

Reflection Report

Group: TruckDriversUnitedz

Product: Svapp

Members: Lina Blomkvist, Josefin Ulfenborg, Alexander Nordgren, Tove Ekman, Emilia Vestlund, Razmus Strandell

1. Application of scrum

1.1 Roles among team members

Förra gången vi gjorde vårt arbete tillsammans så hade vi en platt hierarki. Vi hade ingen som bestämde mer än någon annan och alla hade lika mycket att säga till om och alla kände sig bekväma med detta. Lagarbetet förra perioden var det som vi var mest nöjda med. Våra möten var alltid glada och alltid rättvisa och om någon tyckte det var jobbigt hjälpte vi alla alltid till på grund av vårt första möte om det sociala kontraktet. Det dåliga med detta var dock att det inte blev lika mycket gjort för ingen vågade tvinga på och bestämma att de andra skulle arbeta. Vi tror detta kan vara en av anledningarna till att det inte blev så mycket gjort och att vi inte klarade projektet första gången.

Till andra försöket av projektet drog därför ett par i gruppen mer i trådarna och bestämde att vi skulle ha möte och inte tänkte så mycket på allas behov av sovmorgon och diverse annat. Detta ledde till att alla i gruppen kom tidigt till mötena och vi fick inte en halvtimme av att vänta på gruppmedlemmar. Detta var bra, däremot ledde det till spänningar i gruppen när vissa tog på sig ansvar de inte hade eftersom vi inte bestämde vem som var ledaren.

Därför tror vi att om vi skulle göra om projektet ytterligare en gång, att gruppen tillsammans verkligen bestämmer vem som är ledare och hen bestämmer när och var man träffas och att man lyssnar på det. Vi tror att detta skulle leda till att det blir mycket mer gjort. Gruppens stämning kanske inte skulle bli lika bra med en makthierarki men eftersom de flesta yrken idag har chefer som bestämmer arbetstider och liknande är detta ett vinnande koncept som vi också borde ha i våra projektarbeten.

1.2 Teamwork

Förra gången försökte vi alla respektera varandras åsikter och ha ett demokratiskt arbetssätt där allas åsikter var lika mycket värda. Detta ledde till en bra gruppstämning, men på bekostnad av arbetseffektiviteten. Den här gången blev en gruppmedlem självutnämnd projektledare och vi fick plötsligt en "chef". Vår projektledare skrev därefter egna user cases och vid första sprintmötet gick vi snabbt igenom dessa och det verkade då som majoriteten av gruppen var nöjda med dem.

När sprinten sedan satte igång så märktes det att det inte var särskilt mycket teamwork alls; folk arbetade mest på sina delar själva med väldigt lite kommunikation med

resten av gruppen. Vissa personer gick även in och ändrade andras implementationer utan att kolla av med varandra då de inte riktigt var nöjda.

Ett problem som uppdagades av detta är att alla medlemmar verkar ha haft olika syn på hur varje individuellt user case skulle lösas. Denna skiljaktighet beror på att vi hade väldigt stora generellt formulerade user cases samt att vi inte diskuterade särskilt mycket om detta vid vårt sprintmöte. Det vi borde ha gjort var att istället låta varje medlem beskriva och diskutera sin syn på respektive user case så vi hade kunnat se till att vi alla hade samma vision för slutresultatet. Vi borde dessutom ha skrivit ned vad vi kommit överens om, vilket i sin tur hade stärkt samarbetet oss emellan då man jobbar bättre ihop om man jobbar mot samma vision. Även att bryta ner user casen i mindre, mer konkreta, delar skulle ha bidragit till bättre samarbete, vilket är något vi tar med oss från detta projekt.

1.3 Social Contract

I början av kursen skapade vi ett s.k. Social Contract. Vi tog då tiden att varsamt bestämma ett antal punkter som alla kunde stå bakom.

Denna gången bestämde vi oss dock att inte göra ett nytt sådant kontrakt utan att helt enkelt hålla oss till det förra eftersom det på den sociala fronten hade fungerat mycket bra.

