



# CHALMERS

UNIVERSITY OF TECHNOLOGY



## Software Engineering Project

Reflektionsrapport

Lag Filip

*Nicole Ascard  
Filip Lindahl  
Felix Lissåker  
Peter Möller  
Filip Nilsson  
Henrik Tran*

*DAT255 Software Engineering Project  
Chalmers Tekniska Högskola  
2017-06-02*

# Innehållsförteckning

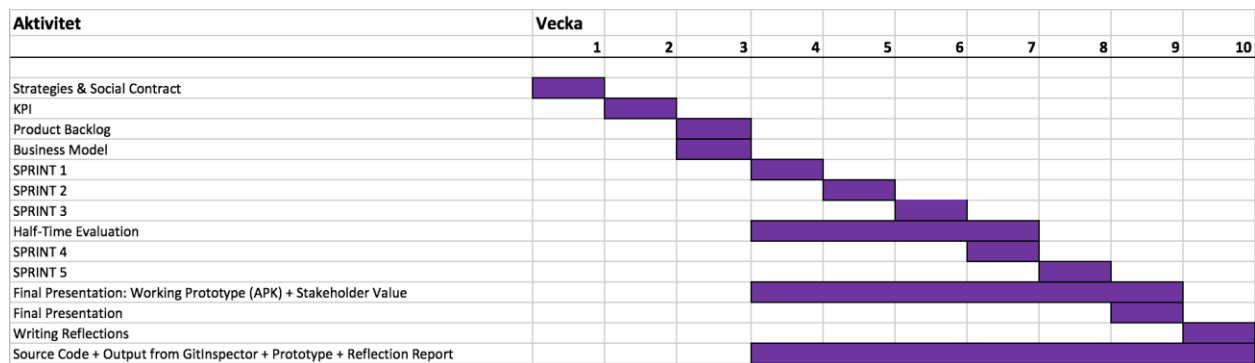
1. SCRUM - applikationsområden	1
1.1 Roller, lagarbete och socialt kontrakt	1
1.2 Använd praxis	2
1.3 Tidsfördelning	3
1.4 Nedbrytning av uppgifter	4
2. Reflektion av <i>Sprint Retrospectives</i>	5
2.1 Vad som generellt varit bra under projektets gång	5
2.2 Vad som kunde gått bättre under projektets gång	5
2.3 Vad skulle gjorts för att uppnå förbättringsförslagen?	6
3. Dokumentation av <i>Sprint Retrospectives</i>	7
4. Reflektion på <i>Sprint Reviews</i>	7
5. Praxis för användning av nya verktyg och teknologier	12
5.1 Programmeringsspråk	12
5.2 Bildredigeringsprogram	12
5.3 Utvecklingsmiljö	12
5.4 GUI verktyg	12
5.5 Scrum Task Board	13
5.6 Kommunikationsmedel	13
5.7 Dokumentationsverktyg	13
6. Reflektion av förhållandet mellan prototyp, process och intressenter	14
7. Relating your own process to literature and guest lectures	15
8. Evaluation of D1 - D4	17
8.1 D1 - Lego	17
8.2 D2 - KPI	18
8.2.1 Psykosocial arbetsmiljö	18
8.2.2. Velocity	19
8.2.3 Stakeholder Satisfaction	19
8.3 D3A - Product Backlog	19
8.4 D3B - Business Model	20
8.4.1 Key Partners	20
8.4.2 Key Activities	20

8.4.3 Key Resources	20
8.4.4 Value Proposition	21
8.4.4 Resterande	21
8.5 D4 - Halftime Evaluation	21
9. KPI charts, 0-1p	22
9.1 KPI No. 1 - Arbetsmiljö	22
9.2 KPI No. 2 - Velocity	24
9.3 KPI No. 3 - Stakeholder Satisfaction	25
10. Acceptanstest och interaktion med intressenter	26

# 1. SCRUM - applikationsområden

## 1.1 Roller, lagarbete och socialt kontrakt

Tillsammans har gruppen planerat och exekverat ett mindre mjukvaruprojekt som ett lag. Genom projektets utveckling har Scrum-gruppen bestått av sex stycken medlemmar. Tre av medlemmarna är i färd med att avsluta sina studier på Industriell Ekonomi och två är i samma stadie men på institutionen för Data- och Informationsteknik. Den sjätte medlemmen håller på att avsluta en masterutbildning med datorsystem och nätverk som inriktning. Gruppen är uppdelad i rollerna *Scrum master*, *front-end utvecklare* och *back-end utvecklare*. Fördelningen har bestått i två stycken *back-end* och tre stycken *front-end utvecklare* samt en *Scrum-master*. Motiveringen till vilken gruppmedlem som fick vilken roll var baserat på individens kunskapsnivå inom det området samt grad av intresse. Detta har gjort att gruppens "bussfaktor" blivit hög inom de olika områdena. *Scrum-master* har haft den övergripande kontrollen och har skapat en projektplan som gruppen förhåller sig till (se figur 1). Även viktiga datum visualiserades tidigt för att ge gruppen en bra översikt över projektet.



Figur 1. GANT-schema med aktiviteter och veckor.

De programmeringsmetoder som använts är parprogrammering, individuell programmering samt en mix av dem båda. Dessa har använts beroende på komplexiteten på uppgiften i fråga. *Front-end* utvecklare har generellt samarbetat bra och har ofta suttit ihop eftersom att parallell programmering med GUI:t visat sig vara ineffektivt. Detta då *front-end* utvecklingen har krävt kunskap på flera håll såsom förståelse för lotsens arbete, kunskap om JavaFX samt Photoshop. *Back-end* utvecklare har fram till Sprint #2 jobbat individuellt eftersom de initialt behövde göras en hel del tester för att lyckas kontakta PortCDM. Under Sprint #3 minskade testperiodens omfattning vilket gjorde att även back-end började samarbeta i sin utveckling. En hel del individuell testande utfärdades fortfarande under Sprint #3 och #4 eftersom att PortCDM har

varit svårt att förstå sig på vilket ledde till en fördelning av parprogrammering och individuell programmering.

Planen från början var att rotera rollerna för att minska risker ifall någon var bortrest eller liknande. Dock var gruppen inte redo för rotation av roller efter de första sprintarna eftersom det fortfarande var oklart hur kontakten med PortCDM skulle konstrueras. För att inte tappa effektivitet i detta avseende fortsatte gruppen med sina respektive roller utan rotation. För att hålla risken på en minimal nivå bibehölls alltid minst två gruppmedlemmar på de mer kritiska rollfördelningarna. I detta avseende har gruppen som helhet förstått och identifierat komplexiteten i mjukvaru design och utveckling.

Det sociala kontraktet har gruppen följt men en förändring har gjorts vilket är att kommunikation inte längre sker primärt via Facebook-chatten utan istället via Slack. Detta gjordes då Facebook-chatten är svår att gå tillbaka i för att se vad som sagt medans Slack-chatten kan skapa trådar för olika konversationer vilket gör den mer lättnavigerad. Gruppen borde ha jobbat mer på att skapa och fördela uppgifter även till den gruppmedlem som är frånvarande vid ett möte. Konsekvenserna av att detta inte gjordes blev oklarheter i rollen för den frånvarande och det påverkade känslan av kontroll. Det hade kunnat förhindras om gruppen såg till att tilldela den frånvarande personen med en tydlig och lättförståelig uppgift.

## 1.2 Använd praxis

Anledningen till att gruppens sammansättning består av både medlemmar från Industriell Ekonomi samt Data och IT är att båda sidorna såg de stora fördelarna med att ha en grupp med olika bakgrunder. Därför skapades ett tvärfunktionellt team, även kallat XFT, med individer som har kunnat tillämpa olika färdigheter från tidigare programmeringskurser och andra relevanta kurser i en projektliknande miljö. Detta ökade kreativiteten och problemlösningssförmågan inom gruppen. Tillsammans har gruppen planerat och genomfört ett mindre mjukvaruutvecklingsprojekt under 8 veckor. Då gruppen upplevt att den tilldelade uppgiften varit aningen oklar så tog det lång tid innan gruppen fått en mer konkret bild av uppgiftens syfte.

