

Reflektion över utvecklingen av mjukvara som  
möjliggör adaptiv farthållare och kolonnkörning  
med arbetssättet Scrum

**Chalmers tekniska högskola**  
Software Engineering Project, DAT255



**Svante Bennhage**  
**Kevin Brunström**  
**Jakob Erlandsson**  
**Alexandra Garrido Jaque**  
**Anton Hägermalm**  
**Anne Keller**  
**Simon Kärrman**  
**Joakim Mattsson**  
**Philip Nord**  
**Sanjin Slavnic**  
**Fredrik Viberg**  
**Jakob Wall**

## **Sammanfattning**

All teknik blir idag allt mer sammankopplad och intelligent. Två resultat av dagens teknik som utgör grundpelarna för självkörande fordon är Adaptive Cruise Control, ACC, och så kallad kolonnkörning. Detta projektet syftar till att genom utveckling av programvara som möjliggör ACC och kolonnkörning på platformen MOPED arbeta med arbetssättet Scrum. Då karaktären av projektet var ny för många av gruppens medlemmar lades tid på att förstå komplexiteten av problemet. Gruppen introducerades till Scrum under en LEGO-övning och denna har lagt grunden för hur gruppen valt att implementera Scrum som arbetssätt. Tre KPI:er valdes ut för att kontinuerligt mäta hur arbetet med Scrum gått. Under processen fattades en mängd tekniska beslut och resultatet av arbetet blev programvara i form av en androidapplikation som möjliggör ACC och kolonnkörning. Processen har kontinuerligt utvärderats och förbättrats. Avslutningsvis har gruppen kunnat reflektera kring de viktiga aspekter som format processen och tagit med sig reflektioner och kunskap inför framtida projekt av liknande karaktär.

# Innehåll

<b>1</b>	<b>Inledning</b>	<b>1</b>
1.1	Bakgrund . . . . .	1
1.2	Syfte . . . . .	1
<b>2</b>	<b>Arbetssätt och strategier</b>	<b>2</b>
2.1	Rollfördelning . . . . .	2
2.2	Tekniskt förarbete . . . . .	2
2.3	Scrum som arbetssätt . . . . .	3
2.3.1	Sprintar . . . . .	4
2.3.2	Arbetsuppdelning . . . . .	5
2.3.3	Interaktion med produktägare . . . . .	6
<b>3</b>	<b>Produkt</b>	<b>7</b>
3.1	Tekniska val . . . . .	7
3.2	Resultat sett till produktägarens målsättning . . . . .	8
<b>4</b>	<b>Utvärdering</b>	<b>8</b>
4.1	Prototyp . . . . .	8
4.2	Process . . . . .	9
<b>5</b>	<b>Diskussion</b>	<b>10</b>
5.1	Sprint retrospective . . . . .	10
5.2	Sprint review . . . . .	11
5.3	Relationen mellan produkt, process och intressent . . . . .	12
5.4	Strategierna ur KPI-perspektiv . . . . .	13
5.5	Strukturerings av grupparbete . . . . .	14
5.6	Interaktion med andra grupper . . . . .	15
5.7	Kontinuerlig reflektion . . . . .	16
<b>6</b>	<b>Slutsats</b>	<b>18</b>
	<b>Referenser</b>	<b>19</b>
	<b>Appendix</b>	
A	Dokumentation av retrospectives . . . . .	
B	Kommentarer på KPIer . . . . .	
C	Referering till kriterier för rapporten . . . . .	

# 1 Inledning

All teknik går idag mot att bli allt mer sammankopplad och intelligent[1]. Detta är en utveckling som inte minst gäller inom fordonsindustrin. Målet är där att ta fram självkörande fordon helt utan mänskliga förare. Drivkraften för detta är att effektivisera fordonstransporter och dra ner på både bränslekostnader och miljöförstörande avgaser. Ett av kraven för att detta överhuvudtaget ska bli möjligt är ett fokus på en hög säkerhet vad gäller fordonens körning. Kraven och önskemålen på det mjukvaruprogram som skall uppfylla detta i framtiden sätter i sin tur höga krav på den utvecklingsprocess som ska användas vid utvecklingsarbetet.

## 1.1 Bakgrund

Två resultat av dagens teknik som utgör grundpelarna för självkörande fordon är "Adaptive Cruise Control", ACC, och kolonnkörning. ACC är en teknik som redan idag finns i de flesta moderna bilar. Tekniken innebär att körmoment som farthållning på motorväg automatiseras. Kolonnkörning innebär att ett antal fordon kör i led och följer ett manuellt styrt ledarfordon i hastighet och styrning.

Ken Schwaber och Jeff Sutherland, författarna bakom "The Scrum Guide™" [2] menar att Scrum är ett ramverk vilket kan hantera komplexa adaptiva problem samtidigt som man på ett produktivt och kreativt sätt levererar produkter av ett så högt värde som möjligt. Scrum ger även upphov till snabba och mer korrekta reaktioner på plötsliga och oförutsedda problem som kan uppstå under arbetets gång.

"Mobile Open Platform for Experimental Design of Cyber-Physical Systems", eller förkortat MOPED, är med hjälp av tre stycken "Raspberry Pi" [4] en programmerbar utvecklingsplattform. MOPED:en är bland annat utrustad med en ultraljudssensor för att mäta avståndet framåt samt en kamera med möjlighet att läsa av omgivningen visuellt.

## 1.2 Syfte

Målet med projektet är att genom utveckling av mjukvara som möjliggör ACC och kolonnkörning för plattformen MOPED arbeta med den agila utvecklingsmetoden Scrum. Vidare är syftet med rapporten att redogöra för de reflektioner som gruppen tillsammans fört kring vad som gått bra, mindre bra och vad som kan förbättras vid en framtida process av liknande karaktär.

## 2 Arbetssätt och strategier

Då syftet med projektet så tydligt är att arbeta med och reflektera kring processen att skapa mjukvara har gruppen jobbat medvetet med att skapa, utvärdera, reflektera och förbättra arbetssättet kontinuerligt.

För att optimera grupparbetet och att sätta grunden för hur arbetet skulle fungera inom gruppen fastslogs inledningsvis ett socialt kontrakt. I kontraktet presenteras gruppens huvudsakliga rutiner tillsammans med åtgärder för potentiella problem under processen. Vidare diskuterades även gruppens ambitionsnivå. Det sociala kontraktet noterades och fick godkännas av gruppens samtliga medlemmar.

Det var tydligt att kommunikationen inom gruppen skulle komma att spela en viktig roll för arbetet. Gruppen valde att använda kommunikationsapplikationen Slack. Genom Slack kunde meddelanden och notiser snabbt nå hela gruppen. I Slack finns det möjlighet att skapa kommunikationskanaler för olika ämnen. För att underlätta arbetet skapades bland annat kanaler för generell information, Git-notiser samt separata kanaler för mindre subgrupper.

### 2.1 Rollfördelning

Vid projektets start tillsattes olika roller inom gruppen. Dessa roller bestämdes genom att gruppmedlemmar frivilligt nominerade sig själva och sedan fick gruppens godkännande.

