

# **Post-mortem Report**

# ***STORYTELLER***

**Av utvecklingsteamet Gråzonen;  
Hannes Lagerroth, Willy Larsson,  
Sara Lingegård, Gustav Olsson,  
Johan Rebner, Erik Sällberg**

**Handledare: Håkan Burden  
Chalmers tekniska högskola  
Software Engineering Project  
DAT255**

# Innehållsförteckning

1	Introduktion.....	1
1.1	Projektets vision.....	1
2	Metod .....	2
2.1	Arbetsprocessen .....	2
2.1.1	SCRUM.....	2
2.1.2	Parprogrammering .....	3
2.2	Disponerad tid.....	4
2.3	Trello och Google Speadsheet .....	5
2.4	Emulatern.....	6
3	Diskussion.....	7
3.1	Problematik under utvecklingen .....	7
3.1.1	Branches som inte gick att merge .....	7
3.1.2	Dubbla serverlösningar .....	7
3.2	Reflektioner om arbetet i grupp .....	8
3.3	Vad vi skulle gjort annorlunda nästa gång.....	10
4	Sammanfattade tankar om projektet .....	11

# 1 Introduktion

Applikationen Storyteller utvecklades i kursen Software Engineering Project (DAT255) på Chalmers Tekniska Högskola. Utvecklingen skedde under april och maj månad 2015. Utvecklingsteamet Gråzonen bestod av Hannes Lagerroth, Willy Larsson, Sara Lingegård, Gustav Olsson, Johan Rebner och Erik Sällberg.

## 1.1 Projektets vision

*Storyteller är en applikation för individer med skrivarglädje som vill få utlopp för sin kreativitet och skriva berättelser i samarbete med andra. Den är till för de som har behov av ett stimulerande tidsfördriv där de samtidigt ges möjligheten att vara delaktiga i att anonymt skapa en berättelse och se produkten av det egna och gemensamma skapandet.*

*Kritiskt för projektet är att kopplingen mellan individer fungerar så att skrivandet kan göras av flertalet personer och att slutprodukten kan lagras och göras tillgänglig för skaparna. På marknaden finns idag ingen motsvarande produkt vilket medför att slutprodukten yttersta orderkvalificerare är att den grundläggande funktionaliteten fungerar. Det som begränsar projektet är att det styrs av fem sprints som alla är en vecka långa samtidigt som inget finansiellt kapital finns tillgängligt.*

Den grundläggande funktionalitet som krävs för att applikationen ska anses spegla visionen och fungera som både kunden och utvecklingsteamet önskar är:

- Det ska gå att skriva en berättelse tillsammans med andra individer
- Kunna vara anonym samtidigt som man skapar berättelser med andra
- Få tillgång till sina egna och andras berättelser och läsa dem i efterhand

## 2 Metod

Under metodavsnittet introduceras de olika arbetssätt och verktyg som gruppen nyttjade under projektet. En kortare förklaring till hur de använts, motivering varför de valdes och hur användandet av specifika verktyg kompletterade gruppens utvecklande av applikationen och inläringen av det använda arbetssättet.

### 2.1 Arbetsprocessen

Eftersom att det viktigaste läromålet för projektet var hur en större grupp kan arbeta med ett IT-projekt och tillsammans med en bra arbetsmetodik producera en fungerande applikation har arbetssättet fått stå i fokus under utvecklingen. Under denna del beskrivs de arbetssätt som nyttjats och utvärderats under processen.

#### 2.1.1 SCRUM

Gruppen hade inför uppstarten av projektet fått lära sig om SCRUM och tagit del av arbetssättet och de metoder som används. Därför bestämdes det redan från start att gruppen skulle testa att använda sig av arbetssättet och testa de olika metoder som erbjöds. Det var ingen av medlemmarna som tidigare varit i kontakt med arbetssättet eller testat någon form av agil utvecklingsprocess.

I processens inledande fas hade gruppen ett möte där user stories listades, dessa värderades efter arbetsbörda och prioritet, vilket utgjorde projektets backlog. Listan har i och med att definitionen av den slutgiltiga produkten förändrats också kontinuerligt reviderats under projektets gång. Detta både när den förväntade och verkliga arbetsbördan inte stämt eller omprioriteringar behövs göras. Gruppen har också insett att vissa av de ursprungliga idéerna som gick utanför den grundläggande funktionaliteten och var extrafunktionalitet för applikationen behövde rationaliseras bort. Det främst på grund av att tiden inte räckte till och att prioriteten inte varit hög. Backloggens viktigaste funktion har under processen varit att klargöra applikationens mål, vilket förhindrat att allt för mycket funktionalitet som inte varit central utvecklats.