Klockan 10:00 varje dag har *Daily Scrum Meetings* utförts. Möten har skett via Slack med de tre frågorna: "1. Vad gjorde du igår? (Sedan förra Scrum-mötet). 2. Vad ska du göra idag? (Fram till nästa Scrum-möte). 3. Har du stött på några hinder?". På grund av att mötena sker digitalt har längden inte varit relevant och ankomsttiderna för svaren har varierat. Dessa aspekter har det funnits overseende för då digitala *Daily Scrum Meetings* har ansetts mycket smidiga. Rutiner var initialt få men har standardiserats och utvecklats under processens gång.

Sprinten har startat onsdag eftermiddag och avslutats nästa onsdag på förmiddagen. Varje onsdag har hela teamet träffats och utvärderat sprinten med *Sprint Retrospectives* och *Sprint Reviews*. I slutet av arbetsdagen på onsdagar görs även en planering av nästa sprint där nya *User Stories* tagits fram. Ett nytt verktyg som gruppen lärt sig att jobba med är just *User Stories*. Det har varit ett bra sätt att styra upp arbetsuppgifter på, men de mer tidskrävande uppgifterna borde ha gjorts tunnare.

Ytterligare metoder som *Scrum Task Board* och *Trello* har här använts som ett digitalt verktyg för detta. *Trello* visade sig vara mycket effektivt och borde därför använts i större utsträckning. Förutom kundens feedback har data rörande gruppens KPI:er samlats in.

### 1.3 Tidsfördelning

Person	Roll	Uppgifter
Nicole Ascard	Scrum Master	Kundkontakt, struktur, <i>Scrum of Scrums</i> möten, dokumentation av projektet, <i>Daily Scrums</i>
Filip Lindahl	Back-end & Front-end	Kundkontakt, Implementation av <i>back-end</i> , implementation och justering av <i>front-end</i>
Felix Lissåker	Front-end	Kundkontakt, utveckling av <i>Mock-GUI</i> , utveckling av <i>front-end</i> , dokumentation av projektet
Peter Möller	Front-end	Kundkontakt, förståelse för lotsprocesser, utveckling av <i>Mock-GUI</i> , dokumentation av projektet, utveckling av <i>front-end</i>
Filip Nilsson	Back-end & Front-end	Kundkontakt, Implementation av <i>back-end</i> , implementation och justering av <i>front-end</i>
Henrik Tran	Front-end & Back-end	Kundkontakt, JavaFX, utveckling av <i>Mock-GUI</i> , utveckling av <i>front-end</i> , utveckling samt implementation av viss funktionalitet

Tabell 1. Personer och dess roller med innehållande uppgifter.

Som nämndes i avsnitt 1.1 *Roller, lagarbete och socialt kontrakt*, roterade gruppen inte sina roller halvvägs som planerat utan fortskred med de första rollerna. En rotation hade resulterat i att alla hade backend-kunskaper, vilket hade möjliggjort att samtliga gruppmedlemmar lade viss fokus på att få *back-end* att fungera i de sista två sprintarna. Nu var tidsfördelningen sådan att *scrum master* och *front-end* jobbade på dokumentation samt sina andra uppgifter medan *back-end* försökte få produkten att fungera externt.

Då det uppkom problem med applikationen kände resten av gruppen en viss maktlöshet då de inte hade nog med *back-end* kunskap för att hjälpa till, vilket resulterade i att *back-end* jobbade övertid för att få igång applikationen. Gruppen borde ha reflekterat mer över nackdelarna med att ha sådan organisatorisk risk. Någon form av informationsöverföring borde ha skett till de andra gruppmedlemmarna, trots att alla behöll sina ursprungliga roller. För att på så sätt öka transparensen så alla kunde hjälpa till vid sjukdom eller andra förhinder. Om alla hade haft lite insikt i varandras arbete hade problematiken med till exempel *back-end*, men även andra uppgifter som försvårade projektet, underlättats.

## 1.4 Nedbrytning av uppgifter

En *Epic* erhålls baserat på *Product Owner's* önskemål och karaktäriseras av att den inte går att genomföra under en sprint. Som följd läggs uppgiften in i *Product Backlog* för att sedan delas upp i tunnare delar som distribueras i de olika sprintarna. I detta avseende har gruppen stött på problem, de uppgifter som erhålls från *Product Owner* har nämligen varit tunna nog att implementera direkt i nästkommande sprint. Som följd har *Product Backlog* använts i mindre utsträckning än vad som är önskvärt. För att öka användandet av *Product Backlog* kunde gruppen fört in de *User Stories* som oönskat och oplanerat spänt över flera sprintar. Dessa hade istället kunnat bli konstruerade som *Epics* för att sedan bli uppdelade i tunnare skivor på flera sprintar. Utöver dessa hade även *Product Owner's* önskemål som inte uppfylldes kunnat bli placerade kontinuerligt i *Product Backlog*. Som följd har *Product Backlog* blivit uppdaterad mot slutet av projektet för att ge en mer överskådlig bild av vad *Product Owner* efterfrågat och vad som prioriterats.

## 2. Reflektion av *Sprint Retrospectives*

Efter varje sprint har möten med gruppen utfärdats för att diskutera vad som kan göras för att förbättra produktiviteten inför nästa sprint. Den generella strukturen för mötena har varit indelad i tre olika delar; 1. Vad som gått bra under sprinten, 2. Vad kunde gått bättre och 3. Vad skulle gjorts för att uppnå förbättringsförslagen.

### 2.1 Vad som generellt varit bra under projektets gång

Efter att ha genomfört de första mötena med *Product Owner* underlättades arbetet med applikationens utveckling. De återkommande förfarande har varit att gruppen, utifrån den information som erhålls från *Product Owner*, utvecklat ett *Mock-GUI*. Det har gått relativt snabbt och har vanligtvis blivit färdigställt direkt efter mötet för att till så stor utsträckning som möjligt få med alla detaljer. Utefter det har det interaktiva GUI:t utvecklats under sprinten med *Mock-GUI:t* som mall. Utifrån en kombination av dessa GUI:n har *Product Owner* beskrivit sin nya bild av produktens utveckling.

Denna strategi har legat till grund för den framgångsrika relationen till intressenterna och deras önskemål. Gruppen har dock något sent fastställt filosofin att *Product Owner's* alla önskemål inte kan uppfyllas utan att prioriteringar måste göras. Samtidigt har den egna visionen av produkten alltid underminerats till fördel för *Product Owner's* vilket till viss utsträckning kan vara destruktivt för projektets kreativitet. Detta har dock i kombination gett insikt i skillnaden mellan kunden, den framtagna lösningen och gruppens individuella strävan. Även samarbetet inom gruppen har fungerat bra och inga konflikter har skett.

### 2.2 Vad som kunde gått bättre under projektets gång

Ett stort problem under projektets gång har varit att bestämma funktionaliteten av produkten. Gruppen har haft fyra olika källor av input; scenariot, lista med *Minimum sends per actor* samt *Product Owner*: Sandra och Anders. Dessa olika källor har indikerat olika åsikter angående produktens funktionalitet såväl som hur lotsens processer faktiskt ser ut. Något som gjordes fel i början var att gruppen vid möten med de olika kundrepresentanterna tolkade deras input som definitiva krav snarare än att det är input från en av fyra informationskällor. Detta gjorde att gruppen efter varje kundmöte fått ändra och addera ny funktionalitet i ganska stor utsträckning.

Det bildades tidigt en uppfattning om att det var lätt att sätta sig in i API:t och PortCDM vilket senare visade sig vara felaktigt. Givet svårigheterna med att kommunicera med PortCDM hade



det varit fördelaktigt med bättre kontakt med utvecklarna från PortCDM. Den kontakt som bedrivits har primärt varit via Slack med långa ledtider och ibland uteblivna svar. Utan dedikerade instruktioner om hur PortCDM's API:er fungerar har projektet utvecklats långsamt och var även nära att inte färdigställas alls.

