

REFLEKTIONSRAPPORT

- Outsourcify

Erik Abrahamsson

Andreas Eckhoff

Jacob Holmén

Marko Ivanovic

Johan Selin

Edvin Tegbrant

Chalmers tekniska högskola

DAT255 Software Engineering Project

2016-06-02

Innehållsförteckning

1. Introduction	1
2. Application of Scrum	1
2.1 Roles, team work and social contract	1
2.2 Used practices	2
2.3 Time distribution	3
2.4 Effort and velocity	3
2.5 User stories and tasks	3
3. Reflection on the sprint retrospectives	4
4. Reflection on the sprint reviews	4
4.1 Feedback from stakeholders	5
4.2 Interaction with stakeholders	6
5. Best practices for new tools and technologies	7
6. Reflection on the relationship between <i>prototype, process</i> and <i>stakeholder value</i>	8
7. Relation to literature or guest lectures	9
8. Comments to D1 and D2	10
8.1 D1	10
8.2 D2	11
Källor	11

1. Introduction

Outsourcify har under våren 2016 arbetat med att ta fram en app för felrapportering, i syfte att möjliggöra en förenklad kommunikation mellan bussförare för linje 55, trafikledningen på Keolis och Volvos verkstad. Gränssnittets enkelhet underlättar förarnas vardag då de flesta problem kan rapporteras in inom två klick på skärmen. Övriga fel som sker mer sällan kan rapporteras genom den inbyggda samtals- eller fritextfunktionen.

Denna reflektionsrapport inkluderar olika delar av arbetsprocessen, med fokus på hur utvecklingsmetodiken Scrum hanterats. Rapportens följande delar ämnar svara på Ronald Smith definition av reflektion: "Assessment of what is in relation to what might or should be and includes feedback designed to reduce the gap".

2. Application of Scrum

Vattenfallsmodellen har länge varit praxis inom mjukvaruutveckling. Dagens högföränderliga marknader ställer höga krav på flexibilitet och kundvärde, varför den modellen blir bristande. Därtill finns ett behov av att reducera risken att projekt misslyckas i ett sent skede och orsakar betydande kostnader.

För att leva upp till behovet av högre flexibilitet har agil utveckling växt fram. En version av agil utveckling är Scrum, vilken också använts som arbetsmetodik i projektet. Den iterativa och inkrementella processen har underlättat synligheten i framstegen och möjliggjort en hög följsamhet efter kunders behov och reaktioner. De nya förutsättningar och förhållningssätt som Scrum innebär tog tid att ta till sig, men nyttan ökade sprint för sprint.

2.1 Roles, team work and social contract

Scrum innefattar de tre rollerna utvecklare, produktägare och Scrum master. Alla i teamet agerade utvecklare. Produktägare utgjordes av kursansvarig Håkan Burden. Inom teamet utsågs dock en produktägarrepresentant, vars uppgifter innefattade att förmedla produktägarens intressen för att stötta prioriteringen av arbete och ifrågasätta val som gjorts, exempelvis gällande appens utformning. I denna roll ingick även huvudansvar för kontakten mot kunderna, exempelvis förare och trafikledning från Keolis. Inom gruppen utsågs också en Scrum master, vilken bland annat coachade teamet i användningen av Scrum och såg till att rätt struktur följdes på möten. Därtill hade Scrum master även kontakt med andra team. Rollerna som användes upplevdes till viss del hämma effektiviteten i arbetsgången genom en stel struktur. Det är möjligt att arbetsroller passar bättre i större organisationer där det finns ett högre behov av struktur. I projektet följdes rollerna halvdant, varför ett alternativ hade varit att slopa rollerna och fördela uppgifterna på alla i teamet.

Det hade varit fördelaktigt i projektet att inte lägga så mycket fokus på att hålla rollen som Scrum master särskild. Som Henrik Kniberg beskriver är risken att Scrum mastern lägger mycket tid på administration och att teammedlemmarna inte blir engagerade i sprint backloggen (s75). Till viss del upplevdes dessa symptom i det genomförda projektet. För framtida projekt kommer alltså större fokus läggas på att involvera alla i teamet för hantering och uppdatering av sprint backlog, i syfte att öka sammanhållningen, engagemanget och effektiviteten. Under projektet var samarbetet lågt med andra team. Ett koncept som varit fördelaktigt att använda för att hjälpas åt och komma fram till bättre lösningar i slutändan skulle kunna vara Scrum of Scrums. På Spotify anges de användas för att hantera beroenden mellan team (s142). I Outsourcify's fall skulle det handla om att skapa beroenden till andra team snarare än att undvika dem, dvs. att inse hur apparna kan använda sig av varandra för att skapa en bättre totalprodukt. För framtida projekt är alltså ett alternativ att planera in regelbundna möten med andra Scrum masters för att gagna synergieffekter.

