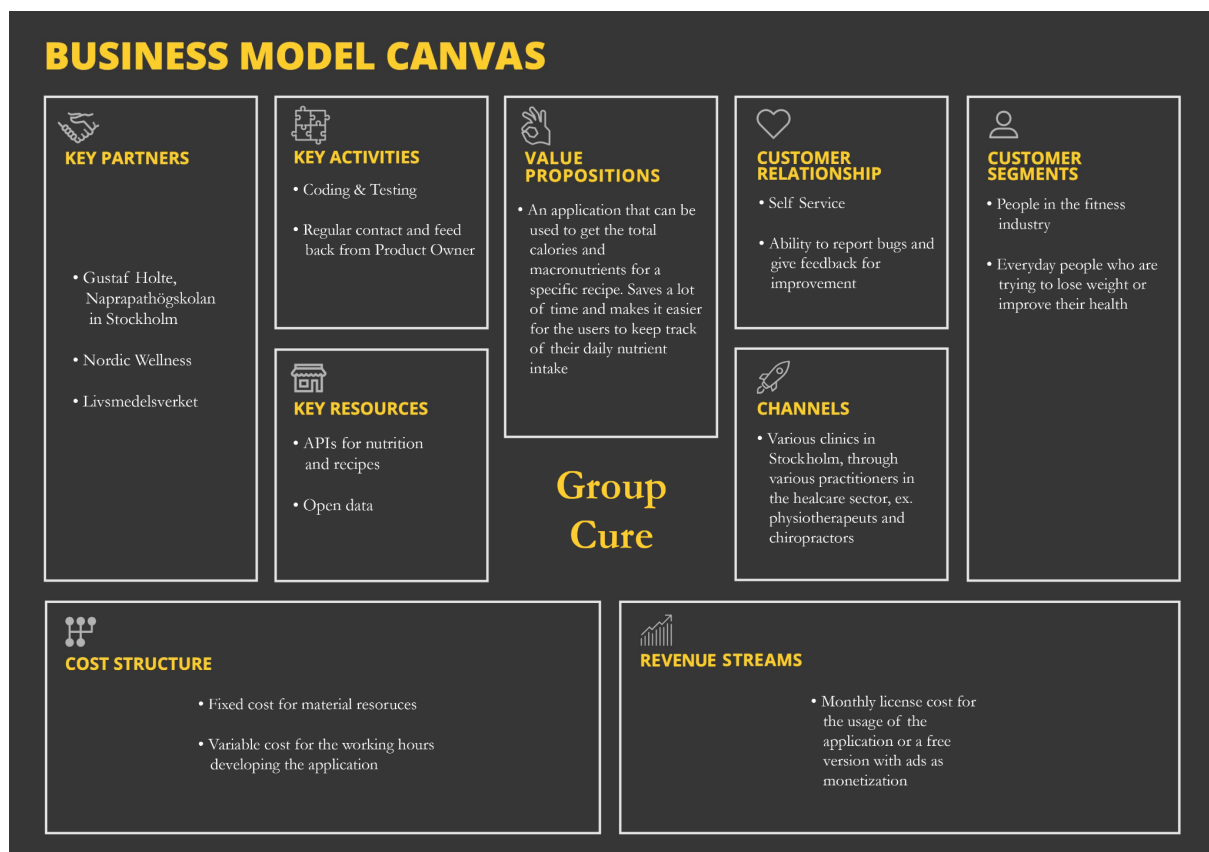


Final Report - The Cure

Abdullah Faez Daboul, Andreas Johansson, Ali Alkhaled, Ali Malla, Jakob Hanak

1. Introduktion

Vårt projekt har varit att utveckla en applikation som hjälper folk att spåra sitt dagliga intag av kalorier och makronutrientier för de måltider som de ätit. Skillnaden med denna applikation i jämförelse med tidigare, är att i denna får du automatiskt hela receptens näringsinnehåll istället för att du behöver lägga till varje ingrediens en och en. Det skulle fungera som en kombination av kokbok och loggbok för ens dagliga näringsintag. Idén kom upp efter att en naprapatstudent i Stockholm som nyligen öppnat egen klinik märkte att kunderna tyckte det var jobbigt och tidsödslande att lägga in varje enskild ingrediens för de måltider de ätit. Dessvärre är detta nästan ett måste om man ska kunna få en uppfattning om hur mycket mat man får i sig under en dag eller vecka. Därav hörde vi av oss till honom och frågade om han skulle vara intresserad av att vara en del i utvecklingen av en applikation som löser just detta problem, vilket han var. Efter ett initialt möte med honom och sedan gruppen togs business canvas model nedan fram av Product Owner (PO) och som vi sedan utgick ifrån under projektets gång.



2. Customer Value and Scope

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

(A)

Tidigt så brainstormade projektgruppen om vad för produkt vi skulle vilja utveckla. Vi ville utveckla för någon extern motpart och inte göra något utan en tydlig målgrupp. Det lades fram ett par förslag där en applikation för att hålla koll på näringsintag i syfte att spåra sin kost. En medlem i projektgruppen har en bekant som kunde ha nytta av en sådan funktion i sitt arbete som naprapat. Vi utsedde en PO som fick hålla kontakten med vår kund. Tack vare vår PO som tidigt höll möten med kunden så kunde vi göra en prototyp tidigt. Detta gav oss ett väldigt tydligt scope tycker vi. Efter uppstarten så var kunden inte lika närvarande utan kommunicerade till stor del genom vår PO och lämnade inte så mycket feedback förutom för de funktioner som angick kunden som användare. Därmed var extra funktioner ibland dåligt definierade.

(B)

I framtiden så skulle vi vilja se en tätare och mer involverad kontakt med kunden. Ett fortsatt tydligt scope, men med mer kundinflytande som dessutom sträcker sig till niche-funktioner. Att utveckla funktioner som berör användare utanför målgruppen skall naturligtvis undvikas, men att avsluta utan att undersöka om målgruppens behov har förändrats visar bara på att man inte har reflekterat en andra gång kring projektets resultat.

(C)

För att nå den situation vi önska att vi befunnit oss i så skall kunden involveras mer, men det skall då också framgå tydligare att det inte är för att vi “måste” ha fler möten eller mer involvering, utan för att det faktiskt är en värdeskapande funktion och en interaktion som är av nytta för resultatet. Detta kan vara svårt ibland, speciellt när man upplever en olikhet mellan teamet och kundens motivation och engagemang i projektet som helhet.