Först valdes en gruppleddare med ett övergripande ansvar för att gruppens och arbetets struktur fungerade, samt för att all tillgänglig information når alla gruppmedlemmar, även andra grupper. Då mycket av gruppens arbete krävde tillgång till MOPED utsågs även en gruppmedlem som ansvarig för kommunikation jämt emot de andra grupperna angående byte av laddare, MOPED samt noteringar av plattformens eventuella hårdvaruproblem. Till sist utsågs en medlem ansvarig för att boka ett stort grupprum inför varje fredagsmöte.

### 2.2 Tekniskt förarbete

Efter introduktionen till projektet blev det tydligt vilket tekniskt förarbete som skulle komma att krävas. Karaktären av både hård- och mjukvara var vid projektets start ny för många av gruppens medlemmar. Det första målet blev att få igång hårdvaran och det operativsystem som styrde den. Till vår hjälp fanns viss samlad dokumentation kring hur MOPED:en fungerade. De tidigare kunskaper som vissa i gruppen besatt delades för att alla skulle få en så jämn grund som möjligt att arbeta utifrån och för att kunna ta till sig ny informationen på bästa möjliga sätt.

Redan tidigt i projektet krävdes att gruppen fattade beslut kring vilka tekniska vägar vi skulle komma att ta. I samråd med andra grupper fattades beslut kring hur vi skulle programmera. Möjligheten att programmera en plug-in via en server fanns. Dessutom kunde pythonscript skrivas och köras direkt på vår MOPED utan att kommunicera med en server. Under rubriken resultat och underrubriken tekniska val går vi igenom vilka val gruppen tog.

Gällande arbetet med Git valde gruppen att ha två olika repositories, ett som var vårt officiella samt ett för testsyfte. Detta gjorde att det fanns en struktur i Git repot samt en tydlig överblick över var det fanns kod under utveckling. Det nyttjades även flera branscher i det officiella projektet för att kunna utveckla specifika delar separat som till exempel applikationen. För att kunna ta del av vad andra medlemmar i gruppen gjorde fick samtliga notiser när något pushades upp till Git via en kanal för detta i Slack.

## 2.3 Scrum som arbetssätt

Då projektet till stor del syftar till att lära sig om process fick medlemmarna i gruppen ta del av arbetssättet Scrum, som är ett beprövat verktyg för att jobba med just en process.

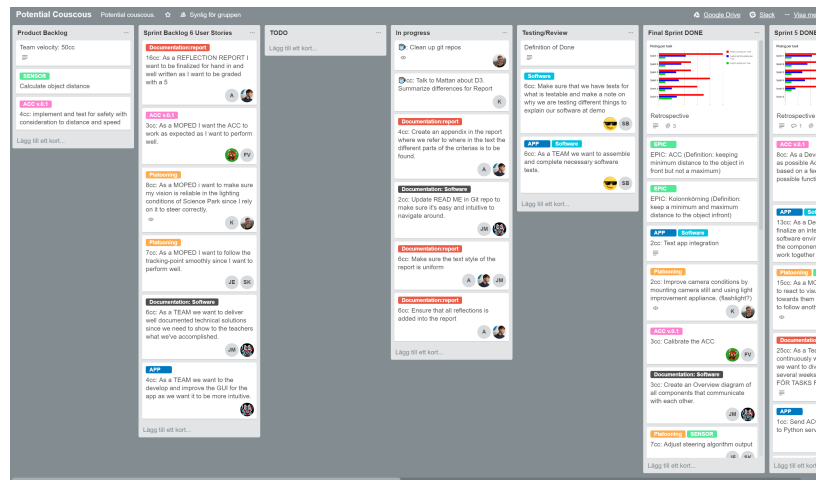
I början av kursen gavs en introduktion till Scrum där vi som deltagare fick ett uppdrag av en produktägare i form av att bygga en stad i Lego. Där visades det tydligt hur Scrum skulle användas i form av att varje del av Scrumarbetet presenterades. Det sattes upp user stories som var önskade av produktägaren, gruppen satte upp en Velocity för att veta hur mycket som kunde utföras under en sprint, mer ingående information om gruppens sprintar för projektet följer nedan. Sprintarna var mycket kortare här än i det stora projektet, här var de endast sexton minuter långa. Det var småskaligt, men det hölls fortfarande en kort reflektion i form av en sprint retrospective i slutet av varje sprint vilket gjorde att det tydligt visades hur en sprint kunde se ut. Det framgick även att det var viktigt att ställa mycket frågor till produktägaren för att vara säker på att man hade samma vision. Utifrån denna övning satte vi som grupp ett antal KPI:er utefter vad vi lärde oss. Gruppen valde att ett KPI där det skulle kollas hur mycket jobb som gjorts i onödan på grund av miss i kommunikationen med produktägaren kändes väsentligt. Det sattes även upp ett KPI för att se hur pass över eller underbemannade de olika subgrupperna var samt det sista KPI:et som beskrev vilka poäng som var högst, respektive lägst på de tasks som bestämts för sprinten i fråga.

Det bestämdes även att varje sprint skulle vara en vecka. Utefter den förutsättningen valde gruppen att ha daily scrums två gånger i veckan för att stämna av arbetet samt ett större planeringsmöte under fredag varje vecka för att planera inför nästkommande sprint samt reflektera över sprinten som varit. Detta var passande då mötet med produktägaren alltid ägde rum torsdagar. Det lades mycket tid i början på diskussion under fredagens möten men desto fler sprintar

### 2.3.1 Sprintar

Det fanns alltså två olika möten under en sprint, både daily scrums och det större sprint retrospective mötet. Gruppen valde att det passade bäst med två daily scrums per vecka där det varje gång ställdes 3 frågor. "Vad har vi gjort?", "Vad ska vi göra till nästa gång? samt "Vilka problem har vi?". Dessa frågor gick igenom effektivt till syfte att inom gruppen stämma av med varandra och se hur de olika smågrupperna låg till i processen.

Ett av de viktigaste verktygen under sprintarna blev den digitala arbetstavlan Trello. Orsaken till att denna valdes var främst för att möjliggöra arbete oberoende av plats och garantera tillgängligheten av arbetsuppgifter åt alla i gruppen. Applikationen gjorde det också enkelt att skapa ett arkiv över vad som avklarats. Dessutom blev det lätt att skapa en product-backlog som med hjälp av färgkoder kunde organiseras väl.



Figur 1: Gruppens Trello-tavla

### 2.3.2 Arbetsuppdelning

Under legoövningen blev det tydligt med vikten av en god arbetsfördelning. För att arbetet under projektet skulle kunna fortlöpa så snabbt och smidigt som möjligt valde gruppen som var bestående av 12 medlemmar att dela upp sig i subteam. Dessa subteam tilldelades olika arbetsuppgifter och på så sätt kunde arbetet parallelliseras och utföras snabbare. En ytterligare fördel med arbete i subgrupper är att det lättare går att hitta en arbetstid som passar för alla inblandade. Viktigt att understryka är dock de fördelar som finns med att sitta flera subteam tillsammans då det ger möjlighet att snabbt kunna hjälpa andra eller fråga andra om hjälp. Något gruppen också valde att göra för att vara effektiva och hjälpas åt var att parprogrammera. För att kontinuerligt utvärdera uppdelningen av subteam valde vi att betrakta uppskattad över- och underbemanning som ett KPI.