Samarbetet i gruppen har fungerat utan problem. Då kandidatarbetet genomfördes samtidigt med samma konstellation underlättades tidsplaneringen. Möten kunde hållas regelbundet och kommunikationen online minimerades, vilket höjde informationskvaliteten. Det sociala kontraktet formades tidigt i projektet. Det mesta följdes utan bekymmer. En vidare diskussion kring det sociala kontraktet går att finna i slutet på kapitel 8.1.

2.2 Used practices

Parprogrammering användes för att framställa koden. Det fanns klara fördelar i att kunskapsdelningen främjade lärandet. Det hjälpte till att överbrygga gapen i ingående kunskapsnivåer. Samtidigt blev arbetet mer effektivt då problem både kunde upptäckas och lösas gemensamt. I slutändan blir kvaliteten högre då allt arbetet kan dubbelkollas. Att koordinera gruppens olika scheman utgjorde inget större hinder med tanke på samarbetet i kandidatarbetet. Periodvis prövades också trippelprogrammering, men i längden hämmades då effektiviteten. Genom att i största möjliga utsträckning utveckla moduler, med få kopplingar av funktionalitet, kunde arbete utföras parallellt. Vissa moduler var dock så pass små att det då hade varit fördelaktigt att implementera dessa enskilt. Effektiviteten med programmering gjord i par blev därför kanske inte lika hög som optimalt möjligt. Under arbetet varierades arbetsparen för att utmana samarbetet och kunskapsspridningen ytterligare. Det var till nytta då sjukdomar uppstod vid olika tillfällen. Henrik Kniberg bekräftar främjad kunskapsdelning, effektivitet och kvalitet (s104-105) men tar även upp en annan aspekt. Genom att rotera paren frekvent under projektet upplevdes fördelen i en känsla av högre upplevt kollektivt ägarskap över koden, vilket leder till robusta team. Riskerna reduceras då för prestationsnedsättningar om någon skulle bli sjuk (s110).

Stand up-meetings användes varje dag som programmering skedde. Det gav en överblick åt hela gruppen och var till stöd för att undvika hinder i den fortsatta kodningen. Samtidigt upplevdes mötena ta tid och inte alltid vara till nytta. Det hade varit fördelaktigt med kortare möten med endast kvalitativ information. Då teamet arbetade nära varandra hela tiden kanske stand-up meetings inte varit nödvändiga överhuvudtaget, alternativt att gruppen kommit överens om ett större fokus i samråd med Scrum master.

2.3 Time distribution

Outsourcify:s sex gruppmedlemmar ämnade arbeta 20h/vecka, inklusive föreläsningar, programmering, kundkontakt och övriga moment. Vissa veckor underskreds denna nivå, exempelvis i samband med inlämning av kandidatarbete. Andra veckor överskreds nivån, men som helhet har storleken på variationerna varit låga. Alla har varit inblandade i alla moment, men eftersom gruppmedlemmarna har olika intresseområden har fokus varierat mellan kodning, kundkontakt och dokumentation. Gemensamt för flera gruppmedlemmar är att ett betydande antal timmar ägnats åt kunskapsinhämtning gällande framtagning och testning av Android-kod.

2.4 Effort and velocity

Estimering av arbetet upplevdes svårt. Fibonaccital användes för att bestämma storleksnivåer, vilka anpassades efter hur mycket tid funktionaliteten upplevdes skulle kräva. Svårigheten visades i att nivåerna sällan stämde med vad som lyckades uppnås. En mer precis matchning hade varit fördelaktigt i planeringsarbetet. Som gruppen uppfattade det finns det inga genvägar till en sådan matchning. Uppskattningarna blev mer precisa för varje sprint, intuitionen för hur lång tid en viss funktionalitet tar att implementera blev bättre och bättre. Ett stressmoment som gruppen upplevde var den så kallade "hockey stick"-effekten, vilken innebär att en stor del av alla user stories blev avklarade i slutet av sprintarna. Eftersom gruppen på grund av detta avvek från "optimal" velocity stora delar av varje sprint uppstod stress.

Vissa user stories tog längre tid än andra beroende på buggar och problem som uppstod samt oväntad ny kunskap som behövde läras in. Samtidigt hade vissa user stories gemensam logik och när den första implementerats gick det relativt snabbt med de andra. Vid estimeringen diskuterades det i gruppen vilket antal story points som är lämpligt, men det hade eventuellt varit bättre att följa en mer strukturerad modell. Exempelvis planning poker, där alla vänder upp kort och motiverar sina val om flera skiljer sig åt. Processen återupprepas tills konsensus har nåtts. Genom att följa detta sätt hade deltagandet och motivationen i gruppen ökat och potentiellt bidragit till en bättre estimering.