Trots detta gäller det att försöka hålla uppe viljan och engagemanget att involvera kunden, genom projektets PO så att teamet kan få feedback på produkten och hjälp att planera sina sprints.

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)

Våra mål med projektet sattes upp på vårt Scrum-board, mest för att ha en vision som teamet kunde översätta i mål under vår sprints. De var troligtvis inte skrivna eller kvantifierade så att de kunde användas riktigt till att mäta framgången, men de har sporrat teamet till att verka för bra stämning och god kvalitet i projektet.

(B)

I framtiden skall success criteria eller framgångskriterier användas för att kunna utvärdera resultatet i ett eller flera projekt, med utgångspunkt i hur projektgruppen/erna presterat. det skulle kunna vara genom att mäta ett antal faktorer i prestationen på ett eller flera projekt eller lag. På samma sätt som ett team använder sig av ett antal KPI. Exempelvis skulle man kunna mäta kundnöjdhet, features per dag eller teamets träffsäkerhet när det kommer till estimerat tidsbehov gentemot lagd tid.

C)

För att komma närmare framtiden så skall vi i kommande projekt ta fram ett antal kriterier, likt en Definition of Done, fast mer mätbara sådana, liknande KPI:er, för att kunna utvärdera prestationen i projektet. Ett projekt kan avslutas och en produkt kan levereras utan att projektet anses framgångsrikt. Till exempel genom att gå över tid eller över budget. Att hålla sig inom ramarna för projektets budget och tidplan, skulle kunna anses vara ett av de mest grundläggande framgångskriterierna.

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)

Teamet använde sig av mjukvaran Trello för att skriva upp och hålla koll på våra user stories. Varje user story var hade färgmarkeringar som visade vilka epics denna user story är kopplad till.

Det är lite oklart om det togs fram en standard för utseendet eller syntaxen på en user storie, mer än det "Som X vill jag Y för att Z" som presenterade i kursen.

Vissa av våra user stories var ganska breda och omfattande i början vilket ibland utgjorde ett hinder för vissa teammedlemmar mot att jobba parallellt. Efter ett fåtal problem som teamet fick uppleva på grund av detta samt med hjälp av feedback av våra handledare ändrade vi storleken på våra user stories och gjorde dem mer tydligt formulerade.

Effort estimation gjordes först och främst med Shirt sizes. Detta upplevdes vara väldigt obekant och stundtals konstigt, även om vi ser hur det, efter att ha arbetat med det ett tag, kan ge ett gott resultat och en god förmåga att uppskatta hur mycket arbete som krävs av en feature.

(B)

I framtiden vill vi ha en tydligare rolltilldelning när det kommer till konstruktionen och ett tydligare ansvarskontrakt mellan stakeholders och team. Vi upplevde vissa svårigheter i att tackla helhets- och designfrågor rörande applikationen. En tydligare rolltilldelning och fler möjligheter till teambuilding efterfrågas, för att kunna lättare tackla helhetsfrågor på ett öppet och konstruktivt sätt.

Har man sedan koll på helheten eller en övergripande design så kan en task breakdown göras enklare och generera ett tydligare resultat, med en säkrare effort estimation som följd.

(C)

För att kunna nå vårt önskade tillstånd måste vi ta ansvar för att alla roller som vi ser att teamet behöver fördela faktiskt görs, oavsett vilket ansvar man har. Är dessutom Scrum Master med i teamet så minskar ju inte den mängd roller och ansvar teamet redan har från de roller och ansvar som måste tilldelas.

Vi vill också ta tid från projektet till att utveckla eller besluta om de verktyg som skall användas under resten av projektet, och alltså inte koda med en gång. I en organisation där sådana redan är framtagna eller bestämda så är det istället tid att lära sig företagets "projektsyntax" som skall finnas, och alltså inte ren kod som skall förväntas av nya kollegor.

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

(A)

Till en början i kursen så var våra acceptanskriterier att se till så att koden vi skrivit fungerade felfritt, utan buggar och att funktionaliteten var faktiskt den som efterfrågades av kunden. Det sistnämnda försäkrades genom avstämning med vår PO efter att han visat kunden de nya tilläggen i applikationen. Vi hade alltså inte diskuterat något närmre i gruppen om exakt hur våra acceptanskriterier skulle se ut, utan bara en muntlig överenskommelse. Närmare mitten av projektet så diskuterade vi med vår handledare hur det kunde vara på sin plats om vi bestämde en mall för hur våra acceptanskriterier skulle se ut, så att alla jobbade efter samma, tydligt stadgade acceptanskriterier. Det innefattade det vi bestämt muntligt i början av kursen, inklusive vår interna acceptanskriterier, det vill säga vår DoD. Utöver detta så bestämde vi också att JavaDocs skulle vara med som kriterium.

(B)

För framtida projekt hade det varit bra om vi varit mer stringenta med de acceptanskriterier som vi bestämt. Det kunde vara så emellanåt att när stressen blev hög och vi hade mycket att göra, i synnerhet mot slutet, så tumrades det något på kriterierna. Till exempel så kunde det vara slarvas en del med JavaDocs eftersom man inte upplever att det ger något ökat kundvärde, och valde därför inte att prioritera det när det blev mycket att göra. Det kanske kan fungera en stund när man gör mindre projekt som detta, men inte fullt lika bra när det handlar om större. Skulle bli oerhört svårt att till exempel hålla koll på vad all kod gör efter en längre tid om det inte finns tydlig dokumentation för det.

Vi skulle även vilja se mer givande och, för båda parterna, värdeskapande möten med acceptanstester.

(C)

Till stor del så tror vi att det krävs en bättre planering, till exempel storleken på user stories för att se till så att det inte blir för mycket arbete under veckorna som man inte hinner med. Av vår egna erfarenhet börjar man istället tumma på de sakerna som vid första anblick kan verka mindre betydelsefulla, som acceptanskriterierna. Dock så leder detta till att produkten man

