

Final Team Reflection

Group Joy

October 29, 2021

Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it.

Through this work we have come to value:

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.[1]

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

I början av projektet var idén att kunna ha en hemsida som visar upp statistik på en karta, med en heatmap som visualiserar och förenklar hur man ser statistiken. Tanken var att man skulle kunna välja flera olika data, t.ex. antal vaccinerade, befolkningstäthet osv. Vi pratade också tidigt om att man skulle kombinera flera olika data för att se ifall det fanns samband mellan viss statistik i de olika områdena. Vi satte en grund med att kolla antalet vaccinerade i varje län, och i LV6 diskuterade vi huruvida det skulle vara möjligt att implementera mer statistik, men kom i LV8 fram till att detta skulle vara för mycket arbete för att hinnas med, och nöjde oss till slut med att enbart visa antalet vaccinerade per capita i varje län.

Meningen med produkten var ett enkelt sätt för folk att visualisera data som annars kan vara svår att hantera i enbart siffror. Detta influerade även vårt val att sätta upp det som en hemsida. Hemsidor är lättillgängliga och vem som helst kan när som helst nå den för att se statistik från både sitt eget län men även sverige i övrigt. Vi valde att använda antalet smittade i Covid-19 eftersom detta är något som är högaktuellt och anknyter tydligt till FNs tredje mål för hållbar utveckling, god hälsa och välbefinnande.

Vi är överlag nöjda med vad vi åstadkommit med projektet, trots att vi motvilligt fick dra ner våra ambitioner och vårt scope. I framtida projekt vill vi förbättra oss i vårt arbetssätt för att gemensamt nå de mål vi sätter i början. Dessa aspekter kommer vi att reflektera över i senare delar av texten.

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)

Våra mål i kursens början var breda och vaga. Vi ville leverera en interaktiv karta som visualiserade data och lade mycket fokus på att alla inblandade skulle känna att de kunde bidra. Efter prat med vår mentor satte vi en mer koncist minimal produkt, vilket var att ha en karta med en heatmap som uppdaterade sig beroende på ett urval av åldersgrupp. Vad gäller learning outcomes var det en del individuellt, beroende på vad man valde att jobba med under projektet, vissa ville t.ex. lära sig JavaScript, medans vissa ville jobba mer med data- och Serversidan, men överlag känner vi alla att vi har fått lära oss en del av det som intresserar oss.

Vad gäller vårt teamwork har det inte varit på topp. Det blev bättre mot slutet, men vi har alltid haft lite problem med att få ihop alla till träffar och möten. Ibland dök någon inte upp, utan förvarning, vilket gjorde att det lätt blev någon miss med hur vi skulle jobba i sprinten.

Tills nästa projekt ska vi bli bättre på vårt teamwork och vår kommunikation. Vi måste ta mer personligt ansvar vad gäller kommunikation och samarbete. Mer om detta kommer i 2.

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

Under hela projektets gång har våra user stories haft samma struktur. De har strukturerats på så vis att de är sorterade efter våra epics, med storlek, sprints de arbetats på och vem/vilka som har ansvar för uppgiften. I början av projektet försökte vi göra så små user stories som möjligt, så att vi inte skulle ta oss vatten över huvudet när det kom till arbetsbelastning. Vi märkte ett problem med detta senare, vilket var att flera user stories ofta byggde på varandra, vilket gjorde det svårt att kunna jobba parallellt. Eftersom vi var så långt in i projektet fick vi lite stressat tunnelseende och istället för att förbättra den agila processen och våra user stories körde vi på som innan för att bli klara. Detta ledde till problem så som att flera arbetade på samma del av projektet.

I framtida projekt kan vi bli bättre på att "dela tårtan" så att alla känner att de har något att göra på projektet. Vad vi hade behövt göra är att bygga user stories som hela tiden gav små inkrementella framsteg som inte nödvändigtvis byggde på varandra. Vi var nöjda med hur varje user story gav små framsteg, men vi borde ha delat in dem på ett sätt som inte gjorde att de berodde på varandra.