Då projektets syfte till stor del är att arbeta med agil mjukvaruutveckling och framförallt med Scrum startade gruppen med att ta ett beslut om att rollen som scrum master skulle roteras mellan sprintarna. Detta gav möjlighet för fler att testa hur det var att leda gruppens process. För att kontinuerligt förbättra scrum master-rollens ledarskap delades erfarenheter mellan de tidigare och den nuvarande scrummastern.

När arbetet i stort parallelliseras är det viktigt att arbetsuppgifterna delas upp på ett sätt som gör dem oberoende av varandra. Allt arbete delas upp i flera user stories som beskriver det önskade utfallet för sprinten i fråga.

Eftersom att det valts att jobba i mindre subgrupper så skulle det även planeras för vad dessa mindre grupper skulle göra. För att dela upp subgrupper delades först varje user story in i tasks. Dessa tasks uppskattades genom tekniken ”poker planning”. Efter detta kunde poängen för varje user story summeras. Beroende på varje user stories storlek formades olika stora subteams. Då en user story var för liten för ett helt subteam kunde flera user stories kombineras. Ett mål för gruppen har varit att skapa så små och tydliga tasks för varje user story som möjligt. För att mäta utfallet av denna önskan valde vi att utvärdera storleken av tasks som ett KPI.

Hur uppdelning av både user stories och tasks skulle göras på bästa sätt visades väldigt tydligt i början av kursen då Jan-Philipp fick samtliga medlemmar i gruppen att bryta ner user stories i mindre och mindre tasks i form av sin elefantcarpaccio-övning vilket visade sig vara effektivt då gruppen upplevde att de kunde ta lärdom av detta och jobba vidare med tekniken efter ny kunskap från denna övning.

Att låta alla arbetsuppgifter ingå i sprintplaneringen gynnade arbetsfördelningen i gruppen. Uppgifter som inte uppenbart hade direkt koppling till MOPED:en summerades ändå i vår sprint-backlog. Rapportarbete, planering och testning är några av de saker som gruppen planerade för. Detta gjorde att vår sprint-backlog på ett bättre sätt speglade den faktiska arbetsbelastningen för projektet

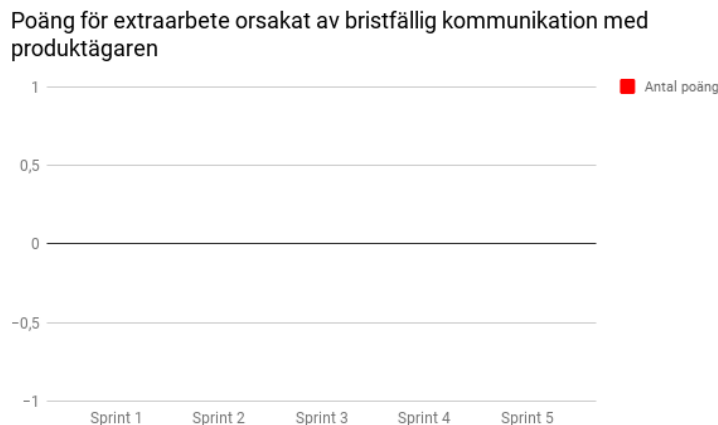


i stort.

### 2.3.3 Interaktion med produktägare

Sista dagen i varje sprint hade gruppen ett möte med produktägaren för att visa upp sprintens resultat, få feedback på arbetet samt diskutera vad som skulle fokuseras på närmast.

Under Legoövningen i början av kursen upplevde gruppen att det fanns mycket specifika åsikter och önskemål från produktägaren. Detta gjorde att en viss mängd extrajobb behövde utföras på grund av bristfällig kommunikation. Därav valde gruppen att skapa ett KPI för att kunna följa just kommunikationen med produktägaren i det faktiska projektet. Detta baserat på våra lärdomar från övningen där gruppen tydligt såg att det blev mindre effektivt om produktägarens önskemål inte kom fram tillräckligt tydligt. Det visade sig dock att denna KPI blev väldigt tom på grund av att produktägarens vision var ganska öppen och det därav inte kunde uppstå några direkta konflikter, till skillnad från Legoövningen där produkten till exempel skulle byggas i vissa färger. Det var svårt för gruppen att visa värde för produktägaren då hans vision enligt gruppen ansågs vara två stora epics. Detta gjorde att det från början av projektet fanns en tydlighet i vad som skulle uppnås och det uppstod inga nya uppgifter under projektets gång som det gjorde i Legoövningen.



Figur 2: Diagram för KPI 1

## 3 Produkt

Produkten som tagits fram under projektet är mjukvara i form av en androidapplikation för plattformen MOPED. Förutom styrning av fordonet möjliggör applikationen även körning med ACC samt i kolonn. Applikationens användargränssnitt är framtaget med fokus på intuitivitet. En sammanställning av de viktiga tekniska beslut som gruppen fattat följer i texten nedan. Dessutom jämför vi det faktiska resultatet mot produktägarens vision.

### 3.1 Tekniska val

Ett av de tekniska valen som gjordes var hur vi skulle programmera vår funktionalitet och som en direkt följd av det; vilket programmeringsspråk som skulle användas. Som nämnt ovan upptäcktes det under det tekniska förarbetet att det fanns två sätt att programmera MOPED:en. Det första alternativet var att sätta upp en server med syfte att kompilera program. Programmen kunde sedan i sin tur köras på MOPED:en. Detta sätt att programmera hade lett till att programmeringsspråket Java kunde användas vilket är det programmeringsspråk som gruppen gemensamt känner till bäst. Det andra alternativet var att skriva program i Python och köra dem direkt på MOPED:en. För att snabbt uppnå värde för produktägaren valde gruppen att programmera i Python och köra programmen direkt på MOPED:en. Efter diskussion med andra grupper gavs intrycket att det var både tidskrävande och svårt att sätta upp en server vilket bidrog till valet att fortsätta på samma spår under de första sprintarna. Det verkade inte heller finnas några tydliga fördelar med att programmera MOPED:en via servern. Dock valdes det att hålla beslutet öppet och ifall det uppkom incitament att byta från Python skulle frågan tas upp och diskuteras igen.

Med projektets småskaliga omfattning i åtanke valdes en enkel arkitektur för kodbasen. Gruppen avgjorde att de komponenter som finns inte behöver något robust eller utökningsbart system då all funktionalitet kan hållas överskådlig och kortfattad. En enklare arkitektur är även mer passande för gruppens begränsade kunskap inom Python, och ger stöd för de snabba omstruktureringar som kan förekomma när man arbetar med SCRUM.