Däremot så bestämde vi att en kontraktpunkt som vi i förra kursomgången hade lagt mycket vikt på inte längre gällde. Denna var regeln att man inte fick ändra någon annans kod utan att diskutera detta med personen. Detta beslut var inte riktigt alla bakom vilket senare resulterade i viss osämja bland gruppens medlemmar då vissa började göra om i andras kod utan att de själva visste om det eller hade gett tillåtelse till detta.

Vi har av detta lärt oss att alla punkter i det sociala kontraktet behöver konsensus för att fungera som bäst som det då gjorde i första projektomgången.

1.4 – Used practices

1.4.1 – Pair programming

Under första delen av kursen avsåg vi tid att sitta och programmera ihop åtminstone en gång i veckan, vilket inte riktigt blev av så ofta som vi hade planerat. Några gånger under kursens gång lyckades vi dock få tid till det, som exempelvis när vi satt i två par och gjorde databasen. Den senaste sprinten så blev det dock ännu mer avskärmat; folk åkte iväg och liknande under julen vilket resulterade i att gruppens medlemmar mest suttit och jobbat på egen hand. Två medlemmar som hade tid satte sig dock tillsammans vid ett par tillfällen för att programmera i par, vilket gjorde deras delar mer enhetliga och synkade vilket gav högre kvalité på deras implementationer till skillnad från hos de som jobbat helt ensamma. Detta till trots så var det ändå tillfällen då de satt tillsammans men ändå "jobbade enskilt", vilket var ett slöseri på det gyllene tillfället som ges av att ha tillgång till pair programming. Vad detta ändå visar är att det gynnade oss positivt att programmera tillsammans, så att inte ha utnyttjat detta mer än vi gjort är helt klart en förlust för alla inblandade samt för vår applikation i sig.

1.4.2 – Daily scrum

Första delen av kursen försökte vi först ha daily scrum meetings genom att använda oss av ett google-dokument. Detta användes i princip aldrig vilket fick oss att slopa det och att istället uppdatera varandra via vår gruppchatt. Denna gång har vi tyvärr inte haft någon form av daily scrum alls utan vi har helt enkelt bara fokuserat på oss själva och våra egna user cases utan att ha någon direkt koll på vad som händer runt omkring.

1.5 – Time distribution

När vi delade upp arbetet den här gången försökte vi att titta på ungefär hur mycket som skulle vara rimligt för varje person att bidra med. Då vi hade fått i uppdrag att göra ungefär fem user cases såg vi det som lämpligt att varje person gjorde ungefär ett user case var. När vi delade upp arbetet såg vi att några av våra tänkta user cases var för stora och behövde delas upp i flera mindre, men vi ansåg ändå inte att det skulle bli för mycket arbete för någon av oss. Även om vi har fler user cases än planerat, så är det bara fem som har prioritet 1.

Då de olika personerna i gruppen har haft väldigt olika villkor under arbetsperioden har vi valt att vara väldigt flexibla med hur var och en har jobbat. Vissa hade mycket att göra med tentor innan jul, medan andra var mer upptagna med arbete och annat under jul och nyår. Vi försökte att dela upp arbetet i relativt lika delar, men de som kände att de hade möjlighet att ta på sig mer arbete gjorde det frivilligt. Vissa som fick mindre user cases på grund av att de hade tidsbrist i början av projektet har försökt att kompensera det med att arbeta vidare med att fixa till småsaker i koden under perioden mellan att våra user cases skulle vara klara, dvs den 26 december, och tiden för projektets slut.

Ett stort problem som uppstod på grund av att vi hade så pass olika tider vi kunde arbeta med projektet på var att vi valde att ha en lång sprint så att alla skulle kunna delta i projektet under den tid då de hade möjlighet. Den långa sprinten ledde till en lång rad problem, som dålig kommunikation, irritation på grund av detta och på grund av att vi delade ut olika stora user cases till olika personer. Vi märkte att även om de som tog på sig större arbetsbörda gjorde det frivilligt, så är det ändå olämpligt att ha en obalanserad arbetsfördelning, då det lätt leder till spänningar och missnöje inom gruppen.