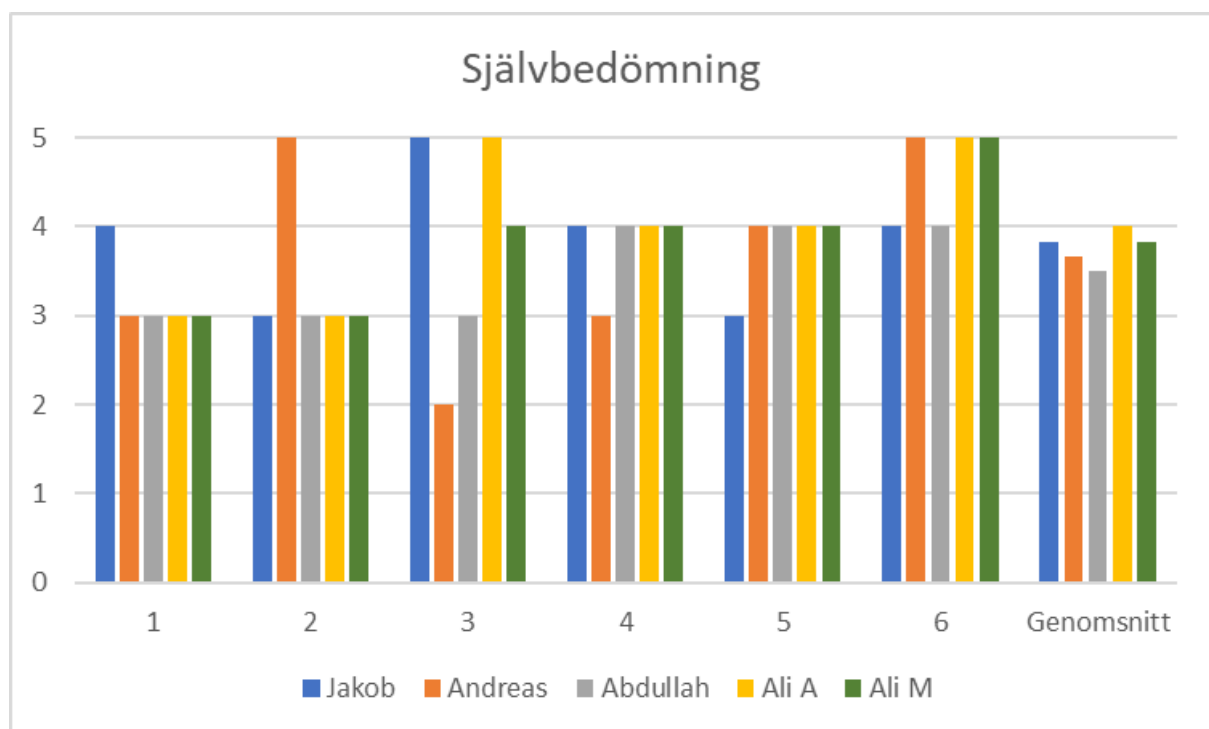
producerar som helhet i slutändan kanske inte håller den standard som man hoppats på, just av att man inte haft tydliga ramar som alla jobbat inom.

Vi skall också involvera kunden mer och trycka på att kunden har lika mycket att säga till om acceptanskriterierna, eller till och med skall ha varit med och skriva dem. Det är trots allt dennes produkt som utvecklas och därmed bör acceptansen av resultatet och troligen en betalning för detta vara av stort intresse för kunden. Om det är en annan form av projekt eller bestämmelse mellan team och kund, så är det fortfarande viktigt att kunden är involverad och engagerad i en utsträckning som gör att teamet kan tillföra värde till produkten.

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

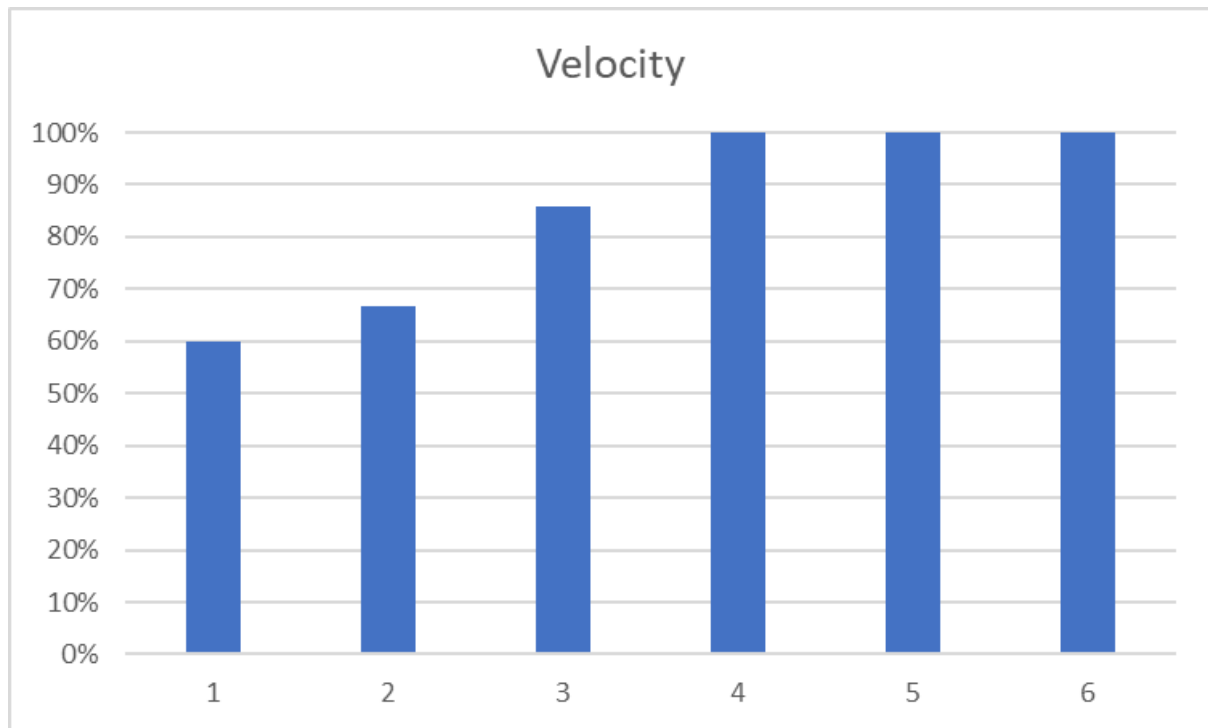
(A)

Till en början valde vi att använda oss av de tre KPI:erna självbedömning (skala 1-5), kundnöjdhet (skala 1-10) samt velocity (andelen avklarade user stories). Ungefär halvvägs igenom projektet valde vi dock efter en diskussion med handledaren att byta ut KPI:n kundnöjdhet mot burn-up chart. Detta eftersom vår kund inte gav oss feedback i den utsträckningen som vi hade hoppats på och blev då tämligen meningslös för att ha med och mäta vår prestation. Burn-up chart skulle demonstrera hur antalet commits hade sett ut total för gruppen till vårt repo på GitHub. Eftersom det hade varit lite problematik gällande frekvensen på commits inom gruppen, skulle detta ge betydligt mer insiktsfull information om hur vår prestation förbättrats efter omställningen.