För att uppnå den kolonnkörning som produktägaren beställt krävdes ett sätt att följa efter andra MOPED:er både lateralt och longitudinalt. Det var även ett krav att MOPED:erna kunde utföra detta utan att direkt kommunicera med varandra. Eftersom gruppen redan hade utvecklat ACC för longitudinal kontroll så valdes det därmed att utnyttja den kamera som satt monterad på MOPED:en för att identifiera och lateralt följa ett framförliggande objekt med hjälp av ett bildigenkänningsprogram. Andra alternativ för kolonnkörning övervägdes inte då små resurser redan tidigt åsidosattes för att utforska kameran och det snabbt fastställdes att lösningen var lovande.

Fjärrstyrning av MOPED:en löstes med hjälp av att skapa en ny applikation med influenser från den som redan fanns. Skälet till att skapa en egen var att viss funktionalitet saknades i den befintliga. Exempel på dessa är funktionalitet för att kunna aktivera samt avaktivera både kolonnkörning och ACC. I samband med att en ny applikation skapades skulle mer data skickas. För att kunna ta emot mer data krävdes också att en enklare server skapades på MOPED:en som i sin tur skickade informationen vidare till berörda program.

### **3.2 Resultat sett till produktägarens målsättning**

Efter sex veckosprintar och mycket arbete med både process och utveckling kunde gruppen visa en till stor del fungerande ACC-funktionalitet samt en kolonnkörningsfunktion då MOPED:en anpassar både hastighet och styrning till framförvarande fordon. Detta var en vision redan från start från produktägarens sida.

Gruppen samtalade med produktägaren efter demonstrationen av ovanstående funktioner. Han uttryckte tydligt att han var nöjd och imponerad av det som gruppen presterat. Vidare ansåg han att gruppens resultat gick helt i linje med hans målsättning.

Vi känner själva att vi presterade över vår egen förväntan sett till vad vi trodde att vi skulle ha möjlighet att visa fram vid demonstrationen. Det skall självklart nämnas att det finns möjlighet för utveckling samt förbättring men att gruppen är nöjda med helheten taget de problem som stötts på under vägen i beaktning.

## **4 Utvärdering**

Att både utmanas genom ny hårdvara samt ett nytt sätt att driva en process har varit spännande och gruppen har från sprint till sprint fått nya insikter och kunnat ta med sig lärdomar till nästa veckas arbete både när det gäller att jobba med vår MOPED samt när det gäller att utveckla hur vi arbetade med Scrum.

### **4.1 Prototyp**

Som grupp var vi måna om att så snabbt som möjligt skapa värde för vår produktägare genom ett första inkrement av programvara. Att vi valde att programmera direkt på MOPED:en utan en server gjorde att vi redan vid första mötet med produktägaren kunde demonstrera programmatisk körning. Det första inkrementet byggdes sakta på under nästföljande sprintar. Vid försök att programmera en mer kontrollerad körning uppstod dock flera svårigheter. Dels

integrationen med den ursprungliga kodbasen och dels hur en effektiv feedback-loop skulle utformas.

Under sprint tre delade vi in gruppen i mindre subgrupper som kunde jobba parallellt och där igenom fokusera på speciella områden i vår prototyp så som applikationsintegration, arkitektur och ACC-algoritmer. Detta gjorde att prototypen byggdes ut från flera håll samtidigt. Alla medlemmar kunde även bidra på ett tydligare sätt då de visste mer konkret vilka förväntningar som fanns på dem. I detta skede var det viktigt att ha en bra och tydlig kommunikation kring inkrementen för att alla olika delar skulle passa in med varandra. I slutändan ville vi leverera en stabil och enhetlig produkt utan känslan av flera ihopklistrade delar. För att undvika detta hölls daily scrum två gånger i veckan då det nya arbetet som gjorts på prototypen stämades av. Olika idéer kring produkten kunde tas upp för senare diskussion.

För att säkerställa leverans av en stabil produkt vid demonstrationstillfället valde vi att hålla en daily scrum varje dag under den sista sprinten. Detta resulterade i att arbetet med prototypen kunde parallelliseras trots att alla trådar samtidigt skulle knytas ihop till en enhet. Vid demonstrationstillfället hade vi en välfungerande prototyp med funktionalitet för både ACC och kolonnkörning.

## 4.2 Process

Under projektets gång utvecklades både arbetssättet och gruppens förhållningssätt till uppgiften. Den första sprinten kom att innebära en hel del kunskapssamlande då projektets natur krävde det. Då vi valde vägen som inte involverade någon server kunde vi redan vid första demonstrationstillfället visa upp värde för produktägaren i form av två simpla script vi skrivit och kunde demonstrera på MOPED:en.

Redan från första sprinten fokuserade vi på att använda tekniken elephantcarpaccio som presenterades under ett övningspass. Tekniken handlar om att bryta ner ett mjukvaruprojekt i mindre delar. Att applicera detta i vårt projekt innebar att bryta ner produktägarens vision i olika user stories som sedan kom att bli flera tasks. Användandet av tekniken har gjort projektet mycket lättöverskådligt och att det alltid funnits en klar bild över arbetsgången. Det har aldrig varit svårt att veta vad det är som behöver göras.

Redan under den andra sprinten stötte gruppen på problem med hårdvaran. Detta gjorde att user stories fick flyttas med in i nästkommande sprint. Det var även viktigt att gruppens velocity inte gick över max när både gamla och nya user stories sattes upp under sprintplaneringen. Efter att problemet med hårdvaran blivit känt för gruppen diskuterades en strategi för att undvika detta i framtiden. Dels skapade vi ett testscript som vi kunde köra på MOPED:en för att direkt se vad som inte fungerade. Ett annat sätt var att föra en tydlig diskussion med gruppen som tidigare arbetat med hårdvaran och därigenom få infor-

mation kring vad som fungerat respektive inte fungerat. Om hårdvaruproblemen inte verkade gå att lösa kunde frågan tas upp med tech-support.

Ett annat problem som identifierades under projektet var att det passade dåligt att ha daily scrums på måndagar och onsdagar. Efter att gruppen identifierat problemet valde vi att flytta våra stand-up-meetings till tisdagar och torsdagar. Anledningen till ändringen var att det mesta av arbetet gjordes mellan måndag och onsdag då torsdagar innebar sprint review och sprintplaneringen gjordes under mötet på fredagar. I efterhand kan vi diskutera kring hur det hade varit bättre att lägga våra daily scrum på tisdagar och onsdagar då detta skulle möjliggöra mer tid mellan veckans sista daily scrum och torsdagens demo.

Under en gästföreläsning av Christian Frithiof från företaget "8 Dudes in a Garage" fick vi presenterat för oss hur han använder Scrum i de mjukvaruutvecklingsprocesser han är involverad i. Frithiof berättade att han experimenterat med olika längd av sprintar. Längden på en sprint hade varierat mellan bara några dagar upp till flera veckor. I vår process har varje sprint pågått i en vecka. Även om vi inte fått möjligheten att förändra detta är den gemensamma upplevelsen att en vecka har fungerat bra som längd. Det optimala hade varit att låta en sprint pågå mellan måndag till fredag med planering på måndag och review och retrospective på fredag. Arbetsveckan hade då fått en tydligare start- och slutpunkt.