Gruppen hade mått bättre om det fanns tydliga roller. Som nämndes i *1.1 Roller, lagarbete och socialt kontrakt* har det skett en viss förvirring kring rollerna och vilka uppgifter varje medlem har. Ledarskapet har varit bristande eftersom ingen utsågs till officiell projektledare och det är inte så populärt att vara inofficiell ledare samt att ledandet blir svårare. Varje projekt mår bättre om det finns ett tydligt ledarskap.

## 2.3 Vad skulle gjorts för att uppnå förbättringsförslagen?

För att undvika problemen som uppstår med att inte till fullo veta vilka funktioner som önskas i applikationen skulle det initialt redas ut vilka minimikrav som ställs från de olika informationskällorna. Genom att göra detta hade gruppen i första hand kunnat tillfredsställa alla minimala krav för att sedan kunna tillgodose så många tillägg som möjligt.

Det hade varit fördelaktigt att upprätthålla återkommande möten med PortCDMs utvecklare för att kontinuerligt kunna ställa frågor till dem rörande API:t och varför det inte fungera på ett intuitivt sätt. Detta har åstadkommits till viss del via Slack men för att förklara ett helt problem hade det varit fördelaktigt att möta kunniga inom området och visa upp back-end där logiken felar.

Det hade varit bra om gruppen bestämde vem som hade vilken roll och vad dess uppgifter var samt att medlemmarna var eniga om beslutet. Nu var planen att rotera rollerna men ett sådant byte skedde aldrig. Vad teamet borde ha gjort då är att officiellt säga att nu har medlemmarna sina roller och inget byte kommer ske. Det hade skapat en större tydlighet och högre trivsel i gruppen.

### 3. Dokumentation av *Sprint Retrospectives*

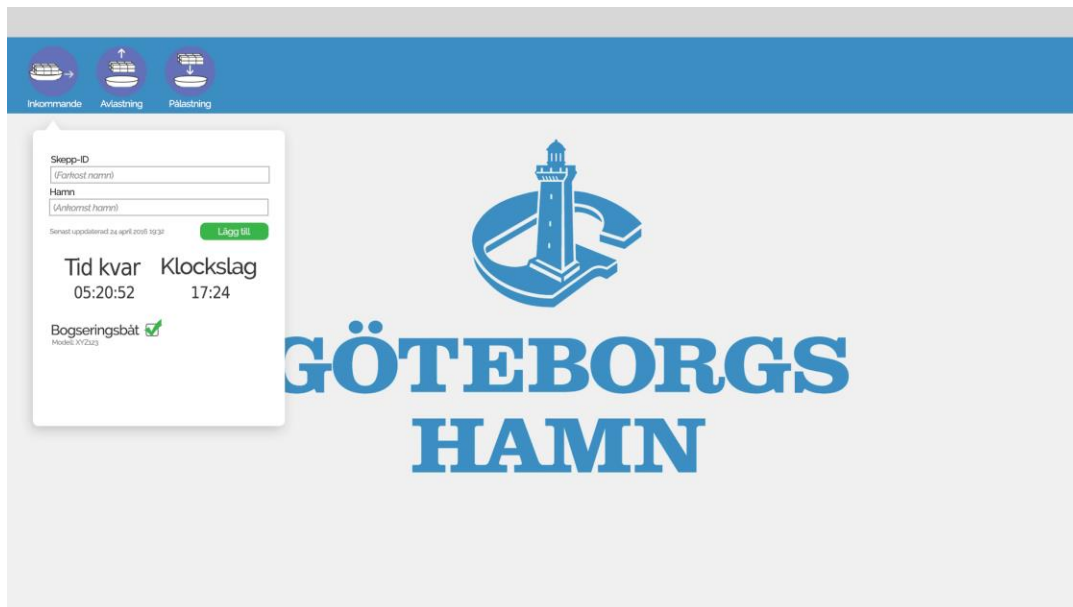
Gruppens sammanställda dokument ligger sparade på GitHub. Genom att besöka länken nedan går det att få tillgång till dokumenten.

<https://github.com/filipni/Software-Engineering-Project/tree/master/documents/Sprint%20Retrospectives>

### 4. Reflektion på *Sprint Reviews*

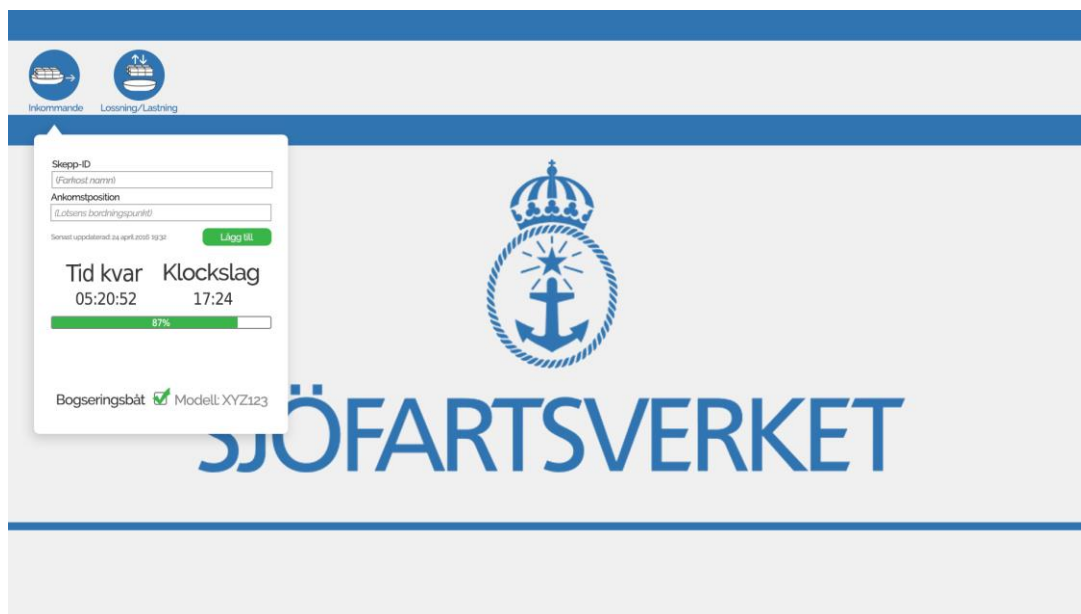
Under projektets gång har gruppen visat upp den senaste versionen av det interaktiva GUI:t samt *Mock-GUI:t* vid varje möte med *Product Owner*. Kunden har då haft möjlighet att testa det interaktiva GUI:t för att bedöma dess funktionalitet samt ge feedback på förbättringar. Utöver det så har *Mock-GUI:t* visat hur produkten är tänkt att se ut i slutändan. Detta gör att kunden kan ge feedback på funktionalitet redan innan den implementeras.

Dokumentation av det interaktiva GUI:t har till stor del baserats på lagrade *User Stories* för varje sprint. *Mock-GUI:t* har sparats som dokumentation på produktens utveckling och *Product Owner's* önskemål genom projektet. Varje sprint har, för back-end utvecklingarna, mer eller mindre varit orienterad kring att förstå sig på hur kontakt och prenumeration med PortCDM bäst ska gå till. Som följd har *back-end* inte haft något att demonstrera vid mötena med *Product Owner* och dokumentationen har blivit bristande. Sammantaget har gruppen vid ett tillfälle i veckan avsatt tid för att demonstrera framgångarna med mjukvaran internt. Dessa möten har primärt baserats på det interaktiva GUI:t och *Mock-GUI:t*. Nedan följer en inkrementell förklaring av utvecklingen av dessa möten.



Figur 2. Första mock-GUI:t i första sprinten.