I framtiden skulle vi kunna ändra våra stories, antingen göra dem större, eller göra fler spår av stories. Vi var alldeles för statiska och rädda för förändringar i vårt projekt. Nästa gång ska vi ta oss tiden att stoppa, se oss omkring, reflektera mer över vad vi gör, och vad vi kan ändra i vårt arbetssätt för att effektivisera vårt arbete.

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

Detta är ett ämne vi inte rört på i våra Team Reflections. Det vi gjort som acceptance tests är att vi visar upp produkten på våra möten för respons, tycker gruppen det ser bra ut gör man sedan en Pull Request med sin kod. Innan koden pushas upp ska den

kollas igenom så den följer standard (mer om vår kodstandard kan läsas på 3.4).

Vi föreslår att man lägger in ett steg efter code review, innan pull requesten mergas in huvudbranchen, där product ownern behöver testa och godkänna att systemet som helhet fungerar som önskat med kodförändringarna. Man bör kanske även lägga in automatiska tester som testar själva slutprodukten, för att undvika regressioner i framtiden. När sedan product ownern är nöjd och godkänner pull requesten så mergas koden in och levereras till användarna.

Till nästa projekt ska vi se till att implementera acceptance tests där vi mer tekniskt analyserar arbetet. Vi tror också det kan vara bra att implementera dem i kod för att köras automatiskt i en CI-pipeline för att förbättra processen och kodstandarden.

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

De KPI:er som vi valde var: Estimated v.s. actual velocity, Backlog finishable with current velocity och User value estimate. Under projektet så använde vi väldigt sällan våra KPI:er. Faktum är att de var knappt mer än ungefärliga tankebanor innan vår TA påpekade, några veckor in att vi borde använda dem och att vi måste ha tre definierade. Vi var snabba med att sätta upp estimated v.s. actual och backlog finishable. Vi satte numeriska värden på hur mycket ansträngning vi gissade att varje user story skulle kräva för att bli klar och jämförde helt enkelt hur mycket vi fick gjort med vad vi hade planerat att få gjort. Detta gick ganska lätt att implementera eftersom vi redan hade "tröjstorlekar" på våra user stories och vi kunde snabbt översätta XS-XL till 1-5. Detta betydde dock inte att vi använde den något vidare. Vi diskuterade ofta översiktligt och ungefärligt, hellre än att exakt avgöra hur många poäng vi gissade mot vad vi gjorde. Vi kollade på hur vi hade presterat och konstaterade att vi borde hinna klart ett par gånger.

Först i kursens sista skede så spikade vi vår sista KPI, user value estimate. Varje user story fick ett värde emellan ett och fem där vi estimerade hur mycket värde just den user storien gav en bättre produkt för slutanvändaren. Denna spikade vi mer för att vi var tvugna och använde den knappt annat än att prioritera vad vi ville få klart inför redovisningen.

Vi borde förmodligen ha bestämt oss för tre KPI:er direkt i början av projektet. Vi var väldigt ivriga med att komma igång med själva koden och glömde bort att etablera allt i vårt agila ramverk. Hade vi etablerat dem tidigt och aktivt använt dem så kunde KPI:erna ha hjälpt oss att prioritera och planera bättre.

För framtida projekt så kan det vara relevant att i grupp bestämma de KPI:er man

använder och föra i en stående punkt i sin mötesordning att man analyserar dem. Trots att vi sällan uppleve det nödvändigt att använda dem så kunde de helt klart ha varit en stor tillgång om vi hade använd dem från början.

2 Social Contract and Effort

2.1 Your social contract

Vi har varit nöjda med det sociala kontraktet och har inte tillfört några ändringar till det under kursens gång. Redan vid vecka 3 märkte vi dock att det inte hade följts. Vi förväntade oss att det skulle bli bättre eftersom de arrangemangen som en del av gruppmedlemmarna var med och arrangerade var över. Det upplevdes som att det inte blev någon förbättring mellan vecka 4 och 7 då samma problem uppstod, att sociala kontraktet inte följdes helt. Det var inte alla som hade koll på när mötena var och missade att meddela om man inte kunde komma eller missade mötet. Dock vill vi ta upp de saker vi tycker vi gjort bra. Vi införde en dagordning för att göra våra möten smidigare, och i början av varje möte gjorde vi "rund bordet" då alla kunde prata om hur man mätte, om man var stressad eller egentligen vad som helst. Detta gjorde att vi höjde stämningen och lagkänslan.

