Final Reflection

Det finns väldigt många människor världen över som har problem med att hitta nya vänner, speciellt vuxna. Detta var problemet vi ville lösa med MeetApp, en plattform att **träffa** nya människor, inte bara det enkla knapptrycket “Lägg till som vän” eller “Följ”, utan ett verkligt möte mellan människor. Målet var att MeetApp skulle bli go-to plattformen att träffa nya människor och vi valde att lösa detta via evenemang som vi kallar för MeetUp:s.

Vår bild av kursen är övervägande positiv. Vi tar framförallt med oss två stora positiva aspekter. Den första är att de multifunktionella teams vi arbetat i har fungerat väldigt bra. Det är sällan man i skolan tvingas in i den situationen, men i arbetslivet kommer vi troligtvis bara arbeta på det viset, vilket gör att utmaningarna med olika kunskaper blev ett trevligt problem att lösa, vilket också diskuteras senare i rapporten. Det andra vi tar med oss är upplägget med fokus på ramverket och arbetssättet istället för produkten. Grupprojekt består ofta av en uppgift, men ingen metod att lösa den. Det har varit väldigt kul och uppfriskande att istället få metoden och få utforma uppgiften själv. Vi tar också med oss lite förbättringspunkter, men dessa beskrivs i rapporten nedan, då de kunde bindas till en specifik fråga. Trevlig läsning!

1. Customer Value and Scope

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

A: *För att testa applikationen i liten skala, likt Facebook, valde vi att främst rikta oss mot studenter. Detta gjordes av två anledningar, dels är det mer ekonomiskt att börja i liten skala för att sedan expandera, men främst ansåg vi att det är lättare att utveckla något där man är en del av målgruppen. Tack vare att vi är en del av målgruppen kan man utgå från sig själv och hur man värderar olika funktioner. Vi valde ändå att genomföra en enkätstudie som skickades ut i slutet av projektet till studenter på Chalmers för att få stöd på att vi lagt fokus på rätt funktionalitet. Detta då ens egna åsikter blir färgade under utvecklingsarbetet och man förmodligen värderar funktionalitet som man själv utvecklat högre än det egentliga kundvärdet. Enkäten visade att vi i stora drag lagt fokus på rätt funktionalitet, dock med vissa avvikelser som vi tyvärr lagt mycket tid på.*

B: *Inför framtida projekt vore det fördelaktigt om man i ett tidigare skede kan få reda på hur den tänkta målgruppen värderar olika funktionalitet. I detta fall var träffsäkerheten på vald funktionalitet relativt hög, förmodligen på grund av att vi är en del av målgruppen, men så är inte alltid fallet. Det är också viktigt att identifiera sin målgrupp och veta vilket problem man löser.*

A → B: *För att tidigt få reda på vad användarna värderar bör en marknadsundersökning utföras i början av projektets gång. Detta kan då ligga till grund för vilken funktionalitet som bör prioriteras. Som tidigare nämnt var träffsäkerheten hög i detta projekt, men i framtiden då utvecklingsteamet inte nödvändigtvis är en del av målgruppen är en tidig marknadsundersökning av ännu högre vikt. För att kunna veta för vem man skapar värde är det viktigt att identifiera sin målgrupp tidigt i projektet och gärna, i så hög mån som möjligt, kommunicera med sina användare för att hela tiden se till att man skapar värde.*

1.2. The success criteria for the team in terms of what you want to achieve within the project (this can include the application, but also your learning outcomes, your teamwork, or your effort).

A: *Vi hade två huvudsakliga mål som sattes upp tidigt i projektet. Det första av dem var att vi ville skapa en fungerande applikation som går att använda. Vad som definieras som "kan användas" diskuterade vi inte ingående men huvudtanken var att det ändå skulle finnas så pass mycket funktionalitet att det gick att navigera i tjänsten och inte bara öppna en startskärm. Det andra övergripande målet var att vi som grupp ville lära oss om Scrum som utvecklingsmetodik och hur Scrum kan och bör appliceras i ett projekt. Vidare hade sedan varje individuell gruppmedlem sina egna mål, där exempelvis I:arna ville lära sig mer om programmering, tidsaspekten i utvecklingsarbete och liknande. IT:arna hade andra målbilder men som grupp hade vi alltså två huvudsakliga mål. Vi uppnådde båda av de stora målen vi satte upp initialt i*

projektet då vi både skapat en applikation som går att använda och vi känner att vi lärt oss om Scrum. Givetvis har det varit utmaningar på vägen, både programmeringsmässigt och Scrum-mässigt, något som vi går igenom i andra delar av denna reflektion. Men sett till det stora hela är vi ändå nöjda med vad vi åstadkommit, hur vi arbetat som grupp och hur vi applicerat Scrum för att komma framåt med projektet. Vi var alla lite nervösa inför det faktum att samarbeta med personer från andra sektioner än den vi själva tillhör. Nervositeten berodde mest på att det var väldigt nytt för alla men samarbetet har fungerat mycket bra hela vägen. Detta nya samarbetsformat ledde också till andra typer av utmaningar än de vi är vana vid, exempelvis att vi har väldigt olika förkunskaper vad gäller utvecklingsprojekt. I:arna hade väldigt begränsade förkunskaper vad gäller programmering men desto mer erfarenhet från grupparbete och projekt, medan studenterna från IT hade det omvända. Sammantaget går det att sammanfatta att vi lärde oss av varandra på bästa sätt och samtliga i gruppen känner att de lärt sig något som de inte kunde från början, både vad gäller metoden Scrum, programmering, GitHub och en massa andra saker. Om man ska nämna en negativ aspekt är det Scrum där vi var lite för dåliga på att reflektera över vad som gick bra och dåligt eftersom vi var ville framåt och skapa värde för de potentiella användarna. Att ha tålamodet att stanna upp, diskutera efforts och velocity och planera noggrant är något vi tar med oss.

***B:** I framtida projekt, oavsett om projektet är professionellt eller i skolmiljö, kommer alla inte att ha samma förkunskaper om ämnet som ska utredas. Vidare kommer gruppen troligen alltid bestå av medlemmar man kanske inte känner sedan tidigare. Därför är det av yttersta vikt att tidigt sätta upp tydliga läromål eller allmänna mål med projektet, både som grupp vad gäller outputen som projektet ska leda till men också individuella mål, både mjuka och hårda. Vi tror att det som legat till grund för att vi lyckades beror på att vi hela tiden haft en tydlig kommunikation, respekt och ödmjukhet inför varandra och de delar varje gruppmedlem är bra på och delar gruppmedlemmen är mindre bra på, vågat kommunicera denna svaghet för att sedan kunna gå vidare. Denna approach är något vi tar med oss till framtida projekt. Gällande Scrum-metodiken så går det att dra lärdomar till framtida projekt genom att oavsett vilket metodverktyg man använder, är det viktigt att stanna upp och reflektera över positiva och negativa aspekter med verktyget. Vidare behöver man verkligen ta tid till att reflektera över och utveckla användningen av verktyget för att verkligen optimera utnyttjandet av det.*

***A → B:** För att kunna nå samma framgång gällande tydliga målsättningar som grupp samt uppfyllandet av dem är det viktigt att sätta upp tydliga mål tillsammans som sedan följs upp och utvecklas under projektets gång. Vidare är det viktigt att varje gruppmedlem är med i diskussionen och framtagningen av målen så att målen speglar hela gruppens syn på projektet. Det är också viktigt att specificera vilket output projektet ska ge och när detta är tänkt att ske så att hela gruppen är medvetna om tidsramen för projektet vilket gör det enklare att både planera sin tid och lägga upp sprintar och user stories, eller vilket projektledningsverktyg man nu använder. Vad gäller användningen av Scrum är detta verktyg inte helt enkelt att hantera i början, även om man lär sig det över tid. Men en viktig poäng är att verkligen reflektera över användningen och försöka utreda vad som gjorde att sprintarna var lyckade vissa sprintar och misslyckade andra för att verkligen finna roten till framgångarna och misslyckandena. Detta är något som vi tror man kan lösa genom att ta 15-20 minuter extra varje sprint retrospective*

till att reflektera över appliceringen av Scrum. En annan fråga att ta hänsyn till är hur man bör adressera problematiken med olika förkunskaper. Detta fungerade bra i vårt projekt men vi tror nyckeln är att vara öppen med vad man är bra och mindre bra på samt kommunicera detta på ett tydligt sätt. Det är bättre att säga ifrån om en tror att en inte blir klar med user storyn och därför antingen behöver hjälp med den eller eventuellt öka effort på user storyn. Att be om hjälp och själv bistå med hjälp på de delar där gruppmedlemmen är stark är viktiga aspekter som vi alla tar med oss till framtida projekt.