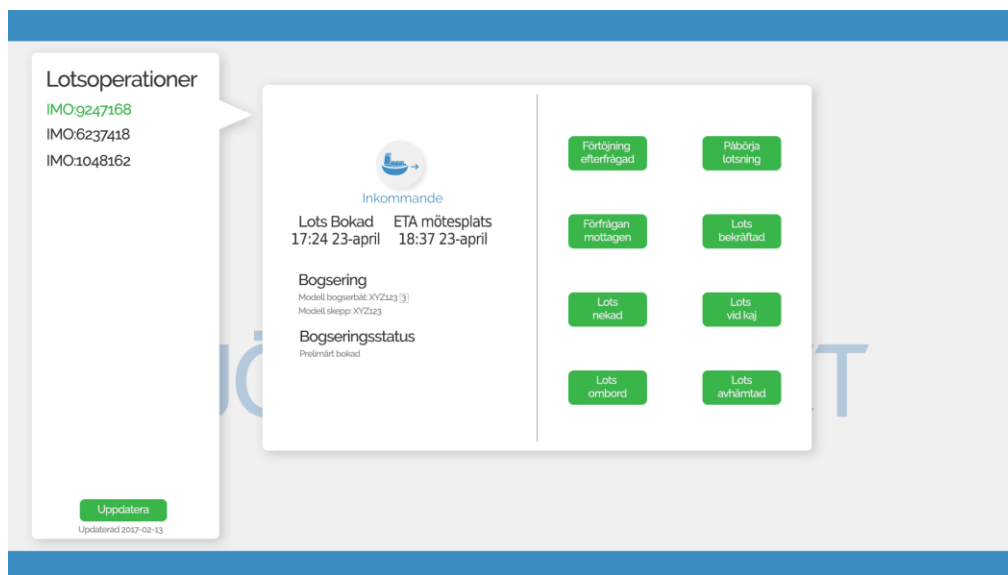
Som figur 2 visar, innehöll det första GUI:ts fartyg tre olika statusar; inkommande, avlastning samt pålastning. I varje statusgrupp kunde man söka efter ett fartyg och dess destinationshamn och då få upp ETA till hamn samt klockslag då fartyget anländer till hamnen. Det fanns även en ruta som visar om bogseringsbåt var bokad till uppdraget samt information om vilken typ av bogseringsbåt som var bokad. Funktionaliteten ovan är inspirerad från ett samtal med Anders som är en av två *Product Owners* gruppen haft kontakt med.



Figur 3. Andra mock-GUI:t i andra sprinten.

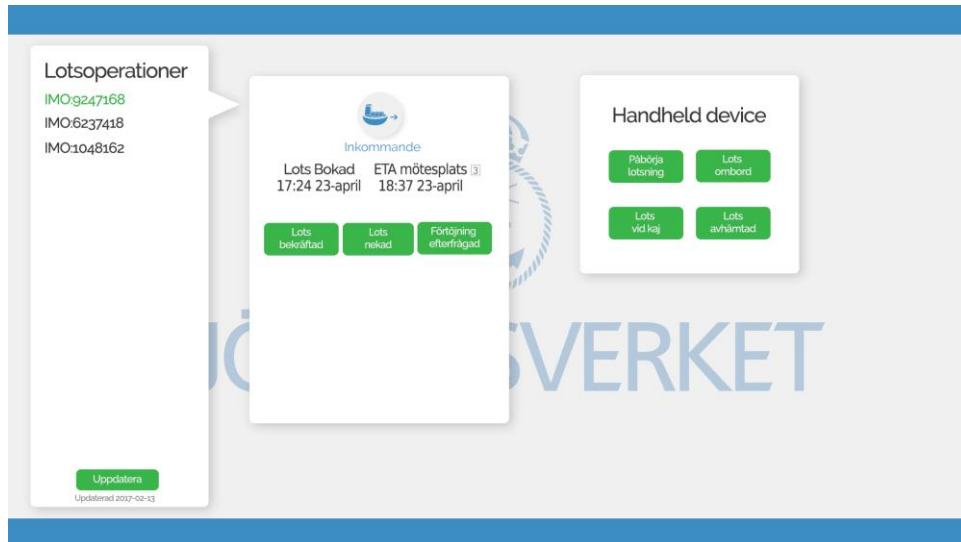
Efter att ha demonstrerat det första *Mock-GUI*:t ville *Product Owner* lägga till och förändra en

del funktionalitet. Istället för att dela in uppdragen i tre olika kategorier valdes istället en gemensam för Avlastning och Pålastning som istället kom att heta Lossning/Lastning. Utöver det kritiserade kunden även valet av inmatning från första *Mock-GUI*:t. Istället för att mata in vilken hamn som är aktuellt för anlöpet matas lotsens position för bordning in. Denna position kan vara en av tre möjliga i Göteborgs skärgård och avser den plats som lotsen ska möta upp skeppet för inlotsning respektive lämna skeppet efter utlotsning. Slutligen ville kunden även ha en visuell representation av i vilket skede Lossning/Lastning befinner sig i för att lotsen ska kunna bedöma när det är dags att lotsa skeppet igen.



Figur 4. Tredje mock-GUI:t i tredje sprinten.

I samband med att *Mock-GUI No.2* visades för *Product Owner* träffade gruppen en ny representant. Vid det här mötet ändrades bilden av hur den slutliga produkten skulle se ut totalt. Den nya representanten hade helt andra önskemål. Bland annat var en mer överskådlig syn över flera lotsoperationer önskat vilket resulterade i listan av lotsoperationer till vänster i fig. 5. Först vid det här mötet erhöles listan med *Minimum sends per actor* vilket gjorde att det lades till ett antal knappar som utförde just detta.



Figur 5. Fjärde mock-GUI i fjärde sprinten.

Efter att *Mock-GUI:t No.3* visades för *Product Owner* flyttades en del av funktionaliteten till ett nytt fönster. Förändringen baserades på att kunden egentligen ville ha den förflyttade funktionaliteten i en mobil enhet. Eftersom det inte fanns tid att skapa två separata applikationer gjordes kompromissen att den aktuella funktionaliteten flyttades till ett externt fönster inom den initiala applikationen. Som följd har *Mock-GUI:t No.4* en lista med alla relevanta skepp längst till vänster i programmet. Genom att klicka på önskat fartyg visas mer information om det anlöpet. Denna funktionalitet kommer lots operatören ha tillgängligt. Fönstret döpt "*Handheld device*" kommer alltid att vara synligt då det representerar den funktionalitet som lotsen har ombord på skeppet.



Figur 6. Femte och slutgiltiga mock-GUI i femte sprinten.

Vid det fjärde mötet tyckte kunden att produkten började närma sig något de kan va nöjda med. Feedbacken var att kunden tyckte det va viktigt att veta till vilken plats ETA:n syftade på då denna kan skilja sig. Detta fixades genom att lägga till en "Från" och en "Till" ruta med information om vilken plats som avses. Kunden ville även kunna mata in till vilken tid lotsuppdraget är bokad vilket löstes med att lägga in en inmatningsruta. Tiden som matas in fyller i informationen som visas upp under "Lots bokad". Det klargjordes även att förtöjning efterfrågad bestod av två olika efterfrågningar beroende på om det gäller ett ankommande eller avgående fartyg. Därför delades knappen "Förtöjning efterfrågad" upp i "Förtöjning inkommande" samt "Förtöjning utgående". Bilden ovan är alltså en screenshot på det slutgiltiga GUI:t. Både kunden och gruppen själva är nöjda med den slutgiltiga produkten. Något som gruppen i efterhand hade velat lägga till är en text som förtydligar att inmatningsfältet är menat för att skriva in tidpunkten då lotsning är bokad till.

## 5. Praxis för användning av nya verktyg och teknologier

Under arbetets gång har ett antal olika verktyg och teknologier använts. Några av dem var bekanta för gruppmedlemmarna medans andra var helt nya. Detta har gjort att viss tid behövts läggas på att lära sig dessa.