I en ideell värld hade alla som kunnat delta på möten göra det, och de som inte kan skulle ha meddelat detta i god tid.

I framtiden bör vi vara strängare med att inte missa mötena och att meddela i tid om man inte kan delta. Vi borde även ha haft konsekvenser ifall man inte följde kontraktet. Ett sätt att se till att göra att folk tar konsekvenser för att komma sent är ett så kallat "strecksystem". Där den som är sen får ett streck och efter ett visst antal streck så skaffar denne fika för nästa möte. I retrospekt borde vi ha infört detta i vårt kontrakt under projektets gång, men vi misstänker att anledningen till att vi inte gjorde det var att vi ville fokusera på att bli klara, och hamnade i något sorts tunnelseende.

2.2 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)

Vi har inte hållit reda på antalet timmar vi har lagt ner under projektets gång. För det mesta har vi haft regelbundna möten där vi har diskuterat vad vi vill få gjort på kommande sprinten och hur mycket vi kommer hinna med att implementera. Det har uppstått krockar med våra user stories då en del har byggt på varandra. Detta har lett

till att en del av oss har behövt vänta på att kod ska pushas upp till Github vilket har minskat vårt produktivitet.

I våra första sprints lades mycket tid på att lära oss de olika teknologierna som vi behövde för att genomföra projektet. Vi bestämde oss för att ta en vecka utan att producera kod och enbart fokusera på att lära oss Javascript. Även under projektets gång har väldigt mycket tid lagts ner på att lära oss och förstå D3.

Istället för ta en hel sprint för att enbart lära oss Javascript borde vi direkt ha börjat med att försöka producera användbar kod då man lär sig bäst genom att testa sig fram och det hade lett till en mer produktiv sprint.

Om vi hade fått börja om från sprint 1 med projektet hade vi sett till att så snabbt som möjligt kunna påbörja arbete med user stories istället för att ta en break för att lära oss teknologierna men även sett till att en del av våra user stories inte byggdes på varandra för att lättare kunna jobba parallellt på dem. Inför ett liknande projekt i framtiden skulle man kunna välja samma teknologier för att snabbt komma igång och inte ha en break / sprint för att lära oss nya saker. Men i fallet man har en ny teknologi bör man börja arbeta med dem i ett sätt som bidrar till projektet och inte stannar upp processen.

3 Design decisions and product structure

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

Från projektets början har vi tänkt att vi ska göra statistik lättkonsumerbart för våra användare. Vår tanke var och är att genom att visa aktuell statistik från SCB så skulle användarna kunna bilda sig en uppfattning i olika frågor baserat på tillförlitlig data. Vi tänkte att om vi kunde kombinera bra data från SCB och öppna källkodsprojekt som är kraftfulla samt lättanvända, skulle vi kunna bygga en bra och användbar tjänst för vanliga medborgare. Vi testade de olika öppna källkodsprojektet, vilka krävde olika sätt att strukturera upp projekten rent källkodsmässigt. Men till slut hittade vi vårt primära verktyg, D3, som vi kände uppfyllde våra krav bäst. Därefter omstrukturerade vi projektet för att det skulle fungera så bra som möjligt med D3.

Vi känner oss nöjda med valet av verktyg. Teknologierna som vi valde passade väldigt bra in på de problem vi ville lösa. Vi har ingen ånger i vad vi valde i slutändan.

Då vi känner att vi gjorde rätt designval i början för att maximera kundnyttan känner vi att vi inte skulle behövt ändra något till nästa projekt. Till nästa projekt vet vi att D3 är ett bra bibliotek ifall vi vill skapa något liknande. I början tittade vi även på

biblioteket Leaflet, men vi bedömde att D3 skulle tjäna oss bättre. Vi har inte kunnat mäta detta på något konkret sett, mer än att verktygen har uppskattats när vi har pratat under våra gruppmöten.