Gruppen bestämde att alla skulle ses två gånger i veckan under totalt sex timmar och utöver det fick de som skulle arbeta tillsammans samordna hur de skulle träffas för att klara av sina uppgifter.

Det första mötet som hölls varje vecka var på måndagar klockan 8.00, den första timmen användes för att avsluta den tidigare veckans sprint och planera den kommande veckans sprint. Gruppen gick igenom den föregående sprinten och samtliga medlemmar fick berätta hur deras arbete gått, vad de lyckats och inte lyckats med samt vilka problem de stött på under den gångna sprinten. De problem som inte lösts under sprinten diskuterades och förslag till lösningar togs fram. Efter att den gångna sprinten gått igenom planerades den kommande sprinten. I denna inkluderades den feedback och kritik som gruppen fått av projektets handledare och product owner. Gruppen sammanställde vad som ansågs möjligt att klara av under kommande sprint

och delade ut ansvar över de olika delarna. På det första mötet utsågs en av medlemmarna till SCRUM-master, dock insåg gruppen efter någon sprint att den rollen fungerade bäst om den inte var fast utan tilläts rotera mellan personerna i gruppen. Detta för att öka känslan av inflytande och alla individers aktivitet på mötena.

Gruppen hade även ett gemensamt möte på onsdagar, detta fungerade även det som ett SCRUM-möte där medlemmarna fick berätta vad de gjort sedan sist, vad de planerade att göra tills kommande måndag och om det fanns något som hindrade dem. Detta var också ett möte som användes för att sammanställa eventuella frågor inför handledningen senare samma dag. Vid båda mötestillfällena fördes noggranna mötesprotokoll framförallt för att ha en bra mötesstruktur och underlätta processen att gå tillbaka och kolla. Det var också användbart för frånvarande medlemmar så att de snabbt skulle kunna sätta sig in i arbetet.

Vid båda mötestillfällena fanns också tid avsatt för att gruppen gemensamt skulle kunna sitta och arbeta med sina respektive uppgifter. Detta för att kunna hjälpa varandra med problem som uppstått och säkerställa att arbetet fungerade för samtliga medlemmar. Utöver denna tid så fick medlemmarna i gruppen själva planera sin tid och arbeta på med projektet.

Den största fördelen som gruppen identifierade med att använda SCRUM var att de redan efter första sprinten hade en applikation som kunde testas och byggas vidare på. Att åstadkomma inkrementella förbättringar och införa en regelbunden arbetsprocess istället för att söka efter en perfekt lösning från början gjorde det lättare att anpassa arbetet varje sprint. Det gav gruppen en tydlig prioritering och planering för varje vecka. Då gruppen anpassade användandet av SCRUM för att fungera med gruppens process har inga större nackdelar identifierats. En förändring som gjordes var att vi inte använde oss av korten som introducerades vid SCRUM introduktionen i kursen för att bestämma effort och prioritet utan istället diskuterade gruppen fram detta gemensamt.

### **2.1.2 Parprogrammering**

Parprogrammering var något som alla i gruppen använt sig av tidigare i andra programmeringsprojekt, vid andra kurser på Chalmers tekniska högskola, och som alla då hade upplevt som något bra. Detta för att koden betraktas av någon utöver den som kodar, vilket hjälper paret att undvika fel och kraftigt minskar tiden för felsökningar i koden. Att mer än en person har ansvar och kännedom för kodens alla delar är också något som underlättar det fortsatta arbetet, speciellt om en person av någon anledning är frånvarande. Det är också positivt att ha två synvinklar på ett problem innan försök görs för att lösa det. Med det som utgångspunkt bestämde gruppen i projektets början att parprogrammering skulle användas.

Gruppen har haft lite problem med frånvaro både på grund av sjukdom och resor. Detta har gjort att gruppen för att ta sig vidare i processen och utveckla applikationen som planerat behövt frångå den ursprungliga visionen om att enbart nyttja parprogrammering. För att göra detta på bästa sätt har gruppen gått igenom de olika uppgifterna och urskilt det som ansetts

enklast att utföra på egen hand och delat ut uppgifter baserat på möjlighet att använda parprogrammering.

Att gruppen i största möjliga mån när det fungerat använt sig av parprogrammering tros vara en av anledningarna till att projektet inte haft problem med allt för mycket buggar. Gruppen anser att nyttjandet av parprogrammering minskat tidsåtgången för projektet och gjort att önskvärt resultat varit möjligt att nå. Det är också något som alla medlemmar önskar använda sig av i kommande programmeringsprojekt och som gruppen skulle rekommendera till andra.