### 5.1 Programmeringsspråk

Initialt diskuterades vilket språk som var mest lämpligt för projektet. De mer erfarna i gruppen talade för att det finns fördelar med vissa språk för vårt syfte. Till slut togs det beslut om att Java och Eclipse var mest lämpligt. Detta beslutet grundades i att gamla erfarenheter från tidigare kurser ville utnyttjas, Java var då det enda alternativet. Utöver detta gav Java, som programmeringsspråk, tillgång till en uppsättning API:er med syfte att underlätta kommunikationen med PortCDM.

### 5.2 Bildredigeringsprogram

Eftersom att en stor del av utvecklingsstrategin låg i att visa upp ett *Mock-GUI* vid möten med *Product Owner* krävdes kunskap i bildredigeringsprogram. En del av medlemmarna i gruppen hade kunskap i *Adobe Photoshop CC* sedan tidigare kurser vilket då föll sig naturligt att använda. Resterande medlemmar fick hjälp att utveckla tillräcklig kunskap för att kunna använda verktyget vid framtagandet av *Mock-GUI*.

### 5.3 Utvecklingsmiljö

Trots att projektet ses som relativt litet är sex medlemmar delaktiga i utvecklingen av produkten. För att göra så att teamet jobbar under samma förutsättningar satte en medlem upp en Oracle VM VirtualBox med Ubuntu och vissa applikationer förinstallerade. På så sätt kunde gruppen använda likadana virtuella datorer, med samma inställningar och förutsättningar. Förutom att systemet var byggt på Linux, som var nytt för många i gruppen, underlättade den virtuella boxen en hel del i utvecklingsarbetet. För att sätta upp och använda den virtuella maskinen krävdes en del självinläring från gruppens medlemmar.

### 5.4 GUI verktyg

För att utveckla det interaktiva GUI:t, front-end, användes JavaFX tillsammans med SceneBuilder. Ungefär halva gruppen tog med kunskap från tidigare kurser och undervisade de

som inte hade koll på programvaran. Tillsammans kunde de flesta medlemmarna i gruppen vara med och utveckla det interaktiva GUI:t till applikationen med *Mock-GUI:t* som mall.

## 5.5 Scrum Task Board

Under utvecklingen av produkten har ambitionen varit att mata in *User Stories* i Trello för att enklare kunna visualisera vad som är åtgärdat och vad som fortfarande behöver göras. Trello var nytt för samtliga gruppmedlemmar vilket gjorde att alla behövde lära sig detta verktyg individuellt. Tyvärr har gruppen misslyckats med att aktivt använda detta. Det vanliga förfarandet har varit att user stories skapats och antecknats under det veckovisa scrum-mötet och sedan inte skrivits in i Trello. Detta eftersom varje medlem har haft tillräcklig bra koll på vad deras *User Story* innefattar.

## 5.6 Kommunikationsmedel

Innan första sprinten beslutades att Slack skulle användas som primärt kommunikationsmedel. På så sätt kunde gruppen som lots organisation på ett enkelt sätt kommunicera med alla aktörer i kursen. Motsvarande kommunikationsmedel som Messenger eller e-mail hade inte kunnat erbjuda detta på ett lika tillfredsställande sätt.

## 5.7 Dokumentationsverktyg

För att lagra färdiga dokument och programkod har GitHub använts. Däremot har Google Docs använts en del vid färdigställandet av olika dokument i kursen. Detta för att det möjliggjort för flera personer att dels skriva på samma dokument men även följa live vad som skrivs av andra gruppmedlemmar och kommentera på detta.



## 6. Reflektion av förhållandet mellan prototyp, process och intressenter

Ett stort problem inom sjöfartsindustrin är kommunikationen och flödet av information mellan de olika aktörerna. I dagsläget sker stor del av informationsutbytet mellan aktörer via e-post vilket gör att samtliga aktörer inte får tag på informationen samtidigt och den blir lätt inaktuell. Detta är något gruppen vill effektivisera med detta mjukvaruprojekt genom att göra information tillgänglig för samtliga aktörer via PortCDM.

Att jobba agilt och att jobba med Scrum var helt nytt för alla gruppmedlemmar i startet av projektet. Trots att teamet har haft processer och verktyg, exempelvis Scrum-processen, har fokus legat på individer och interaktion inom gruppen, men även med de andra projektgrupperna via Slack och Scrum of Scrums-möten. Det har dokumenterats en hel del men teamet har lagt mer fokus på att ha en fungerande mjukvara. Teamet har koncentrerat sig mer på att samarbeta med kund över att hålla sig till de ursprungliga instruktionerna. Trots att teamet har haft en projektplan att följa har de varit duktiga på att reagera på förändringar. Gruppen har lärt sig att bli mer agila än förut och har fått en större insikt i konsekvenserna av att jobba agilt och med Scrum.

Gruppen har gjort sitt bästa för att få prototypen att ge så mycket stakeholder value som möjligt. Önskemål och feedback från kund har applicerats på prototypen i största möjliga mån men ibland har prioriteringar behövts göras. Detta beror till stor del på, som tidigare nämnt, att informationen kommit från två olika kundrepresentanter, ett scenariodokument samt en lista med minimal send requirements varav samtliga källor haft olika åsikter av vad som skapar kundvärde hos prototypen. Givetvis har prioriteringar även behövts göras på grund av projektets tidsram.

## 7. Relating your own process to literature and guest lectures

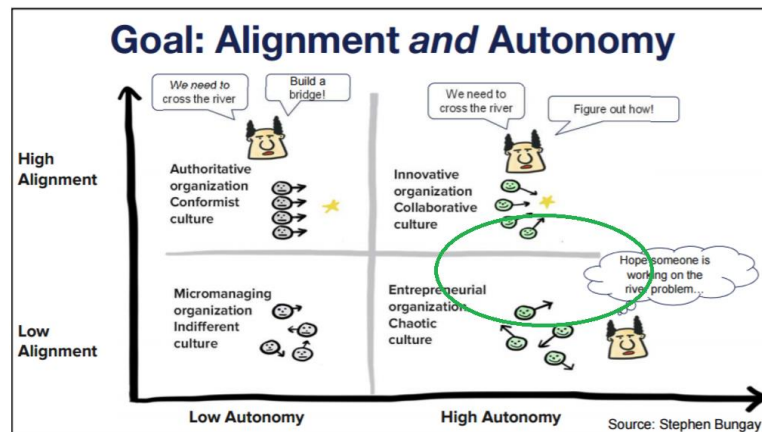
Eftersom att gruppens tidigare erfarenhet av projekt av den här omfattningen är av bristande karaktär kan utformningen av gruppens arbete förbättras i ett antal avseenden. Denna aspekt är extra tydlig i jämförelse med stora, erfarna, aktörer på mjukvarumarknaden. Simon Hofverberg (Spotify) demonstrerade en rad förhållningssätt som exemplifierar fördelar och nackdelar i gruppens sätt att arbeta med projektet.

En viktig aspekt av Spotify's riktlinjer är tvärfunktionalitet inom gruppens medlemmar samt mångfald. I det här avseendet har gruppen lyckats väl. Gruppen har bestått av medlemmar med olika funktionalitet, bakgrund och inriktning. Som följd har även utvecklingen av produkten varit uppdelad i tre segment av gruppmedlemmar, en tredjedel mer tekniskt inriktade, en tredjedel mer visuellt och grafiskt lagda och en tredjedel mer inriktade mot ledning och dokumentering. Varje gruppindelning har haft stor frihet att utveckla produkten enligt egna värderingar och tidigare erfarenheter. Detta är ytterligare en av Spotify's riktlinjer, att värdera gruppen högst men att också ge individerna stor frihet och mycket support. Till en början innebar denna strategi att de initiala minimum viable product versionerna utvecklades effektivt. Mot slutet av projektet uppstod dock svårigheter med denna strategi. Grunden till problematiken var att två av tre gruppindelningar var klara med sin utveckling mot kundens efterfrågan. På grund av viss isolering mellan varje gruppindelning kunde den tredje gruppindelningen inte få mycket hjälp från de resterande två vilket orsakade asymmetri i arbetsbelastningen.