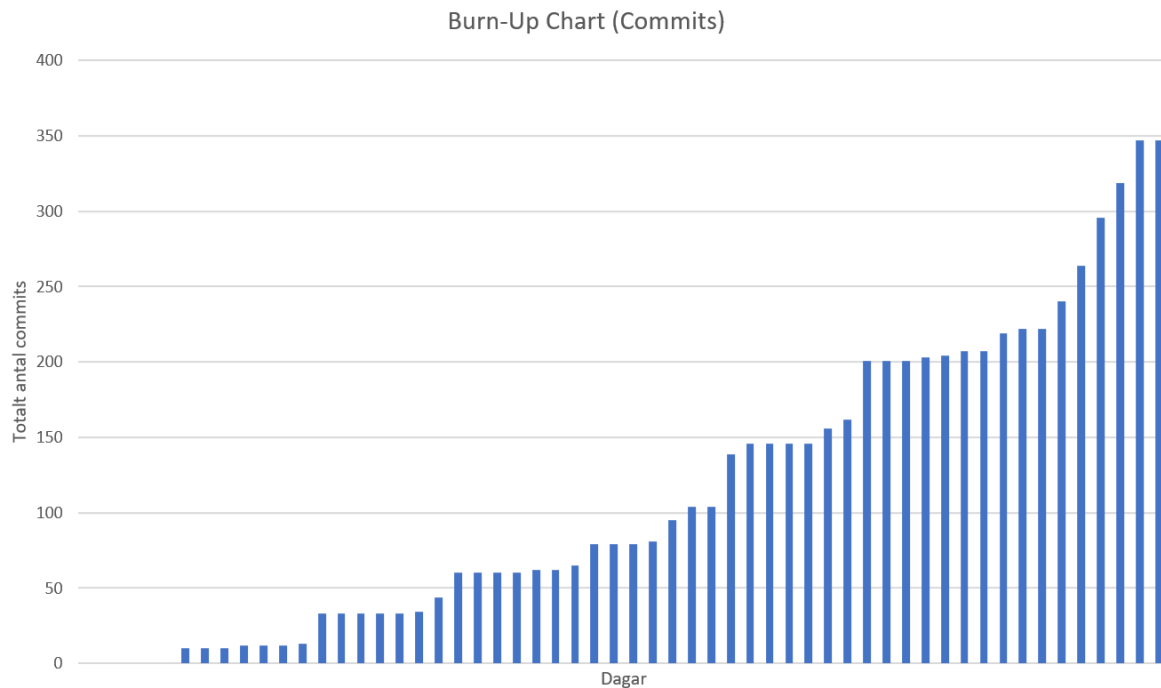
Figur 1 - Självbedömning

I figur 1 ovan ser vi hur den individuella prestationen har sett ut över de 6 veckorna som vi jobbat med Scrum, samt hur genomsnittet har sett ut över dessa veckor. Något som vi tyckte var intressant var att även om självbedömning för varje person har fluktuerat från den ena veckan till den andra så ligger samtliga i gruppens genomsnitt väldigt nära varandra. Detta är någonting som ligger helt i linje med hur den generella uppfattningen i gruppen har varit. Vissa veckor har det varit mer att göra för någon, beroende till exempel på vilka user stories man jobbat med, men sett på övertid jämnar det ut sig.



Figur 2 - Velocity

För figur 2 ser vi istället hur velocity:n har sett ut över de 6 veckorna vi jobbat. Som man kan se tydligt så har velocity:n bara ökat för varje vecka. Detta är inte så konstigt eftersom vi i början hade ganska svårt att uppskatta hur mycket tid varje user story skulle ta. Efterhand som vi blev bättre på att portionera upp dem, samt fick ett effektivare flöde i arbetsprocessen så blev följden av en tydligt förbättring i velocity:n.



3. Social Contract and Effort

Your social contract, i.e., the rules that define how you work together as a team, how it influenced your work, and how it evolved during the project (this means, of course, you should create one in the first week and continuously update it when the need arrives).

(A)

I början av kursen satte vi oss ner och gjorde ett socialt sinsemellan. I kontraktet stipulerade vi ingående regler som gruppen skulle förhålla sig. Områden som berördes var möten, kommunikation, dokumentation och arbete. När kontraktet var färdigt och alla läst igenom, dubbelkollat så allt såg rätt ut skrev samtliga i gruppen under på det på datorn med sin signatur, för att slutligen ladda upp det på vår GitHub. Eftersom det är svårt i början att avgöra vad som är viktigt och värt att ha med i kontraktet bestämde vi oss för att uppdatera det löpande när vi väl fått lite mer kött på benen. Detta var dock inte någonting som blev av förrän närmare slutet av projektet eftersom vi var mer fokuserad på att klara av våra user stories för varje vecka.

(B)

Om vi hade haft en liknande konstellation av människor i ett projekt hade vi nog inte behövt ett socialt kontrakt, eftersom det mesta flöt på smärtfritt tack vare att samtliga i gruppen var flexibla. Däremot så är vi införstådda att fallet inte alltid kommer att vara så i framtida projekt och därför hade ett socialt kontrakt ändå varit på sin plats. Speciellt om det hade varit större projekt som sträcker sig över en längre tidshorisont. Det vi hade gjort annorlunda då jämfört med detta projekt är att vi varit betydligt mer frekventa med uppdateringen av det. För som tidigare nämnt är det svårt att veta allt som behövs innefattas innan man börjat. Därför kan det vara på sin plats att smälta av eller lägga till delar allt eftersom projektet fortlöper.

(C)

För åstadkomma denna förbättring är det viktigt att under veckorna anteckna saker som man märker är viktigt att ha med. Det är lätt att tänka för sig själv att "det ska jag komma ihåg till sprint-mötet" för att sedan glömma bort det en stund senare. Därför hade det varit bra om man hade ett dokument där man direkt kunde anteckna förbättringar i kontraktet som man sedan kunde gå igenom gemensamt i slutet av veckan. Än bättre att dedikera en kort stund varje vecka för att snabbt bara kolla igenom om allt ser bra ut eller om man tycker att någonting behöver skrivas om i kontraktet.

The time you have spent on the course and how it relates to what you delivered (so keep track of your hours so you can describe the current situation).