## **2.2 Disponerad tid**

Gruppens medlemmar spenderade ungefär 20 timmar i veckan på projektet. Fem sprints arbete, en veckas förberedelse för presentation och sista färdigställande av avslutande dokumentation resulterade sammantaget i att varje medlem lagt runt 120 timmar på projektet. Medlemmar från gruppen har dock varit sjuka samt att några varit ute och rest under projektets gång, vilket gjort att de under vissa veckor inte kunnat arbeta 20 timmar. Detta har dock kompensats av extra arbete innan och efter frånvaron. Tiden varje medlem lagt ned har också varierat mellan veckorna beroende av vilket ansvarsområde varje individ haft. Total nedlagd tid för projektet är således drygt 720 timmar, i den tiden är kodande, dokumentation, användande av arbetssättet SCRUM, presentation och reflektion över processen inkluderat.

Det som tog mest tid för gruppen var allt administrativt arbete med att hålla scrummöten och dokumentera processen. Inkluderat i detta är också den testning som utförts på applikationen och förberedelser inför och deltagande på presentationen av applikationen, totalt beräknas denna typ av arbete tagit 240 timmar. En ytterligare del som upptog mycket tid för gruppen var inlärningsprocesser där samtliga medlemmar i gruppen skulle lära sig alla programvaror som användes för framtagandet av applikationen. Detta uppskattas ha tagit 170 timmar. Att sedan använda den nya kunskapen för att ta fram den grundläggande funktionaliteten i applikationen beräknas för gruppen ha tagit 160 timmar. Den externa serverlösning som gruppen valde att ej använda, läs mer om det under problematik under utvecklingen, antas ha tagit 50 timmar att utveckla till det stadie där utvecklingen avslutades. Den andra serverlösningen med Parse som gruppen valde att använda sig av beräknas ha tagit 110 timmar att utveckla och använda i applikationen.

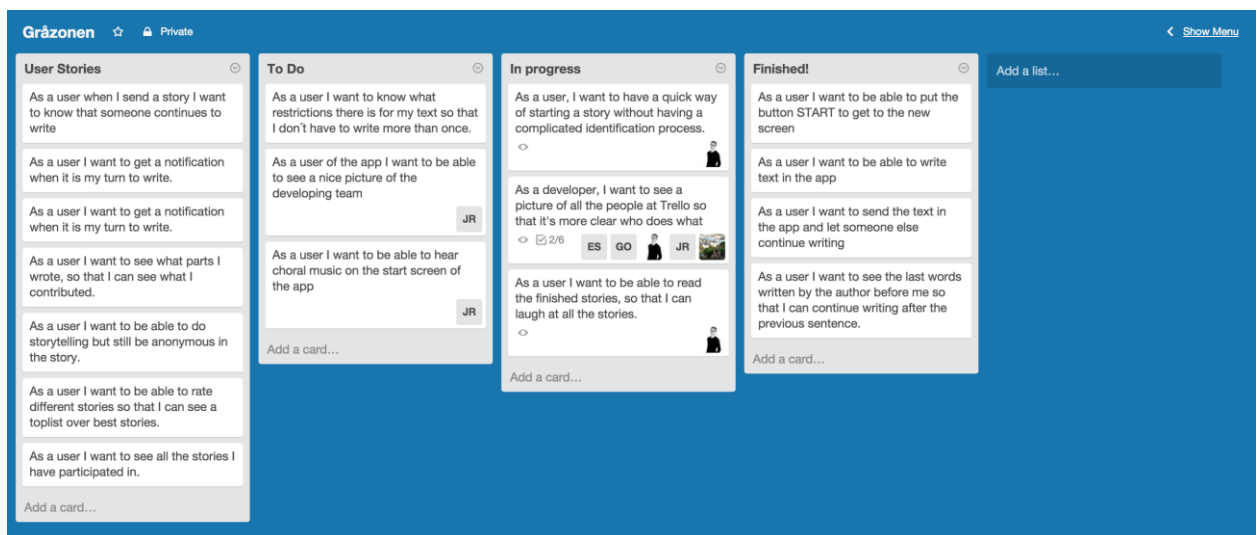
Inom gruppen fördelades tidigt ut ansvar för att alla medlemmar skulle känna sig delaktiga och kunna känna att de tillförde något till gruppen. Dessa områden byggdes på och förändrades allt efter som att projektet utvecklades och efter sprint fem hade gruppmedlemmarna följande ansvar:

- ❖ Gustav var ansvarig för den externa serverlösning som ej nyttjades av gruppen men som fortfarande blev en viktig del i dokumentationen och en lärdom från projektet.
- ❖ Hannes var ansvarig för gruppens användande av versionshanteraren GitHub och var efter halva projektet den enda i gruppen som fick lov att göra commits till projektets masterbranch.