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

Inledningsvis skrev vi knappt någon dokumentation alls, det ända vi skrev var några rader i en readme-fil, och några enstaka kommentarer i källkodfilerna. Anledningen är att vi fram till slutet av projektet fokuserade nästan helt på att ta fram olika prototyper, vilket därmed ledde till att vi hade en hög omsättning på koden. Det diskuterades inte explicit, men vi misstänker att vi inte hade velat lägga tid på dokumentation då vi inte visste om vi skulle använda koden vi skrev veckan därpå. Det vi gjorde för att förmedla hur olika moduler fungerade till varandra var att några satte sig ner då och då för att demonstrera hur koden fungerade, och hur det var strukturerat. I slutet av projektet, när vi var mycket mer säkra på att koden vi skrev skulle vara med i resten av projektet, dokumenterade vi samtliga funktioner vi skrivit. Vi lade även till instruktioner hur man startade och stoppade applikationen när vi implementerade att man inte längre behövde köra koden i vissa IDE:er.

Till nästa projekt så tänker vi att vi skriver dokumentation i form av källkodskommentarer som beskriver vad funktionerna gör, med exempel. Vilka funktioner man dokumenterar kan man vara selektiv med, till exempel kan man välja att endast dokumentera funktioner som fungerar som API:er för andra submoduler i projektet. Utöver det så tänker vi att vi även ska ha ett övergripande dokument som beskriver projektets arkitektur. Eventuellt om domänen är komplex behöver vi inkludera dokumentation även om den, men vad för dokumentation vi ska skriva vet inte förrän vi vet vilken domän. Tiden lagd på att dokumentera projektet behöver dock begränsas, då projektkurser bygger mycket på att man utvecklar MVP:er.

För att nå till våra mål om dokumentation tänker vi att man i början av projektet skriver ner sina tankar om arkitekturen man tänker sig. Dokumentet ska inte vara detaljerat, utan ge en snabb överblick. Tanken är att man lägger in det som ett acceptanskriterium i user stories där man ändrar arkitekturen att man uppdaterar arkitekturdokumentationen. Det ska även ligga på code reviewen att kontrollera att all relevant dokumentation har uppdaterats. En liknande regel inför när man har skapat en ny funktion, alternativt ändrat i en befintlig funktion, behöver man dokumentera detta i form av kodkommentarer.

3.3 How you use and update your documentation throughout the sprints

Se avsnitt 3.2 för hur vi hanterat och använt teknisk dokumentation.

3.4 How you ensure code quality and enforce coding standards

Vi hade redan från början ett arbetsflöde som såg ut som nedan

1. skapar en *fork* av huvudrepositoriet (Albins repository);
2. skapar en featurebranch på sin fork;
3. skriver kod;
4. laddar upp koden till sin kopia av repositoriet på Github;
5. skapar en pull request till huvudrepositoriet;
6. en annan gruppmedlem gör en code review; och slutligen när kodkvalitén är tillräckligt hög
7. mergar in koden.

Genom att tillämpa det här arbetsflödet så hade vi möjlighet att lägga in ett steg i flödet där vi kan kräva en code review. Vi förlitade oss på code reviews för att upprätthålla kodkvalité.

Vi skaffade inte en kodstilsstandard förrän i slutet av kursen. Det ledde till att vi behövde göra många ändringar i slutet för att följa kodstilen.

Till nästa projekt hade vi lagt tid på i början att sätta upp en CI-pipeline som triggas varje gång någon skapar en pull request mot repositoriet. Till pipelinen hade vi lagt till ett verktyg som kontrollerar koden mot en satt kodstil. Vi hade även tillämpat regeln att man skapar en ny pull request så fort man har skapat en commit på en feature branch. På det sättet så ser de andra gruppmedlemmarna direkt hur det går med koden, och man kan korta ner feedbackcykeln. Sedan ska samtliga gruppmedlemmar använda något versionshanteringsystem kompetent tillsammans med vårt arbetsflöde.

Först och främst efter man hade bestämt vad man ska göra hade man behövt bestämma vilket programmeringsspråk och vilken kodstil man ska följa. Därefter hade man behövt sätta upp en CI-pipeline för projektet. Man hade även antagligen behövt gå in mera

på detalj hur man ska använda git och git forgen som vi väljer att använda (antagligen Github eller Gitlab). Sedan hade man för säkerhetsskull ställt in på git forgen i inställningarna så att så många som regler som möjligt ligger inprogrammerade i våra verktyg, för att förhindra att någon missar något steg eller gör något de egentligen inte får göra. Vi har möjlighet att göra detta mera då de gruppmedlemmar som har varit obekväma med git vid kursstart nu är mycket mera bekväma.