(A)

Under början av projektet hade vi relativt låg output i relation till antalet insatta arbetstimmar. Detta var för att vi hade problem med att få API:n som vi skulle använda oss av att fungera. Utöver det hade vi även färre problem den nästkommande veckan med att genomföra operationer på de API-anrop vi gjorde, vilket gjorde att vi fortfarande hade en relativt låg output

i relation till arbetstimmar. Att vi även hade föreläsningar samt fler möten gjorde också att vi inte producerade fullt lika mycket det första två veckorna. Däremot efter det, när alla började bli varma i kläderna och fått ett bättre grepp om gruppdynamiken och Scrums ramverk flöt det på betydligt bättre.

(B)

Något som alla i gruppen är överens om inför framtida projekt är att vi skulle velat haft en betydligt jämnare output över veckorna. Som vi nämnde ovan var det väldigt trögt i början med att försöka få projektet i rullning. Detta gjorde att vi kom efter i schemat för vad vi planerat att göra i början av kursen.

(C)

Eftersom startsträckan tenderar till att vara längre än andra delar av projektet kan det vara på sin plats att lägga in flera timmar i början av projektet. Detta eftersom att det ofta är oklart vad som behövs göras eller hur lång tid det kommer att ta. Sen allt eftersom projektet fortlöper kan man dra ned på timmarna då det tenderar till att bli enklare att producera output effektivare när alla blivit lite varma i kläderna. Det är i alla fall vår erfarenhet efter det här projektet.

4. Design Decision and Product Structure

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

(A)

För vårt projekt använde vi oss av Edamam API. Eftersom vår applikation skulle fungera som en dagbok för hur mycket man äter och vilka makronutrierter man får i sig föll det sig tämligen naturligt att använda sig av deras API, eftersom den hade den innehöll flera miljoner recept och ingredienser med denna informationen. Vi använde oss av Retrofit bibliotek för att den erbjuder enkla funktioner till att ansluta API:n, vilket skapar kundvärde eftersom det sparade tid för oss i utvecklingsprocessen genom att vi inte själva behövde utveckla sådana anslutningsfunktioner. Som architecture pattern använde vi oss av Model-View-Viewmodel (MVVM), eftersom utvecklingen av applikationen skedde i Android Studio. MVVM är ungefär densamma som Model-View-Controller (MVC) förutom att det är betydligt enklare att göra förändringar i MVVM än MVC, vilket gjorde den lämpligare för vårt projekt. Utöver detta så använde vi oss av en lokal SQLite-databas som innehöll en enda tabell med två kolumner. Valet av detta var för att vi försökte göra en minimal databas för att undvika att spara en massa onödig information. Till exempel så sparade vi receptens ID-nummer istället för att spara all information om det på databasen, vilket uppskattades av våra handledare.

(B)

För framtida projekt hade vi definitivt velat göra ett bättre förhandsarbete gällande systemarkitektur och val av API. Även om API:n hade det mesta som vi behövde, märkte vi allt eftersom att det fattades saker som vi behövde den eller att där var begränsningar. I vår applikation var tanken att det även skulle finnas med tillagningsbeskrivning över respektive

recept, vilket inte fanns med i den API:n vi valde. Även om det inte var en core-feature i applikationen är det något som skulle kosta oss extra mycket arbete, eftersom vi skulle behöva hårdkoda det själva. Det skulle även visa sig att (i alla fall gratisversionen) Edamam API hade en begränsning på 10 st anrop/min och 60000 st/månad, vilket också begränsade oss när vi skulle utveckla applikationen.

(C)

För att knyta an till tidigare reflektion över insatta arbetstimmar i relation till output, hade det varit på sin plats att lägga mer tid på att utmejsla en tydlig design och struktur på applikationen. Det hade visserligen krävt betydligt mer arbete och research i början av projektet, men det hade underlättat oerhört senare i arbetet eftersom att man kan undvika att hårdkoda en massa i onödan. Samtidigt är vi medvetna om att det är svårt i början av ett projekt att veta exakt allting som ska innefattas och således vilken struktur eller API som lämpar sig bäst. Det hade däremot hjälpt att lägga mer tid på detta än vad vi gjort. Nu blev det istället att vi fick lösa problem som uppkom ad hoc, vilket hade kunnat undvikas om vi gjort ett bättre förhandsarbete och gjort mer medvetna val gällande design. Det hade i sin tur gjort så att vi kunnat öka kundvärdet mer under samma tid.

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

(A)

I början av projektet använde vi oss bara av JavaDoc för att enkelt kunna förstå varandras kod utan att behöva sätta sig in hur den fungerar på detaljnivå. Ungefär halvvägs igenom insåg vi att det kunde vara på sin plats att använda sig av "System Architecture"-dokumentation för att enklare få en överskådlig blick över hur koden är designad och vilka designmönster vi använt oss av. Det har varit användbart att ha något att kontrollera sin kodstruktur mot, även om vår dokumentation inte har varit allomfattande. Eftersom vi valde att spara ytterst lite information lokalt i hårdvaran, så fanns det inte någon anledning att t.ex. använda oss av ER-diagram. Vår databas innehåller endast två kolumner i endast en tabell. Då vi inte hade någon tydlig DoD direkt i början av projektet kom vi överens med vår handledare att det kunde vara på sin plats att göra ett kort dokument för vad som måste vara uppfyllt för att en user story ska klassificeras som klar.

(B)

I framtiden så vill vi att det är tydligt inom teamet vad vi har för krav från kunden på dokumentation, och vad vi ämnar ställa för krav på oss själva och vår produkt i form av teknisk dokumentation. Viss teknisk dokumentation är dessutom till stor hjälp under projektets gång.

(C)

För att nå framtiden så kommer vi att i uppstarten av projektet att ha separata möten för att ta fram den design och dokumentation för design som hjälper till i utvecklingsarbetet. Att avsätta tid för projekteringsarbete görs i de flesta industrier och företag, och bör inte vara undantaget i mjukvaruutvecklingsprojekt. Med en systemskiss och en tydligare arkitektur så kan

utvecklingsarbetet gå fortare framåt och man kan lättare diskutera och planera sin kod. Därigenom uppnås ett bättre samarbete och en högre tillförlitlighet till sin egen velocity.

How you use and update your documentation throughout the sprints.