Det har under projektet blivit tydligt hur Scrum kan appliceras på olika typer av arbetsprocesser. Till exempel har rapportarbetet strukturerats genom att bryta ner tasks och sammanställa dessa i en trello-board. På detta sätt har arbetet med rapporten lätt kunnat överskådas.

## 5 Diskussion

Under projektets gång har vi kontinuerligt jobbat med att dokumentera våra framsteg och motgångar inom projektarbetet, vår applicering av Scrum, vårt samarbete inom gruppen och samarbete med andra grupper. Med hjälp av denna dokumentation har vi sedan reflekterat över och diskuterat kring exempelvis hur vår Scrum-process har utvecklats från första till sista sprinten, vad som har gått bra, mindre bra, varför vi tror att det har gått som det har gått samt hur vi kan göra för att det ska gå bättre.

### 5.1 Sprint retrospective

Att kontinuerlig reflektion är av stort värde har visat sig tydligt för hela gruppen genom detta projekt. Förståelsen för hur viktigt det är att reflektera över det arbete man gjort och ta med sig de lärdomarna framöver är av större vikt än vad många i gruppen trott innan vi gick in i projektet. Genom att reflektera tvingas man som grupp att tänka över vad man presterat under veckan som gått och hur

effektiva man varit. Gruppens reflektioner har skett under fredagens stormöten där vi haft sprint retrospectives.

Vid projektets start var reflektioner över utfört arbete fortfarande något som gruppens medlemmar inte jobbat aktivt med i tidigare arbeten. Vi blev under arbetets tid bättre och bättre på att reflektera över sprintens prestationer men fortfarande finns plats för förbättringar. De mest konkreta reflektionerna försökte gruppen attackera på en gång genom att hitta en direkt lösning på problemet som upplevts.

Under tiden för reflektion var det dock lätt att komma in i en rutin där gruppmedlemmarna inte lagt lika stor tid för eftertanke när det gällde reflektionerna och istället för att tänka över vad som verkligen gått bra och vad som gått dåligt och vilka möjligheter till förbättringar som fanns, så blev kommentarerna till veckans sprint istället att allt hade gått bra eller att man valde att hålla med en tidigare medlem i dennes åsikter. En möjlighet till förbättring skulle kunna vara att lägga mer tid för eftertanke över vad som faktiskt gått bra, inte bara att det gått bra. Något som skulle kunna ha motverkat detta var att på ett bättre sätt förbereda sina reflektioner. Det hade varit möjligt att reflektera över vad som gick väl eller mindre väl under veckans gång, anteckna de tankarna och ta med sig till mötet. Då hade tiden för reflektion varit längre och det hade inte blivit lika påtvingat att komma fram till en bra reflektion under den korta tid som fanns tillgänglig på mötet.

Något som gruppen hade kunnat förbättra för projekt av samma kaliber i framtiden är uppföljning. Vi har inte följt upp och sett efter att det som faktiskt kom upp under fredagens sprint retrospective verkligen åtgärdades och förbättrades. Självklart tog alla medlemmar till sig de utvecklingsmöjligheterna som kom upp men det fanns ingen rutin i gruppen för att kolla upp att dessa förfiningar av processen faktiskt utfördes.

## 5.2 Sprint review

En av de viktiga beståndsdelarna i arbetssättet Scrum är det som kallas sprint review. Händelsen kan liknas vid en demonstration och en avstämning. Teamet som arbetat med att skapa ny funktionalitet får en möjlighet att visa upp denna för produktägaren. Vidare ges också tillfälle för direkt kommunikation mellan de två parterna då exempelvis produktägaren kan ställa frågor kring det nya inkrement som presenteras.

Under vårt projekt väljer vi att se de aktiviteter som skedde på torsdagar ute vid Chalmers på Lindholmen som projektets sprint review. Alla dessa innebar avstämning gentemot olika externa aktörer. Vi hade möjlighet att demonstrera ny funktionalitet för produktägaren och samtala kring den fortsatta processen. Dessutom gavs tillfälle att på ett konstruktivt sätt diskutera processen och arbetssättet med någon av lärarna på kursen. Eftersom syftet med projektet förutom att skapa en produkt också var att jobba och reflektera kring Scrum

blev samtalet med lärarna en slags avstämning av processen. Till sist hölls ett möte mellan de tolv gruppernas scrum master. Mötet kallades Scrum of Scrums. Under mötet stämde grupperna av processen mot varandra och diskuterade viktiga gemensamma riktlinjer.

Projektets natur gav upphov till en inledningsvis ganska tävlingsinriktad ton mellan grupperna. Eftersom alla grupper gjorde samma projekt kunde resultaten lätt jämföras mot varandra. Dessutom befinner vi oss i en skolmiljö där många studenters fokus är mycket goda studieresultat. Gruppernas samarbete förbättrades under processen och vi kan konstatera att ett mer öppet samarbete mellan grupperna gynnade alla inblandade. Kanske hade grupperna kunnat komma ännu längre om samarbetet kommit igång direkt vid projektstart.

Det visade sig svårt att vid varje sprint review visa nytt värde för produktägaren. Produktägarens vision var två komplexa och stora uppgifter. Dessa fick gruppen själv bryta ner till mindre deluppgifter. Varje deluppgift var inte utformad att på ett tydligt sätt ge ett ökat värde för produktägaren utan snarare att komma närmre produktägarens vision. I vårt projekt var produktägaren så pass insatt i komplexiteten av de uppgifter som fanns att han kunde se värdet i det vi gjorde utan vidare argumentation eller förklaring. Om produktägaren inte hade varit lika insatt hade vi på ett bättre sätt behövt förklara hur vi delat upp uppgifterna och vilket värde som uppnås efter varje avklarad deluppgift.

Under hela processen har hårdvaruproblem skapat försenade leveranser och försvårat arbete. Detta har även försvårat momentet sprint review då vi under vissa demonstrationer inte kunde visa ett konkret inkrement utan fick snarare berätta om vad vi gjort. Problemen hade kunnat minska om grupperna tillsammans snabbare hade försökt lösa hårdvaruproblemen och jobba runt dessa. Till exempel kunde initiativ tagits för att skapa ett dokument med samlad information kring hårdvaran tidigare.

### 5.3 Relationen mellan produkt, process och intressent

Redan när projektet tog sin avstamp i form av Legoövningen som introducerade Scrum trodde sig gruppen ha en tydlig bild av hur arbetet skulle fortlöpa samt hur interaktionen med produktägaren skulle ske. Det visade sig dock inte vara fallet.

De möten med produktägaren som var under Legoövningen skiljde sig från de möten som var med produktägaren under projektet med MOPED:erna. Under Legoövningen upplevdes kraven vara mer specifika samt det upplevdes att det vid varje möte kom en påbyggnad i form av nya uppgifter som skulle utföras i den kommande sprinten. Därav hade gruppen en tydlig förväntan om att de möten som skulle hållas med Kenneth skulle vara av liknande karaktär. Dock visade det sig att Kenneth redan från start lade samtliga kort på bordet och tydligt beskrev vad han förväntade sig att se under slutpresentationen. Detta var något gruppen inte trodde skulle ske och därav blev förväntningarna inte

uppfyllda och de möten som hölls inte som gruppen tänkt sig. Mötena som under torsdagarna hölls gjorde dock att gruppen istället tydligt kunde försöka visa och framföra värde till produktägaren i form av utveckling av produkten samt stämna av att han fortfarande hade samma vision och tanke gällande slutdemonstrationen. Gruppen fick alltså ut information varje möte men hade förväntat sig mer instruktioner.