Ett bättre sätt att hantera problemet med olika arbetstider hade varit om vi trots allt hade haft veckolånga sprints som tidigare, men att vi vid projektstarten kom överens om under vilka sprints var och en skulle jobba. Då hade vi kunnat ha fler möten och därmed bättre kommunikation, diskutera mer vad som hade hänt under varje sprint och vad som skulle göras under nästa, samt se till att all var med under lika många sprints så att arbetet blev uppdelat på ett mer rättvist sätt. Vi hade därmed kunnat slippa en stor del av de spänningar som uppstod under projektets gång.

Trots problemen som uppstod på grund av olika arbetstider och en för lång sprint, så kan vi säga att vi tycker att vi egentligen har haft tillräckligt med tid för att hinna med våra user cases. Flera av oss har känt att de egentligen skulle ha haft tid att arbeta mer med projektet och göra sina user cases mer noggrant, men att de valt att prioritera annat. Vad vi kan se av detta är att vi borde ha varit mer tydliga med vad ett user case innebär och hur det ska se ut när det är färdigt. Det hade kunnat göras genom mer diskussion kring vilka förväntningar som finns på ett färdigställt arbete och genom att spalta upp user cases i mindre och mer väldefinierade delar.

1.6 Effort, Velocity and Task Breakdown

I höstas, när kursen började, så tog vi oss tid till att ha ett långt sprintmöte där vi bestämde en velocity samt delade ut siffror till de olika user casen. Detta ändrades sedan under de första veckorna då vi märkte hur både velocityn och user case siffrorna hade blivit felevaluerade. Under vårt extrainsatta sprintmöte nu i vintras bestämde vi oss för att ännu en gång ägna tid åt att bestämma en ny velocity och dela ut siffror på nytt eftersom vår nya sprint skulle komma att vara mycket längre än de vi tidigare haft. Vi behöll däremot skalan som vi använt oss av under första delen av kursen då vi kände att den till kursavslutet hade varit tillräckligt bra utvecklad.

Denna sprinten hade vi egentligen fått i uppdrag att göra fem user cases. Däremot tyckte flera medlemmar att vi kunde göra vår applikation ännu bättre och satte då till ännu fler user cases. Vi kände också att det, om user casen var allt för få, inte skulle kännas som ett seriöst arbete, vilket hade kunnat leda till sämre arbetsmoral. User casen blev dock indelade efter prioritet från 1 (ska hinnas) till 3 (om tid finns) där antalet user cases som har en etta är exakt fem, antalet vi blev tillsagda att göra.

Då vi, som sagt, nu hade en mycket längre sprint än tidigare sprintar, satte vi nu en betydligt högre velocity än vi haft förut. Att vi höjde velocityn med så pass mycket som vi gjorde berodde delvis på ovannämnda längd på sprint men också på att vi hade känt att vi programmeringsmässigt hade kunnat göra mer på den avsatta tiden under höstterminen.

Sättet vi sedan delade ut siffror till user casen bestämde vi oss dock för att behålla från tidigare sprintar då vi lärt att man inte ska ändra både velocity och use case siffror på samma gång. Om vi hade gjort det hade vi kanske istället höjt både siffror och velocity och då gjort precis samma mängd jobb som i föregående sprintar.

Vi märkte dock i sinom tid att siffrorna än en gång hade delats ut obalanserat och vi hade delvis samma problem som under kursen, nämligen att vissa fick väldigt mycket att göra och vissa väldigt lite. Detta skapade en viss irritation bland gruppens medlemmar och vissa bestämde sig då för att göra minimalt jobb på större user cases. Detta skapade en ännu större obalans jämfört med dem som inte bara gjorde klart funktionerna utan också såg till att deras user case passade in i resten av appen på ett konsekvent sätt.

Lösningar på detta problemet hade kunnat vara att man antingen hade tagit ännu längre tid på sig att varsamt dela ut siffror eller att de som känt obalansen verkligen gått fram och sagt till att de inte går med på att göra så stora user cases när andra fick så lite. Då hade man kanske kunnat snacka ihop sig om att överlåta vissa delar av stora user cases till de medlemmar som redan hade klarat av sina mindre och på så sätt uppnått en bättre jobbharmoni.