(A)

Användningen av dokumentationen berodde helt på vilken typ det var. Till exempel så använde vi JavaDoc betydligt mer frekvent eftersom det sparade betydande tid för oss när vi skulle fortsätta bygga på någon annans kod, då vi inte behövde sätta oss in i varje liten del av deras kod. Dokumentation som KPI:n "Självbedömning" användes och uppdaterades däremot endast en gång i veckan istället, vanligtvis i anslutning med sprint-mötet. Användningen och uppdatering av Trello skedde på daglig basis och hade en positiv inverkan på arbetsprocessen genom att tydligt visa vad och vart någon jobbade med veckans user stories.

(B)

Även om JavaDoc användes frekvent under arbetets är detta någonting som vi ändå skulle vilja se en förbättring inför framtida projekt. Stundtals när det var mycket att göra i veckorna blev det lätt att man skippade eller på grund av rent slarv missade att använda sig av JavaDoc. Detta gjorde att andra i sin tur att det kostade ännu mer tid och arbete för andra i gruppen eftersom man var tvungen att lägga mer tid på att förstå delar av koden eller vart delarna av koden man behövde låg. Det hade även kunnat vara på sin plats att kontinuerligt använda sig av UML-diagram för att enklare hålla koll på relationen mellan klasserna. Det var något som vi övervägde mot slutet av projektet, men som inte blev av eftersom vi kände att det skulle kosta mer tid än vad vi skulle tjäna på det.

(C)

För att uppnå ovanstående hade det varit bra om vi utformat tydliga ramverk för hur JavaDoc-dokumentationen skulle genomföras för att få en jämn kvalitet mellan samtliga gruppmedlemmar. Annars blir det lätt som det blev nu, det vill säga att användning av den varierar från person till person, samt från vecka till vecka beroende på arbetsbördan. För UML-diagram hade det varit bra om vi från början använt oss av det, istället för att komma på det halvvägs. Vi gjorde tyvärr misstaget den här gången att tro att vi inte behövde ett UML-diagram eftersom vi uppskattade att vår kodbas inte skulle vara så stor i början av projektet. Detta var någonting som vi insåg senare inte var fallet som sagt. För att summera det hela hade det varit bättre att föra mer dokumentation än vad man tro sig behöva i början av projektet och hellre kassera längre in om det faktiskt inte är någonting man behöver. Gör man som vi nu och inser att det hade varit bra längre in i projektet, blir det betydligt svårare att genomföra det eftersom man behöver bygga upp allt från grunden, vilket är betydligt svårare än att göra små inkrement veckovis.

How you ensure code quality and enforce coding standards.

(A)

För att bekräfta kodkvalitet var det viktigt för teamet att främst testa koden, genom att bland annat skriva enhetstester (Unit tests). Enhetstestning kunde dock endast användas för att bekräfta kvalitet av Backend-koddelar. Visuellt testning gjordes därmed i hög grad för att exempelvis testa kodens kvalitet som berör användargränssnittet.

(B)

En strategi som vi skulle velat använda inför framtida projekt för att höja kvaliteten är att låta kunden testa värdet som produceras i slutet av varje sprint. En ytterligare viktig förbättring som bör nämnas är att teamet i nästkommande projekt ska begränsa sig till tydliga *Coding standards* som ska bestämmas redan i början av projektarbetet. Detta gör att koden blir mer läsbar och lättförståelig vilket i sin tur effektiviserar hela arbetsprocessen bland samtliga teammedlemmar och därmed sparar tiden för andra uppdrag som kan skapa kundvärde.

(C)

För att kunna utföra förbättringarna som nämndes ovan, måste kunden vara mer delaktig med teamet. Dessutom kan vi även ha ett möte innan vi börjar koda för att bestämma en standard kodningsstil som alla teammedlemmar känner sig bekväma med.

5. Application of Scrum

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

(A)

Direkt från första veckan så delade vi ut de två Scrum-roller som vi valde att använda och tilldela personligen till någon, nämligen scrum master och Product Owner (PO). Båda rollerna var statiska genom hela projektet, det vill säga att vi inte bytte runt dem. Anledningen till detta var för att vi upplevde att det skulle ta för mycket onödig tid att byta roller och sätta sig in i dem, samt skapa onödig förvirring. Samtliga tyckte att det var bättre om vi istället utnyttjade allas expertis fullt ut. Eftersom vi hade en i gruppen som tidigare haft erfarenhet av Scrum, föll det sig naturligt att utdela scrum master rollen till honom och PO till den som tagit kontakt med vår stakeholder. Vi var dock överens om att om vi hade extra tid under veckorna så kunde man prova på varandras ansvar för att öva på dem om man ville. Den tredje rollen i Scrums ramverk, utvecklingsteamet var ingenting som vi tilldelade specifikt till någon eftersom alla skulle vara delaktiga i utvecklingen av applikationen. Genom att ta tillvara på allas styrkor i gruppen och dela ut roller därefter, upplever gruppen att det gett positiv utdelning på outputen för projektet.

(B)

Inför framtida projekt skulle det vara bra med om man går tillbaka och påminner sig lite om vad varje enskild roll har för uppgift inom Scrums ramverk. Även om vi delat ut rollerna och alla tyckte det kändes bra, märkte vi efter ett tag att vi började glida lite med rollerna. Det blev lätt så när man blev bekväm med arbetsprocessen, att man tummade omedvetet på vissa delmoment i Scrum och tappade därav fördelarna med det. Till exempel så märkte vi att när vi började med regelbundna stand-up mötena via Zoom eller Discord genom att vår scrum master styrde upp

det löstes problem som uppstått under veckan mycket snabbare än när vi försökte lösa det över chatt. Kortfattat hade det varit bra om vi inför nästkommande projekt hade en mer stringent förhållning till de utdelade rollerna. Vi märkte att det gav en betydligt bättre output än när man stundtals tappade fokus på det.

(C)

För att nå dit kom vi fram till att det hade varit på sin plats att regelbundet de första 2-3 veckorna kolla tillbaka på vad som sagts under föreläsningarna samt komplettera med annan litteratur för att verkligen förstå vad varje enskild roll har för syfte i grupp. Det hade även underlättat om man gjort ett dokument med korta punkter för varje roll som man enkelt skulle kunna gå tillbaka och kolla på om någonting skulle vara oklart. För scrum mastern skulle det kunna vara att säkerställa dagliga stand-up möten och för PO att ha regelbunden kontakt med stakeholders eller säkerställa att rätt features prioriteras i produkten.

