, där ena gruppen fokuserade på att reda ut hur front-endsidan skulle utvecklas och den andra på hur back-endsidan skulle utvecklas. Utifrån detta kunde efterhand också många user stories utvecklas och idéer för hur appen skulle se ut och fungera kunde tas fram.

Under den andra sprinten bytte vi till tre grupper med två studenter i varje. Detta då det med tiden dök upp fler och fler uppgifter på våra to do-lists, samtidigt som nästan samtliga epics och user stories bröts ned till flera mindre user stories. I gruppen tog vi alltså beslutet att bemöta en ökande arbetsbörda med att sprida ut oss ytterligare, vilket var ett beslut som i detta skedet möjliggjordes av att en god grund nu hade arbetats fram i projektet. Detta fungerade bra i bemärkelsen att många av alla user stories blev genomförda. Samtidigt blev integrationen i appens olika delar sämre vid detta stadie. Vi upplevde att det var svårt att hålla koll på vad de andra grupperna hade gjort trots uppdatering och samordning genom den gemensamma dagboken. Vi hade i detta stadie lyckats skapa experter på olika delar av mjukvaran, men få eller ingen av oss hade mer övergripande koll.

Inför den tredje sprinten hade många olika delar av appen utvecklats, såsom databasens uppbyggnad och funktionalitet, en mängd klasser och metoder i Java samt de JavaScript som skulle fungera som "mellanhand" mellan databasen och Java-koden. För att i sprint tre kunna integrera dessa delar på ett effektivt sätt delades gruppen åter in i två grupper om tre personer. Detta också för att maximera gruppmedlemmarnas lärande och insyn in appens design. Dessa två grupper höll sig under arbetstillfällena nära varandra för att enkelt möjliggöra kommunikation. En fördel med denna gruppindelning var att samtliga gruppmedlemmar fick bättre kontroll över hur appen var byggd och hur olika delar hängde samman. En nackdel var å andra sidan att tempot för utvecklingen minskade i förhållande till föregående sprint. Vi bedömer dock så här i efterhand att denna fas möjliggjorde den välfungerande integration som vi lyckades åstadkomma med appens olika delar.

I den avslutande delen av projektet, alltså i sprint fyra och fem, varierade vi mellan att arbeta två studenter med dokumentation och test, samtidigt som övriga fyra gruppmedlemmar fokuserade på att bugfixa och finjustera appen. I det absoluta slutskedet ansåg vi att det var viktigt att vi alla arbetade tillsammans för att i bästa möjliga mån kunna utvärdera hela projektet med samtliga gruppmedlemmars åsikter i betraktelse, samt att vi tillsammans kunde planera, skapa och genomföra presentationen av projektet.

3.2 Tid spenderad på projektet

I planeringsfasen kom gruppen överens om att målet var att följa de riktlinjer som föreslagits, vilket resulterade i att samtliga gruppmedlemmar arbetade 12-15 timmar per sprint. Under varje sprint arbetade gruppen vid flera tillfällen, vilka var allt mellan fyra till sju timmar långa, beroende på hur väl

gruppmedlemmarnas individuella scheman korrelerade. Då en sådan struktur följdes arbetade varje gruppmedlem cirka 75 timmar med projektet sammanlagt, vilket sedermera innebär att den totala arbetsinsatsen från gruppen var cirka 450 timmar.

3.3 Fördelar med vald teknik baserat på erfarenhet från projektet

Scrum har varit en väldigt god erfarenhet då rutinen har hjälpt oss att ständigt hålla rätt kurs och vid behov har vi kunnat parera för uppgifter eller störningar som kommit upp i scrum meetings under hela processen. Hade vi inte använt oss av scrum hade det varit svårt att ständigt anpassa arbetet under processens gång med avseende på eventuella förändringar under projektet. Exempelvis möjliggjorde scrum meetings för att ständigt uppdatera varandra inom gruppen på status på olika delar av processen. Det var av stort värde då vi under hela processen roterade arbetsområden, arbetslag och uppgifter. Scrum gjorde det enkelt för samtliga gruppmedlemmar att rotera då vi hela tiden kunde ta vid där någon annan lämnat och samtidigt vara uppdaterade på vad som hade gjorts, hur det hade gjorts, vad som borde göras och hur detta potentiellt skulle kunna göras.

Att arbeta antingen i hela gruppen, i grupper om tre eller par om två har varit en mycket lärorik erfarenhet för samtliga inblandade. Då vi inför projektet hade olika färdigheter på olika områden samt olika intressen och preferenser kunde vi genom ett gott samarbete tillgodoräkna oss samtliga gruppmedlemmars färdigheter genom att dela in gruppen i mindre grupper. Genom att undvika att arbeta helt själva gavs möjligheter för att hela tiden diskutera samt bolla idéer med varandra i grupperna inom projektet. På så sätt kunde vi ta oss fram med större effektivitet och kvalitet, vilket vi samtliga känner har varit en stor orsak till att vi kunnat uppnå de mål vi tillsammans skapade redan under inledande delen av kursen.