Ytterligare en problematik, som Spotify har som riktlinje att motverka, har varit en avgörande faktor till att projektet var nära att misslyckas i vissa avseenden. Denna aspekt har att göra med förmågan att evaluera arbetstekniken inom gruppen och ständigt utvärdera och testa nya metoder. Redan under Sprint 3 (av 5) upptäckte gruppen ett eventuellt problem med versionen av de meddelanden som skickades in till PortCDM. APIerna och meddelandena var av en tidigare version som inte stämde överens med versionen av servern. Som följd var dessa två inte kompatibla med varandra. Hade då råd sökts från andra grupper och utvecklingsmiljön ändrats kunde problematiken med back-end lättare lösas och andra aspekter av programmet utvecklats istället.

Ett återkommande problem som erfarits under projektets gång är *Product Owner's* varierande efterfrågan. Initialt var det problematiskt eftersom *Product Owner's* efterfrågan försökte följas till punkt och pricka vilket resulterade i stor variation i produktens ursprungliga versioner. Detta var en aspekt som misslyckades flera gånger innan det blev tydligt att kundens efterfrågan inte kan mötas i full utsträckning (ytterligare en aspekt som Spotify försöker motverka i sina

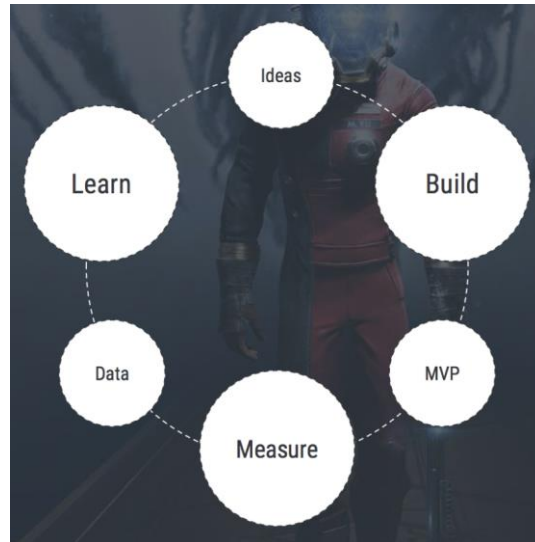
riktlinjer). Därför prioriterades de mest vitala aspekterna från de olika kundrepresentanterna och som sedan implementerades.



Figur 7. Gruppens positionering under projektets process.

Sammanfattningsvis summeras gruppens översiktliga positionering i fig. 8 ovan (grön ellips). Anledningen till att gruppen positionerar sig i den fjärde kvadranten är för att gruppens funktionalitet varit så strikt indelade i de olika rollerna. Eftersom gruppen trots den konkreta indelningen varit framgångsrika i sättet att repetitivt generera *Mock-GUI* lutar placeringen även mot den första kvadranten. Anledningen till det är att *Mock-GUI* har gjort att gruppen trots individuellt arbete inom sina segment haft ett tydligt mål att rikta utvecklingen mot.

Arbetsprocessen skulle i stora drag kunna liknas med *Lean Development Process* som IGDB använder sig av, se fig.9. Varje sprint har motsvarat en cykel i *Lean Development Process*. Det hela började med *Learn*-fasen där gruppen bringade kunskap om det arbetet som en lots organisation sysslar med. Detta gjorde att idéer kunde utarbetas i *Ideas*-fasen rörande hur arbetet kunde underlättas. Efter det byggdes idéerna in i produkten under *Build*-fasen och skapade en *MVP* som sedan presenterades för *Product Owner*. Detta gjorde att KPIer kunde mätas i *Measure*-fasen, samt att ytterligare information om vad kunden ville ändra i *Data*-fasen. Grundat på denna informationen cykeln repetitivt genomförs i nästa sprint.



Figur 8. Lean Development Process.

## 8. Evaluation of D1 - D4

### 8.1 D1 - Lego

Efter LEGO-föreläsningen identifierades tre fördelar med att använda SCRUM. Här ges en reflektion över hur väl gruppen förhållit sig till dessa fördelar och eventuella förbättringsmöjligheter.

Den första aspekten som beaktades var relationen mellan kund och utvecklare. Utan att ha ett etablerat system (t.ex. SCRUM) för hur utvecklingen ska fortskrida kan utvecklarnas syn på produkten lätt avvika från kundernas. Anledningen till detta kan bl.a. bero på att kunden är vag eller otydlig i sin definition av produkten eller att utvecklarna uppfattar önskemålen fel. Gruppens användning av SCRUM-metodiken beaktar detta problem genom att etablera sprint-visa möten med kunden. Efter varje sprint fick kunderna utförligt bedömde förra sprintens förbättringar samt förslag på hur nästa sprint bör utformas. Användningen av SCRUM underlättar på så vis att vår produkt, vårt mål och kundens förväntningar går åt samma håll.

Den andra aspekten som beaktades under projektets gång är prioriteringsordningen av *User Stories* i utvecklingen av slutprodukten. Vid varje möte med *Product Owner* delgav de alla sina önskemål rörande produktens funktionalitet och utseende. Under dessa möten var gruppen, framför allt mot slutet, även mer benägna att rådfråga *Product Owner* om prioriteringsordningen av produktförbättringarna. Denna utveckling ter sig naturligt eftersom att gruppen i början kunde uppfylla kundens alla önskemål men fram mot slutet fick mindre tid och mer detaljerade, mer tidskrävande uppgifter. Andemeningen här är att *Product Owner* ofta,

och även i gruppens fall, efterfrågar funktionalitet utan att förhålla sin efterfrågan med sannolikheten att efterfrågan går att genomföra. Som följd lärde gruppen sig, den hårda vägen, att samla in *Product Owner's* önskemål och sedan fråga hur de väljer att prioritera önskemålen för att slutligen bedöma hur genomförbara åtgärderna är.

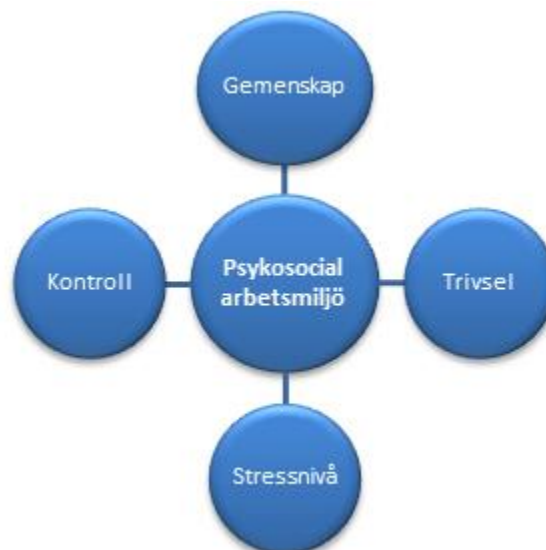
Den tredje aspekten beaktar bland annat hur väl *User Stories* planeras och hur tidsåtgången är utspridd över projektet. Alla dessa är, om de genomförs på ett bra sätt, fördelar med SCRUM. *User Stories* har tagits fram i början av varje sprint. Det har dock tagit en del övning innan de blivit "sliceade" på ett tillräckligt tunt vis, i början var de för omfattande. När det gäller tidsdistributionen över projektets gång har det oundvikligen blivit asymmetriskt på olika sätt. I början av projektet var arbetsbördan jämt distribuerad mellan gruppmedlemmar för att mot slutet av projektet bli väldigt förskjuten mot back-end och rapportskrivande.

## 8.2 D2 - KPI

Tre stycken key performance index har använts: *psykosocial arbetsmiljö*, *velocity* och *stakeholder satisfaction*. Dessa har i sin tur brutits ner i egna KPIer för att möjliggöra mätning av gruppens huvud KPIer.