- ❖ Sara var ansvarig för dokumentationen av processen under varje sprint och den slutgiltiga dokumentationen för projektet.
- ❖ Johan var ansvarig för applikationens övergripande funktionalitet och att det som skulle finnas med fungerade. Var också ansvarig för att koden var dokumenterad enligt Javadoc.
- ❖ Willy var ansvarig för den serverlösning som gruppen valde att använda sig av för projektet.
- ❖ Erik var ansvarig för att de buggar som uppkom löstes och dokumenterades.

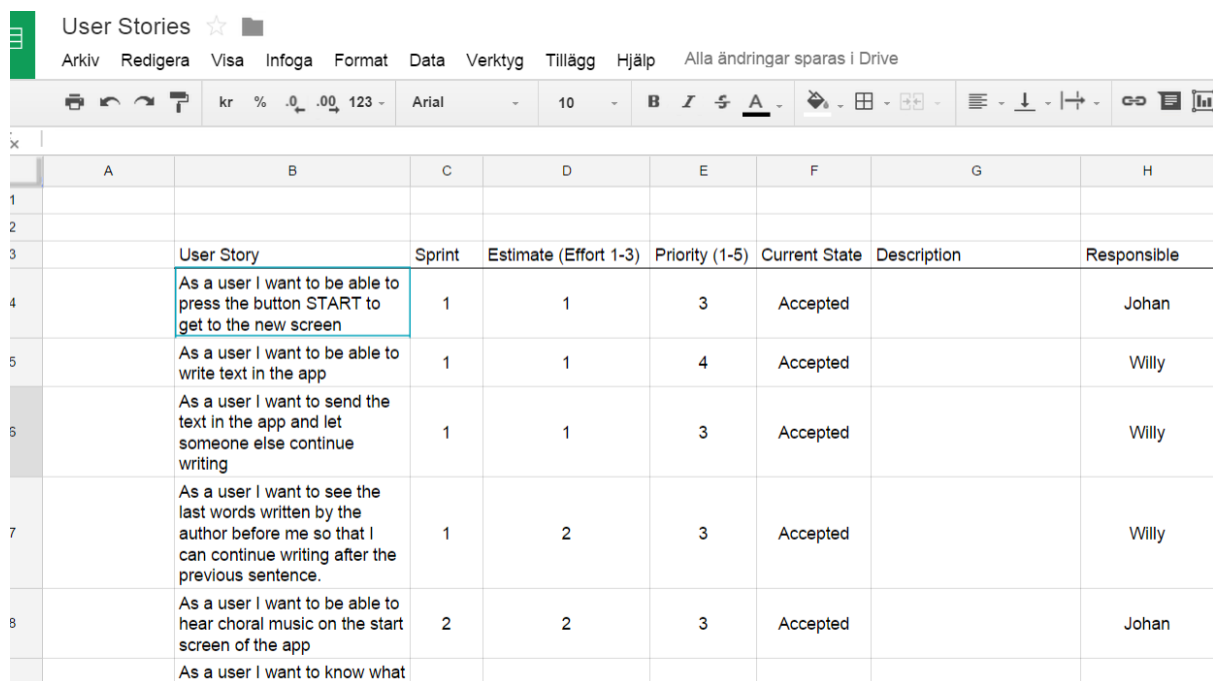
## 2.3 Trello och Google Spreadsheet

För att strukturera och arbeta med projektets user stories använde sig gruppen av två olika verktyg. Google Spreadsheet användes för att kunna kategorisera alla user stories och värdera arbetsbördan på respektive. För att sedan visualisera gruppens *SCRUM Board* användes Trello där ansvar kunde fördelas till bestämda personer samtidigt som det blev tydligt vilka user stories som var startade, avslutade och vilka deluppgifter som var uppfyllda hos vissa (figur 1).