Scrum tillämpades i projektet genom sprints som var en vecka långa var, vilka påbörjades måndagar kl 12. Att använda korta sprints var till nytta då inkrementen snabbt visade vilket värde som skapas. Fokus var på att snabbt leverera funktionalitet. En risk som upplevdes var att teknisk skuld kunde uppstå om allt för mycket fokus lades på velocity. Funktionalitetsstillväxten blev hög men koden blev svårare och svårare att arbeta med allteftersom. Samtidigt sker projektet under en begränsad tidsperiod, att arbeta upp en teknisk skuld kan vara motiverat eftersom huvudmålet har varit att leverera så mycket värde som möjligt i slutet av kursen. Det är motiverat eftersom snyggt designad kod tar längre tid att implementera, men snygg kod tillför nödvändigtvis inget kundvärde.

2.5 User stories and tasks

User stories beskriver användarbehoven och har varit till nytta för att visualisera det som ska göras. En inledande product backlog skapades första arbetsveckan, men under arbetets gång har ny funktionalitet tillkommit och viss gammal fallit bort. Anledningen är en tydligare förståelse av kundens behov. Product backlog ordnades efter prioritet, men det var ibland svårt att göra det. Ett alternativ hade varit att låta kunden göra prioriteringen och på så sätt levereras högt värde tidigt i projektet.

I början av varje sprint lades user stories över i en sprint backlog samtidigt som de graderades utifrån komplexitet. Vissa user stories implementerades aldrig, till följd av prioriteringar, och en fördel med detta arbetssätt är alltså att ingen tid slösas på onödig estimering (dvs. user stories estimeras först när de plockas från product backlog till sprint backlog). Ett alternativ som inte använts under projektet men identifierats i efterhand är att göra en grov estimering av product backlog, t ex S/M/L. På det sättet fås en överblick och estimeringen i sprint backlog kan potentiellt gå snabbare. En fördel med user stories är att de är tydligt kopplade till ett kundvärde. De user stories som använts har teamet eftersträvat att skapa utifrån tårbitsprincipen. På så sätt blir alla utvecklare delaktiga vid skapande av både GUI och logik och det möjliggör även att en potentiellt levererbar produkt finns färdig efter varje sprint.

3. Reflection on the sprint retrospectives

Sprint retrospectives utfördes efter varje avslutad sprint. Inledningsvis var de tidskrävande då strukturen och agendan inte var självklara. Det fanns tydliga nyttor i att kunskap kunde spridas och läget uppdateras för övriga i teamet. Nuläget diskuterades, varigenom problem och hinder kunde identifieras. Genom att ta hänsyn till dessa i planeringen underlättades det vidare arbetet. Fokus på ständiga förbättringar ökade både effektiviteten och produktiviteten i längden. En viktig faktor var att alla fick komma till tals och tala ur sin synvinkel i en öppen miljö. Samarbetet förbättrades och ägarskapet över projektet förstärktes.

Det hade dock varit fördelaktigt med mer tidseffektiva möten där alla även är på plats på utsatt tid. Det finns olika alternativ för att nå detta. Exempelvis hade mötena kunnat hållas online för att eventuellt öka tillgängligheten och möjligheten för alla att delta. Nackdelen är att det försvårar och ger en mindre tidseffektiv diskussionen samtidigt som tolkningen av åsikter kan bli svår. Ett annat alternativ hade varit att ha en Scrum master som påpekar riktlinjerna tydligare. Att undvika onödiga sidodiskussioner och följa en agenda gör det lättare för alla inblandade i slutändan.

I en grupp som träffas dagligen och där daily stand-up meetings användes går det att ifrågasätta användandet av sprint retrospective. De frågor som besvarades - vad som gått bra under sprinten, vad som gått mindre bra och vad som kan göras för att nästa sprint ska gå bättre - hade gruppen en bra uppfattning om ändå. Det fanns fördelar med att få det nedskrivet officiellt dock, och även med att låta personer som vanligtvis håller inne på sina åsikter komma till tals.

4. Reflection on the sprint reviews

Interna sprint reviews har genomförts efter varje avslutad sprint. Allas framsteg har visats upp och resultatet har diskuterats. Kontinuiteten för dessa avstämningar har varit till nytta för att följa utvecklingen av det kundvärde som kan levereras. Vid sidan av interna sprint reviews har teamet också visat upp statusversioner av appen för andra intressenter, däribland produktägare, bussförare och handledare från näringslivet.

4.1 Feedback from stakeholders

Inledningsvis i projektet togs idéer och förslag fram över hur appen skulle se ut och vilka funktioner den skulle innehålla. Vid det första handledningstillfället mottogs betydelsefull feedback. Håkan Burden menade att en prioritering var nödvändig över olika delar. Från början var tanken att både behandla vanliga fel på en buss samt även bygga ett slags uppslagsverk med en komplett uppsättning över varningssymboler som kan dyka upp på förarpanelen. Det senare prioriteras därefter ned eftersom störst kundvärde antogs kunna levereras med en app med olika typer av problem snarare än endast symboler. Att kunna leverera en mindre app med ett tydligt fokus ansågs vara mer värdefullt än en bred, komplex app. Feedbacken förde teamet i rätt riktning och låg till grund för prioriteringen av arbete. Likaså handledaren från näringslivet, Gustav Sohtell från Techno creatives, höll med om att ett mindre scope var att föredra med tanke på projektets begränsade tid och resurser. Här diskuterades också olika designval för appen, och Gustav poängterade att enkelhet är av stor vikt i systemet som ska byggas. Inledningsvis experimenterade arbetsgruppen med olika kombinationer av färger, men för att följa enkelheten och undvika störningsmoment reducerades skalan till endast svart och vitt. Efter diskussioner valdes också Skype som väg för att möjliggöra en samtalsfunktion, GPS-position för att ange plats och mailfunktionen för att visa vilken information som är tänkt att skickas till Keolis trafikledning. På så sätt undviks beroenden till andra system och enkelheten bibehålls i projektet, samtidigt som appen och dess funktionalitet går att visa upp för kunden och andra intressenter.

Vid flera tillfällen var arbetsgruppen i kontakt med kunden. Inledningsvis med Keolis för att få information om vanliga inrapporterade fel. Ett komplett utdrag för 2016 mottogs, vilket efter tolkning och sedan analys med pivottabeller visade vilka de vanligaste kategorierna för inrapporterade fel var. Denna information användes för att bestämma utformningen av appen.

Kundkontakten gick vidare till intervjuer med bussförarna. Informationen om vanliga fel som rapporteras bekräftades till stora delar av förarna. En tidig version av appen ämnade dock täcka in alla fel som en förare kan tänkas stöta på. Det skapades en lösning med komplicerade menyer i flera steg, vilket förarna kritiserade när detta presenterades. Ett fel som sker en gång om året ansågs inte nödvändigt att kunna rapportera med appen, då kan istället samtals- eller fritextfunktionen utnyttjas i detta sällsynta fall. De frågor som ställdes under första intervjuomgången presenteras i figur 1 nedan.

Figur 1. Feedback från första intervjuomgången

	Förare 1	Förare 2	Förare 3	Förare 4
Hur länge har du kört elbussar?	1 månad	2 månader	10 månader	10 månader
Vilka är de vanligaste problemen du rapporterar?	Dörrar	Dörrar, laddstation	Laddstation och klimat	Dörrar, GPS, motorproblem
Hur ofta rapporterar du in problem?	2 gånger i veckan	1-2 gånger i veckan	Osäker	2-3 gånger i veckan
Hur länge står du stilla på varje busshållplats?	1-2 minuter	Osäker	1-2 minuter	40 sekunder
Finns det några problem som du inte rapporterar?	Nej	Nej	Problem som går att fixa på plats	Nej
Har du någonsin velat rapportera ett problem utifrån lukt, ljud, känsla eller liknande?	Nej	Nej	Nej	Nej

Appens slutgiltiga utseende efterliknar paretoprincipen, eller 80/20-regeln, dvs. att en liten andel av möjliga fall som kan inträffa också kan rapporteras in med appen, men dessa står i sin tur för en stor andel av antalet fel som uppstår. En intuitiv design var alltså målet där enkelhet prioriterades. Att kunna rapportera fel snabbt var också högt prioriterat och därför blev appen utformad för att förare ska kunna rapportera redan vid nästa busshållplats. Bussförarna var också till hjälp för att specificera undernivåer till felkategorierna. De sade detaljer som Volvos verkstad efterfrågar, men som inte fanns i utdraget som skickas av Keolis. Exempelvis om det är ett dörrfel, då ska det rapporteras var i bussen det uppstod samt om problemet är att dörren inte kan stängas eller om det är att den inte kan öppnas. Feedback från den andra intervjuomgången presenteras i figur 2 nedan.

Figur 2. Feedback från andra intervjuomgången

	Förare 1	Förare 2	Förare 3	Förare 4	Förare 5
Tycker du att designen är enkel att förstå?	Fantastisk meny, enkla bilder, förstår vad appen handlar om. Design 5/5.	Bra och tydlig app.	Bra och enkelt att förstå designen. Liten men tydlig. Detta är framtiden.	Tydlig app, bra idé.	Okej app, ganska enkel app.
Är det något som kan förbättras eller läggas till i appen?	Möjligheten att rapportera om dörrar inte kan öppnas eller inte kan stängas.	Möjligheten att specificera om laddaren inte går upp eller om den inte åker ner.	Appen behöver inte kunna meddela om platserna nästan är slut.	Möjligheten att rapportera om det är för varmt eller för kallt i bussen.	Saknar inget särskilt.
Skulle den här appen underlätta ditt dagliga arbete?	Ja.	Absolut, appen kräver mindre fokus än ett telefonsamtal.	Skulle underlätta arbetsdagen.	Väldigt bra app, skulle underlätta arbetsdagen.	Skulle underlätta arbetsdagen.

4.2 Interaction with stakeholders

Interaktionen med intressenter skedde kontinuerligt under arbetets gång. Kontakten med Keolis skedde över mail och kontakten med bussförarna var i samband med att de laddade bussen på Johanneberg. Till förarna hade ett kortare frågeformulär förberetts och en läsplatta togs även med som hade en prototyp av appen. Interaktionen med handledare från Chalmers och näringslivet skedde i samband med organiserade handledningstillfällen.

Som helhet användes feedbacken från intressenter främst till att utforma appen och fundera över vilken design som var att föredra. Det hade varit fördelaktigt att fokusera mer på arbetsprocessen i samband med handledningstillfällena. Även om arbetet flöt på bra och ny funktionalitet växte fram fanns det ibland en osäkerhet i om rätt arbetsprocess följdes. Därtill om för mycket eller för lite arbete planerats för den närmsta tiden. Ett alternativ för kommande projekt är att föra diskussionen med intressenter utifrån product backlog och sprint backlog. Genom att stämma av prioriteringen av user stories med produktägaren hade en större säkerhet kunnat bildas kring om rätt problem tacklades för stunden.

5. Best practices for new tools and technologies

Under projektet användes flera verktyg som stöd. Rapporteringsverktyget JIRA användes huvudsakligen för samordning av user stories, arbetsfördelning och uppföljning av vad som avklarats. Anledningen till att detta verktyg valdes var att hela arbetsgruppen hade erfarenhet av det sedan kandidatarbetet. Det är brett använt bland mjukvaruutvecklande företag. Gruppen över-skattade dock nyttan av verktyget. Eftersom det krävdes en viss inläring blev det snarare ett hinder att all rapportering var tvunget att ske där. Vid ett projekt som bedrivs under en längre period kan antagligen nyttorna med verktyget realiseras bättre men gruppen glömde att ta hänsyn till den learning-curve som det innebär att sätta sig in i ytterligare ett verktyg.

Ett alternativ till JIRA hade enligt Henrik Kniberg kunnat vara exempelvis Trello, MS Excel eller Google kalkylark (s8). Alla hade kännedom om JIRA sedan innan, men inte tillräckligt för att jobba efter full kapacitet utan hinder. Vid sidan av de digitala verktygen hade också en fysisk lösning varit bra. Exempelvis en whiteboard, med en Scrumboard och post-it lappar, på ett fast ställe. Ett hinder är den begränsade tillgängligheten på skolan. Där hade social contract även kunnat sättas upp för att erinra alla gruppmedlemmar om vad som är överenskommet.

För att samordna koden användes GitHub, enligt instruktion från kursen. Verktyget gav möjligheter till versionshantering och att arbeta parallellt med olika funktionaliteter i appen. Tidigt i projektet fick alla medlemmar i gruppen lära sig hur GitHub fungerar, men senare kom uppladdningarna att fokuseras till en användare för att öka effektiviteten genom tydliga ansvarsområden. För vidare beskrivning av användningen utav GitHub, se dokumentet “Kommentarer till GitInspector”. GitHub har fungerat väl i slutet av projektet, men i början utgjorde det en learning-curve och ett hinder för integrering av kod. För övrigt hade det varit fördelaktigt med mer regelbunden uppdatering av kod för att säkerställa att den senaste versionen finns tillgänglig för alla.

Samordning av dokument hanterades med hjälp av Google Docs. Anledningen var att det är lätthanterligt, alla känner till det och att alla kan redigera samtidigt. Därtill är det positivt att kunna spåra ändringar och chattfunktionen kan vara till hjälp vid de tillfällen alla inte skulle befinna sig på samma plats.

Programmeringen sköttes i Android Studio, då detta skulle ge bäst möjligheter till handledning under kursens gång. Om svårigheter stöttes på kunde dessa problem lättare lösas och fokus kunde läggas på värdeadderande aktiviteter. Fokus har även varit att, i största möjliga utsträckning, använda “best practice” inom Android, därför har Espresso, Hamcrest, JUnit och Androids standardbibliotek använts. Som emulator användes dels Android Studios egna, men även en läsplatta (Samsung Galaxy Note 10.1) som är representativ för målenheterna.

För viss uppföljning av kvalitet användes programmet Findbugs. Genom att kontinuerligt stämma av och även inkorporera detta i Definition of Done kan en hög kodkvalitet säkerställas. I de första versionerna av Definition of Done var Findbugs ej inkluderat. Då projektet fortskred blev dock behovet av det tydligare för att säkerställa kodkvalitet.

Det som eftersträfvats med verktygen är att samla allt på ett ställe för att få en bättre överblick. Samtidigt är det viktigt med enkla verktyg som blir ett stöd i det dagliga arbetet och inget extra moment som ses som onödigt. Till stora delar har detta uppnåtts, men exempelvis har JIRA upplevts något för arbetsamt. En annan sak att ha i åtanke för framtida projekt är att tydliggöra nyttan för alla inblandade och på så sätt skapa en motivation kring användandet av verktygen.

6. Reflection on the relationship between *prototype*, *process* and *stakeholder value*

Prototyp, process och kundvärde är tre nära sammankopplade begrepp. Den arbetsprocess som följts under projektet har varit iterativ och inkrementell. Från en idé om funktionalitet har prototyper i olika versioner arbetats fram som visats upp för kunden. Efter varje sprint mottogs alltså feedback som kunde hjälpa till att höja kundvärdet inför nästa iteration. På så sätt bekräftas också att rätt arbete utförs, dvs. värdeadderande aktiviteter som har en påverkan i slutändan. I större mjukvaruutvecklingsprojekt är det vanligt att all funktionalitet som överenskommits till en viss deadline inte levereras. Då är det viktigt att välja och prioritera arbetet rätt så att så högt värde som möjligt fortfarande levereras. Samtidigt måste produkten i slutet av sprinten vara tillräckligt komplett för att kunna utvärderas helt. Det krävs alltså ett fokus på att ta fram “minimal viable products” för att intressenterna ska kunna testa kontinuerligt och förmedla sina åsikter.

Under projektets gång förändrades designen på appen mycket. En prototyp skapades tidigt, vilken var tänkt att täcka in alla möjliga fel som kan uppstå i en buss. De tre vyer som utvecklades då går att finna nedan. Med kundkontakt som en del av arbetsprocessen följde reaktioner om att appen inte blev användarvänlig. En komplicerad och heltäckande app ansågs inte vara den bästa lösningen. Det visade sig att kunden snarare ville ha en enkel app som täckte in de flesta problemen. Scrumprocessen tillät alltså gruppen att ta fram ett så högt kundvärde som möjligt, genom kontinuerlig feedback kring prototypen.

Figur 3. GUI för den första prototypen



Michael Öhman från Spotify underströk vikten av att testa lösningarna mot kunder. Modellen ”Think it – Build it – Ship it – Tweak it” hade varit fördelaktig att applicera i projektet. Genom user research redan i Think it-stadiet hade den första versionen av appen kunnat dödas i ett ännu tidigare skede än den gjorde. På så sätt hade tid sparats in som hade kunnat läggas på funktionalitet som faktiskt levererar värde till kunden.

En lärdom att ta med till nästa projekt är att det finns ett stort värde i att visa upp en prototyp i ett tidigt stadiet, även fast att all funktionalitet inte är implementerad. Genom att få tidig feedback minskas projektets kostnad och risk, samtidigt som fokuset på kundvärde bibehålls. Det går exempelvis att reda ut missförstånd om kundbehov och önskemål om designen. Att komma till diskussionen med intressenter med en tidig prototyp underlättar kommunikationen och visualiserar arbetet och dess status. Då kan intressenterna lättare komma med åsikter som ökar det värde som produkten kan tillföra dem. Samtidigt blir intressenterna delaktiga redan från början, vilket skapar engagemang och ökade chanser för att slutprodukten accepteras av slutanvändarna när projektet är färdigt. Vid sidan av ökad motivation kan också viss träning och utbildning ske redan under utvecklings- och diskussionsfasen. En risk är dock att intressenterna fäster sig vid en tidig prototypdesign, vilket kan bli ett problem om utvecklingarna senare väljer att inte gå vidare med den idén/strukturen. Den tid och de resurser som krävs för att skapa prototypen bör vägas mot nyttorna som fås i form av ett förhöjt, slutligt kundvärde. Ännu större nytta fås om processen upprepas kontinuerligt under projektet, vilket ger ytterligare en dimension av det iterativa arbetssättet. Det har utnyttjats i det aktuella projektet, och kan även beskrivas som evolutionary prototyping. Viss risk med mycket kundkontakt på detta sätt är att kunderna kommer på fler attribut/egenskaper i appen som från början inte ingår i "avtalet". Samma situation upplevdes under legoövningen. Fokus ska ligga på att utvärdera den funktionalitet som från början var överenskommet, snarare än att tillföra ny funktionalitet.

7. Relation to literature or guest lectures

Relationen till litteratur och gästföreläsningar har till viss del behandlats integrerat i rapportens olika delar, men flera delar har samlats här nedan.

John Lantz från Volvo pekade på komplexiteten som uppstår i mjukvaruutvecklingsprojekt. Fullständig planering från dag ett fungerar inte längre, utan experimentering och flexibilitet är nödvändigt. Detta är också en av anledningarna till att arbetsgruppen jobbar med arbetsmetodiken Scrum i projektet. Ett sätt att minska komplexiteten, som använts mycket i projektet, är det som Lantz beskriver som rödpennan. Alltså att stryka funktionalitet och överflödiga modeller som inte adderar värde. På samma sätt drar han en koppling till Steve Jobs och iPhone. Därutöver menade han att första versionen inte behöver kommenteras, då den ändå kommer att slängas. Igenkänningsfaktorn är hög i det aktuella projektet, då en större förståelse för kundbehoven ledde till en radikalt omarbetad design. Tips om testningen har också utnyttjats, exempelvis att göra det klart för en användare när en knapp har tryckts samt att stresstesta applikationen genom att klicka flera gånger på en knapp. Under projektets gång fanns en stor försiktighet i att bygga upp teknisk skuld, men Lantz påpekade att det i vissa fall kan vara värt det om det innebär att utvecklingen går snabbare. Samtidigt är det svårt att skapa framtidskompatibilitet.

Lantz sade att diversitet är viktigt för framgång, dvs. att kombinera olika människor som inte gör samma misstag. Detta är en faktor som eventuellt hade ändrats om projektet gjorts om. Alltså att kombinera olika kompetenser, så som både studenter från I och från IT, och inte bara samarbeta inom samma program.

Under gästföreläsningen nionde maj, då en representant från Interaktionsbyrån var och diskuterade framtidens mjukvara, nämndes att applikationer ska utformas för att manipulera användare till att bruka produkten som det är tänkt. Vissa funktioner är nödvändiga för att ge ett heltäckande scope, men om de inte är det huvudsakliga ändamålet med applikationen så ska de döljas för användaren. Exemplet som togs upp på föreläsningen var att Spotify uppmuntrar sina användare till att lyssna på ny musik. Därför är användarens egengjorda spellistor “gömda” och det krävs navigation genom flera vyer för att hitta dem. I Outsourcify's felrapporteringsapp har detta designtänk applicerats genom att rutnätet av ikoner är ordnade i fallande prioritetsordning. De ikoner som är vanligast förekommande ska vara lättast att hitta och därför placerades de högst upp till vänster. Ikoner som inte är prioriterade, exempelvis knappen “Övriga problem”, ligger istället långt ner till höger.

En sak som inte använts i projektet för sprint-planering är sprint goal. Scrum Guide rekommenderar exempelvis detta (s24). Anledningen till att det inte användes var på grund av okunskap. Henrik Kniberg menar att det inte alltid är nödvändigt (s24), att det kan räcka med ett övergripande mål för flera sprintar. Under projektets gång har det ändå funnits ett framväxande behov av att ha något gemensamt att sträva mot. Det tros kunna inge ett lugn i gruppen om att rätt saker hela tiden utförs. En annan aspekt på sprint-planering mötena som Kniberg tar upp är längden. Ett problem som upptäcktes under projektet var att de tog för lång tid. En bättre lösning än att låta dem dra ut på tiden hade enligt Kniberg varit att sätta en gräns och låta mötet gå dåligt (s20). Isåfall skulle man lära sig av det och förbättra nästa vecka. För framtida projekt ska därför en fast tidsgräns sättas för mötena för att säkerställa effektiviteten och att tid ägnas åt värdeadderande aktiviteter.

8. Comments to D1 and D2

8.1 D1

Reflektionen över Legoövningen berörde de tre nedanstående punkterna.

- Svårt uppskatta velocity
- Viktigt med en Scrum master
- Kundkontakt

Svårigheten i att estimerar velocity har även upplevts under projektet. Likt Legoövningen var det speciellt svårt tidigt i projektet, eftersom det inte fanns någon prestation att jämföra med bakåt. En skillnad gentemot övningen är att alla i gruppen hade tidigare erfarenhet av att bygga Lego. Det fanns alltså en intuitiv uppfattning om hur ansträngande det är att exempelvis bygga ett hus. Motsvarande intuition saknades i projektet då tidigare erfarenhet av att skapa en app i Android Studio saknades. Väl under projektets gång justerades estimeringen av velocity snarare än definition av en story point.

För det andra påpekades vikten av att ha en Scrum master. Detta följdes upp genom att också utse en Scrum master i projektet. Rollen skapade struktur och säkerställde följsamhet till Scrumprinciperna. För det tredje påpekades hur kundkontakten skulle hanteras. Denna fråga bedöms ha större tillämpbarhet under Legoövningen än i ett verkligt projekt, men bör ändå ha i åtanke. Under intervjuerna med bussförarna anpassades frågorna för att förtydliga deras behov av en app, snarare än fråga om allt de vill ha in i en app. En enkel app är att eftersträva som assisterar förarna i felrapportering, snarare än en komplett app som digitaliserar och presenterar all tänkbar information för förarna.

Visionen som skapades första veckan är fortfarande representativ för vad appen ämnar uppnå. Fokus har dock skiftats något i appen, vilket har uttryckts i version två av visionen. Hög informationskvalitet är fortfarande viktigt, men huvudmålet i dagens läge är att appen är snabb och enkel för förarna. Tanken är alltså att felrapporten som skickas ska innehålla den information som krävs för Keolis och Volvo att agera, men samtidigt vara så enkel som möjligt att använda för förarna. På så sätt bibehålls enkelheten och appen går snabbt att använda.

Den product backlog som skapades första veckan var en bra start för att komma igång med programmeringen. Ett fel som gjordes var att utveckla allt för mycket funktionalitet från denna innan kundvärde bekräftats för just en sådan version. Intervjuer med förare där penna och papper tas med hade varit ett alternativ.

Angående det sociala kontraktet upplevdes problem med att alla inte höll tiderna. Högre effektivitet i arbetet hade uppnåtts om alla faktiskt kom i tid. När man kommer sent pekar Henrik Kniberg bland annat på två alternativ som inte använts i projektet. Det ena är att bestraffa med pengar och ett annat är att tvingas till att sjunga en sång (s76). Stämningen tros kunna bli negativ, men i längden är det värt mycket att ha gruppmedlemmar som kan lite på varandra och följa uppsatta regler. En pengapott som i slutändan går till sociala event ska alltså införas i kommande projekt. En sådan lösning diskuterades vid utformningen av det sociala kontraktet men bedömdes då inte vara nödvändig.

8.2 D2

Vad gäller produkten följs de steg som rekommenderas i halvtidsrapporten mycket väl. Då fanns en app som upplevdes fylla kundkraven väl, men efter närmare kundkontakt visade det sig inte stämma. Genom att banta ned innehållet för att minska omfånget i appen kunde den upplevda nyttan höjas, med fokus på hög användarvänlighet. Detta inkluderar designval för appens förstasida samt annan funktionalitet som en hemknapp. Att kundkontakten genomfördes som den gjordes grundades i John Lantz föreläsning om rödpennan, vilken också kommenterades i halvtidsrapporten. Att ta fram en Definition of Done (DoD) var en annan förbättringsåtgärd som identifierades. Detta underlättade strukturen för det vidare arbetet och säkerställandet av att endast helt färdigställda uppgifter klarmarkeras. Inledningsvis i projektet användes ingen DoD, vilket försvårade och skapade förvirring om när nytt arbete kunde påbörjas. Henrik Kniberg menar att en DoD är mycket viktigt (s37). En aspekt som användes i projektet var att kräva användarfeedback för klarmarkering av uppgifter, vilket även Kniberg tar upp som en möjlig del att ha med i DoD (s38). Det tog längre tid och upplevdes krångligt till en början, men när feedbacken väl kom föddes nya idéer som gav stöd till ständiga förbättringar för appen. Projektgruppen känner igen sig i Knibergs beskrivning av att programmerare är otåliga och snabbt tar på sig nya uppgifter (s 137).

Dessutom sköttes GitHub mer kontinuerligt efter halvtidsrapporten. Som stöd till klarmarkering av uppgifter var acceptance criteria för user stories. Dessa upplevdes dock överflödiga i vissa fall, då user stories varit enkla att förstå och i princip endast kan utvecklas på ett sätt. Det blir då lätt att förstå när uppgiften är klar. Samtidigt, eftersom parprogrammering utnyttjats, har en slags extra kontroll alltid skett över utvecklingen.

Källor

Kniberg, Henrik (2015) *Scrum and XP from the Trenches - 2nd edition*