Vår task breakdown har också ändrats från hur den såg första omgången vi hade sprintas. En skillnad från hur vi gjorde under kursen är att istället för att skapa väldigt små user cases utifrån större tasks skapades nu större user cases som då också innehöll ungefär en hel task vardera. Detta på grund av att vi hade en längre sprint istället för flera mindre. En förändring som då uppkom var att user casen blev mycket diffusare då de var väldigt generaliserade, vilket bidrog till den tidigare nämnda oklarheten av vad det faktiskt innebar att göra klart ett user case. Generaliseringen gjorde det troligtvis också svårare att dela ut balanserade siffror till user casen och detta ledde, som ovan nämnt, till en mindre rättvis arbetsbelastning mellan medlemmar. Det uppkom dock också positiva förändringar

med de större user casen. Dessa var att det blev större möjlighet att enskilt låta kreativiteten flöda och att gruppmedlemmarna fick möjligheten att känna sig mer ansvariga över sina områden, d.v.s. sina större tasks.

En annan förändring i hur vår task breakdown såg ut var att vi, istället för att på första sprintmötet bestämma vilka nya funktioner som skulle införas och hur de skulle delas upp, helt och hållet använde oss av mer eller mindre färdiguppdelade user case-förslag. Dessa, skapade av en enskild gruppmedlem, blev efter en mindre diskussion godkända av majoriteten av gruppmedlemmarna även om vissa ansåg att det hade varit bättre att göra egna från början. Att det endast var en person som hade gjort user casen visade sig dock inte vara den bästa idén, då user casen som tidigare nämnt blev för generaliserade och oklara, samt inte heller tog hänsyn till de andra medlemmarnas synpunkter på hur vår applikation skulle utvecklas vidare. Att andras villkor och åsikter inte togs upp kan också ha varit en bidragande effekt till tidigare ambitionsskillnader inom gruppen.

Lösningar som troligtvis skulle kunnat ha löst våra problem med sprintens task breakdown är att vi först och främst inte borde tagit en medlems enskilda förslag på user cases. Det är självklart att en ensam person inte resonerar lika bra som en hel grupp då det gäller att skapa user cases. Hade hela gruppen diskuterat hade vi säkert kommit fram till en mer klar bild över vilka funktioner gruppens medlemmar, som kollektiv, skulle vilja ha och med det möjligtvis hjälp att höja ambitionsnivån i hela gruppen. Genom att diskutera i grupp hade vi säkert också kommit fram till att user casen var för stora. Vi hade säkert då bestämt oss för att skala ner dem och på så vis hade medlemmarna haft lättare att veta den exakta betydelse av att göra klart ett user case. Då hade det med säkerhet blivit mycket enklare att dela ut korrekta siffror till vardera user case och hade då kunnat bli ännu mer balanserad mellan medlemmar.

Som sammanfattning har det känts som att vår task breakdown nu gentemot hur det tidigare sett ut blivit sämre, detta troligtvis på grund av gruppens beslut att ta förslags-user-cases från en enskild person istället för att tillsammans arbeta fram egna.

2. Reflection on the daily scrum

Första gången vi gjorde projektet hade vi en dagbok där vi skulle ha skrivit ner allt vi hade gjort under dagen. Vi använde den dock inte till någonting, vilket vi borde ha gjort. Vi borde ha kollat på daily scrums på våra veckomöten och sett om vi kunde effektivisera någonting och om vi verkligen gjorde så mycket per dag som vi borde ha gjort.

Under den nya sprinten hade vi många andra kurser och jullov samtidigt så vi kunde inte jobba varje dag. Därför tog vi beslutet att inte skriva dagbok varje dag utan bara berätta i gruppchatten de dagarna vi jobbade. Vi lade nämligen mer tid dagarna vi arbetade istället för att göra lite varje dag. Detta sätt att rapportera kändes bättre, eftersom det är inte så intressant att kolla på en daily scrum för de dagar som inte innehåller arbete med projektet utan istället står tex "idag jobbade jag med matematiken istället". Vår metod istället för daily scrum-metod kändes mer värd eftersom vi bara såg arbetet som hade gjorts. Det var dock svårt att hålla koll om man inte kollade igenom chatten ordentligt eftersom det flöt samman med andra konversationer. Vi inser att vi har haft dålig konversation genom projektet och detta har lett att det blivit osämja eftersom man inte vet hur olika personer har arbetat och hur mycket tid de lagt ner.