### 8.2.1 Psykosocial arbetsmiljö

Den psykosociala arbetsmiljön består av gemenskap, trivsel, stress och upplevd kontroll över arbetet, se figur 9.



Figur 9. Psykosociala arbetsmiljöns fyra mätområden gemenskap, trivsel, stressnivå och kontroll.

Dessa har mätts vid de veckovisa mötena med anonyma enkäter. Anonymitet har använts för att svaren ska vara så sanningsenliga som möjligt och inte påverkas av eventuellt gruppträck. De

valda KPI:erna har fungerat bra, de har varit enkla att greppa och sätta en siffra på. Varför arbetsmiljön mättes är på grund av att välmåendet i gruppen har stor påverkan på gruppens prestation och på processen. Syftet var att först när gruppens välmående är kartlagt och visualiserat, kan förbättringsåtgärder initieras. I efterhand anses dock att de psykosociala KPI:er hade kunnat användas på ett bättre sätt. Vissa medlemmar har upplevt att KPI:erna har mestadels bara samlats in för sakens skull och sett det som något som ska reflekteras över i reflektionsrapporten. En bättre användning hade varit att se dessa som ett verktyg för att förbättra gruppens prestation och jobba aktivt med dem. Efter insamling av gruppens svar hade det genomsnittliga svaret kunnat vara en indikator till att se om något behöver förbättras och hur det isåfall skulle ske, detta har inte gjorts. Resultatet hade kunnat bli en effektivisering i arbetet som görs som en följd av att samtliga gruppmedlemmar trivs.

### 8.2.2. Velocity

Varje User Story fick tilldelat sig effort points och ett acceptanskriterium som representerar projektets *Definition of Done, DoD*. Målet var att mäta hur mycket jobb som är gjort under en tidsram och att ha fokus på kvalitet. Generellt upplevde gruppen det svårt med att göra bra uppskattningar och fick träna under fem sprintar på att försöka estimeras bättre. Under Sprint No.1 överestimerades även Velocity-KPI:t på grund av feltolkning och därför fick den 100%.

### 8.2.3 Stakeholder Satisfaction

Stakeholder satisfaction har delats upp i *Completeness, Relevance* och *GUI*. Dessa är menade att ge en kontinuerlig bedömning av produkten från *Product Owner's* perspektiv. Alla kriterier bedöms på en skala från ett till tio där ett motsvarar ett negativt omdöme och tio är positivt. Så varje kriterium kan få högst 10 poäng och tillsammans kan de max få 30 poäng under en sprint. *Product Owner* har i det här fallet bestått utav olika personer som har bedömt subjektivt. Poängsättandet varierade beroende på representant men det gav en uppfattning om att gruppen var inne på rätt spår och vilken riktning fokus borde ligga på. Som tidigare nämnts upptäcktes det ganska sent att alla *Product Owner's* önskemål aldrig kan uppfyllas. En lärdom som drogs av det här var att önskemål måste prioriteras så att det går att fokusera på den mest kritiska funktionaliteten istället.

## 8.3 D3A - Product Backlog

Under arbetet har gruppen missuppfattat konceptet med produktbacklogg till viss utsträckning. Ny och uppdaterad funktionalitet har inte kontinuerligt lagts upp i produktbackloggen efter kundmöten utan snarare antecknats. Från dessa anteckningar har gruppen sedan skrivit *user stories* på den funktionalitet som prioriterats och tänkt genomföras. Dessa har sedan lagts till i product back-loggen. Alla de synpunkter *Product Owner* har haft har alltså inte slutförts utan en

del av dem har gruppen prioriterat bort, dels på grund av olika åsikter från olika kundrepresentanter men även på grund av den begränsade tiden. Dessa *User Stories* skulle lagts till i *Product Backlog* för att representera all den funktionalitet som önskats av kund.

Den slutliga Product Backlog kan besökas via länken:

<https://github.com/filipni/Software-Engineering-Project/blob/master/documents/D1-D4/ProductBacklog.pdf>

## 8.4 D3B - Business Model

### 8.4.1 Key Partners

Göteborgs Hamn samt STM var initialt sedda som de två största samarbetspartnerna i kursen. Det har visat sig att detta var felaktigt. De aktörer som gruppen samarbetat närmast med är inte STM, utan mer precist PortCDM. Göteborgs Hamn har heller inte varit inblandade i framtagandet av produkten, det har snarare varit Sjöfartsverket. Utöver dessa har naturligtvis Chalmers, eller mer specifikt, Håkan Burden, varit en stark samarbetspartner.

### 8.4.2 Key Activities

Eftersom att uppgiften för kursen var att utveckla en front-end applikation sattes det som huvudaktivitet. Det har dock visat sig att projektet till minst lika stor del består i att utveckla back-end och förstå hur PortCDM fungerar samt att dokumentera och driva arbetet på ett hållbart och effektivt sätt.

### 8.4.3 Key Resources

För att förstå och tackla komplexiteten i mjukvarudesign och -utveckling har internet varit den största källan till hjälp. Lärare och handledare har varit till mindre hjälp i de tekniska aspekterna av utvecklingen och till större hjälp i de mer överskådliga frågeställningarna i projektet. Resterande kunskap har erhållits från tidigare kunskaper i IT samt i projektledning.

Utöver de initiala verktygen som gruppen haft tillgång till vid kursens början har en rad nya verktyg utvecklats under kursens gång samt en del gamla förbättrats. Framförallt har dessa nya verktyg varit till för att underlätta arbetet med utvecklingen av programmet samt dokumentationen av projektets utveckling, t.ex. JavaFX, Photoshop och Trello.

#### 8.4.4 Value Proposition

Från början var det tänkt att det skulle utvecklas totalt en applikation för lots organisationen. Efter övervägande med *Product Owner* har detta ändrats till två stycken teoretiska applikationer, en för lotsen och en för lots organisationen.

#### 8.4.4 Resterande

Överlag har resterande aspekter av Business Model förblivit oförändrade. *Customer Relationships* är fortfarande lotsen i Göteborgs hamn, *Channel* är fortfarande STM (möjligtvis PortCDM), *Customers Segments* är fortfarande aktörer inom hamnverksamheten samt att *Cost Structure* och *Revenue* är obefintlig i och med att detta är ett teoretiskt fall i en kurs.

### 8.5 D4 - Halftime Evaluation

En av de aspekter som tas upp i Halvtidsutvärderingen är gruppens höga "bussfaktor". Denna aspekt är intressant eftersom strategin, med uppdelning baserat på kompetens och inriktning, varit både positiv och negativ. På den positiva sidan har gruppen, i och med de specifika gruppindelningarna, haft en mycket effektiv första halva av projektet. På den negativa sidan har dock den höga "bussfaktorn" lett till att gruppens resurser inte utnyttjas fullständigt och att projektet löpt en ganska hög risk när till exempel back-end utvecklarna varit sjuka eller otillgängliga.

Kunskapsutbytet med aktören *Tug Life* var mycket givande för utvecklingen av projektet. Eftersom att gruppen tidigare inte genomfört något projekt av denna storlek var det intressant att se hur en annan grupp angripit deras respektive problem. Idealt sett skulle en sådan process genomförts varje sprint. På så sätt kanske gruppen hade kunnat undvika problematiken med att back-end inte fungerade förrän på demonstrationsdagen.