Då vi under hela utvecklingsprocessen programmerat tillsammans har vi också haft en viss kvalitetssäkring genom att flera ögon alltid funnits på koden. Vi tror att vi därigenom fick god överblick, vilket underlättade vid de bugfixar som i slutskedet av utvecklingen genomfördes då vi alla tillsammans kunde hjälpas åt att reda ut problem som uppstod någonstans mellan appens olika delar. Samtidigt var det utvecklande för gruppmedlemmarna att få chansen att lära sig lösningar som den andra studenten i de mindre grupperna kände till.

Under utvecklingen gjordes kontinuerligt mindre tester av koden genom att den testades praktiskt på en telefon eller i en emulator. Dessa tester genomfördes i enlighet med de beskrivningar som finns i "Acceptance test"-dokumentet. På så sätt kunde de många olika user stories som byggde upp appen kontrolleras innan de slogs ihop med den senaste versionen av appen. Detta gjorde också att de buggar som uppstod i developerbranchen uppstod av integrationsskäl snarare än logiska fel inom de mindre delarna. Genom att vi dock arbetade såpass tätt inom gruppen kunde dessa integrationsbuggar för det mesta lösas utan några större ansträngningar. Eftersom appen nästan uteslutande bestod av användargränssnitt valde vi att inte göra några autotester. Detta tror vi inte försämrat eller bidragit till några buggar som inte hade tagit väldigt mycket tid att testa och som antagligen betydligt mer effektivt kunde testas likt beskrivet ovan.

3.4 Nackdelar med vald teknik baserat på erfarenhet från projektet

Som vi nämnt tidigare använde vi oss inte av en scrum master. Detta var kanske ett av skälen till att vi stötte på mjukvaruintegrationsproblem och behövde återgå till att arbeta i grupper om tre. Samtidigt gjorde det faktum att vi inte använde en scrum master att vi inte fördjupade oss för mycket i speciella

områden utan roterade vilket gav god insikt i koden. Det faktum att många ögon hela tiden fanns på koden bidrog antagligen till att koden kunde "strömlinjeformas".

Vi programmerade i princip inte ensamma vid något skede i utvecklingen. Istället gjorde vi det i par eller i större grupper. Detta kan mycket väl ha påverkat tempot på programmeringen. Samtidigt bidrog det till att kvaliteten kunde hållas hög och jämn, vilket också kan ha hjälpt till att uppehålla ett bra tempo när det kom till att integrera de många olika delar som mjukvaran bestod av.

Samtliga gruppmedlemmar upplever att gruppen inte fått några negativa erfarenheter relaterade till scrum. Detta kan tänkas bero på att projektet vi genomfört, appen och tidsramen verkar ha passat arbetstekniken väl. Samtidigt följde vi inte rakt av en scrum metodik då vi bland annat inte använde oss av en scrum master. Detta kan ses som resultatet av utvecklingsgruppens storlek. Vi tror att om vi haft svårare att kommunicera med varandra, arbetat mer på skilda håll och individuellt så hade en scrum master kommit till mer nytta.

3.5 Teknikens effektivitet och tidsåtgång för att använda den

Att vi arbetade med programmering i grupp har säkerligen medfört att utvecklingstempot för enskilda user stories minskat. Samtidigt, som också diskuterats ovan, så bidrog denna metod för programmeringen till att koden kunde hållas jämnare och mer lättförstådd, vilket också gjorde integrationen av de olika delarna enklare. Här finns alltså antagligen en viss trade-off mellan att arbeta i grupp och arbeta individuellt i förhållande till tiden utvecklingen tar för projekt.

Vidare kan nämnas att vi lagt en del tid på att formulera tydliga user stories och därefter delat in dessa till de olika utvecklingsgrupperna. Denna tid tror vi däremot framförallt gynnat oss eftersom vi därigenom till stor del kunnat undvika redundant arbete och konflikter i Git vilket vi tror hade tagit mer tid.

Totalt har vi inte upplevt av teknikerna tagit mer tid än den tid de sparat oss. I det stora upplever vi att den scrummetodik som vi arbetat efter sparat oss tid genom att vi fått tydlig struktur i hur vi arbetade. Att först förstå hur vi skulle arbeta med scrum var däremot något som tog viss tid och fortsatte utvecklas efterhand i projektet.