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

(A)

Genom hela projektet har vi praktiserat flera olika agila tillvägagångssätt för att åstadkomma optimalt arbetsflöde. Till att börja med så delade vi in vårt projekt i epics som vi sedan konstruerade user stories runt. För att skapa en ännu tydligare bild färgkodade vi även epicsen och user storiesen för att se vilken epic som en user story relaterade till. Detta gjorde vi med hjälp av vår Trello-board. Gav oss en lättöverskådlig översikt på vilka delar som skulle prioriteras och vilka komponenter som ska vara med i vår slutprodukt. Här hade vi även en product backlog tar vi enkelt kunde se vad som behövde göras för nästkommande sprint.

Vidare har vi även kommunikationen varit ett annat område där vi jobbat agilt. I början av varje vecka har vi haft ett sprintmöte som gruppen tycker har varit oerhört viktig för projektets framgång. Under detta möte går till exempel scrum mastern igenom hur föregående vecka sett ut, vad som har varit bra och vad som har varit mindre bra. En ny plan läggs upp för nuvarande vecka där scrum mastern har en mer överskådlig plan på vad som ska göras, medan PO till exempel ger en mer detaljerad plan över vilka user stories som ska prioriteras för att möta stakeholders krav. PO:n kan även gå igenom vad som behövs förbättras från föregående veckas output gällande inkrementellt kundvärde. På onsdagarna hade vi ett mid-week möte för att stämma av med gruppen hur alla ligger till med veckans uppgifter. På det mötet kunde vi diskutera om det till exempel varit en user story som varit mer omfattande än vad vi trott vilket skapat flaskhalsningar för veckans planer. Slutligen så hade vi en sprint review i slutet av veckan där vi sammanställer hur samtliga medlemmar har presterat på en individuell nivå, men också en stund för att ventilera generellt hur vi tycker det har gått och reflektera över det.

Till sist har vi även använt oss av KPI:er för kvantifiera vår prestation i projektet. Som vi berättade tidigare i KPI avsnittet använde vi oss av personlig prestation, burn-up chart, samt velocity (hur mycket av det utsatta arbetet under sprinten som vi färdigställt). Dessa var till god nytta för gruppen då den gav en insikt till exempel hur den individuella prestationen relaterade till hur gruppen presterat för veckan.

(B)

För kommande projekt skulle vi vilja förfina de agila arbetsprocesser vi jobbat med. Som tidigare nämnt kunde det stundtals skippas något stand-up möte för att någon inte kunde vid en specifik tid eller att man inte upplevde att det behövdes. Nu i efterhand märkte vi dock att det ändå kunde vara på sin plats, eftersom vi såg hur produktiviteten blev sämre på grund av att problem som uppstått inte löstes lika snabbt. Även bättre på att uppdatera vår Trello-board. Det kunde emellanåt bli så att man glömde bort att uppdatera på Trello om man tog sig an någonting, vilket kunde skapa förvirring för gruppens andra medlemmar om vart vi befann oss för veckan. Även skapa en prioritering för vilka delar som är viktigast i backloggen hade varit bra, något vi insåg mot slutet.

(C)

Att göra klart för hela gruppen att stand-up mötena är ingenting som tummas på. Är det inte någon som inte kan komma till utsatt tid kan man försöka flytta på det, alternativt genomföra det ändå och sedan summera det i text för den som är frånvarande. Detta bör vara också någonting som innefattas i det sociala kontraktet, så det är tydligare för samtliga medlemmar. För prioriteringsproblemet hade vi enkelt kunnat lösa det genom att numrera dem till exempel 1-10 för att enkelt se vilka delar som bör prioriteras.

The sprint review and how it relates to your scope and customer value (Did you have a PO, if yes, who?, if no, how did you carry out the review? Did the review result in a re-prioritisation of user stories? How did the reviews relate to your DoD? Did the feedback change your way of working?).

(A)

Som tidigare nämnt hade vi en och samma PO genom hela projektet, vilket har tjänat oss väl. Anledning till att ha en dedikerad PO är för att några i gruppen från tidigare erfarenhet är medvetna om att programmerare och kunder pratar helt olika språk. På grund av detta tyckte vi att det var sin plats att ha en medlar mellan dessa två parter, samt någon som sköter den interna logistiken i projektet. PO:n har veckovis (ibland flera gånger) haft kontakt med kunden för att kontrollera att gruppen styr utvecklingen av applikationen i rätt riktning. Utöver detta har PO:n utformat user stories efter den information som fåtts av kunden vid mötena. Detta har möjliggjort att gruppen snabbare kunnat ställa om och prioritera vad kunden efterfrågar än om det skett sporadiskt med olika representanter från gruppen. Gruppen anser också att det bygger bättre relation till kunden om man har en och samma representant. PO:n styrde till stor del hur den detaljerade planeringen för veckan skulle se ut gällande user stories. Beroende på vilken information som gavs kunde tidigare user stories som ansågs vara klara, läggas in på nytt med nya tasks och kriterier om kunden till exempel upplevt att det inte fungerade som han hade tänkt sig eller om han bara ändrat sig gällande någon funktionalitet.

Den allmänna uppfattningen i gruppen om att vi haft en dedikerad PO har gett positiv effekt på arbetsflödet eftersom de som primärt jobbar med utveckling kan fokusera bättre på det än om de hade behövt hoppa mellan de två arbetsuppgifterna. En bit in i kursen insåg med hjälp av vår handledare att det även att det kunde vara på sin plats att baka in en check-up med PO i vår

DoD. Detta var någonting som vi hjälpte oss att säkerställa att varje enskild user story blev korrekt utförd.

(B)

En förbättring gällande sprint review till nästkommande projekt hade varit för PO att pratat med kunden och visat de framsteg vi gjort med applikationen under veckan innan sprint reviewn. Anledningen till detta var att det stundtals kunde bli ganska mycket på måndagens sprintmöten att gå igenom vad kunden tyckte om föregående veckas framsteg samtidigt som man ska gå igenom vad som ska göras i veckan. Det hade alltså varit bättre om PO kunnat visa kunden vad som gjorts på fredag morgon innan sprint reviewn för att få feedback på det och sedan dedikera måndagen enbart åt vad som ska göras. En annan fördel med detta är att kunden snabbare får feedback vad som gjorts, vilket i sin tur skulle underlätta om någonting råkat bli fel så man inte behöver fixa allting på samma gång under måndagen. Tror även det hade kunnat vara bra för kunden att testa applikationen själv för att få en känsla för hur det är att navigera i den, istället för att man demo:ar för honom.