Figur 1: SCRUMboardet i Trello i början av sprint 2

Anledningen till valet att använda två olika verktyg var för att ett värde sågs i båda lösningarna. Fördelen med Trello var att det gav en visuell överblick och kunde användas som ett verkligt SCRUM Board, vilket ansågs värdefullt för att dra lärdom om att arbeta med ett projekt enligt SCRUM. Verkettyget erbjöd även en möjlighet att enkelt dela upp varje user story i *tasks*, vilka kunde “checkas av” när de var klara, vilket sågs som en stor fördel för att erhålla total kontroll över hur arbetet fortlöpte. Det ena verktygets fördel var dock det andras nackdel och det som Trello inte kunde erbjuda var ett smidigt sätt att bedöma hur viktiga user stories var för projektet och hur stor arbetsbörda det beräknades ta att färdigställa dem. Detta var något som också ansågs vitalt att lära sig för att få en bättre förståelse för SCRUM som arbetssätt och efter det kunna utvärdera det. Figur 2 visar hur gruppen använde sig av Google spreadsheet för att komplettera användandet av Trello.



	A	B	C	D	E	F	G	H
1								
2								
3		User Story	Sprint	Estimate (Effort 1-3)	Priority (1-5)	Current State	Description	Responsible
4		As a user I want to be able to press the button START to get to the new screen	1	1	3	Accepted		Johan
5		As a user I want to be able to write text in the app	1	1	4	Accepted		Willy
6		As a user I want to send the text in the app and let someone else continue writing	1	1	3	Accepted		Willy
7		As a user I want to see the last words written by the author before me so that I can continue writing after the previous sentence.	1	2	3	Accepted		Willy
8		As a user I want to be able to hear choral music on the start screen of the app	2	2	3	Accepted		Johan
		As a user I want to know what						

Figur 2: Det kompletterande verktyget Google spreadsheet

Den största risken med att använda två verktyg är att ingen av dem används ordentligt av alla parter och att verktygen inte automatiskt kan synkroniseras med varandra. Detta var också något som blev ett problem i inledningen innan alla lärt sig hur de båda verktygen fungerade, men det gick bättre i slutet och för att förstå arbetsprocessen och värdera fördelar och nackdelar med den ansågs det ha varit ett bra alternativ att nyttja båda verktygen. Det är dock inte troligt att någon i gruppen skulle välja att nyttja dubbla verktyg i ett senare projekt med tanke på risken att inte ha uppdaterad och riktig dokumentation. Ytterligare en anledning till att två verktyg kunde användas var för att god kännedom fanns hos alla vad gällde Google spreadsheet samt att en av medlemmarna i gruppen tidigare testat Trello och kunde intyga att inlärnings tiden för verktyget inte var lång och att det inte var komplicerat att använda.

## 2.4 Emulatorn

För att testa de funktioner som ändrades i applikationen använde sig alla i gruppen av den emulator som finns att ladda ner med Android studio. Problematiken med denna är att den ofta tar tid att ladda, vilket kan medföra att testandet av den nya funktionaliteten kan ta omotiverat lång tid. Gruppen fick förslag om att använda en alternativ emulator men detta var det dock ingen som gjorde. Emulatorn var för alla i gruppen tung att ladda men det skiljde sig väldigt mycket mellan datorer med olika prestanda. För några av medlemmarna i gruppen, de med "sämre" datorer, var det nödvändigt att använda en äldre telefonmodell i emulatorn för att kunna ladda applikationen inom en rimlig tid, medan det för andra gruppmedlemmar inte spelade någon roll vilken telefonmodell som användes.



## 3 Diskussion

Under detta delkapitel diskuteras de problem som gruppen stötte på under projektets gång och hur dessa löstes. Efter det diskuteras också hur arbetet i grupp fungerade, vad som gick bra och vad som var problematiskt. Kapitlet avslutas med en kortare redogörelse för vad gruppen i efterhand skulle valt att göra annorlunda för att erhålla ett bättre resultat.

### 3.1 Problematik under utvecklingen

I följande del förklaras de problem som gruppen under processen stött på. Det förklaras hur de uppstod, hur gruppen valde att lösa problemet och vilka lärdomar som kunde dras av problemet.

#### 3.1.1 Branches som inte gick att merge

Något som gruppen insåg i slutet av sprint ett och början av sprint två var att de hade börjat utveckla olika funktionalitet i olika branches som inte längre var kompatibla, vilket medförde att de ej gick att merge. Detta var något som gruppen fick lösa manuellt men gruppen insåg då att ett gemensamt arbetssätt med olika branches var nödvändigt. Efter incidenten fick gruppen som tips att i framtiden enbart använda branches för att utveckla mindre funktionalitet och så fort detta var gjort merge samman förändringen med projektets developer-branch. När utvecklingen på en viss branch var klar skulle också den tas bort för att undvika att inaktuella branches användes. Efter detta inledande misstag lärde sig gruppen snabbt att hantera och strukturera upp de branches som fanns på projektets GitHub. En av gruppens medlemmar valdes då till *Gitmaster* och var efter det ansvarig över strukturen och användandet av versionshanteraren. Personen var också den enda i gruppen som tilläts att göra commits från branchen develop till master.