3.5 Situationer då tekniken skulle kunna tillämpas i framtiden

Bortsett från just projekt liknande det genomförda i denna kurs hade scrum även passat sig bra i projekt där förutsättningar kan förändras under processen. Sprintmeetings hade möjliggjort för gruppen att identifiera och analysera sådana förändringar i nära anslutning till det tillfälle dessa dykt upp och med hög flexibilitet anpassa projektets förutsättningar, resurser och mål därefter. Gruppen anser att scrum hade varit väldigt bra i framtida projekt, i vilka antalet gruppmedlemmar är stort. I sådana projekt tänker gruppen att det kan vara svårt att hela tiden kommunicera förändringar på spridda områden till många personer utan att ha en väldefinierad och funktionell arbetsteknik att följa likt metodik då scrum "skalas upp".

Samtliga gruppmedlemmar anser även att det hade varit intressant att få erfarenheter av att applicera ett sådant arbetssätt på större projekt över längre tid. Exempelvis hade det varit lärorikt och intressant att få applicera sådan teknik i ett kandidatarbete, examensarbete eller senare även i arbetslivet. Några av de stora fördelarna med scrummetodik som vi kan ta med oss är att dela in arbeten i mindre delar

som därefter kan fördelas på de olika medlemmarna i gruppen, att sätta en veckobaserad plan för indelningen snarare än att hela tiden ta fram nya deadlines, och snarare dela ut ett rimligt antal uppgifter per vecka och medlem än att bestämt sätta deadlines för varje liten del. På så sätt lämnas utrymme för ansvarig för ett område att självständigt planera arbetet såsom passar denne. Samtidigt var en av de viktigaste momenten i utvecklingen av SnoozeLoze den att vi programmerade i grupper. Även detta skulle kunna prövas i andra sorters projekt för kvalitetssäkring.

3.6 Situationer då tekniken inte bör tillämpas i framtiden

Gruppen anser att scrum inte nödvändigtvis hade behövts vid ett projekt med kort tidsram och väldigt få deltagare. Om projektet hade varit väldigt tydligt, inneburit få user stories, så hade scrum kunnat bli överflödigt. Likaså hade kommunikationen kunnat skötas kontinuerligt under processen om antalet gruppmedlemmar hade varit exempelvis två till tre personer och projektet enbart fortgått cirka en till två veckor.

Att arbeta i mindre grupper hade också kunnat anses som överflödigt vid tillfällen där färdigheter och kunskaper som krävs för projektet är av väldigt basal eller generell karaktär. I sådant fall menar vi att det kanske hade varit bättre att arbeta individuellt vid varsin dator och bli klara med projektet på kortare tid. Detta då förutsatt att projektet och dess innehåll är väldigt tydligt och grundläggande.

3.7 Fördelar med arbetssättet

Vi har alla fått tillfälle att lära oss av varandra då vi inför projektet hade olika kunskaper och färdigheter. Detta tydliggjordes i och med att vi allihopa tilläts testa samtliga arbetsområden då vi under hela processen jobbat med att rotera inom gruppen. Rotationen innebar också att vi hela tiden kunde få fräscha ögon på alla delprojekt då gruppmedlemmar vid flera tillfällen fick ta vid där någon annan gruppmedlem redan inlett arbetet. Ett sådant arbetssätt har i hög grad möjliggjorts av att samtliga i gruppen visat stort intresse för att lära sig nya saker och uppnå vårt gemensamma mål. Vidare möjliggjorde även tillfälliga arbetsgrupper inom gruppen i sig att vi kunde arbeta i par där varje individ kunde olika saker och på så sätt blev den samlade kunskapen större.

Dessutom kan också nämnas att de flesta av de projektaspekter som nämnts ovan såsom programmering i grupp, kontinuerlig testning av ny funktionalitet, varierad gruppstorlek med flera menar vi fungerat bra enligt de resonemang som förts kring dessa aspekter ovan.

3.8 Nackdelar med arbetssättet

Grenarna i GitHub kan nämnas som ett av områdena som fungerat mer problematiskt. Genom att det inte fanns någon tydlig struktur för när vi skulle skapa nya grenar för features, hur ofta vi skulle genomföra commits och merges och så vidare så gjordes här antagligen en del arbete som hade kunnat undvikas eller effektiviserats. Master-branchen användes inte heller för sitt tilltänkta syfte då det inte användes för att merga ut varje veckas sprint, alltså färdiga eller funktionerande versioner av appen.

Dokumentationen var till en början spridd eftersom vi hade olika idéer och inte visste hur vi skulle samla den (se vidare avsnitt 3.9). Detta medförde att dokumentationen kan ha tagit mer tid än vad som är önskvärt, men samtidigt användes denna effektivt som en del av utvecklingsarbetet genom att den samlade information kring utvecklingen som gjordes.