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

A: *Vi formulerade alla user stories gemensamt i början av projektet. Samtidigt satte vi Definition of Done på alla olika backlogs som syftade till att ge tydliga regler för när ett kort skulle få flyttas från en backlog till en annan, exempelvis från "In progress" till "Testing". Vår Definition of Done innehöll även att för att ett kort ens skulle få skapas, måste det tillföra värde till projektet på något sätt. Task breakdown gjordes också i början av projektet. Dock bestämdes att alla user stories skulle utvärderas innan de flyttades till sprint backlog beroende på vilka delar av appen som den senare skulle integreras med. Vi skulle också utvärdera och eventuellt uppdatera effort för varje user story när vi flyttade den till sprint backlog. Arbetet fungerade bra i början av projektet, när delarna ofta var separata och inte behövde byggas ihop. Senare under projektet gick det sämre, speciellt under de två sista sprintarna. User stories som vi valde blev större och större efterhand på ett sätt vi inte räknat med. Troligtvis berodde detta på att ju större appen blev, desto fler delar behövde integreras med varandra. Det blev också svårare att bygga ihop appen när andra user stories som behövde integreras med inte var helt klara. Vi upptäckte även att vi gärna lämnade över olika delar av user stories som vi hade problem med till någon som hade bättre koll på området den skulle kopplas ihop med. Vi var inte helt klockrena på att utvärdera när arbetet gick bra, varför det var svårt att återgå till bra arbete när det gick sämre. Dock var vi duktiga på att utvärdera efforts och uppdatera dessa efter vad som behövde kopplas ihop och vad som kunde användas från tidigare user stories.*

B: *Till nästa projekt tar vi framförallt med oss vikten av att utvärdera arbete och user stories efterhand. Vi tar också med oss att även om det till en början kan vara svårt att slutföra en user story på egen hand, är det viktigt att lära sig det. Projektet var inte så stort att alla bara kunde ha koll på en del var, utan vi hade kunnat fråga varandra och lära oss själva. Detta hade troligtvis lett till ett mer effektivt arbete i slutet av projektet i och med att alla då hade haft koll på en större del av appen. Det är även viktigt att ta hänsyn till olika förkunskaper när man bestämmer efforts, vilket vi också tar med oss. En user story som har en effort på 5 för en person kanske har 10 för en annan, beroende på hur väl de är insatta på området den behandlar. Till nästa projekt bör vi också tänka på att uppdatera tasks under sprint planning på ett bättre sätt än vi gjorde denna gång. Nu blev vi i slutet av projektet lite överväldigade av hur många tasks vi ytterligare behövde för varje user story, varför sprintarna i slutet gick sämre än de tidigare map antal avklarade user stories per sprint.*

A → B: *Det är svårt att göra user stories som inte beror på varandra, men genom kontinuerlig utvärdering borde det gå att tydligare visa vilka som beror på varandra och därmed kunna slutföra dem i rätt ordning. Ett sätt att göra detta kan vara att lägga dem i en viss ordning i backlogen eller att göra ett separat flödesschema som visar hur de hänger ihop. Olika personers förkunskaper bör i framtiden tas i beaktande när efforts bestäms och även i de kontinuerliga utvärderingarna och uppdateringarna av efforts som görs. I sprint retrospective bör mer fokus läggas på konkretisering så att man alltid får med sig nyttiga tankar som är lätta att gå tillbaka till vid senare utvärderingar.*

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

A: *I inledningen av projektet skrev vi ihop en definition of done för alla olika backlogs. För att få flytta ett kort till "Testing" var följande kriterier tvungna att vara uppfyllda:*

- *User stories ska uppfylla alla acceptance criteria*
- *Alla tasks på user story:n ska vara gjorda*
- *Programmet går att kompilera*

Detta utfördes av den eller de gruppmedlemmar som tagit på sig user storyn. Acceptanskriterierna och tasks:en var lite ihopflytande och inte tydligt skilda från varandra, mestadels för att vi inte var helt säkra på skillnaden. Vi resonerade så att om alla tasks är gjorda uppfylls automatiskt acceptanskriterierna. Detta fungerade bra, och vi hade inga stora problem att se ifall en user story var klar eller inte. För att få flytta ett kort från "Testing" till "Done" var dessa kriterier tvungna att vara uppfyllda:

- *Alla tidigare DoD:s ska vara uppfyllda*
- *Åtminstone en annan gruppmedlem ska ha testat din kod*
- *Koden är avlusad*
- *Alla edge cases är testade*

Det viktigaste med vår formulering var att vi ville kvalitetssäkra genom att låta en annan gruppmedlem testa koden. På så vis kunde vi se om koden var tillräckligt intuitiv för att testas av någon som inte skrivit den. Dessutom tänkte vi att det är lättare för någon annan än författaren att kritiskt granska en kod och hitta fel. Angående avlusningen hänger det ihop med tidigare kriterier om kompilering. Om programmet inte kan kompilera är det svårt att testa koden. Har vi inga kompileringsfel är det däremot lättare att avlusa koden. Det vi missade lite på med våra formuleringar var att alltid låta någon annan testa koden. I många fall testades den av författaren för att sedan pushas upp till master-branschen. Senare testades den av andra i gruppen som en del i det vanliga arbetet med appen. Vi försökte gå igenom appen lite grann och demonstrera det vi gjort för handledare under måndagarna, vilket gjorde att hela appen blev testad så småningom men kanske inte nödvändigtvis innan kortet flyttats till "Done". Ofta berodde detta på att det gick snabbare och lättare för författaren till koden att testa den, varpå

vi missade vår viktigaste kvalitetssäkring. Vi lade inte så mycket tid som vi kunde gjort på testningen, vilket har lett till en del buggar som fått åtgärdas i efterhand.

B: *Till nästa projekt tar vi med oss att strukturera testerna bättre. Vi vill ha hunnit testa alla edge cases innan vi visar upp applikationen för någon för att vara säkra på att den fungerar som vi vill. Vi tar också med oss att lägga mer tid på tester istället för att alltid pusha framåt. Vi tog faktiskt en sprint till att arbeta på mer bugfix och back-end, men kommentarerna vi fick från handledare då var att vi borde alltid kunna visa upp ny funktionalitet och visa att projektet gått framåt. Denna attityden har också funnits lite inom projektgruppen, varför testerna hamnat lite i skymundan. Till nästa projekt tror vi därför att det är viktigt att alltid ta sig tid till testerna, oavsett handledares rekommendationer. Det känns bättre att leverera lite mindre, men stabila, delar än fler och instabila. Detta speciellt om vi i nästa projekt faktiskt ska leverera till en "riktig" produktägare. Hade detta varit fallet i vårt projekt hade vi nog varit mer noggranna med alla tester innan vi lämnade bort produkten.*

A → B: *Genom att lägga mer tid på testning istället för att hela tiden pusha framåt i projektet hade vi nog kunnat undvika en del onödiga buggar, som vi nu fått åtgärda i efterhand. En viktig del i att utföra välfungerande tester i framtiden är att strukturera upp vad som ska testas och skapa ett ramverk, exempelvis med checklistor för att se till att alla tester utförs på liknande sätt. Vi hade kunnat använda oss av färdigskrivna metoder som testade programmet åt oss, eller haft en testansvarig som såg till att alla tester blivit gjorda under våra sprint retrospective-möten. Det kan också vara en bra idé att förklara vilket värde testerna ger både produktägaren och användarna. Om alla förstår vikten av att utföra tester och buggfixar blir det automatiskt motiverat att lägga tid på det.*