4 Application of Scrum

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

I början av vårt projekt så bestämde vi oss för att ha ett rullande schema för rollerna som project owner och scrum master. Idéen var att vi kunde förvänta oss mindre kod från dessa under veckan då de fokuserade mer på code reviews och den agila processen. Senare under kursen så kunde vi konstatera att vi inte var tillräckligt bekväma i rollerna för att ordentligt veta exakt vad vi skulle göra. Eftersom våra sprints bara var på en vecka så hade vi dåligt med tid och låg motivation för att lära oss vad vi skulle göra eftersom vi inte skulle vara i sitsen igen. Detta är troligtvis en av anledningarna till att vår process var skakig och saknade många viktiga bitar, framförallt i början. Mot slutet av kursen så började vi ifrågasätta vårt beslut och i vecka 7 så konstaterade vi att om vi hade fått börja om så hade det varit bättre med två fasta poster. Vi ville se till att alla skulle få känna på rollerna, men det resulterade i att ingen av oss faktiskt fick erfarenhet av rollerna, varken som PO/SM eller som någon som jobbar med dem.

För framtida projekt så tror vi att det är viktigt att tydligt definiera roller och att vara försiktig med att rotera dem. Om man ser en stor vinning i att flera ska få testa på olika roller så är det viktigt att alla förstår vad som förväntas av den rollen och att de kanske rent av testat att behålla rollerna över flera sprints om syftet är att lära sig.

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

Vi jobbade med en scrum-board och frekventa möten och diskussioner i fokus. För kommunikation så använde vi Discord och när vi kunde så försökte vi ha våra möten på campus. Vi hade även (iallafall senare i kursen) tre KPI:er som man kan läsa om i 1.5. Vi kan inte hävda att ha utnyttjat alla hjälpmedel eller processer till sin fulla potential. Till exempel så var vi inte jättebra på att använda scrumboarden under pågående sprints utan brukade använda den mindre som ett levande bräde och mer som en checklista på vår sprint review.

Däremot så upplevde vi att vi fick ut en hel del från våra möten. Framförallt efter att vi skapat en fast mötesordning för att få struktur på våra möten. Det var med dessa möten som grund som vi ändrade takt och struktur för att alla skulle kunna göra sitt bästa. Under början av kursen till exempel så hade vi möten måndag och fredag. Vi märkte dock att detta blev väldigt stressigt och ändrade vår planering till möten måndag och söndag, eller att ha längre möten på måndagen för att avrunda den tidigare sprinten och starta den nya.

I 1.5 så diskuterar vi våra KPI:er och hur vi borde ha använt dem mer och tidigare så vi kommer inte att gå in djupare på dem. Som sagt så upplevde vi en stor hjälp av att ha en frekvent och öppen dialog om hur vi låg till. Däremot så var vår kommunikation via discord något bristfällig. Som vi nämnt i 2.1 så hade vi vissa svårigheter med att förmedla information eftersom flera av oss inte var duktiga på att hålla koll på discord-servern.

Inför kommande projekt så kommer vi att fortsätta ha frekventa möten med tydlig mötesordning. Vi kommer även att vara öppna för att förändra i pågående strukturer för att se till att allting funkar. När det kommer till att bryta det sociala kontraktet så diskuterade vi det i 2.1 och kom fram till att man hade kunnat vara strängare med konsekvenser av att inte följa reglerna vi satte upp för oss själva. KPI:er diskuterar vi också mer ingående i 1.5.

4.3 The sprint review and how it relates to your scope and customer value

Våra sprint reviews bestod av att varje person kort redovisade vad man jobbat på under veckan. Sedan såg vi över den gångna veckans user stories. Det var vanligt att våra effort estimations inte hade stämt och detta blev bra tillfälle att reflektera över detta. Det hände både att våra user stories var svårare och lättare än väntat. I och med att vår sprint planning var med i samma möte gick vi ofta snabbt till att diskutera ändringar på vår kanban board. Här fick veckans PO också tillfälle att diskutera med gruppen vad som var högsta prioritet inför nästa vecka.