Gruppen insåg längre fram i projektet att det var tur att problemet uppstod i en tidig fas då det fortfarande gick förhållandevis enkelt att sammanfoga de olika varianterna. Hade problemet uppstått senare kunde det gett upphov till ett mer svårlöst problem, där koden i de olika branscherna eventuellt var så olika att de inte längre var möjliga att sammanfoga utan att skrivas om. I och med detta insåg gruppen också hur viktigt det var att lära sig att arbeta effektivt med versionshanteringen. I framtida projekt skulle gruppen använda sig av en *Gitmaster* eller liknande då gruppen ansåg att detta var en viktig del i att det gick så smidigt i projektet. Det behöver inte vara till just Github heller utan detta går att applicera på andra verktyg eller program där gruppmedlemmarna inte är familjära redan från start. Gruppen anser det alltså viktigt att ha en ansvarig som snabbt lär sig det för gruppen nya verktyget och sedan sätter regler för hur det ska användas samt lär ut detta till resten.

#### 3.1.2 Dubbla serverlösningar

Efter det att andra sprinten var avslutad hade gruppen stött på ett problem som främst berodde på misskommunikation. En i gruppen hade fått som uppdrag att börja kolla på en serverlösning, dock hade denna person under sprint ett blivit sjuk varför två andra i gruppen började kolla på en alternativ lösning till problemet. Resultatet var dock att vi efter två sprints stod med två olika lösningar. Den ena lösningen var genom Parse och den andra var genom en extern server

med MySQL som grund till databasen. I den första lösningen kommunicerar Android med Parse backend via funktionalitet som hämtades från Parse hemsidan. Android kommunicerade i den andra lösningen med databasen genom PHP. De båda lösningarna fungerade ungefär lika bra men det fanns vissa viktiga skillnader.

Efter ett möte med handledare fastslogs att de två personer som hade bäst koll på respektive server skulle sätta sig ner och gå igenom varandras lösningar vartefter de skulle lägga fram ett gemensamt förslag till gruppen hur arbetet skulle fortsätta. De framtagna förslaget blev att använda Parse-lösningen, detta för att den till högre grad var inkluderad med den befintliga funktionaliteten i applikationen vilket möjliggjorde för gruppen att studera vad som hände i applikationen. Det betydde att gruppen kunde se vad som fungerade i applikationen och direkt kunde testa och debugga det som inte fungerar som det ska.

Den riskanalys som gjordes innan beslutet var att det med PHP-lösningen fanns en möjlighet att applikationen aldrig skulle fungera medan med Parse-lösningen såg gruppen redan att en del av den önskade funktionaliteten var på plats och fungerade. Om gruppen hade velat utveckla en applikation för att sälja på marknaden så hade troligen den andra lösningen med PHP valts, detta för att det förmodligen skulle generera ett bättre resultat och en mer givande funktionalitet. Men i och med att tanken med kursen är att medlemmarna ska lära sig att arbeta i projekt ansågs inte det argumentet som vinnande. Det var också ur en lärande- och kunskapssynpunkt mer värdefullt att fortsätta med Parse eftersom att fler medlemmar ur gruppen arbetat med lösningen och det fanns därav fler som hade någorlunda koll på lösningen.

Gruppen gjorde därmed en bedömning att det med en thin client blir svårare att hitta information och lösningar på problem som kan uppstå, vilket kunde resultera i att om gruppen körde fast i sitt arbete kunde applikationen riskeras att aldrig bli färdig. Däremot med en rich client fanns det en större möjlighet att finna rätt dokumentation om problemet då det finns mycket mer dokumenterat och stor tillgång på information samt guider som gruppen kan följa, vilket skulle underlätta fortsatt problemlösning.

### **3.2 Reflektioner om arbetet i grupp**

Att arbeta sex personer i ett mjukvaruprojekt var något nytt för hela gruppen. Ingen av medlemmarna hade varken gjort ett så omfattande projekt som applikationen innebar eller gjort det i en så stor grupp som krävdes. Tidigare erfarenheter begränsas till programmeringsuppgifter givna i kurser på Chalmers tekniska högskola, vilka lösts med parprogrammering, kontrasten som därför skapades vid projektet var påtaglig. Däremot hade gruppen stor erfarenhet av att arbeta med andra projekt i större grupper, vilket antas ha underlättat processen och varit till stor fördel jämfört med om det också varit helt nytt.

Det var under projektet svårt att samla gruppen som helhet speciellt då det var medlemmar från gruppen som var ute och reste under stora delar av projektet samtidigt som sjukdom drabbade två personer under utvecklingsprocessen. Att inte kunna ses som helgrupp försvårade både uppgiftsfördelning och kommunikation. Bortfallet av individer på veckomötena försökte

hanteras genom väl dokumenterade mötesprotokoll som gjordes tillgängliga, genom Google drive, samt direkt kommunikation via Facebook. Svårigheten att ses som helgrupp är ändå något som måste lyftas fram då de anses ha haft en stor negativ effekt på det slutgiltiga resultatet. Detta då mycket tid har fått läggas på att uppdatera individer om projektet och vad som bör göras näst. Det har också varit problematiskt när någon av de gruppmedlemmarna som varit borta besuttit nödvändig information för att fortsätta med projektet eller gruppen inte haft full vetskap av vad som blivit gjort och vad som är kvar att göra.

Det var också svårt att få alla i gruppens bidrag till den slutgiltiga produkten att bli lika stort. Detta berodde främst på att intresset var högre hos vissa i gruppen, vilket ledde till att de gärna tog på sig fler uppgifter än de blev tilldelade i sprinten. Detta ledde således till att attribut som kanske inte var centrala för applikationen utvecklades vid sidan av framtagningen av den grundläggande funktionaliteten. Samtidigt var det detta brinnande intresse för att lära som tog projektet framåt och resulterade i det goda resultat som kan uppvisas.

Gruppens bristande erfarenhet både av att arbeta med ett stort IT-projekt i grupp och de programvaror som behövdes gav upphov till de största missberäkningarna gällande tidsåtgången. Projektet började med mycket inläring både av program och att få all programvara att fungera för hela gruppen. Det var också en inlärningsprocess att börja arbeta enligt arbetssättet SCRUM, något som gruppen tog förgivet att de skulle klara av från början men som visade sig behöva en viss inläringstid. Det tog några sprints innan gruppen var inne i tankesättet och hade förstått hur mycket som kunde planeras in och vad som kunde förväntas av gruppen under en sprint. Att projektet höll på under en så begränsad tid medförde att gruppen kunde arbeta effektivt och väl med tillvägagångssättet först när projektet skulle avslutas och färdigställas. Gruppen antas däremot ha lärt sig mycket och vara mycket bättre förberedda inför ett liknande projekt i framtiden. Arbetssättet är också något som många tror sig delvis kunna överföra i projekt som ligger helt utanför mjukvaruutveckling. Framförallt anser gruppen att användandet av *SCRUM-möten* kan översättas till andra projekt för att ge samtliga medlemmar i ett projekt en bättre överblick över vad de själva och andra har gjort inför det aktuella mötet och ska göra tills nästa möte.

Ansvarsfördelningen anses ha varit bra för gruppen då alla ansvar ofta blir ingens ansvar. Det som däremot varit negativt är att kunskapsinhämtningen inom olika områden koncentrerats till en eller ett fåtal individer. Detta i sin tur har gjort att kunskapsdelning har tagit en stor del och att kunskapsspridningen inom gruppen inte blivit optimal. Däremot har alla i gruppen på ett bra sätt delat med sig av sin kunskap och gruppen löste inledningsvis många av de problem, vilka uppstod vid inläringen av den nya programvaran, gemensamt och lärde av varandra.

Sammantaget anser gruppen ändå att arbetet fungerat bra med tanke på de förutsättningar som funnits. Medlemmarna har varit engagerade och velat ta på sig uppgifter att utföra, det personer har sagt att de ska göra har också blivit gjort även om det stundtals skett med fördröjning på grund av sjukdom. De problem som gruppen har ställts inför har lösts på bästa möjliga vis och gruppen anses ha gjort ordentliga riskbedömningar innan beslut fattats. Den tydliga ansvarsfördelning har gjort att alla medlemmar har känt ett ansvar för slutprodukten och aktivt

arbetat för att slutföra de uppgifter som tilldelats, det anses alltså ha effektiviserat processen snarare än lett till motstridiga lösningar i majoriteten av fallen.