Om vi skulle göra om projektet i framtiden hade vi nog använt oss av daily scrums,

eftersom det visar ganska tydligt hur mycket tid en person lägger ner och de andra medlemmarna då kan skärpa sig om de inte lagt ner lika mycket tid. Detta tror vi skulle förebygga att olika personer arbetar olika mycket, i alla fall om hela gruppen reflekterar tillsammans över alla medlemmars dagbok.

3. Reflection on sprint retrospectives

Innan kompletteringen hade vi scrum-möten en gång i veckan, vilket verkade fungera rätt så bra men om man ser tillbaka på det hade vi kunnat göra det så mycket bättre. Vi kände att vi inte gjorde fullt så mycket som vi kunnat genomföra, men att vi ändå fått tilldelat tillräckligt många energipoäng baserat på vår velocity. Antingen så borde vi haft kortare sprints eller lagt ner ännu mer tid än vad vi gjorde på att fördela om energipoängen, så vi hunnit med fler user cases.

Den här gången ändrade vi vårt arbetssätt på så vis att vi använde oss av färre men längre sprints. Vi började med ett övergripande möte om vilka user cases vi skulle implementera, vilka som skulle göra dem, samt vilken del av rapporten varje person skulle stå för. Som tidigare nämnts använde vi oss fortfarande av samma sifferskala. Skälet till färre sprints ligger till grund för att vi haft en kortare tidsperiod på att implementera våra uppgifter, samt att det varit jullov och att folk varit bortresta och inte alltid i Göteborg. Vi försökte planera våra sprints utifrån det, där vår första sprint var att bli klara med kodimplementationerna.

Den andra sprinten var att bli klar med rapporten, och den sista då vi möts en sista gång och ser över detaljerna och fixar det som behöver justeras.

Det här sättet att arbeta med sprints är inte optimalt, då vi får väldigt stora uppgifter att utföra och ingen tid däremellan att ses eller diskutera kring appen. Det vi hade behövt göra var att ha fler och kortare sprintar, samt att uppgifterna vi blev tilldelade var mindre. Problemet med att vi varit på olika håll hade kunnat lösas med hjälp av att ha Skype-möten eller liknande. Hade vi fått mer diskussion under vintern hade de personer som ansvarat för ett visst område själv kunnat justera på det området, istället för att andra medlemmar i gruppen gör det på fri hand.

Generellt sett, både under hösten och vintern, har vi haft väldigt få och långa möten vilket orsakar att många blir trötta och ofokuserade. Eftersom det var många gånger vi fick vänta på några av medlemmarna hade det kanske varit bättre att ha fler och kortare möten, samt att några av dem hade kunnat vara via Skype. Då kan de flesta medverka och man riskerar inte längre att vissa personer blir sena.

4. Reflection on the sprint reviews

Under förra projektets gång hade vi kontinuerlig kontakt med några handledare från masterprogrammet Interaction Design. De hjälpte oss väldigt mycket med designen av vår applikation och det kändes bra att ha någon utomstående att kunna gå till för checkups då och då.

Denna gången hade vi inga sprint reviews alls då vi bestämde oss för att ha en lång sprint istället för flera korta eftersom folk reste bort under julen. Negativa resultat av detta blev inkonsistens i applikationsdesignen samt obalans i hur mycket olika gruppmedlemmar jobbade.

Lösningar till detta kunde ha varit att ha kortare sprintar och då på något sätt ha mötena via internet, till exempel skype, eller att ha någon typ av progressmätare så att man kunde se hur långt alla kommit på sina tilldelade uppgifter. En annan lösning hade kunnat vara att ha någon i gruppen som hade i uppgift att hålla koll på att alla arbetade lika mycket samt att applikationen blev konsistent i design, kod och funktionalitet. En typ av Scrummaster helt enkelt.

5. Best practices for using new tools and technologies

Efter den första implementationen kände sig alla i gruppen ganska säkra på hur Android Studio och Git fungerar, detta medförde att gruppmedlemmarna kunde arbeta mer självständigt på sina user cases.

Då funderingar kring olika implementationssätt har uppkommit så har dessa lösts gemensamt av gruppen med hjälp av chatt på Facebook. Detta har fungerat bra.

I den första implementationen av appen använde vi Trello flitigt. Trello gjorde att vi enkelt hade uppsikt över vem som gör vad samt vad som är gjort och vad som finns kvar att göra. Då vi hade så få user cases i den här, andra, implementationen valde vi att helt strunta i Trello och endast dela ut user cases till olika personer i gruppen och att rapportera i Facebook-chatten när vi var klara med våra uppgifter. Detta var dock ett misstag och ledde till en hel del av våra problem under kompletteringen nu i vintras.

Att kunna se vad andra håller på med och hur långt de har kommit har visat sig mycket viktigt för att en grupp ska kunna samarbeta på bästa sätt. Trello hade kunnat vara en lösning på problem som ojämn arbetsfördelning, då man hade kunnat se när vissa varit klara och hur mycket andra hade kvar att göra. Man hade då kunnat dela upp det som tog längre tid och på så sätt kunnat göra applikationen ännu bättre. Trello hade också kunnat användas till att se när folk kände sig klara och då hade vi troligtvis fått upp diskussionen om vad det faktiskt betydde att göra klart ett user case. Det sistnämnda har också varit ett större problem inom gruppen som lett till inkonsistens i appens design och beteende.

Kort och gott hade Trello kunnat vara en mycket bra lösning på en hel del problem, speciellt då vi jobbade på separata håll.

6. Relationship between prototype, process and stakeholder value

Vår vision för hur vi ville att appen skulle se ut var nog inte i sig för orealistisk, däremot så var den väldigt generell med väldigt få detaljer om vad vi egentligen ville få ut från den. Det vi lade upp som user stories, men inte nämnde i vår vision, var att vi ville få med vokaler, konsonanter och ljudkombinationer men i slutändan implementerade vi endast vokalerna, vilket är något som de berörda parterna förmodligen hade blivit besvikna på.

Under hösten jobbade vi tyvärr inte så mycket med visionen utan hade den mest i bakhuvudet under tidens gång. Det vi ville ha var en app som nybörjare till svenska skulle kunna använda, genom att inte använda för mycket text utan fokusera på ljud, vilket i slutändan var något vi ändå fullföljde. Dock så borde vi reflekterat mer över färgvalet vi valde att ha i appen. Den slutade väldigt färgglad med starka färger, vilket vi insåg nu under vintern att vi borde lagt mer tid på att ändra, då färgerna vi valt inte riktigt passar den breda målgruppen vi riktade in oss mot. Det här hade vi förmodligen kunnat lösa om vi kontinuerligt arbetat med vår vision under implementationen.

Nu under vintern så gjorde vi misstaget att vi inte gick igenom visionen, och vad som står i den, innan vi började komma på user stories. Det vi dock insåg var att det är oerhört svårt att få fram allt vi som vi ville få fram enbart med bilder, det vill säga informationen som användare behöver veta om appen. Vi bytte ut bilder som var svåra att förstå men det hade kanske till och med varit bättre om vi också skrivit namnen på orden ovanför bilderna. Vi hade med största sannolikhet kunnat åstadkomma en lösning för dessa brister mycket tidigare om vi arbetat mer aktivt med visionen. Vi känner generellt att vi hade kunnat utveckla vår vision rätt mycket från början, då den i nuläget är för vag för att kunna leverera en bra produkt till målgruppen.

9. Burndown-chart

Vi har valt att inte fylla i vår burndown-chart eftersom vi inte hade många sprintar utan vi hade en som var väldigt lång. Vi har heller inte haft haft daily scrums så vi vet ju inte hur många poäng folk klarat av per dag. Vi blev dock klara med de flesta av våra poäng på vår sprint men vet inte hur vi ska distribuera den.

Vi inser att vi borde haft med detta att vi borde haft daily scrum för att det är bra att efter projektet är klart när vi arbetade på det, men har som sagt inte varit praktiskt genomförbart. Vi borde också haft korta sprintar för att kunnat få till burndown-charten.