Detta visar tydligt på hur viktigt det är att stämna av sina förväntningar samt att det är viktigt att inte göra antaganden baserade på tidigare erfarenheter, trots att de kan vara av liknande karaktär. Något som detta lärde oss är alltså att man bör gå in till exempelvis möten utan några antaganden då de antagande man gör kan vara baserade på felaktig information. Detta är något som gäller processen i stort och som samtliga medlemmar tar med sig till framtida projekt.

## 5.4 Strategierna ur KPI-perspektiv

Från Legoövningen var en av de största problemen som identifierades vår bristande kommunikation med produktägaren angående implicita krav som ställdes på produkten. Under denna övning bestod dessa krav av bland annat färgenhetlighet och dimensionskrav. Utifrån detta utformades den första strategin som syftar till att förbättra kommunikationen med produktägaren. Den tillhörande KPI:n skulle fånga upp gruppens felaktigt förbrukade resurser på grund av antaganden istället för kommunikation och feedback. Däremot märktes det redan efter ett par sprintar hur denna KPI:n inte var relevant utformat för detta projektet. Inte en enda sprint review identifierade några missuppfattningar mellan gruppen och produktägaren vilket resulterade i ett helt tomt KPI-diagram. Detta skulle visserligen kunna tolkas som att gruppen helt eliminerat kommunikationsproblem redan innan projektet börjat. Men troligtvis handlar det snarare om skillnaderna mellan de olika projektens produktägarnas karaktär. Mängden implicita krav skiljde sig kraftigt mellan de olika produktägarna vilket i grunden beror på utformningen av projekten. I Legoövningen fick gruppen bygga ett verk från grunden, vilket ökar antalet variabler som slutprodukten beror på. När det gällde att bygga en mjukvara för styrning av MOPED:en, där produktägaren inte bryr sig någonting om de enstaka komponenterna utan ett resulterande beteende, minskar det kommunikationsklyftan då beskrivningen av MOPED:ens beteende är simplare.

Någonting som gruppen upptäckte fungerade väldigt bra på Legoövningen var nedbrytningen av en user story i mindre parallella uppgifter. Detta ökade produktiviteten markant då gruppen hade ett överskott av resurser i början av övningen. För att säkerställa att alla tolv medlemmar i gruppen skulle kunna bidra och arbeta autonomt definierades den andra strategin. Att bryta ner alla user stories i mindre uppgifter som tidsuppskattas och sedan summeras förenklar dels uppskattningen för hur omfattande user storyn i fråga är samt uppdelning inom den grupp som jobbar med just det kortet. För att mäta gruppens ut-



veckling gällande väldefinierade och fristående uppgifter valdes den tillhörande KPI:n att mäta tre värden: poängen av den största uppgiften, minsta uppgiften samt medelpoäng för alla uppgifter som definierats i en sprint.

Ett problem som följer med parallellisering är arbetsfördelningen mellan de mindre grupperna. Under Legoövningen kände vissa gruppmedlemmar att de inte alltid visste hur de skulle kunna bidra. Trots en uppskattning av arbetstid för user stories sker vid sprintplaneringen ville gruppen med den tredje strategin synliggöra vem i gruppen som skulle ansvara för vad. Detta var menat att möjliggöra kartläggning av över och underbemanning. KPI:n som skulle mäta detta baserades på en subjektiv bedömning varje delgrupp fick göra i slutet på en sprint. KPI:n kan i efterhand kritiseras då det uppmätta värdet är fullständigt subjektivt, men den skapade en diskussion vid varje sprint review då den tvingade grupperna att reflektera över hur veckans arbete gått och hur mycket tid som lagts i förhållande till vad som faktiskt var planerat.

Av de tre KPI:er som gruppen satte anses de alla kunna förenklas för att bli mer effektiva och då uppnå att de blir ännu tydligare samt att de blir mer mätbara. Flera gästföreläsare har pekat på hur KPI:er fungerar bäst när de minimeras ner till ett nyckelvärde istället för att ta fram irrelevanta KPI:er som inte blir tillräckligt mätbara. Om vi skulle tänka om och ordna en ny strategi som skulle passa bättre in på projektet hade kommunikation med andra grupper skapat större nytta för oss att analysera med hjälp av ett KPI då samarbetet mellan grupperna fick en stor betydelse.

## 5.5 Strukturering av grupparbete

En viktig del för att uppnå ett lyckat projekt var att säkra att grupparbetet fungerade väl. Detta försökte gruppen att säkerställa på flera olika vis. Ett första steg i rätt riktning var att sätta ett tydligt gruppkontrakt där gruppen även stämde av sina förväntningar. Det kändes viktigt att alla förstod vilka förutsättningar gruppen gick in i projektet med samt vad alla andra kände i förhållande till arbetsinsats så att vi tillsammans kunde veta vilken nivå som vi skulle lägga oss på.

I en grupp med så många olika medlemmar så är struktur otroligt viktigt. Genom att sätta en tydlig struktur i arbetet så visste alla hela tiden veta vad som behövdes göras. Ett sätt att strukturera upp detta på var genom att nyttja verktyget Trello. Trello var ett bra verktyg för gruppen att kunna implementera Scrum. Den stora gruppen delas upp i mindre delgrupper där varje grupp tog på sig en eller flera av veckans uppgifter. Under den första sprinten valde den stora gruppen att endast dela upp sig i två större grupper, men det visade sig tydligt att detta inte gav någon bra struktur i arbetet. För att förbättra detta och ändra inför veckan efter delades tolvmannagruppen in i fyra mindre grupper med tre i varje grupp som tog på sig att lösa uppgifter inom ett särskilt område tillsammans. Detta visade sig vara ett effektivt sätt att arbeta

på och grupperna kunde prestera mer än i första sprinten. Med det sagt så såg alla grupper självklart till att hjälpas åt över gränserna mellan grupperna vid problem.

Uppgifterna som skulle utföras under veckan sattes upp under gruppens stormöte på fredag där huvudfokus var att planera inför nästa sprint, men där det också var viktigt att se till att samtliga medlemmar i gruppen mådde bra samt att det inte varit en för hög arbetsbelastning i föregående sprint. För att stämma av hur de olika smågrupperna låg till under sprintens gång hade de mindre grupperna möjlighet att berätta vad de gjort, vad som skulle komma närmast och vad för problem som stötts på under daily scrum.