(C)

Den enkla lösningen på detta är att PO försöker boka in ett extra möte i slutet av veckan för att presentera vad som gjorts till kunden och även ta med en demo som gör det möjligt för dem att testa själva än att göra som PO gjorde nu, det vill säga att visa den på sin dator via Zoom/Discord. Om det skulle vara så att det är tidsbrist för kunden på fredagarna skulle man istället kunna ta det på torsdagar istället. Även om veckans mål inte är helt uppklarade tills dess är det bättre än att inte visa någonting alls.

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)

Under projektets gång och utveckling av applikationen har vi använt oss primärt av Android Studio som IDE eftersom det var tänkt att applikationen skulle främst användas på mobiltelefoner. Android Studio har dessutom en Git-integration som tillåter oss att direkt push/pulla eller commita i vår IDE, vilket underlättar en hel del. Framför allt för de som inte har haft så mycket tidigare erfarenhet av Git.

Iomed användandet av Git som versionshanteringssystem så har vi suttit och utvecklat parallellt i separata grenar, en för varje utvecklare. Detta har medfört lite krångel då man har upplevt problem med att arbeta på, eller hålla isär, mer än en user story åt gången. Det har dessutom maskerat för andra vad man faktiskt jobbar på eller vad för förändringar som finns i en gren.

Utöver Android Studio så använde vi oss också av Figma för att göra mock-ups som PO kunde använda sig av för visa kunden hur applikationen skulle se ut när den väl var färdig. Det var fördelaktigt eftersom kunden direkt kunde säga om det var någonting han inte gillade utseende eller funktionsmässigt, utan att en massa onödig utveckling av applikationen gjordes. Trello

användes som det primära verktyget för att strukturera vårt arbete. Underlättar enormt när man har en visuell representation av arbetets olika delar.

Eftersom det fanns en stor variation av hur mycket har personer i gruppen programmerat tidigare såg vi till att utnyttja de med god erfarenhet till att göra de som inte haft fullt lika mycket mer bekväma i att använda till exempel Android Studio eller bara programmering generellt. Uppdelningen av uppgifter skedde således i relation till hur kunnig man var. Var det någon som upplevde att de ändå inte kunde leverera det arbete som de tagit på sig för veckan, fick man hjälp av andra mer kunniga medlemmar. Stundtals satt vi även och parprogrammerade för att underlätta om någon kört fast. På grund av detta tycker vi att best practice är att ta tillvara på den expertis som finns inom gruppen och om ingen där skulle kunna hjälpa en att antingen vända sig till någon tillförlitlig källa på internet eller någon av de resurser som kursansvariga gett oss.

(B)

För framtida projekt skulle vi vilja kolla på mer best practices för saker som inte relaterar till de tekniska aspekterna av arbetsprocessen. Till exempel hade det varit intressant hur man gör för att optimera gruppens effektivitet eller hur man minskar stressen. Utöver detta hade varit bra om vi hade lite extra tid på att lära oss de best practices som vi redan lärt oss, för att förstå dem och utnyttja dem bättre.

En konkret förbättring som vi skulle vilja se är möjligheten att istället för developer branches, ha story branches, där man på ett tydligt sätt kan se varför en branch finns och därigenom också lättare kunna föreställa sig vilka ändringar som görs när den branchen sedan sammanförs till main (Daly, i.d.).

(C)

För att göra detta möjligt och så intressant som möjligt tror vi att det hade kunnat vara bra att kolla på ett riktigt IT-företag som använder sig av till exempel Scrum för att se hur de opererar den dagliga verksamheten. Nog för att man kan hitta mycket information på internet om det, men vi tror ändå att det hade kunnat ge betydligt mer att se det appliceras på något konkret. Det hade också skapat en bättre och djupare förståelse på ett interaktivt sätt.

Gällande förbättringen med branches hade det varit bra om man i början av kursen kommer överens om att man använder sig av denna standarden så att inga misstag uppstår.

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

(A)

Till stor del har vårt arbetssätt baserats på de föreläsningar och övningar som vi hade i början av kursen. Föreläsningarna hjälpte oss att få en generell förståelse för hur en agil arbetsprocess ser ut och hur man åstadkommer det med hjälp av Scrum. Övningarna gav oss bland annat insikt till hur man ska tänka när man gör user stories. Vi använde oss av en artikel på Medium (Drill, 2019) som hjälpte oss med att skriva sociala kontraktet. Utöver detta så har vi haft möte med vår handledare Irina Levkovets varje måndag morgon för att berätta hur det gått för oss och få

vägledning om det var någonting hon tyckte vi missat i vårt utövande av Scrum. Hon var till stor hjälp och gav oss flera bra tips för hur vi skulle ta oss framåt i vårt projekt. Bland annat så hjälpte hon oss att definiera DoD, hur code review skulle gå till samt feedback gällande våra val av KPI:er.

(B)

Något som vi med facit i hand skulle ha gjort bättre till nästa projekt är att använda oss av material från gästföreläsningar och övrig litteratur gällande agilt praktiserande och hur Scums ramverk ser ut, löpande genom kursen. Även om Håkan och Jan-Philipp föreläsningar i början av kursen var till stor nytta är det lätt att man tappar vissa delar. Även om Scrum är relativt lätt att ta till sig är det ett nytt arbetssätt med många nya termer, vilket gör att det tar tid innan allting sätter sig. Vi tror att det hade snabbat på inläringen av Scrum och således fått upp flödet i arbetet genom att veckovis ta hjälp av till exempel föreläsningar på YouTube. Det ökar också förståelsen för ämnet genom att få flera infallsvinklar.

(C)

Den enkla lösningen på detta är att tillvara på de nästan ändlösa resurserna som finns tillgängliga på internet. Utöver detta så hade man kunnat utnyttja de tips som vi fått av handledare och kursansvariga på ämnet bättre. Ett annat alternativ hade varit att prata med folk i ens omgivning som jobbar på något utvecklingsföretag inom programmering för att se hur de jobbar och få en bild av hur det faktiskt ser ut i verkligheten när man använder sig av en agil arbetsprocess.

Länkar

Scrum Board (Trello)

<https://trello.com/invite/b/MT5jy3S1/bf9fccbe1592264a5667d7683cee31c3/dat257-the-cure>

GitHub

<https://github.com/AliMalla/Cure>

Referenser

Daly, L. Learn how Git fits into an agile workflow. Atlassian.

<https://www.atlassian.com/agile/software-development/git>

Drill, M. (5 maj, 2020). Agile ways of working- social contracts. Medium.

<https://medium.com/swlh/agile-ways-of-working-social-contracts-ab6b8c93429f>

Ken Schwaber & Jeff Sutherland (Nov 2020). The Scrum Guide.

<https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf>