

EcoDrive Group Team Reflection

DAT256

Rasmus Claesen
Sigbjørn Drøsshaug
Thomas Frödin Larsson
Carl Lundborg
Eric Mossberg
Niclas Tauson

November 2, 2018

Contents

1	Customer Value and Scope	3
1.1	the chosen scope of the application under development including priority of features and for whom you are creating value	3
1.2	the success criteria for the team in terms of what you want to achieve with your application	4
1.3	your user stories in terms of using a standard pattern, acceptance criteria, task breakdown and effort estimation	5
1.4	your acceptance tests, such as how they were performed and with whom	6
1.5	the three KPIs you use for monitoring your progress and how you use them	7
2	Social Contract and Effort	7
2.1	your social contract, i.e., the rules that define how you work together as a team (this means, of course, you should create one in the first week)	7
2.2	the time you have spent on the course (so keep track of your hours so you can describe the current situation)	8
3	Design decisions and product structure	8
3.1	how your design decisions (e.g., choice of APIs, architecture patterns, behaviour) support customer value	8
3.2	what you document and why, by using e.g. use cases, interaction diagrams, class diagrams, domain models or component diagrams, text documents etc.	9
3.3	how you use and update your documentation throughout the sprints	10
3.4	how you ensure code quality, enforce coding standards, and . . .	11
4	Application of Scrum	11
4.1	the roles you have used within the team	11
4.2	the agile practices you have used for the current sprint	12
4.3	the sprint review (either in terms of outcome of the current week's exercise or meeting the product owner)	12
4.4	best practices for using new tools and technologies (IDEs, version control, scrum boards etc.)	13
4.5	relation to literature and guest lectures (how do your reflections relate to what others have to say?)	13

1 Customer Value and Scope

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

Den första iterationen av vårt scope gjordes vecka 3 efter att tillsammans med produktägaren diskuterat våra idéer och vilka features vi skulle jobba mot. Vårt scope var här en app som skulle ge feedback för körningen normaliserat för topografi i realtid. Denna feedback skulle färgkodas och presenteras. Vårt scope innehöll även simuleringsutveckling och statistik feedback.

Vi skulle utgå ifrån Vovlos modeller för liter / mil presentation och addera värde med flerdimensionell färgkodad feedback.

Vecka 4 gör vi nästa iteration och prioriterar vårt scope DESC i två delar;

- Färgbytt UI baserat på körkvalité avseende ECO-driving
- Historisk statistikpresentation
- Vidare funktionalitet, bl.a simulering, får lägre prioritet och blir mer av extra features givet vårt scope

Detta förblir vårt scope för var vecka genom konsensus. Vi gör nya prioriteringar vecka 7 efter feedback från produktägaren;

- Implementera en hjälpfunktion som förklarar appens syfte och funktion
- Implementera alternativ feedback för färgblindhet
- Förbättra den visuella upplevelsen av appen
- Vidare utveckling av statistikpresentation bortprioriteras

vi upplevde att det tog lång tid innan vi fastställt vårt scope vilket försvårade för det tidiga arbetet med bl.a user stories och sprints. Scopet vi fick fram den andra iterationen förblev det överliggande scopet för projektet, enstaka ändringar gjordes vid reflektioner samt en tydlig prioritering vecka 7.

I retrospekt hade vi velat arbeta mer och tidigare med vilka feautres vårt scope innehar och dess prioriteringar. Något vi har inte drog nytta utav förens sent in i projektet. Det generella scopet var vi nöjda med och vi hade även utrymme för att lägga till några extra features sista veckan.

Vi upplevde också att vi var dåliga på att ta hjälp av handledare och att föra dialog med produktägaren, vilket hade kunnat hjälpa oss i våra iterationer av scope och feautres.

1.2 the success criteria for the team in terms of what you want to achieve with your application

Den första iterationen av success criteria gjordes vecka 2 och formulerades

Vi vill att appen ska fungera som ett användbart verktyg för föraren som vill spara pengar på bränsle och tänker på miljön. Den ska vara lätt att använda med minimal interaktion under resans gång och vara något föraren vill nyttja varje dag.

Vi reflekterar mycket över vilka kriterier detta innebär och skapade oss ett konsensus om vilka dessa kriterier konkret är. Under iterationen av dessa kriterier vecka 6 utvecklar vi med bl. a att applikationen skall klara simulering med scopets definierade funktionalitet under en minuts körning.

Likt tidigare är vårt success criteria att ha en app som skiftar färg beroende på hur föraren kör ur ett ECO perspektiv och även titta tillbaka historiskt för att se resultat från tidigare körningar. Men vi vill också kunna simulera och demonstrera detta på ett bra sätt. Detta för att kunna verifiera dess funktion.

Under den sista iterationen uppdaterar vi även dessa kriterier med krav på den visuella framställningen. Med begränsad kunskap inom gruppen kring utmaningarna i ett android-projekt så började våra iterationer med relativt låga kriterier, vilket vi tycker fungerade bra både vad vi siktade på resultatmässigt för applikationen men även för vårt arbete och utveckling.

I retrospekt hade vi velat jobba mer med att konkretisera dessa kriterier i form och dess testbarhet i applikationen. Detta är något vi mest gjorde informellt och det uppstod i vissa fall oklarheter vilken funktionalitet som förväntades klara av vad vid vilken tidpunkt. Med tydligare definitioner hade vi kunnat bryta ner delmål som underlättat vid ex. vertical slices.

1.3 your user stories in terms of using a standard pattern, acceptance criteria, task breakdown and effort estimation

När vi började skriva våra user stories var vi, som man kan förvänta sig, väldigt nya till hela konceptet. Detta skapade problem såsom att vi hade väldigt stora user stories, vilket i sin tur ledde till att vi hade viss overlap mellan dem och dess tasks. Sagda tasks hade saknade även utförliga konkreta beskrivningar och acceptanskriterier. Som följd så hamnade vi i en sits där vissa personer hade utvecklat funktionalitet som liknade varandras, vilket uppenbarligen inte var effektivt för gruppens produktivitet.

Vi insåg senare att vi hade ett problem när vi hade haft två sprints, men inte avklarat en enda user story, och hade med nöd och näppe lyckats bocka av ett fåtal tasks. För att åtgärda detta tittade vi igenom alla våra user stories, och trimmade ner dem där vi kunde. Detta hjälpte oss med att bättre avgränsa vårt arbete, men vi hade fortfarande problem med att många tasks hade implicita krav som inte alltid var uppenbara, och att det fortfarande var svårt att avgöra när man var klar med dem.

Det var dock inte förrän vi fick ordentlig feedback på våra user stories som vi skrev om dem på ett sätt som vi slutligen blev nöjda med. Vi skrev dem på ett mycket tydligare sätt, men viktigast av allt så fokuserade vi på att vara väldigt noggranna med att alla tasks skulle vara väldigt konkreta i vad de efterfrågade och när man kunde anse dem klara. Detta ökade vår produktivitet märkvärdigt i den sista sprinten vi hade.

Vad vi hade kunnat göra bättre är först och främst att vi borde varit mer uppmärksamma på att vi inte riktigt kom någon vart med våra ursprungliga user stories. Redan på vår första sprint review borde vi diskuterat faktumet att vi hade svårt med att avklara user stories/tasks, och att vissa gruppmedlemmar ibland behövde fråga de andra på möten vilken funktionalitet det de jobbade på faktiskt behövde. När vi väl identifierat det borde vi ha bett om feedback efter första gången vi skrev om våra user stories/tasks så att vi hade kunnat försäkra oss om att de inte skulle bli ett hinder i framtiden.

Med det vi vet nu så borde ju såklart även våra user stories/tasks varit mer konkreta från första början. Om en situation någonsin uppstår där gruppmedlemmar måste fråga vad som verkligen behövs för att avklara sin uppgift så har man redan där misslyckats.

1.4 your acceptance tests, such as how they were performed and with whom

Vi har tyvärr inte haft några formella acceptanstester under kursens gång. Detta beror till del på, som nämnt ovan, att våra tasks har varit väldigt odefinierade. Tack vare att det var svårt att veta när man var klar så har det varit svårt att sätta sig ner och skriva tester för sin färdiga kod, och när vi väl nådde punkten då vi hade renskrivit våra tasks så var unit tests och dylikt tyvärr så låg prioritet gentemot att få fram en shippable product.

Alla gruppmedlemmar hade dock ett eget ansvar att se till så att deras kod var funktionell, och när någon väl hade något som var redo att visas så tittade vi alla på det för att bekräfta att det såg bra ut.

Det är väl det närmsta vi kom att ha ett +2 system (system där andra developers review:ar pushad kod och antingen ger +2, vilket innebär att koden är ok för att pushas till master, +1 som innebär att det behövs förändringar, men annars bra, och slutligen 0 som innebär att koden måste göras om) som Volvo diskuterade. Mot slutet av kursen så testkörde vi även programmet med events från våra simulationer för att bekräfta att programflödet flöt på som det skulle.

För framtiden så borde vi, som tidigare nämnt, ha mer konkreta tasks. På så sätt hade det varit mycket lättare för varje gruppmedlem att utveckla funktionalitet och sedan testa den.

Vi borde även ha använt oss av ett formellt +2 system som förklarat ovan. Det hade hjälpt oss med vårans tidsbrist som vi hade problem med under projektets gång ifall vi kunde titta över och review:a varandras kod utan att behöva vara på plats. Tiden som hade krävts för att sätta upp ett sådant system, vilket vi ursprungligen trodde vi inte hade tid med, hade betalats av sig i slutändan.

1.5 the three KPIs you use for monitoring your progress and how you use them

De första KPI'erna som vi tog fram var tid spenderat, effort och rader kod. Vi märkte ganska snabbt att rader kod inte funkade för oss som KPI då det blev väldigt missvisande för tasks som inte var så mycket kod men tog mycket tid för att leta information eller att tänka ut hur en lösning skulle göras. Vi gick sedan vidare med endast tid spenderat och effort.

Problemet som dock uppstod var att, som vi tidigare nämnt, våra user stories var för omfattande och vi fick därmed ingen effort förrän i sista veckan efter att vi omdefinierat user stories. De blev därmed mindre och mer väldefinierade och vi kunde då tillslut logga effort under vårt KPI.

För framtida projekt skulle vi som tidigare nämnt gjort väldefinierade user stories redan från början alternativt korrigera dom i ett tidigare skede när vi märkte att vi ej loggade någon effort. På detta sättet hade vi även fått bättre tasks definierade och vi hade därmed kunnat plocka nya tasks när vi satt sysslolösa och därmed kunnat logga mer tid på projektet.

2 Social Contract and Effort

2.1 your social contract, i.e., the rules that define how you work together as a team (this means, of course, you should create one in the first week)

Vi definierade väldigt tidigt i projektet vilka tillfällen vi skulle träffas och vilka tider samt när vi skulle vara tillgängliga i vår Slack. Detta lade en bra grund för vårt arbete under kursen. Vi märkte även en bit in att vi behövde lite extra tid på fredagar för att slutföra reflektionen och vi lade därmed till det. Vi gick därmed från 1 timme på fredagar till 2 timmar. Dock var det ganska vanligt att tiderna inte passades på fredagar. Detta gjorde att det ändå blev ont om tid trots ökningen av tid.

Våra sprint planeringsmöten var på måndagar under handledningstiden. De var tvungna att vara då på grund av våra väldigt olika scheman. Vi fick därför inte lika stor möjlighet att ta hjälp av handledningstillfällena. För framtida projekt i arbetslivet skulle vi dock inte ha denna typen av problem då scheman är mer synkade.

Vad vi skulle behöva ha gjort bättre var att följa de tider som vi utsatt för att vara så effektiva som möjligt när vi väl hade möjlighet att jobba ihop.

2.2 the time you have spent on the course (so keep track of your hours so you can describe the current situation)

Genom våran tidslogg har vi mätt hur mycket tid vi lägger på kursen och en generell kommentar är att man kan tydligt se hur väl vi strukturerat arbetet med scrum och tiden vi lagt på kursen. I tidigt skede loggade vi ganska lite tid och detta beror nog till stor del på konkreta deadlines från andra kurser medans scrum fortfarande var väldigt abstrakt och vi saknade tydliga tasks att arbeta med och detta påverkade tiden vi lade som grupp på kursen.

Allt eftersom att kursen fortlöpte blev vi mer insatta i arbetsmiljön för detta projektet och kunde därmed mer konkret arbeta med våran applikation. Vi märkte att med hjälp av feedback så fick mer konkreta uppgifter att arbeta med vilket hjälpte oss att strukturera och driva vårat arbete framåt.

För framtiden hade vi som nämnt tidigare behövt tydligare tasks där vi själva borde ha satt ‘deadlines’ för att motverka att andra kursers deadlines tar prioritet från detta vilket hade kunnat verka för bättre utjämning av tiden vi lade på kursen. Hade även kunnat hjälpa vid dependencies mellan implementation med tydliga ‘deadlines’ för tasks.

3 Design decisions and product structure

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

Vi valde som ett grundläggande design-pattern att använda oss av model view controller då detta är ett väl tilltaget sätt att strukturera och producera modulär kod. Vi ville även ha en modulär struktur på koden för att kunna uppfylla kundens behov i form av återanvändbarhet.

Efter att ha suttit ett tag med att försöka implementera MVC insåg vi dock att det inte var helt lätt att göra just med Androids bibliotek. Efter att ha satt sig in i Android lite insåg vi istället att MVP(model view presenter) blev ett betydligt lättare sätt att jobba med men också att implementera med just Android. Det medförde också precis det vi sökte, vilket var modularitet och en struktur som gjorde det lätt för alla att implementera sitt arbete med.

Vi valde även vissa andra designval, så som användning av tredjepartsbibliotek för databashantering och grafritande. Dessa val gjordes för att vår fokus inte låg på att bygga en lösning med egna bibliotek. Vi ville helt enkelt inte uppfinna hjulet på nytt, utan istället lägga vår fokus på en helhetslösning och på så sätt uppfylla customer value genom att hinna få ut en produkt som fanns inom vårt scope.

För framtida projekt tar vi med oss att det är väldigt viktigt med olika designval för att skapa ett projekt som är lätta för alla parter att ta del av. Vi insåg att det var ett väldigt bra val att använda oss av MVP modellen för just detta projektet som tillät oss att fokusera på customer value då det ökade vår produktivitet och gjorde det lätt för alla att merka ihop sitt arbete på ett strukturerat och modulärt vis.

3.2 what you document and why, by using e.g. use cases, interaction diagrams, class diagrams, domain models or component diagrams, text documents etc.

Generellt sätt har vi använt oss en stor del av google drive för att samla våra dokument i. Vi har dokumenterat och fört anteckningar över viktiga beslut under möten, använt oss av diagram för överblicka user stories och tasks samt loggat vår tid spenderad i kursen via ett diagram. Denna del av vår dokumentation har dock mest rört administrativa frågor och scrum-relaterade ämnen. Detta sätt att föra dokumentation utvecklades under arbetets gång till en ganska förfinad version, som vi hade lätt för att använda oss av och det blev ett väldigt bra verktyg för oss att dela information mellan alla i gruppen.

Då vi beaktat detta som en akademisk övning har vi dock valt att spendera majoriteten av den tiden vi haft på andra delar än just dokumentationen och vi har gemensamt valt att lägga en högre prio på att arbeta med scrum processen som det arbetssättet medförde. Detta har tyvärr medfört att vår dokumentation för applikationen och utveckling blivit lidande (SCRUM overflow) då vi bl.a. inte analyserat vår tidigare dokumentation tillräckligt för att på så vis lära oss av våra misstag och ändra på det som behövdes för att nå ett önskvärt resultat.

För dokumentation av appen har vi främst (informellt) använd b.la whiteboard för jobba mot en samförståelse av applikationen bland alla i gruppen. Detta har fungerat men vi har ofta funnit oss i situationer där vi haft väldigt olika uppfattningar om hur olika delar av applikation bör se ut och te sig och saknat dokumentation som man kunnat gå tillbaka till och jobba utifrån.

Vi vill i framtiden bli betydligt bättre på att föra dokumentation. För att förbättra integrationen mellan olika delar av programmet hade vi behövt tydligare dokumentation som överskådar de olika klasserna och dess integration, detta hade hjälpt oss både att skapa unit-tests för de olika delarna och slutgiltiga programimplementationer hade förenklats.

Om vi vidare reflekterar kring huruvida ett helt nytt team hade kunnat fortsätta vårt arbete och hur väl den överlämningen hade kunnat ske; Vi saknar en hel del dokumentation gällande vad som rör konkret information kring vår applikation och vad den ska innefatta då detta varit informellt mellan oss. En överlämning till ett nytt team skulle därför bli väldigt svår. Även dokumentation såsom uml och javadocs skulle brista för detta scenario och därför vara något som vi vill lägga mer vikt vid i framtida projekt.

3.3 how you use and update your documentation throughout the sprints

Dokumentationen under sprints bestod främst av att vi använde oss av programmet trello i form av ett scrum board med relaterad dokumentation. Under sprints gjorde vi få uppdateringar och dessa berörde oftast status kring tasks som fanns i våran sprint backlog.

Vi gjorde många reflektioner kring dokumentationen men hade ibland svårt att genomföra förbättringarna till nästa iteration, mycket på grund av hur vi utformat våra use cases och tasks. Vi använde oss av dagordning för att ta upp relevanta frågor och dokumentera detta men detta verkar ha minskat över tid då vi blev mer bekväma med vad som behövdes tas upp och även mer självständiga i vårt arbete.

För framtida projekt vill vi ta med oss att arbeta mer iterativt med våran dokumentation för sprints och vi bör även söka mer feedback kring hur vår arbetsprocess ser ut, för att på så sätt föra en mer ingående dokumentation som troligtvis leder till ett mer produktivt och genomgående arbete.

3.4 how you ensure code quality, enforce coding standards, and

Vi hade inget direkt behov av kodkvalité förrän i vecka 5. Detta pga att vi inte hade någon kod att tala om innan dess. I diskussionen kring detta kom vi fram till att man bör lägga till en liten kommentar till sin kod ifall funktionsnamnet inte är självförklarande. Det har till synes räckt för oss i det här projektet, eftersom det aldrig kommit upp som ett problem i våra sprint reviews.

Kanske hade alla redan en god kodningsstandard, men vi misstänker att det kan bero på det designmönstret vi valde. Vi valde MVP och varje medlem såg till att hålla sin kod inom ett av de tre delarna i MVP. Vi kan tänka oss att en kodbas med unit tests för alla klasser och funktioner är ett sätt att hålla styr på ett projekt av större magnitud. Att ha som krav att varje task ska ha ett unit test som verifierar att task:en lever upp till sin definition of done.

4 Application of Scrum

4.1 the roles you have used within the team

Vi behövde aldrig uttryckligen fördela arbetet. Varken när det gällde att organisera projektet enligt agila metoder eller vilka tasks som skulle göras av vilka. När ett något dök upp, såsom schemaplanering, scrum board, git och google drive, utfördes uppgiften av den personen som kände att den hade kolla på det.

Vi utsåg däremot en scrum master. Thomas tog den rollen och varken han eller resten av teamet kände att vi behövde ändra på detta. Det ställdes inte så höga krav på rollen, men han levde upp till rollen av en scrum master genom att vara drivande på möten och styra oss i rätt riktning när vi drev för långt ifrån de agila metoder vi skulle följa.

I framtida projekt kanske man vill utse roller som kodare, kvalitetssäkrare eller dokumentationsansvarig, och rotera dessa roller på en sprintbasis. Det här kanske skulle vara mer applicerbart på större projekt.

4.2 the agile practices you have used for the current sprint

Vi har haft short iterations, kanske lite för korta. Sprintplaneringarna ägde rum på måndagar och print review på fredagar. Vilket effektivt gav oss 3 dagar per sprint, eftersom mötena tog så mycket energi. En digital scrum board har vi använt. Däremot kom den inte i ordentligt bruk förrän vecka 5 då vi hade etablerat vår user stories.

En sustainable pace har vi haft trots att den har eskalerat. Vi har ändå legat under 20h/vecka per person. Eskaleringen har främst berott på att vi allteftersom har lärt oss att jobba enligt scrum. Vi har sett mest utveckling i och med att vi har skapat och förbättrat våra user stories. Vi har haft mest problem med att bryta ner våra idéer i mindre user stories. Vi lyckades bättre med detta närmare slutet av projektet. Vilket också ledde till att teamet spenderade mer tid på det eftersom vi fick en tydligare bild om vad som behövde göras.

Customer value blev lite försummat eftersom vi inte gick på många feedbackmöten. Vi hade inte lyckats producera något PSPI att visa för produktägaren så vi tog beslutet att inte dyka upp. Vi inser att detta inte var bra eftersom det ledde till att teamet blev mer och mer osäkra på vilken riktning vi skulle gå allt med att tiden gick. Vi skämdes nog lite och vågade inte fråga om hjälp. Vi skulle åtminstone dykt upp och pratat med lärarna om arbetsmetoden i allmänhet, eftersom vi i efterhand inser att vi hade en bristfällig implementation av scrum.

4.3 the sprint review (either in terms of outcome of the current week's exercise or meeting the product owner)

Because of limited scheduling options (we had badly clashing schedules) we did the strategic error of scheduling our sprint planning meeting on top of the monday help sessions with the intent that we would do our sprint planning as a group in a group room, and - if needed - send our scrum master to the meeting to ask questions. This was done from week 4 and out, with the exception of week 7.

In hindsight we see that we absolutely should have gone to the monday meetings to get help, or just talk about our group progress in general, as we spent a lot of time in our meetings discussing scrum processes and trying to figure things out on our own – a process that would have been made a lot easier by consulting with a teacher.

We feel that our sprint reviews worked pretty well for us, in the sense that we had good reflections and thoughts about the sprints, but we feel we fell a bit short when it came to implementing the ideas in the following sprint.

In future projects, we would definitely make sure to attend help sessions and consult teachers when we feel we are in a bind, instead of trudging on by ourselves. This would have saved us a lot of time, and probably accelerated our scrum-learning process greatly.

For our sprint reviews, we felt we had a good workflow, that improved slightly with every sprint, but we would like to work more on being consistent with implementing the ideas that we come up with. Some ideas we had to drop because of time constraints, while others were given a low priority, because of more pressing concerns. Overall we feel we had a good scrum-workflow, considering our limited time window for meetings, and relatively short sprints.

4.4 best practices for using new tools and technologies (IDEs, version control, scrum boards etc.)

During the first few weeks we decided on the different programs and tools we were going to be using throughout the course. We decided to use Trello for our scrum-board, git for version control, google docs for editing documents together as a group, google calendar to synch our calendars, as well as slack and discord for communication. In meetings we have also used whiteboards for meeting agendas and notes, as well as "scrum-poker" for estimating task effort.

These tools have helped us stay on top of our progression as a group, and helped us stay synchronized when we have worked remotely, or members have been absent from meetings.

This setup remained unchanged throughout the duration of the course, and we think it worked out very well for us overall, and we would most likely go with the same if not a very similar setup for future projects possibly replacing slack with a different chat client. It's a bit easier to use on the go since slack does not sync between devices, you have to make sure to join chat rooms, and channels on every device, otherwise you might be losing out on information without even knowing it.

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

In the first few project weeks we mostly winged it based on what we had learned from the lectures in the first two weeks of the course, and the LEGO exercise. After we started reading more of the literature available (from the course page and otherwise) we picked up some pointers on how to estimate our velocity a bit better (this felt quite abstract to us in the beginning), as well as tips on how to proceed with our scrum processes. This helped us get a little bit more on track to learning to apply the scrum processes properly.

The semcom lecture brought up a lot of good points regarding common situations which you often find yourself in as a programmer. This helped alleviate some of the stress we felt regarding diving down into a mostly undocumented, and completely new to us API, as well as learning many tools concurrently while working in a new programming environment. Some important points to take away from the lecture was the emphasis on the fact that it is better to do something than to do nothing, and that we will have to act on incomplete information a lot of times. Essentially it is better to do something and fail fast, rather than over-plan and spend way too long on discussing things to death. Which we also feel fits rather nicely with the agile philosophy of short sprints, and its iterative nature.

We feel that we did relatively well comparing our experiences with how the Spotify guest lecturer described how they did their sprint plannings, and at the same time drawing comparisons to Scrum and XP from the trenches – where the author talk about scrum planning meetings that lasts for 4 hours. Considering our limited time – and clashing schedules, as well as the fact that we were learning at the same time as we were applying – we feel that we had a good learning experience when it comes to applying scrum processes, and that we improved a little each sprint.

During our sprints we used different literature sources to help us decide on our MVP pattern, and how to split user-stories up into tasks. A combination of the elephant-slicing lecture and the Hamburger method article linked on the course page was helpful in this regard, as we felt pretty confused when it came to these points. In hindsight it would have helped to consult the literature even more (having read more of XP from the trenches at this point) but at the time, it felt pressing to “just get going” so we winged a lot of it. This is definitely something we would want to work on improving more in future projects.

When considering how these points can be improved moving forward, we definitely feel that picking up extra tips and tricks from people in the field is very useful, as they have often been through it a couple of times already, and have weeded out what works and what does not. This might provide useful insight into situations that we might not think about when reading directly from the literature. It is also important to consult literature when available, since we could have spared some time in meetings by reading up on subjects beforehand, and not spent so much time on discussing it during our sprint plannings.

Comparing our experience from this course with the info from the guest lecturers, and the literature available, we feel that we have done pretty well all things considered. We have seen an improvement with every sprint, and would use a similar approach for future projects with a heavier emphasis on the points mentioned above.