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

A: *En KPI (härefter KPI 1) vi använde var att under varje sprint självvärdera vår motivationsnivå, stressnivå och job satisfaction på en skala från 1 till 5. Genomsnittet i varje kategori användes för att mäta stämningen i gruppen. Vid mindre motivationsnivå såg vi till att höja den genom att arbeta mer tillsammans, medan vi vid höga motivationsnivåer kunde arbeta mer självständigt. Ofta sammanföll också hög job satisfaction med hög motivationsnivå. Stressmomentet var viktigt för att mäta hur mycket press vi lade på oss själva och gruppen, vilket vi följde upp genom att utreda om stressen berodde på detta projekt, andra kurser eller privatliv. Vi kunde också ta på oss mindre de veckor där stressnivåerna var högre. Den andra KPI:n (härefter KPI 2) vi använde oss av var antal avklarade user stories per sprint. Denna var tänkt att ses som en motivation för oss själva att pusha oss lite mer varje sprint. Den tredje KPI:n handlade i början om hur många buggar vi hittat och åtgärdat varje sprint. Denna visade sig dock svår att använda och att få något värde av, varför vi (runt sprint 4) bytte denna till att istället handla om hur många efforts som låg i vardera backlog vid sprintarnas början och slut (härefter KPI 3). Detta gjorde i sin tur att KPI 2 blev mindre användbar, eftersom det kändes mer relevant att mäta efforts mot velocity än bara hur många separata user stories vi slutfört.*

B: *Vi drar slutsatsen att KPI 1 var användbar och dessutom intressant att se över tid. Dock är det inte klockrent att alla gruppmedlemmar ser varandras svar, eftersom det kan leda till att man inte svarar helt sanningsenligt, utan mer som resten har svarat. Detta tar vi med oss till nästa gång. KPI 2 och 3 borde ha slagits ihop, men eftersom instruktionerna var att ha tre KPI:er och vi redan bytt ut en kändes det bättre att köra på för att få rimlig data från iallafall två stycken KPI:er, istället för att byta ut två. Hade vi slagit ihop dessa tidigare hade vi kunnat införa en KPI som behandlade varje individs velocity per vecka, exempelvis genom att fylla i hur stor effort:en var på de user stories man slutfört per sprint. Data på detta hade kunnat användas för att avhjälpa problemet som beskrevs tidigare om olika förkunskaper och inlärningskurvor. Till nästa gång vill vi utvärdera användandet av KPI:erna tidigare i projektets gång, för att upptäcka brister i dem.*

A → B: *Vid ett längre projekt är KPI 1 särskilt viktig eftersom att den visar på hur varje person mår. En reflektion vi har är att vi eventuellt borde gjort denna anonym för alla utom projektledaren, för att få mer ärliga svar. Detta skulle i så fall leda till att ansvaret för gruppens välmående hamnar mer på projektledaren, vilket är rimligt men inte optimalt; hela gruppen bör ju bidra till allas trevnad. Dock kan det kännas bättre att endast prata om eventuella personliga problem med en ledare istället för hela gruppen. Angående KPI 2 och 3 kan det, i nästa projekt, vara en poäng att utvärdera KPI:erna tidigare än sprint fyra för att se om de bidrar till arbetet i gruppen. Detta skulle göra att vi kunnat få mer tillförlitlig data eftersom de använts under en ännu längre tid. Vi hade därmed också kunnat införa en ny KPI tidigare som dessutom hade kunnat bidra till gruppens utveckling.*

Figur 1. Urklipp från KPI 1

190404-190411	**Mejborn**	**Nils**	**Baum**	**Axel**	**Alexander**	**Hassan**	**Ludwig**	Genomsnitt
Motivationsnivå	4	5	4	4	3	5	4	4,14
Stressnivå	4	5	4	4	4	4	4	4,14
Job satisfaction	2	4	4	3	3	4	4	3,43

5	Bäst
4	Bättre
3	Lagom
2	Sämre
1	Sämst

190411-190418	**Mejborn**	**Nils**	**Baum**	**Axel**	**Alexander**	**Hassan**	**Ludwig**	Genomsnitt
Motivationsnivå	5	5	4	4	4	5	5	4,57
Stressnivå	3	3	3	3	3	3	4	3,14
Job satisfaction	4	2	4	4	4	3	3	3,43

190418-190425	**Mejborn**	**Nils**	**Baum**	**Axel**	**Alexander**	**Hassan**	**Ludwig**	Genomsnitt
Motivationsnivå	5	5	3	3	4	5	3	4,00
Stressnivå	4	5	5	4	3	3	4	4,00
Job satisfaction	2	2	2	4	5	2	5	3,14

190425-190502	**Mejborn**	**Nils**	**Baum**	**Axel**	**Alexander**	**Hassan**	**Ludwig**	Genomsnitt
Motivationsnivå	5	5	2	3	4	5	3	3,86
Stressnivå	2	3	5	4	2	2	4	3,14
Job satisfaction	4	4	2	2	3	3	3	3,00

Figur 2. Urklipp från KPI 2.

Sprint #	Mål för veckan	Avklarade	
190404-190411	4	3	
190411-190418	12	5	
190418-190425	8	4	vi råkade glömma att räkna våra user stories innan vi flyttade på dem så vi fick göra en "rough estimate"
190425-190502	11	8	
190502-190509	8	6	
190509-190516	9	1	
190516-190523	5	4	
190523-190530	-	-	

Figur 3. Urklipp från KPI 3.

Sprint #	Kvar sen förra sprinten	Velocity	Sprint backlog i början av veckan	Kvar i sprint backlog	Kvar i in progress	Kvar i testing	Done	Sprintens mål	Kommentar
190425-190502		100		0	25	10	107	142	
190502-190509	35	100	95	0	35	0	48	83	
190509-190516	25	100	108	10	85	0	13	98	
190516-190523	95	100	95	0	20	0	75	95	
190523-190530	-	-	-	-	20	-	-	-	Allmänt fix med final reflection, ingen effort sattes

2. Social Contract and Effort

2.1. The rules that define how you work together as a team, how it influenced your work, and how it evolved during the project. The time you have spent on the course and how it relates to what you delivered.

A: *Vi införde tidigt i projektet ett socialt kontrakt. Den huvudsakliga anledningen var att det efterfrågades av handledarna. Eftersom vi var från olika program och kom från två kompisgäng var det positivt att utforma kontraktet, eller åtminstone diskutera ambitionsnivå, hur vi förhåller oss till möten och liknande scenarier. Ju längre projektet fortlöpte desto mer positivt inställda blev vi till det sociala kontraktet. Det är en bra plattform att utgå från i början av ett projekt, eftersom det blir en anledning att adressera scenarier som lätt blir irritationsmoment, exempelvis vad som händer vid sena ankomster, om gruppmedlemmar inte meddelar frånvaro från möte i tid och liknande. Nu fick vi tidigt istället möjlighet att definiera ramverket vilket gjorde att vi fick lära känna varandra och använda det sociala kontraktet så att alla började på samma sida. Från tidsrapporteringen har det framkommit att gruppen lagt ungefär 800 timmar på projektet från start till slut. Det är såklart svårt att veta huruvida det är tillräckligt med tid eftersom vi ändå lyckades leverera något som vi är stolta över. Vidare kan också sammanfattas att det borde vara så att vi hunnit med att skapa mer om vi lagt ner tid men vi har samtidigt lagt kontinuerligt med tid och sett till att vi kommit någonstans varje vecka vilket är positivt.*