### **3.3 Vad vi skulle gjort annorlunda nästa gång**

Om gruppen distanserar sig till projektet så anser alla att arbetet gick förvånansvärt bra men att det finns saker som borde gjorts annorlunda för att ett optimalt resultat skulle uppnås.

Den största förbättringspotentialen gruppen identifierat för projektet var att mer gemensam tid borde avsatts, då hade gruppen kunnat säkerställa att delar som krävde större delar av gruppen för att bli bra färdigställdes tidigare. Det hade också gjorts att de problem som uppstod eventuellt kunnat behandlas tidigare och inte löstes först vid nästa gemensamma möte. Däremot fungerade Facebook som en bra kontaktväg för gruppen för att försöka substituera möjligheterna att ha möten mer frekvent. Det fungerade dock inte lika bra som gemensamt arbete hade gjort. Något vi hade gjort annorlunda i framtiden är att se till att om en individ ska vara borta under en period så måste detta planeras noga så att arbetet kan fortsätta framgångsrikt utan denne. Detta genom att se till att inte bara en individ har en viss nödvändig kunskap utan att se till att denna kunskap delas hela tiden men framförallt att det säkerställs att detta sker när någon skall vara borta.

En annan förändring som gruppen anser borde gjorts är att fokus borde legat mer på den grundläggande funktionaliteten i applikationen. Nu koncentrerades mycket energi på att försöka färdigställa och utveckla roliga komplement till applikationen, vilka i slutändan ändå blev bortrationaliserade då det inte tillförde något direkt värde till slutprodukten. Gruppen borde alltså ha fokuserat mer på den funktionalitet som krävdes för att applikationen skulle fungera, det hade gjorts att denna funktionalitet kunnat färdigställas tidigare i processen. Information om detta var något som gruppen fick vid handledningstillfällena, gruppen var dock så ivriga att få testa alla sina nya idéer och extra funktioner i applikationen att goda råd ibland blev lite åsidosatta.

Kopplat till det tidigare problemet borde gruppen gemensamt identifiera vad de viktigaste uppgifterna är och därigenom definiera vad som skulle vara färdigställt och fungera när. Detta är viktigt vid olika deadlines för att visa att gruppen kan leverera vad som utlovas och i tid till kunden vilket också är en viktig del inom agil utveckling. Det hade troligtvis också kunnat minska ner den individuella drivkraften att förverkliga sina egna idéer och istället drivit projektet framåt i rätt riktning. Nu var många beslut baserade på individers vilja att ta på sig olika uppgifter och en känsla av vad de skulle klara av och tycka var roligt.

Hade gruppen då börjat definiera väl fungerande delmål som möjliggjorde för gruppen att veta hur de låg till hade medvetenheten om när tiden började bli knapp för att hinna klart i tid blivit tydligare. Dessa mål borde definierats i enighet med SMART goals, alltså mål som är specifika, mätbara, accepterade, realistiska och tidsatta, för att anses vara realistiska att uppnå.

## 4 Sammanfattade tankar om projektet

Avslutningsvis så är gruppen som helhet nöjda med det resultat som de under den relativt begränsade tiden lyckades åstadkomma. Det känns också som om arbetet med det för gruppen nya arbetssättet SCRUM, givit alla medlemmar nya verktyg för att strukturera upp ett projekt. Även om alla delar kanske inte kommer inkluderas i framtida projekt känns det ändå värdefullt att fått möjligheten att testa och utvärdera dessa. Det känns också som om de problem som gruppen under projektets gång stötte på var värdefulla för att erhålla ett komplett lärande även om det för stunden givetvis var betungande. Arbetet i grupp har fungerat förvånansvärt bra och de meningsskiljaktigheter som funnits i gruppen har tidigt lyfts och behandlats på ett bra sätt utan att påverka det fortsatta arbetet.

De lärdomar som gruppen tar med sig utöver kunskapen om den agila arbetsprocessen är också hur en grupp kan arbeta för att skapa en applikation. De grundläggande kunskaper om de verktyg som använts under utvecklingen är också betydligt högre nu än vad det var inledningsvis. De verktyg som gruppen har lärt sig mest om anses vara AndroidStudio och GitHub. Även om dessa kanske inte kommer användas flitigt av gruppen senare anses det vara av stor nytta att förstå hur en applikation är uppbyggd samt hur en kan använda sig av ett versionshanteringsprogram. Gruppenmedlemmarna har även fått mersmak när det gäller att bygga appar vilket gör att det säkerligen kommer att produceras fler appar antingen med samma gruppkonstellation eller med delar av den.