En vanlig väg framåt var att först ändra/bryta ner överblivna user stories i syfte att få till ett jämnare flöde av uppdateringar på hemsidan under nästa sprint. En utmaning var att ordna så flera user stories kunde tas om hand parallellt för att maximera den kommande sprintens bidrag till customer value. Det hände att user stories som hamnat hos olika gruppmedlemmar visade sig bero på varandra och kom att hämma arbetsflödet.

Vi tog sällan explicit upp perspektiven scope och customer value på mötet. Detta hade eventuellt kunnat bidra till ett bättre fokus i diskussionerna. Detta tar vi med oss till kommande projekt.

4.4 Best practices for learning and using new tools and technologies

En de stora valen för oss var att implementera applikationen på en hemsida. Få av oss hade vana inom webbutveckling sedan tidigare. Tidigt i kursen så tog vissa av oss en vecka för att lära oss nödvändiga grunder i javascript. Detta var lärorikt men det märktes större skillnad när vi faktiskt vågade börja prova att skriva och leka med koden. Detsamma gällde de biblioteken som vi använde. Vi var mycket mer bekväma i att använda dem efter att vi hade fått sätta tänderna i dem än vad vi hade lärt oss av att läsa dokumentation kring dem.

Vår användning av Git och Github innebar också en del att lära för vissa av oss. För att snabbt få alla på banan höll Albin i en workshop under första veckan. Därefter lärde vi andra det som behövdes på vägen av skriva och commita kod. Vi bestämde första veckan att var och en skulle ha en fork från huvudrepot och skapa pull requests för att kunna ha code reviews och enklare kunna hantera merge-konflikter. Detta gjorde också att var och en fick jobba i git självständigt och lära sig det nödvändiga utan att riskera att förstöra ordningen i huvudrepot. Det var en lyckad strategi, men krävde en viss ansträngning från de flesta för att alla kunna bidra med konkret kod och utvecklas inom programmeringen.

Vi tycker oss därmed ha noterat att när det kommer till utveckling av program och applikationer så är en av de viktigaste sakerna att göra för ditt lärande att pröva dig fram. Kodning i vilket språk som helst är en färdighet som behöver tränas genom att utforska för att ta stora framsteg. Detta är troligtvis en stor poäng med att kursen är strukturerad som den är. Vi fick väldigt lite ledning i hur vår slutprodukt skulle se ut, vad den skulle handla om eller ens vad det skulle vara. Vi fick heller ingen vidare djupgående genomgång kring de tekniska färdigheter man bör ha för att bygga en hemsida. Detta gjorde att vi var tvungna att leta upp egna källor och komma fram till vad som funkar bäst för det projektet vi ville göra och hur vi lär oss det på bästa sätt. I många fall så tror vi att man bara behöver en grundläggande förståelse för att lära sig mycket snabbare genom att experimentera än att läsa.

För framtida projekt bör vi ta efter vårt sätt med inläringen av git. Att tidigt ha en liknande workshop på plats för exempelvis javascript och d3 hade kunnat hjälpa fler av oss att komma igång mycket snabbare.

4.5 Relation to literature and guest lectures

Vi baserade vår arbetsmodell mestadels på de föreläsningar, slides och övningar som vi hade i början av kursen. Exempelvis i fallet User stories togs principen rakt av från en föreläsning första veckan.

Överlag upplever vi oss ha jobbat med ganska minimal teoretisk bakgrund. Antagligen hade vi kunnat undvika ett par onödiga misstag på vägen om vi följt föreläsningar/seminarium mer flitigt. Däremot upplever vi att vi har lärt oss mycket under kursen från våra praktiska erfarenheter, och tack vare våra kontinuerliga träffar med TA höll vi oss med någorlunda röd tråd. Som tidigare nämnt 4.4, vi upplevde att vi lärde oss väldigt mycket genom att själva pröva oss fram.

References

- [1] Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J. & Thomas, D. (2001). Manifesto for Agile Software Development. Available from: <https://agilemanifesto.org> (Accessed: 25 Oct 2021).