B: *I framtida projekt är ett socialt kontrakt ett bra verktyg för att ena gruppen kring vilka regler som gäller. Vi tror inte nödvändigtvis att det behöver heta "socialt kontrakt" utan riktlinjer skulle fungera alldeles utmärkt. Det viktigaste är att gruppen tar sig tid att noggrant diskutera igenom vad som förväntas av varje gruppmedlem i form av kommunikation, närvaro och liknande. En annan viktig aspekt med ett gruppkontrakt är att kontinuerligt utvärdera och uppdatera kontraktet efter hur projektet fortlöper. Det är inte rimligt att gruppkontraktet blir helt rätt från början eftersom det kan uppkomma scenarier under projektets gång som måste adresseras i takt med att de uppkommer. Därför är det av yttersta vikt att gruppkontraktet är dynamiskt och utvecklas kontinuerligt. I framtida projekt borde man sätta upp en plan för hur ofta gruppkontraktet bör utvärderas för att skapa den dynamik som krävs för att kontraktet ska användas så effektivt som möjligt. I detta projekt användes vidare tidsrapportering för att hålla koll på den tid som varje gruppmedlem lade ner på projektet. Det finns många goda anledningar till detta, en av dem är att varje gruppmedlem blir lite mer engagerad och mån om att lägga ner tid på projektet när denne ser att resten av medlemmarna har lagt ned tid. Vidare är det positivt att allokera tid på förhand till projektet och mäta denna. Många gånger kan det vara så att man kanske känner att man lägger ned mycket tid på projektet, men egentligen har man kanske suttit och gjort andra, icke-värdeskapande aktiviteter samtidigt vilket minskar effektiviteten. Det blir alltså bättre fokus på uppgiften och det blir enklare att planera sin vecka. Sammanfattningsvis bör tilläggas att det är viktigt att ha en bild av hur många timmar man vill lägga på projektet varje vecka för att se till att projektet fortlöper i den takt som eftersträvas, oavsett om man väljer att använda tidsrapportering som*

dokumentationsformat eller inte. Detta bygger förväntningar från gruppen och vidare vore det inte fel att lägga till en sådan klausul i det sociala kontraktet ifall det är så att gruppen vill ha en tydligt definierad nivå.

A → B: *För att i framtida projekt uppnå ett optimalt socialt kontrakt bör gruppen ta tag i kontraktet tidigt, ungefär som vi gjorde i detta projekt. Som vi varit inne på tidigare är det inte alltid så att det mest optimala är ett skriftligt kontrakt utan att det troligen gått att uppnå liknande konsensus genom att skapa riktlinjer alternativt endast utföra diskussionen muntligt i gruppen. Det viktiga är dock att en diskussion förs i gruppen gällande sen ankomst, mötesnärvaro och eventuellt också hur mycket tid varje gruppmedlem förväntas planera in varje vecka. Vidare bör gruppen också diskutera hur man vill mäta den tid varje gruppmedlem lägger på projektet och hur detta bör följas upp. En annan aspekt är uppföljning och kontinuitet. Det sociala kontraktet bör med fördel vara på mötesagendan minst varannan vecka under projektets gång för att se till att kontraktet är uppdaterad med utvecklingen gruppen har.*

3. Design decisions and product structure

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

Java & Android

A: *En av applikationens värdegrunder är att den ska vara enkel för användare att komma åt och ta med sig överallt, vilket gjorde det självklart att utveckla den för mobila enheter. Eftersom alla gruppmedlemmar var mest bekväma med att använda Java som huvudsakligt utvecklingsspråk togs beslutat att initialt utveckla applikationen för plattformen Android.*

B: *Ett framtida mål är givetvis att den ska finnas tillgänglig för alla användare, oavsett vilken plattform deras mobila enhet använder. Därför vill vi såklart även utveckla applikationens motsvarighet för iOS, men det är tyvärr utanför projektets scope.*

A → B: *I ett framtida projekt hade detta däremot kunnat enkelt realiseras enligt den tekniska beskrivning som används för Android-versionen. Med det färdig är det endast att implementera applikationen likadant men enligt den standard iOS använder. Enda hindret här är att hitta utvecklare som är kunniga inom att utveckla iOS.*

Model-view-viewmodel

A: *Applikations bakomliggande arkitektur är byggt enligt en tolkning av designmönstret Model-View-Viewmodel (MVVM). I den tolkningen består modellen av ett självständig modul och views av diverse xml-filer. Dessa sammankopplas av viewmodels bestående av aktiviteter och fragment, som även kommunicerar med databasen med hjälp av ett services-paket. Att applikationen utvecklats enligt MVVM stöder inte explicit något användarvärde. Däremot är mönstret vanligt förekommande och väl fungerade gentemot utveckling mot Android. Detta tillåter användare att enkelt vidareutveckla applikationen med ny funktionalitet och enkelt anpassa den efter alla krav som användarbasen kan tänkas ha. En användare kan därför förlita sig på att applikationen följer strömmen efter hur kraven ändras, vilket indirekt stödjer dess användarvärde.*

B: *Eftersom applikationen enkelt kan vidareutvecklas med MVVM som bakomliggande designmönster, är det ett hållbart alternativ även i framtiden. Det hade däremot kunnat ske bättre planerat och på ett mer enhetligt sätt.*

A → B: *Med en tydlig planering av övergripande aktiviteter i viewmodel, och hur datan ska kommuniceras mellan dessa hade resulterat i en mer förståelig design som enklare kunnat byggas ut. Viewmodel kan likaså delas upp internt, vilket hade ökat dess abstraktionsnivå.*

Google Firebase

A: *En bakomliggande grundpelare för applikations användarvärde är en väl fungerande databas. Detta realiserad med Google Firebase, vilket är en plattform som utlovar en databas med enkel auktorisering, snabb åtkomst och säker lagring. Detta passar applikationens syfte perfekt, då en effektiv applikation som kan snabbt hämta och visa upp data för användaren starkt stöder dess användarvärde. Likaså kan en användare enkelt registrera sig med ett telefonnummer och uppleva att dess data ligger säkert förvarad, vilket är viktigt för många användare. Att just telefonnummer används som metod för auktorisering är för att det stöder applikationen värdegrund gällande mobilitet, samt att minska risken att en användare skapar flera konton.*

B: *Användandet av Firebase har i sig fungerat mycket bra, och ett bra alternativ att använda även i framtiden. Däremot har utvecklingsprocessen med databas-kommunikationen haft återkommande problem, med många nästlade user stories och bristande kommunikation mellan gruppmedlemmar om logistiken gällande dess implementation. Exempelvis vilka fält som finns sparade och hur dessa ska hämtas.*

A → B: *Implementation ska framförallt följa "slicing the cake" på ett bättre sätt där user stories kan implementeras oberoende av varandra. Gällande databas-kommunikationen kan det göras genom att abstrahera det direkta beroendet till kommunikationen genom att en utvecklare kan anta att den är färdig men tillfälligt hårdkoda datan den behöver, för att sedan "koppla på" den riktiga kommunikationen när den är slutförd. Detta samtidigt som man för en aktiv dialog med utvecklaren som parallellt implementerar kommunikation så att ens egna kod är korrekt formaterad för att enkelt kunna koppla på kommunikationen när den är slutförd. Givetvis kan inte direkta beroenden mellan user stories undvikas helt och hållet, men det kan minimeras genom välplanerade stories och tydligt strukturerade objekt kring den data som ska användas.*

Google Maps

A: *Applikationens kanske viktigaste funktionalitet och en grundpelare för vad som gör den unik, är dess karta. Större delen av en användares tid inne i applikationen läggs på kartan, och en stor del av dess användarvärde ligger därför på att kartans funktionalitet ska fungera bra. Som skal till kartan användes därför Google Maps API för att utgå från en redan fungerande och robust grund. API:t används sedan för att realisera en utbyggd karta där användare kan leta, hitta, och visa MeetUp:s och vänner, vilket är hela grundidén med applikationen.*

B: *Google Maps funktionalitet är i stort sett all funktionalitet som kartan behöver ha tillgång till enligt dess nuvarande specifikation. Därför hade den använts även i framtiden, ifall kartans specifikation inte förändras, och utvecklas därefter.*

A → B: *Kartan hade fortsatt vidareutvecklas så att den upplevs mer som MeetApp än som Google Maps. Detta hade realiserats genom att komma överens om en enhetligt design som stämmer med applikationens övergripande känsla, och implementerat det därefter. Google Maps har även stöd för att göra detta.*

QR-koder

A: *En viktig del av applikationens grundidé är att främja att användare ska träffa nya människor. Vi ville därför att användare ska uppleva att deras vänner i MeetApp faktiskt är riktiga personer och inte bara "vänner över internet". Lösningen till detta blev att användare endast kan lägga till vänner genom att skanna deras QR-koder när de träffas. Detta anser vi göra att användare faktiskt kommer hemifrån och träffar personer i verkliga livet, istället för att endast sitta hemma. Denna lösning realiseras med ett API för att skanna QR-koder.*

B: *Så länge ovanstående värdegrund är aktuell för applikationen är QR-koder en bra lösning enligt oss. Den hade därför funnits kvar i framtiden och vidareutvecklas. I nuläget är läsaren simpel och QR-koderna ganska neutrala. Vid vidareutveckling av applikationen skulle målet vara att hela tjänsten blev enhetlig. Detta för att skapa en mer familjär och professionell känsla.*

A → B: *För att skapa en mer familjär och professionell känsla i appen kan läsaren och koderna anpassas till applikationens design och på så sätt ge en mer personlig känsla,, likt Snapchats QR-koder, se figur 4. Beroende på hur detta realiseras kan API:t vara överflödigt, och koden behöva göras på egen hand.*

Figur 4 - Exempel på Snapchats QR-kod

Bitmoji

A: *För att ge applikationens användare en personlig touch i gjordes ett försök att tillåta dem koppla en Bitmoji till sin profil. Detta hade gjort att användare upplevts vara mer levande och*

på så sätt gjorde applikationen mer hemtrevlig. Att implementera kommunikation med dess API var däremot fyllt av problem. Efter att det även uppdagades att användarna ansåg att detta hade en låg prioritet togs beslutet att lägga vidareutvecklingen av Bitmoji tillfälligt på is.

B: *Ifall ett intresse hos målgruppen hade funnits hade vidareutvecklingen av Bitmoji återupptagits och förhoppningsvis färdigställts. Däremot visar den användarundersökning som genomfördes att intresset inte finns i nuläget, och Bitmoji kanske därför inte färdigställs alls.*

A → B: *Vi ska även undvika att göra samma misstag två gånger, det vill säga att lägga onödig tid på ett problem med låg prioritet. Hade en liknande situation inträffat i framtiden hade vi därför utvärderat hur viktigt problemet är att lösa, tydligt bestämma vilka som fokuserar på en lösning, och sedan ge det en tidsram på hur mycket tid och ork som är värd att lägga på att lösa problemet.*

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

A: *I detta projekt har vi använt oss av Trello, Google Drive, Tidsrapportering, KPI:er i Google Sheets, Arkitektur, Databas-layout och GUI-layout för att strukturera vårt arbete. Användningen av samtliga av dessa verktyg har fungerat bra och vi är i stort nöjda med hur dessa har hjälpt oss komma framåt med projektet. För att kommentera den faktiska kod vi producerat har vi använt oss av Javadoc. Dock kan vi konstatera att utnyttjandet av detta sätt att kommentera kod har varit högst sporadiskt och väldigt beroende på vem som kodar. För IT-studenterna där det tjatats om att man ska använda Javadoc i alla tidigare utvecklingskurser tyckte de det var skönt att för en gångs skull inte vara tvungna att använda sig av Javadoc. Vad gäller I-studenterna har de aldrig blivit utbildade i hur Javadoc bör användas och därför har de inte haft det i åtanke överhuvudtaget under projektets gång. Detta har lett till att vi i vissa klasser varit väldigt duktiga på att kommentera koden men på vissa ställen har detta inte skett överhuvudtaget.*

B: *I framtida projekt tar samtliga i gruppen med sig vikten av att kommentera kod med verktyg, likt Javadoc. Det har blivit en bra påminnelse om varför det är viktigt och nu kan vi konstatera att ältandet från tidigare examinatorer på kurser är klart befogat. Också I-studenterna tar med sig vikten av att lära sig kommentera kod med Javadoc, återigen handlar det om enkel kommunikation som möjliggör ett effektivt arbetssätt. Kommentering av kod möjliggör enklare testning av kod, något vi tidigare skrivit om och dessutom varit dåliga på i detta projekt. Vad gäller de andra verktyg vi använt oss av för att dokumentera våra framsteg (och givetvis utmaningar) har fungerat bra och vi tar med oss att det är viktigt att ha all dokumentation på servrar så att samtliga gruppmedlemmar kan tillgodogöra sig all information och alla beslut som tas på ett effektivt sätt.*

A → B: *Det är värt att lägga tid i början av projektet att lära upp dem i gruppen som inte har någon erfarenhet av kommentering av kod, antingen genom självstudier, lathundar alternativt*

en genomgång. Det är inte komplicerat att lära sig men det inte heller något intuitivt som man kommer på själv i alla lägen. För att möjliggöra god dokumentation genom hela projektet tar vi med oss att ha alla dokument tillgängliga på en server så att den kan användas av alla i gruppen, oavsett var man befinner sig. Genom att använda JavaDoc får alla i projektet en god överblick över vad de olika klasserna och metoderna gör vilket är den primära anledningen till att använda metoden.

3.3. How you use and update your documentation throughout the sprints.

A: *För att hålla koll på mötesanteckningar och utvärderingar av varje sprint har vi använt Google Drive, vilket har fungerat smärtfritt. Drive är också ett system som alla var familjära med innan projektet och det krävdes därför ingen inlärning. Att alla alltid har tillgång till vad som sagts på varje möte ger många fördelar. Dels spelar det ingen roll om alla närvarat på alla möten, eftersom man kan ta del av anteckningarna i efterhand, och dels kan man lätt återgå till vad som sagts vid senare diskussioner om scope eller utvecklingsområden. Ytterligare en fördel är att vi kunnat arbeta parallellt med inlämningsuppgifter och deliverables. Vi har även använt oss av Trello för att visualisera våra backlogs. Även detta har fungerat bra, eftersom vi i inledningen av projektet definierade tydliga definition of done som sade när kort skulle flyttas från en kategori till en annan. Det enda negativa var att alla gruppmedlemmar inte hade använt sig av Trello innan, vilket gjorde att inlärning tog lite tid. Dock är verktyget väldigt intuitivt, och tiden det tog för inlärning var en liten kostnad i utbyte mot fördelarna. En annan dokumentation vi använt oss av är tidsrapportering. Detta för att mäta hur mycket tid varje person lagt på projektet. För att reflektera kring vad som gick bra respektive sämre med tidsrapporteringen i detta projekt är en aspekt att vissa av gruppmedlemmarna var dåliga på att fylla i sina timmar. Detta leder till att det blir svårare att använda materialet i gruppen och systemet urholkas. Vad gäller hur andelen nedlagd tid speglar hur mycket varje gruppmedlem lyckats leverera till projektet är detta svårt att säga. Vi har konstaterat att vi kommer att ha lagt runt 800 timmar på projektet vilket är rimligt. Men eftersom vi gick in i projektet med olika förkunskaper av programmering är det också rimligt att IT-studenternas timmar på utveckling har lett till mer rader kod. I-studenterna har gjort en stark utveckling och framförallt själva känt att de utvecklats vilket är positivt.*

B: *Till nästa projekt tar vi med oss att Google Drive är en bra plattform för att dela anteckningar och inlämningsuppgifter. Dessutom är det tidseffektivt att kunna arbeta parallellt. Den enda nackdelen vi märkt med att använda Drive är att GitHub, som vi använt för kodandet, har liknande funktioner för att dela dokument mellan grupper. Det kan då vara krångligt att använda sig av två olika verktyg som har liknande funktioner. Å andra sidan är GitHub ytterligare ett verktyg som inte alla i gruppen använt sig av tidigare, vilket gör att fördelarna med Drive övervinner den nackdelen. Dessutom kan vi då särskilja dokument som är under arbete och därmed ligger i Drive, från de som är inlämnade och laddats upp på GitHub.*

Även Trello har varit ett värdefullt verktyg för dokumentation. Scrum-boards i allmänhet är någonting vi alla vill använda i senare projekt eftersom att de ger en överblick på hur arbetet fortlöper. Det blir lätt att urskilja framsteg och tillkortakommanden, vilket är viktigt för att kunna identifiera framgångsfaktorer. Konceptet är intuitivt och lätt att använda.

Trots att tidsrapporteringen lämnade övrigt att önska finner ändå gruppen detta som ett bra verktyg för att skapa en välfungerande miljö för att projektet ska fortlöpa på ett produktivt sätt. Som sagt är inte tidsrapporteringen i klockren korrelation till produktivitet, men själva dokumenteringen gör att varje gruppmedlem kan ta ansvar för att planera sin egen tid.

A → B: *Vi är väldigt nöjda med våra olika verktyg för dokumentation under projektet och slutsatsen blir att vi bör arbeta på liknande sätt i senare projekt. För att kunna göra detta krävs att vi tar med oss de positiva aspekterna av våra dokumentationsmetoder och hur de skapat värde för oss utvecklare genom projektet. Vi hade kunnat använda fler verktyg och metoder, men vi ansåg att dessa var tillräckliga och hade den funktionalitet vi behövde, varför det känns onödigt att ändra för mycket i ett koncept som uppenbarligen gruppen är nöjd med och som vi ser fungerar. Angående tidsrapporteringen är det viktigt att vi i nästa projekt tar oss tid att följa upp huruvida alla fyllt i sina delar, eventuellt på varje sprint retrospective.*

3.4. How you ensure code quality and enforce coding standards

A: *Kodens generella kvalitet har under projektets gång bibehållits genom det förutbestämda krav i Definition of Done som funnits gällande godkännande av kod. Kravet definieras som att en annan gruppmedlem snabbt ska ha läst igenom din implementerade kod, och godkänt att den håller en tillräckligt hög kvalitet och följer kodstandarder relativt väl innan den tolkas som "klar". På grund av projektets scope har detta endast följts till viss del då vår huvudsakliga prioritering varit att koden hellre ska fungera än att hålla en god kvalitet och formatering. Likaså ligger det ett större användarvärde i att ha färdig funktionalitet i applikationen än att den ska vara snyggt gjord bakom kulisserna. Detta har i många fall gjort att kodens kvalitet har åsidosatts för att hellre hinna färdigställa funktionalitet, och på så sätt stärka dess användarvärde.*

För att den aktiva huvudgrenen på GitHub ska vara välfungerande och hålla en bra kvalitet har även de merge conflicts som uppstått blivit överlåtna till gruppmedlemmar som känt sig kapabla att lösa dessa korrekt. Detta har garanterat att huvudgrenen inte tappar någon funktionalitet som ska finnas där, och i de flesta fall fungerar korrekt.

För att kunna koda parallellt har vi använt oss av Git. Detta har fungerat bra överlag, men vissa problem har uppstått längs projektets gång. Det främsta problemet var bristande förkunskaper från I-studenterna. Trots en gedigen genomgång i hur programmet fungerar uppstår fortfarande vissa oklarheter kring hur man gör om exempelvis en "merge-conflict" uppstår.

B: *I framtida projekt där en potentiell applikation troligen har ett mer långsiktigt scope och ett mål att kunna bibehållas av framtida utvecklare kommer kvaliteten på dess bakomliggande*

kod vara av större vikt. Med det menat vill vi ha en kod som följer dess platformsstandard, exempelvis Javas kodstandard, och även är lättläst, återanvändbar och enkel att felsöka.

En utmaning vi upptäckt på Git är att alla gruppmedlemmar inte kände sig säkra i programvaran och hur man hanterade merge-conflict. Mål till nästa projekt är således att alla gruppmedlemmar ska känna sig säkra i användandet av Git, vilket innebär att alla känner sig säkra med att pusha, merga och hantera konflikter.

A → B: *För att säkerhetsställa att kraven för god standard skulle man, samtidigt som någon blir tilldelad en user story, också tilldela en annan person granskningsansvar över koden. Vidare är det viktigt att kontinuerligt utvärdera hur det fungerar att testa varandras kod för att göra denna metod så optimal som möjligt. Detta gäller framförallt när teamet består av medlemmar med olika kunskaper gällande utveckling, vilket troligen alltid kommer vara fallet även i framtiden. Huvudmålet är att skapa en iterativ process så att även testningen av koden sker agilt på samma sätt som själva programmeringen av densamma. Med detta menar vi att testa koden under tiden denna programmeras så att all testning inte enbart sker när upphovspersonen anser sig ha färdigställt user storyn, eftersom det då tar längre tid och är mer komplext att testa koden.*

Inför framtida projekt kommer Git användas på samma sätt som i detta projekt. För att det ska fungera smidigare måste alla gruppmedlemmar känna sig säkra i programvaran, vilket fallet inte är idag. För att övervinna detta problem krävs att de personer som känner sig osäkra dels blir bättre på att koda för att på så sätt förstå varför en ”merge-conflict” uppstår och hur man löser den på ett säkert sätt. För att bli bättre på att programmera krävs då att man övar på att koda antingen via parprogrammering eller genom att googla sig fram.

4. Application of Scrum

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

A: *I början av detta projekt delade vi ut ansvarsområden till varje gruppmedlem. Dessa kunde vara till exempel Scrum-ansvarig, som såg till att läsa på om Scrum-metodiken och såg till att vi följde den, Trello-chef, som ansvarade för att alla kort skapades och flyttades i Trello som vi använde som sprint backlog, eller kodchef som ansvarade för att alla såg till att koden blev tydlig och lättläst. Dessa ansvarsområden var bra att ha inledningsvis, då de gav en känsla av ägandeskap över en del av projektet. Det blev tydligt vad man skulle sätta sig in i initialt. Med tiden användes de mindre och mindre allteftersom vi kom in i arbetet, alla tog snarare ansvar för allt än att någon hade ett enda område.*

I efterhand kan det sägas att rollerna vi valde åt oss själva inte var klockrena. Ett ansvarsområde var till exempel kommunikation med andra grupper och handledare, vilket aldrig utnyttjades. Vi insåg också att Trelloansvaret inte bara var onödigt, utan också kontraproduktivt, eftersom arbetet flöt på bättre om alla tog ansvar för att flytta sina egna user stories i vår backlog. User stories skapades tillsammans men tasks inom en vald user story kunde ändras av den som arbetade med den, eftersom det blev för krångligt om det bara var Trello-ansvarig som fick göra det. Planen med det från början var att en person alltid skulle ha överblick över alla kort som var i rörelse, men uppgiften blev för stor för en person och det blev dessutom mer effektivt om alla fick ansvar över sina egna user stories.

B: *Till nästa projekt tar vi med oss att rollerna är viktiga i inledningen av projektet för att komma igång. De ger en "power of ownership" över en egen del av projektet vilket gör att alla känner sig delaktiga. Det vi vill ändra på är själva utformningen av rollerna och vad de innebär. Vi tar också med oss att det är viktigt att utvärdera rollerna efterhand, så att känslan av delaktighet i projektet lever kvar. Får man en roll som senare visar sig överflödig är risken stor att ens engagemang i projektet dalar. Poängen med att arbeta agilt är att vara förberedd på hantering av förändringar i arbetstakt eller -sätt, vilket man löser genom kontinuerlig utvärdering och ifrågasättande så att ingenting ses som självklart.*