Detta var otroligt effektivt då alla snabbt hade möjlighet att få en inblick i vad de andra grupperna gjort sedan sist samt se hur de olika delarna man jobbat på kunde integreras med varandra. För att hålla dessa möten korta och informativa hade gruppen varje vecka utsett en scrum master, som hade ansvar för alla möten under sprintens gång samt för sprinten. Gruppen valde att varje vecka ha en ny scrum master för att olika personer skulle kunna prova på rollen och utvecklas. Detta var fördelaktigt då många personer fick prova på att ta ansvar för att leda processen. Det som kunde ses som en nackdel var att det inte blev en konsekvent struktur för varje sprint på grund av att varje scrum master gjorde en egen tolkning av hur sprinten skulle läggas upp på bästa sätt. Hade vi gjort om projektet hade vi dock valt att göra på samma sätt då detta är utvecklande för personerna som fick leda processen.

Utöver dessa delar var det mycket viktigt att kommunikationen mellan medlemmarna skulle fungera väl och därför valde gruppen som tidigare nämnt att kommunicera genom Slack. Det visade sig vara väldigt behändigt att föra kommunikationen mellan varandra på detta sättet då det enkelt kunde ses var informationen fanns och om det fanns någon ny information att ta del utav.

Enligt ovan har det varit flera delar som har gjort att grupparbetet har varit välfungerande. Det spelar en stor roll med en likvärdig ambition, struktur samt tydlig och frekvent kommunikation. Något som vi också inser vara av stort värde är att medlemmarna i gruppen känner varandra sedan tidigare och därför även litar på varandra. Detta gjorde att vi förlitade oss på varandra samt varandras kompetenser och därav kunde vi fokuserat jobba med delgruppens mål utan oro för de andra gruppernas uppgifter.

## 5.6 Interaktion med andra grupper

Det är efter kursens gång tydligt att det skett en stor utveckling av förmågan att samarbeta grupper emellan. Det första testet av gruppernas förmåga till samarbete var under Legoövningen. Den övningen visade tydligt att ingen av grupperna hade beaktat att samarbete kunde leda till en större gemensam framgång för alla. En stor majoritet av grupperna fokuserade på sin egen framgång och när detta tankesätt skapade problem stod många grupper helt utan lösning. Efter

ett tag förstod bland annat vår grupp att det fanns fördelar med samarbete. Alla tjänade på att byta Lego med varandra. I efterhand kan denna övning ses som en metafor för hur kursen i stort har varit, där Lego kan representera kod och kunskap.

Under projektets början fokuserade många grupper på att förstå systemen. För vår grups del var kommunikationen med andra grupper ytterst begränsad under denna period. Detta ses till en vis del bero på att gruppen inte visste om det fanns något att tjäna på med att dela med sig så tidigt. Dock ledde bristen på samarbete till att många grupper började gå sin egen väg och bestämde själva vad som ansågs viktigt. Här hade ett mer omfattande samarbete antagligen kunnat leda till att det uppstod en gemensam standard tidigare, vilket på lång sikt hade gett bättre förutsättningar för att uppnå produktägarens mål.

Allt eftersom projektet fortskred så ökade dock samarbetet. Till en viss del ses detta bero på att det helt enkelt blev oundvikligt att inte samarbeta. Grupperna blev tvungna att enas om en gemensam standard för vissa tekniska lösningar. Parallellt med detta ökade även samarbetet för att lösa andra problem. Ett av dessa var problem med hårdvaran, där grupperna satte upp gemensamma dokument som innehöll lösningar på återkommande problem. Detta gjorde att varje enskild grupp slapp lägga tid på att hitta samma lösning som en annan grupp redan upptäckt, vilket var gynnsamt för alla. Om gruppen hade fått evaluera sin samarbetsförmåga med andra grupper för att se till hur den skulle kunna förbättras om projektet skulle genomföras på nytt, anser samtliga medlemmar att det hade varit mer effektivt och lönsamt att från start interagera med andra grupper och se till att tidigt jobba tillsammans och kommunicera med varandra gällande tillvägagångssätt samt lösningar på problem. Det skulle i större utsträckning ha delats kunskap och vissa problem hade kanske kunnat undvikas.

Ett tillfälle där det mindre bra samarbetet mellan grupperna kunde tagits i beaktning var under mötet med alla scrum masters, Scrum of Scrums, under varje sprint review. Där hade det funnits en möjlighet för att tillsammans jobba fram en strategi för hur bästa möjliga samarbete skulle uppnås.

## 5.7 Kontinuerlig reflektion

Under kursen har stor vikt lagts på att utvärdera och reflektera över den process vi befunnit oss i. Fyra leveranser har på olika sätt gett oss möjlighet till reflektion. Det kontinuerliga reflekterandet har delvis gynnat projektets framfart men kanske främst arbetet med processen och Scrum. Reflektion blev en väldigt naturlig del av projektet. Processen har tydligt kunnat förbättras då den kontinuerligt utvärderats och förändrats efter gruppens gemensamma reflektion. Förbättrandet av processen kan liknas vid en feedback-loop.

Innan projektet med MOPED:en introducerats fick vi delta i en Legoövning med syfte att introducera oss till Scrum. Efter denna övning skulle lärdomar

tillsammans med tre strategier för att implementera Scrum samt KPIer för att mäta dessa samlas till en första leverans. Legoövningen fungerade som en mycket bra introduktion till Scrum. De svårigheter som fanns kunde lätt identifieras. Leveransen gav oss i gruppen en möjlighet att reflektera över hur vi skulle lösa eventuella liknande misstag. En tydlig lärdom var att verkligen hålla en tydlig kommunikation med produktägaren för att eliminera risken för missförstånd och extraarbete. Detta är någonting vi tydligt reflekterar kring efter Legoövningen. Under projektet har det däremot inte varit ett lika stort problem med missförstånd mellan produktägaren och arbetslaget som det var under Legoövningen. Inför framtida mjukvaruutvecklingsprocesser är dock detta någonting som alltid bör beaktas. Något som gruppen även tog med sig från denna övning var att det visade sig väldigt viktigt att arbeta tillsammans som ett stort team, detta tog gruppen med sig vidare till det stora projektet kring MOPED:en. Summerat så kan svårigheterna kring den första övningen nu i efterhand ses som bra erfarenheter trots det inte var samma problem som uppstod under arbetet kring MOPED:erna.

Projektets andra leverans bestod i att skapa en initial backlog. Gruppen skrev även en vision. Fördelarna med skapandet av backlogen var att bryta ner produktägarens vision i mindre och mer tydliga delar för att få in hela teamet på samma bana. Detta gjorde att det skapades ett värde för gruppen omgående. Visionen skrevs till syfte för att kunna abstrahera från nerbrytningen av tasks och se till det stora målet i projektet. Detta är något som gruppen anser sig kunnat jobba mer med och återgå till oftare i processen. En ytterligare komplikation med detta var att skapa user stories, dock hade gruppen som tidigare nämnt stor nytta av Jan-Philipps elefantövning.