3.9 Reflektioner kring beslut ej relaterade till någon specifik process

Inledningsvis var vi osäkra på vilken metod för att kommunicera och samla data (user stories och liknande) som skulle fungera bäst. Detta gjorde också att kommunikationen blev splittrad under första sprinten då viss kommunikation hamnade på en facebook sida, viss i en chatt, viss i Trello och viss i Google Drive-dokument. Anledningen att det blev så splittrat var antagligen att vi inom gruppen var vana vid olika rutiner. Vi märkte dock efterhand att Facebook-gruppen inte blev använd, medan Facebook-chatten blev den samlade platsen för mötesplanering. Ett gemensamt beslut om att frånga Trello och samla all information i Google Drive gjordes efter en tid för att undvika redundant information och skapa ordning. Anledningen till att vi valde Google Drive framför Trello var för att vi hade större vana av att använda detta verktyg.

3.10 Samarbete och interaktion inom gruppen

Samtliga gruppmedlemmar har under hela projektet varit mycket öppensinniga till uppgiften. Alla bidrog med ett brett utbud av färdigheter, idéer och lösningar för oss att planera, skapa och utvärdera utifrån. Hela gruppen har varit mycket flexibel i dess sätt att arbeta och samtidigt använt rak och tydlig kommunikation. Detta gjorde det mycket enkelt för gruppen att testa olika arbetssätt, utvärdera dessa och sedan vidareutveckla för att bli ännu effektivare. Exempelvis har det varit en stor tillgång för gruppen att hela tiden kunna dela in oss i olika grupper med olika uppgifter under projektets gång. Detta har låtit gruppmedlemmar med liknande scheman att arbeta tillsammans under kortare perioder samt att detta i sig möjliggjorde för oss att lära från varandra då vi alla hade olika färdigheter och kunskaper i det inledande stadiet av projektet.

3.11 Lärdomar inför framtida projekt

En av de största hindren i detta projektet var att det drevs vid sidan om ett större projekt med kandidatarbetet. Utan att det betydde att vi hade lägre ambitioner, så betydde det att det blev svårare fokusera på visionen för appen och de uppgifter som krävdes. Eftersom vi upplevde att det fanns en viss uppstartningsfas i samband med varje möte tror vi att det vore lättare om man i framtida projekt använde arbetspass längre än fyra timmar. Detta speciellt om projekt drivs vid sidan om, då dessa annars tar visst fokus och försvårar uppstartsfaserna under möten.

I nästa scrumprojekt kommer vi att utifrån en liknande, men mer genomtänkt grenindelning i Git för att hantera versionerna av koden. Tanken med en developbranch som genom avknoppningar i form av features kändes intuitiv, men vår implementation kanske inte lyckades eftersom vi hade svårt att avgöra när det var värt att göra en avknoppning eller ej. Därför kanske det vore bra inför nästa projekt att ha någon slags specifikation för när en avknoppning är en bra idé eller ej för att därigenom också få en mer konsekvent struktur i Git.

Något som vi var nöjda med, men ändå menar att det finns utrymme för förbättring för, är hur sprintarna var upplagda. Om vi varit mer utförliga med att utvärdera arbetsbördan för olika user stories kanske vi effektivare hade kunnat dela in arbetet mellan olika grupper i projektet och på så sätt fått högre tempo i utvecklingen. En annan aspekt kopplad till indelningen av user stories är att vi inte använde en scrum master. I detta projektet fungerade det bra, men samtidigt uppstod ju som nämnt integrationssvårigheter med appen olika beståndsdelar. I fallet av en större projektgrupp hade kanske en scrum master varit nödvändig för att åstadkomma den överblick som krävs för att få projekt att fungera.

Även om en av de största fördelarna med projektet var flexibiliteten och dynamiken i gruppen, så är det tänkbart att den inte fungerar lika bra i framtida projekt. Då tror vi att det hade varit av nytta att ha en mer klar struktur över hur arbetet skulle se ut, exempelvis när nya grenar i Git skulle göras, mer specialiserade grupper eller lag i gruppen som stort, samt en scrum master som kan överblicka projektet och dess utveckling.

4. Sammanfattning och slutsats

I projektet och utvecklingen av appen har vi för första gången fått öva oss i scrummetodik i mjukvaruutveckling. Vi har sett stora fördelar i att dela in större uppgifter i mindre och arbeta med dem i mindre grupper. Genom att använda Github har vi också kunna få bättre insyn och förståelse i varför versionshantering är svårt och viktigt i mjukvaruutvecklingsprojekt samt metoder för att testa programmen kontinuerligt under utvecklingen. En av de främsta orsakerna att projektet har gått så bra som vi menar att det gjort tror vi är att gruppmedlemmarna har varit positiva och flexibla att byta runt och lära sig nya saker. Detta har möjligtvis gjort att vissa moment tagit längre tid än andra, men definitivt betytt att vi lärt oss mer inför kommande projekt. Ett sådant kommer bland annat att bli att efter sommaren fortsätta med utvecklingen av SnoozeLoze inför dess lansering.