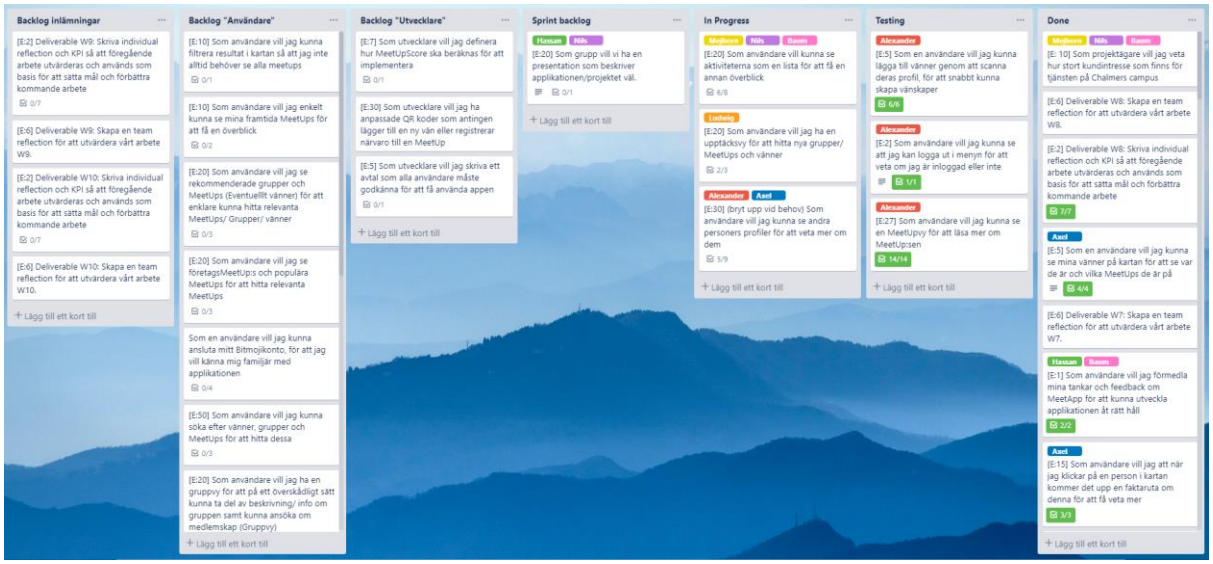
A → B: *För att nå en optimal rollfördelning i nästkommande projekt är det viktigt att snabbt försöka utreda vilka roller arbetet kräver och hur många gruppmedlemmar som finns att tillgå. Till en början kan det vara så att man inte behöver roller till samtliga i teamet men en agil approach bör eftersträvas där roller läggs till efterhand i takt med att projektet fortskrider. I senare projekt skulle man därför kunna införa en punkt på sprint retrospective att diskutera hur man använt sin roll under sprinten, precis som man utvärderar resten av arbetet. Man skulle samtidigt kunna diskutera övriga fält som kommit upp under sprintens gång som eventuellt behöver en ansvarig; behoven för gruppen kan ju vara annorlunda i början, mitten och slutet av projektet. För att detta ska kunna fungera måste det finnas en natur där gruppen kontinuerligt utvärderar och utreder om det behövs fler definierade roller som vi själva är inne på att vi skulle vilja ha i framtida projekt. På så sätt blir det naturligt att addera roller, om*

sådana behövs, efterhand och arbetssättet blir mer agilt. Givetvis kan det också vara att gruppen tidigt i projektet inser att projektet är komplext och att det därför behövs roller för samtliga i teamet på en gång och i sådana fall bör gruppen implementera en rollstruktur direkt. Sammanfattningsvis bör det påpekas att målet med roller är att arbetet ska förenklas och därför får roller aldrig finnas om de inte fyller någon funktion, därför måste funktionen av rollerna alltid tas i beaktning i första hand.

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

A: *Under projektets gång har flera olika agila arbetsmetoder och verktyg använts. För att tydligare visualisera Scrum-boarden har Trello använts. I Trello har flikarna ”Backlog användare”, ”Backlog utvecklare”, ”Backlog inlämningar”, ”Sprint Backlog”, ”In Progress”, ”Testing” och ”Done” funnits. För varje flik definierade vi ett ”Definition of Done”, alltså när ett kort får flyttas från en flik till en annan. Korten har bestått av en user-story, där vi i gruppen har bestämt en effort för varje kort, enligt traditionell Scrum-metodik. Vi har även markerat vem som utför vad på korten genom att olika etiketter representerar olika gruppmedlemmar. Vår velocity ändrades under projektets gång från 140 till 100. Den ursprungliga velocityn bestämdes genom att vi är 7 stycken gruppmedlemmar á 20 h/ veckan vilket blir att projektet totalt har 140 h/ vecka. Efter samtal med handledare var rekommendationen att velocityn ej ska vara bunden eller härledd utifrån timmar. Denna input gjorde att vi förändrade velocityn till 100, främst för att det blir enklare att räkna med. I Figur 5 finns ett exempel på hur vår Trello-board såg ut i mitten av projektets gång. Tack vare användandet av Trello har vi kunnat arbeta strukturerat och det har varit enkelt att få en överblick över vad som ska göras samt vem som gör vad. Vår tydligt definierade ”Definition of Done” har gjort att det inte uppstått någon förvirring kring när ett kort får flyttas och när en user-story anses vara klar. En annan positiv aspekt var att vi delade upp user-stories i olika flikar beroende på vem som får värde av den (användare, utvecklare etc.), vilket även har lett till bättre tydlighet. Det som dock fungerat mindre bra är specificeringen av user-stories och att acceptanskriterierna för vissa var otydliga. Det sammanfattande intrycket av användandet av Trello är positivt och vi kommer helt klart att använda oss av det i framtida projekt.*

Figur 5: Bild på vår Trello-board i mitten av projektet



En viktig aspekt i att arbeta agilt att möjligheten att kunna koda parallellt. Detta har möjliggjorts genom användandet av Git. Problematik kring användandet av detta har redan redogjorts under frågan "How you ensure code quality and enforce coding standards".

De metoder vi använt i projektet är "Daily Scrum"-möten, retrospective och sprint planning. Sprintarna har pågått torsdag till torsdag varför daily Scrum genomfördes på måndagar då arbete till största del skett på måndagar. Retrospective och sprint planning har genomförts på torsdagar vid sprintens början och slut. Dessa metoder har överlag fungerat bra, framförallt i slutskedet av projektet. I början var det svårt att förstå vikten och innebörden av retrospective och Daily Scrum. Detta klarnade dock under projektets gång då vi fick kontinuitet i det. Gällande sprint planning fungerade det överlag okej. Vi var dessvärre dåliga på att uppskatta efforts på user-stories vilket gjorde att vi inte fick klart så mycket som vi planerat varje sprint och därför fick "backorders" till veckan efter. En annan viktig insikt vi kom till under senare delar av projektet var att kontinuerligt utvärdera oss själva och effort på user stories. Vi lade störst fokus då sprinten inte gått som planerat och utvärderade dessa sprintar mer kritiskt och utförligt än när en sprint gått bra. Vid slutet av en bra sprint blev slutsatsen att "vi gör som vi gjort hittills", vilket gjorde att när en sämre sprint var genomförd visste vi inte vad vi gjort bra tidigare, varför det var svårt att hitta tillbaka till det.

***B:** En utmaning vi mötte var att våra user-stories var dåligt formulerade vilket skapade förvirring och misskommunikation. Inför nästa projekt är målet därför att ha tydligare definierade user-stories i Trellon.*

Då användandet av Scrum var nytt för många i gruppen förstod vissa i gruppen inte riktigt vikten av att utvärdera varje sprint. Detta gjorde att vi i synnerhet lade fokus på att utvärdera de sprintar som gått dåligt. Målet inför nästa projekt är därför att lägga större vikt på utvärdering.

Något som vi inte tänkte på under projektets gång var att vi endast hade ett "Daily-Scrum möte", detta var något vi tog för givet då vi endast träffades 2 gånger i veckan. Detta ledde till

att frågor och problem togs upp för sent och det enstaka veckoliga Daily-Scrum mötet kunde kännas överväldigande.

A → B: *De förbättringsområden som nämns ovan gällande user-stories är främst att tydligare specificera dessa, helst innan de ens läggs till i backlogen, men framförallt innan de läggs till i sprint-backlogen. Utöver detta anser vi att uppdelningen och strukturen på vår Trello har fungerat bra. För att inte uppleva samma problem i framtiden kommer därför mer tid att läggas på att specificera beskrivning och acceptanskriterier på varje user-story. Utöver detta borde man även se över dessa en extra gång under sprint planning då de läggs till i sprint-backlogen. Den främsta oklarheten har egentligen uppstått på grund av bristande kunskap i hur man bör formulera en user-story och detta har blivit bättre med tiden och förväntas även bli bättre i framtida projekt.*

För att använda tidigare nämnda metoder i framtida projekt är det viktigt att alla förstår innebörden av varje moment från start. Utöver detta är det viktigt att ha kontinuitet i användandet av metoderna. Exempelvis kan Daily Scrum kännas onödig i början av projektet, men vikten av det syns först i slutet av arbetet. En annan viktig lärdom är att kontinuerligt utvärdera alla sprintar på samma sätt, även de sprintar som gått bra även om det kanske inte känns nödvändigt för stunden. Detta för att konkretisera vad som gått bra och hur man ska fortsätta arbeta för att fortsatt arbete ska gå i samma stil.