Den tredje reflektionsuppgiften var att tänka över hur arbetet gått när halva tiden passerats. Det viktiga med denna uppgiften ansåg gruppen var att kunna reflektera över processen i stort. Detta blev en tid för tanke över vad som gått bra och vad som gått mindre bra samt hur vi under arbets gång förbättrat oss. Det negativa med uppgiften var att reflektionen blev mer generell då vi som grupp inte hade jobbat tillräckligt mycket med reflektionerna i det stadiet och inte förstått vikten av en mer djupgående reflektion. Det blev en väldigt generell bild av vad som gått bra och dåligt och inte så tydligt vilka specifika delar som gruppen ansåg sig förbättrat under processens gång. Att jämföra denna reflektion med en annans grupps reflektioner kring deras process gjorde att vi fick upp ögonen för hur viktigt det var med reflektion då den gruppen vi jämförde med var mer konkreta i deras tankar och hade reflekterat över mer specifika saker som gått bra respektive dåligt.

När det beträffar gruppens sista leverans i form av demon blev vi positivt överraskade över att det blev så bra som den blev. Under den slutgiltiga sprinten hade gruppen inte haft så mycket möjlighet att testa de sista funktionerna som innefattade kolonnkörning vilket gjorde att det fanns en viss nervositet över hur resultatet skulle bli. Dock visade det sig att gick mycket bra vid presentationen och att produktägaren var nöjd.

## 6 Slutsats

I detta projekt har vi fått lära oss hur man tar till sig en stor uppgift och bryter ner den i mindre beståndsdelar för att kunna utveckla den. Vi har även lärt oss hur man jobbar processdrivet med hjälp av att nyttja metoden Scrum.

Vi har tagit med oss många lärdomar när det kommer till vikten av bra struktur i ett grupparbete, samt samarbete i en större grupp för att kunna lösa en uppgift av mer omfattande karaktär. Vi har genom ovanstående också lärt oss vikten av reflektion för att kunna utvärdera och förbättra en process, för att kunna jobba ännu mer effektivt vecka efter vecka oavsett typ av projekt.

## Referenser

- [1] M. Malmqvist, ”Så många miljoner prylar kopplas upp – varje dag”, *ComputerSweden*, 2016. [Online] Tillgänglig: <https://computersweden.idg.se/2.2683/1.650672/uppkopplade-prylar-iot>, hämtad: 2017-10-02.
- [2] J. Sutherland K. Schwaber, *The Scrum Guide<sup>TM</sup>; The Definitive Guide to Scrum: The Rules of the Game*, 2016, [Online] Tillgänglig: <http://www.scrumguides.org/scrum-guide.html>, hämtad: 2017-10-04.
- [3] *MOPED: Mobile Open Platform for Experimental Design of Cyber-Physical Systems*. [Online]. Tillgänglig: <https://moped.sics.se/>, hämtad: 2017-10-04.
- [4] Raspberry Pi Foundation, *Raspberry Pi*, [Online]. Tillgänglig: <https://www.raspberrypi.org/>, hämtad: 2017-10-11.

## Appendix

### A Dokumentation av retrospectives

Dokumentationen av våra retrospectives för varje vecka hittas i vårt git repository under mappen dokument.

### B Kommentarer på KPIer

Våra tasks blir mindre och bättre fördelade, detta i kombination med att både över- och underbemanningen närmar sig noll visar på att vi också blir bättre på att estimerar tidsåtgången för olika typer av tasks. Optimalt är att ha maximal velocity på en task bara strax över medel och minimal strax under. Vi trodde att projektet skulle ha en struktur som mer liknade Lego-övningen där vi hade mer kontinuerlig kommunikation med produktägaren. Därför är vår sista KPI inte så användbar då vi endast träffar produktägaren en gång per vecka och har ganska lång tid att ställa väl formulerade och heltäckande frågor. Detta i kombination med att alla andra grupper jobbar mot samma mål leder till att visionen för både oss och produktägaren är väldigt klar och missförstånd blir knappt några alls.

På sista sidan i Appendix följer bilder på samtliga tre KPI:er.

### C Referering till kriterier för rapporten

Nedan följer en dokumentation över var det i texten kan hittas beskrivning och reflektioner som täcker de givna kriterierna som är presenterade nedan.

#### **Application of scrum**

Se sektion 2.3.

#### **Roles, team work and social contract (relates to D1B)**

Se sektion 2.1

#### **Used practices (pair programming, stand-up meetings, etc.)**

Beskrivs löpande i texten, men kan bland annat hittas under sektion 2.3.1 samt under 2.3.2.

#### **Time distribution (person / role / tasks etc.)**

Se sektion 2.3.2

#### **Effort and velocity and task breakdown**

Kan bland annat hittas under sektionen 2.3.

#### **Reflection on the sprint retrospectives**

Se sektion 5.1

**Documentation of sprint retrospectives, 0-1p**

Dokumentation över samtliga retrospectives hittas i gruppens git repository.

**Reflection on the sprint reviews**

Se sektion 5.2

**Best practices for using new tools and technologies**

Beskrivs under sektion 2.2.

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

Se sektion 5.3

**Relating your own process to literature and guest lectures**

Beskrivs löpande i texten, nämns bland annat i sektion 2.2 samt i sektion 5.4.

**Evaluation of D1 - D4**

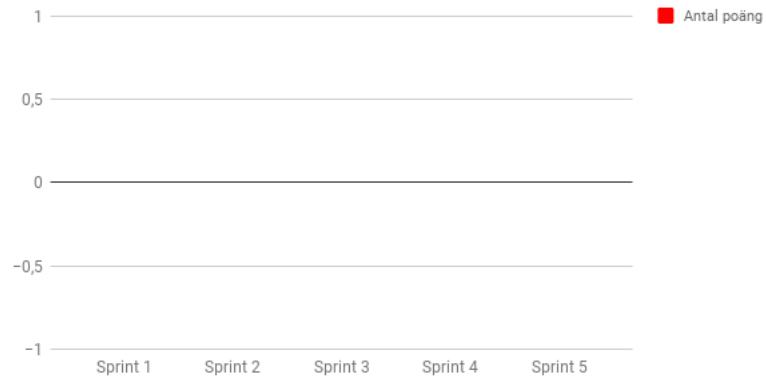
Se sektion 5.7

**KPI charts, 0-1p**

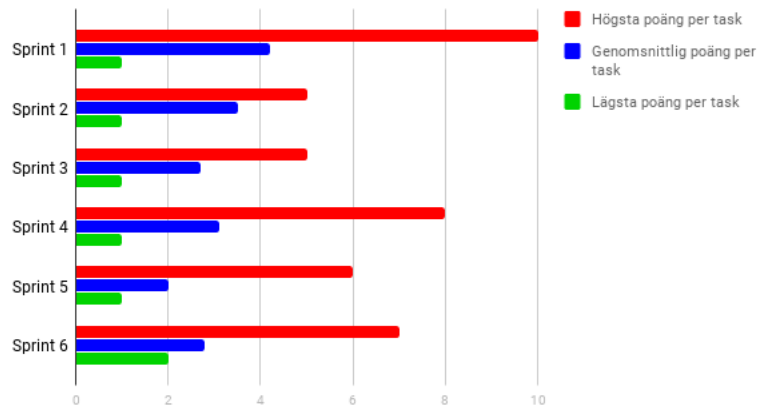
Se bilder på slutgiltiga KPI:er nedan.



### Poäng för extraarbete orsakat av bristfällig kommunikation med produktägaren



### Poäng per task



### Fördelning av arbetsresurser

