

Stakeholder value

Completeness

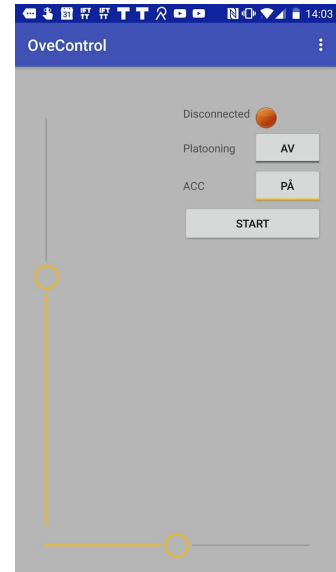
Vision (8/9-2017) *“Att på sex veckor utveckla en mjukvaruprototyp som är startskottet i en revolution av transportindustrin, detta genom att demonstrera kolonnkörning med en MOPED. Mjukvaran kommer när den appliceras under lastbilstransporter öka kostnadseffektiviteten, säkerheten och minska miljöpåverkan.”*

- Jämförelse med vår vision
 - Vi lyckades med enkel platooning
 - För att öka kostnadseffektiviteten, säkerheten och minska miljöpåverkan har vi jobbat med att göra ACC:n och ALC:n mjuk och kontrollerad. Dessa funktioner kan såklart alltid förbättras för att få mopeden köra på en rak linje efter mopeden framför samt minska t.ex. oscillering.
- Hinder under projektets gång
 - Våra 3 första sprintmål var nästan identiska och var svåra att uppnå pga brister och fel i dokumentation och hårdvara. Vi hade även svårt att sätta oss in i projektet från början eftersom de fanns så mycket kod och stora delar av koden fungerade inte, eller behövde uppdateras.
 - (Sprintmål 1) *“Att i slutet av sprinten ha skapat en miljö där saker fungerar och där alla kan programmera och testa sin kod, samt ökad förståelse överlag.”*
 - Vi la ner mycket tid i början för att få servern att fungera vilket den aldrig gjorde pga bristningar i hårdvara. Squawk versionen på servern skiljde sig från raspberry pisen bl.a.
 - För att komma förbi problemen med bristfällig dokumentation och fel i dokumentationen tog vi mycket kontakt med andra grupper, framförallt via slack och scrum of scrums. Detta gjorde att vi kunde skrota vissa saker som inte fungerade och hela tiden ta små steg framåt.
 - Vi gick ihop med en annan grupp för att tillsammans hitta en bra lösning.
 - Hade inte en fungerande moped, första gången var i sprint 4
 - Vi jobbade mycket med att få igång simulatören för att detta inte skulle hindra oss från att arbeta vidare. Till slut fick vi igång simulatören och kunde ladda upp plugins till den för att testa kod vi skrivit. För att simulera sensor värden när mopeden inte fungerade utvecklade vi en mockup för att kunna testa kod ändå.

- Plugins fungerade inte att använda på mopeden eftersom servern inte fungerade
 - Vi började parallellt med att kolla på att utveckla ACC med hjälp av python, i och med detta kunde vi också testa hårdvarans brister även om vi i slutändan valde en annan lösning
- Signalerna för att styra tas inte emot TCUn utan SCUn av original appen (WirelessIno), det gjorde de krångligt att vi skulle in och ändra i SCUn
 - Löste med hjälp av en server istället

GUI

- De GUI:t vi har är appen vi utvecklat. Den är enkel att förstå och baserad på WirelessIno.
- Eftersom det gav mer värde till produktägaren att få en fungerande platooning än att ha ett snyggt GUI därför prioriterades detta bort och hamnade som en epic i vår product backlog. Hade vi haft mer tid hade vi:
 - Gjort en app som kan hantera allt
 - Så att man kan starta platoonläge helt från appen utan att köra via terminalen



Acceptance tests and stakeholder interaction

På demon visade vi upp:

- Kameran:
 - Tester
 - Hitta cirklar men inte rektanglar och andra föremål i bakgrunden
 - Förklarade hur de funkar rent tekniskt med hjälp av en server
 - Problemlösning:
 - Färg, dåliga ljusförhållanden
 - Kamerakvalité
 - Manuell fokus
 - Den hittade flera cirklar i samma bild
 - löste det genom att hitta största cirkeln
 - Hur många bilder den tar per sekund, långsam
 - Oves pi är inte så snabb
 - Optimering:
 - Färgskala
 - Hur många bilder den tar per sekund

- Serverlösning för att få över bilderna snabbare för raspberry pi är långsam och vi vill inte överbelasta den
- ALC:
 - Vi höll upp ett papper med en röd cirkel, motorn av, och visade att den svänger efter cirkeln
 - Förklarade hur det funkar rent tekniskt
 - Vi får in ett offset från kameran (en siffra mellan -100 och 100), där negativa värden innebär att cirkeln är till höger i bilden och till vänster för positiva värden
 - Problemlösning
 - Reagerar inte så mjukt
 - Hjulen ställer in sig lite själva
 - Optimering:
 - Började med att bara få den att svänga med hjälp av ett papper
 - Sen försöka få den att svänga mjukare
 - Vi började med att svänga hjulen med samma värde som offset, vi testade oss sedan fram för att kunna svänga mjukare. Vi kom fram till att svänga 60% av offset i taget vid små svängar och 90% av offset vid skarpa svängar.
- ACC:
 - Vi satte inte på bild servern, så att vi bara visade att den kan köra efter ett papper
 - Problemlösning
 - Vi började utveckla ACC plugins till MOPEDen. Insåg att det inte gick att ladda ner dem till mopeden, och utvecklade då en mockup för att kunna testa koden i simulatören. Testade parallellt en pythonlösning som fungerade halvdant. Därefter kontaktade vi en annan grupp och fick ta del av deras lösning med canbussen, vilket vi använder nu.
 - Förklarade hur de funkar rent tekniskt
 - Vi använder oss av en canbuss för enklare kommunikation direkt med MOPEDen och för att kunna skriva java-kod.
 - Optimering
 - Sensorns värden uppdateras inte tillräckligt ofta samt kan vara väldigt felaktiga, vilket resulterar till ett problem mellan att välja en bra nödbroms och kunna anpassa farten i platoon.
 - Vi hade kunnat försöka koda en mer generell lösning till ACC:n, men eftersom vi fick tillgång till en fungerande moped först i 5:e sprinten fokuserade vi på att få ACC:n att fungera för vårt specifika fall först.
- Appen:
 - Vi visade upp appens funktioner

- Egen server för ACC, Platoon
 - Socket mottagning på moped för djupt inbyggd
 - Lättare att använda egen server
- WirelessIno skillnader
 - Skicka ACC, Platoon on/off
 - Nya sliders för styrning
 - Krångligt att bygga ut gränssnittet runt de gamla hårdkodade
- Problemlösning
 - Signaler, hur dom skickas och tas emot av mopeden
 - Skicka extra information över samma socket
 - Var tvunget att tas emot i samma socket hantering, krångligt att komma åt
- Visade att de går att köra med hjälp av appen enbart, samt sätta på och av ACCn