Anledningen till att vi inte hade dagliga Scrum-möten var att vi kände att som studenter hade det tagit allt för mycket tid. Men faktum är att dessa möten endast skulle ta 5 minuter, vilket skulle göra det fullkomligt möjligt att ha ett 5-minuters möte om dagen. Detta är något vi tar med oss till framtida projekt, förutbestämda tider där vi antingen ses, men kanske mer realistiskt sett ringer varandra via Skype eller dylikt och uppdaterar varandra om hur det går i utvecklandet.

4.3. The sprint review and how it relates to your scope and customer value (in the first weeks in terms of the outcome of the current week's exercise; in later weeks in terms of your meetings with the product owner).

A: *I början av projektet fokuserade vi inte så mycket på kundvärde i sig, i alla fall inte medvetet. Föreläsningarna var tänkta att ge oss förutsättningar för att alltid ha med kundvärde i processen genom att använda oss av Scrum och varje sprint utvärdera hur vårt arbete relaterade till kunden. Detta fungerade bra, även om det som sagt tog emot i början att alltid tänka på vad slutkunden behövde, när vi hellre ville fokusera på det vi ville ha. Under arbetets gång kom vi dock in i det. Lego-övningen var värdefull eftersom att vi då för första gången provade på Scrum-metodiken i praktiken, och dessutom hade en produktägare att leverera till. Dock kändes den lite missvisande eftersom att den krävde en kommunikation med andra grupper på ett sätt som inte alls behövdes senare i projektet.*

Eftersom vi själva, som utvecklare och produktägare, också var potentiella användare till vår applikation tänkte vi oss att vi hade rätt bra koll på vad som skapade värde för användarna.

Vi bestämde mycket själva över utvecklingen, men efter enkäten framgick att vårt scope inte hade varit helt klockrent. Det var efter den sprinten när enkäten genomfördes som vi verkligen förstod hur vi skulle skapa värde för användarna och hur vi gjorde det på bästa sätt. Slutsatsen blev därför att vi borde utrett kundbehoven tidigare och utgått från fler personer än oss själva. Vi kan också konstatera att även om vi skapat kundvärde under de flesta av våra sprintar (enkäten visade att vi hade arbetat på delvis rätt saker) så visste vi inte om det förrän vi faktiskt genomfört en marknadsundersökning.

B: *Vi tar med oss att det är viktigt att tidigt lägga fokus på att utreda vad som skapar värde för att inte riskera att utveckla en app som ingen vill använda. Varje sprint bör skapa nytt värde för användarna och scopet bör justeras kontinuerligt. Vi tar också med oss att vi som utvecklare var lite för drivande i utvecklingen. Detta beror troligtvis på att kursen är uppbyggt lite som en sandlåda för studenter som vill prova på att driva ett IT-projekt, vilket vi alla uppskattade. Dock hade vår grupp nog tagit projektet på ännu större allvar om vi skulle levererat till en riktig kund. Till nästa projekt vill vi fokusera ännu mer på kundvärde och lära oss mer om hur vi kan utreda det. Vi kan också ta med oss att att vi i projektet var både produktägare och utvecklare, vilket troligtvis inte är fallet nästa gång. Vad vi än utvecklat under sprintarna har vi sett det som värde eftersom att det antingen gör det lättare för oss som utvecklare eller kommer närmare målet som produktägare. Det har därför varit svårt för oss att vara objektiva. Till nästa gång kommer vi troligtvis ha en extern produktägare som vill se nytt värde i appen efter varje sprint, vilket gör att vi måste lägga upp arbetet på ett sådant sätt att vi skapar värde varje sprint. Mötena blir därmed viktiga att hålla så att vi får ett fokus på kundvärdet.*

A → B: *För att undvika att som produktägare vara för drivande i utvecklingen är det viktigt att tidigt göra en marknadsundersökning som utreder vad som skapar värde för kunden. Denna bör sedan ligga till grund för fortsatt arbete, vilket gör att nytt värde kan skapas varje sprint. Dock finns en reflektion om vad som egentligen är kundvärde. Saker som underlättar för utvecklarna, tex strukturerade tester och välutvecklade user stories, skapar ju i längden värde för användarna i och med att utvecklarna blir mer motiverade och deras arbete roligare, vilket leder till en bättre produkt. Därför är frågan hur fördelningen mellan att skapa värde för utvecklarna och att skapa direkt värde för användarna bör se ut. Vi kan lösa detta genom att använda oss av KPI 1 som redogjorts för tidigare i rapporten. Att kontinuerligt utreda hur gruppen mår är en viktig faktor för att kunna skapa värde eftersom att ju bättre gruppen mår, desto högre produktivitet kan uppnås.*

För att kunna läsa oss mer om kundvärde är ett sätt att läsa på om kundundersökningar. I-studenterna har lite mer förkunskaper inom detta då många av kurserna på kandidatprogrammet behandlar detta, men det kan finnas mindre förkunskaper hos IT-studenterna. Därför är det viktigt att man som I:are tar mer ansvar för att utreda detta och att lära gruppen om det.

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

A: *Det verktyg som var nytt för samtliga medlemmar i gruppen var Scrum och användningen av denna projektmetod. Vi hanterade detta verktyg på ett bra sätt och nyckeln till det var att vi började köra sprintar så snart vi kunde istället för att vänta för länge innan vi börjar använda verktyget. Detta berodde också på den Lego-övning vi hade tidigt i projektet där vi tidigt fick en grundläggande förståelse för hur Scrum kan och bör användas. En best practice är alltså att våga börja använda verktyget så tidigt som möjligt och lära sig på vägen och under användning istället för att läsa på för mycket innan. En annan best practice är att vara noggrann med reflektion under tiden man använder verktyget, både om användningen fungerar bra men också om användningen fungerar dåligt. Det sistnämnda lyckades vi med men vi var sämre på att reflektera över användningen av Scrum de sprintar vi lyckades bra med.*

B: *I framtida projekt när det kommer till att använda nya verktyg tror vi att det är viktigt att applicera både "learning by doing" men också andra källor till kunskap, såsom experthjälp, litteratur och tutorial-videor. På så sätt lär man sig om olika aspekter som behöver tas i beaktning vid användningen av ett verktyg. Vi tror också kontinuitet är ett viktigt uttryck för att lyckas med implementationen av en ny metod. Om man bara använder exempelvis Scrum vid ett fåtal tillfällen för att man kanske tycker att det är svårt att hantera av olika anledningar kommer gruppen aldrig lyckas med implementationen.*

A → B: *För att lyckas med användningen av ett verktyg eller en arbetsmetod är kontinuitet, en blandning av "learning by doing" och lärande från andra källor, samt kontinuerlig utvärdering av verktyget. Dessa best practices är likadana för vilket alla verktyg eller metoder och detta är något vi kommer applicera för att lyckas med framtida användning av metod och verktyg.*

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

A: *Litteraturen hjälpte oss i början av projektet med att förstå vad Scrum är och hur metoden bör användas på ett optimalt sätt. Föreläsningarna av handledarna var också en bra introduktion och hjälpte oss att förstå hur vi skulle påbörja användningen av Scrum. Också den inledande Lego-övningen var användbar för att förstå konceptet. I takt med att projektet fortgick blev vi sämre på att ta del av extern litteratur och kunskap om Scrum även om vi, vid något tillfälle, läste på om Scrum på Stack Overflow.*

B: *I framtida projekt tror vi att det är en idé att tidigare i projektet läsa på lite på egen hand för att inlärningen ska gå snabbare. Det finns säkerligen en massa misstag vi gjorde som hade kunnat undvikas om vi varit bättre på att ta del av extern kunskap och framförallt om vi lagt mer tid på att läsa in oss på ämnet.*

A → B: *För att nå ett högre utnyttjande av extern litteratur i framtida projekt är det viktigt att teamet hjälps åt att samla in och tipsa varandra om relevant litteratur och kanske till och med skapa en mapp på servern med informationen som varje gruppmedlem får till uppgift att tillgodogöra sig varje sprint. Det kan också vara värt att ta en del av mötet varannan vecka till att diskutera om någon funnit någon relevant information som gruppen borde ta del av. Detta kan liknas vid kompetensutveckling vilket ofta prioriteras högt i företag och därför också borde bli en självklar del i sådana här framtida projekt.*