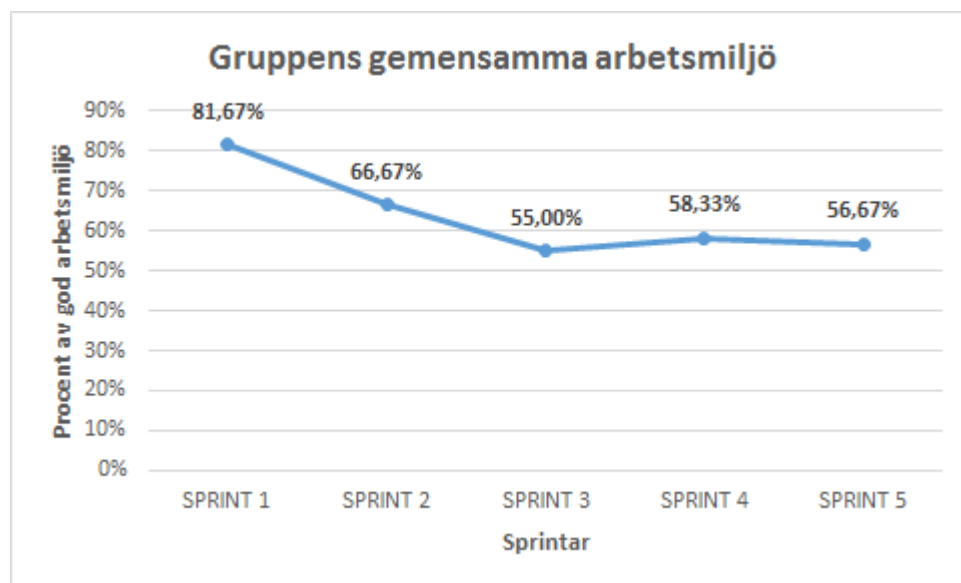


## 9. KPI charts, 0-1p

Nedan redovisas de diagram över de olika KPI:er som har samlats in under processens gång.

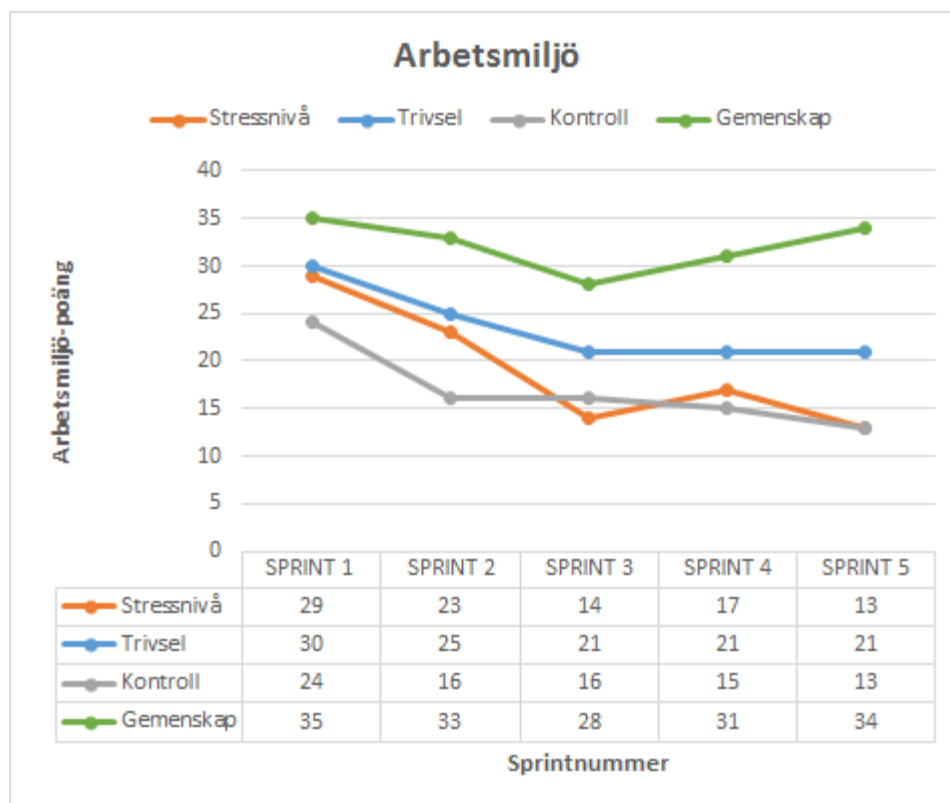
### 9.1 KPI No. 1 - Arbetsmiljö

Arbetsmiljöenkäten bestod utav fyra frågor för att kartlägga gruppens stressnivå, trivsel, kontroll över arbetet och gemenskap inom gruppen. Frågorna i sin tur har haft olika svarsalternativ och svaren har varit graderade i en skala från 1-6 som går från negativ till positiv. Desto högre poäng ju högre arbetsmiljö. Maximal poäng är att få 6 på varje fråga och därför har arbetsmiljöpoängen räknats ut med totalen delat på vad gruppen svarat och resultatet visas i procent.



Figur 10. Linjediagram över gruppens gemensamma arbetsmiljö under fem sprintar.

I början av projektet mättes arbetsmiljön till nästan 82% av 100% möjliga, se figur 11. Gemenskapen och känslan av kontroll var hög samt stressen var låg vilket genererade en hög trivsel, se figur 12. I efterföljande sprintar sjönk arbetsmiljön till nästan 50%, där det är främst en högre grad av stress och minskad känsla för kontroll som ligger bakom fallet.

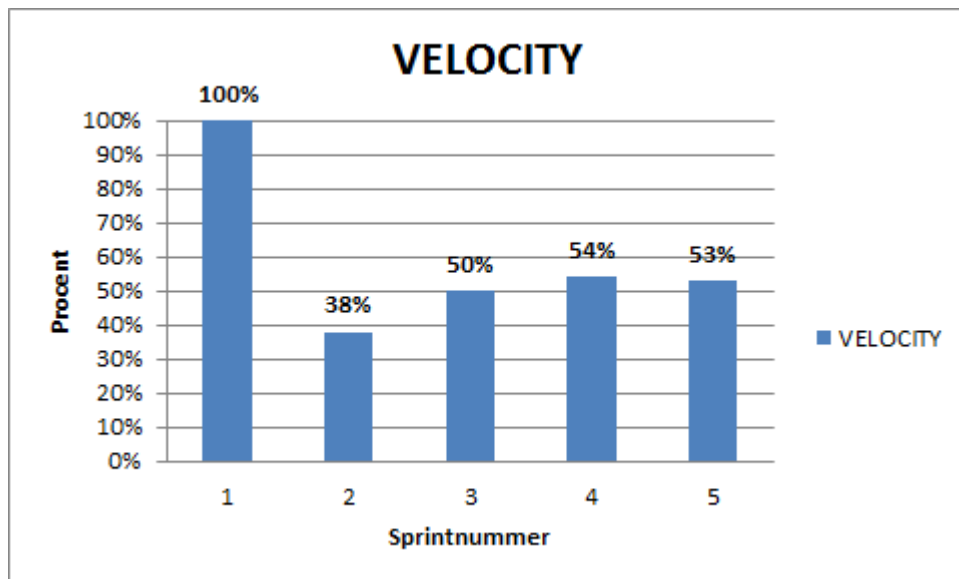


Figur 11. Linjediagram över arbetsmiljöns fyra mätetal under fem sprintar.

Gemenskapen har fått höga mätetal under hela processen vilket är beundransvärt eftersom gruppen inte har träffats så mycket i helgrupp utan mest jobbat i mindre team inom gruppen eller individuellt. Däremot har de andra faktorerna bara sjunkit under processens gång som kan avläsas i figur 12, har Sprint No.3 låga mätetal. En stark bidragande faktor till de lägre mätetalen i Sprint No.3 är att det uppstod en viss förvirring och kontrollbrist. Det var ett planerat rollbyte som inte skedde och gruppen hade fortfarande inte helt greppat uppdraget vilket påverkade känslan av kontroll. När känslan av kontroll minskade ökade stressen och även trivseln minskade. Stressnivån inom teamet var som högst i slutet av processen men gruppdynamiken var bättre.

## 9.2 KPI No. 2 - Velocity

Under Sprint No.1 överestimerades Velocity-KPI:t på grund av feltolkning och därför fick den 100%. Gruppen blev bättre på att splittra upp user stories inför Sprint No.3 jämfört med Sprint No.1 och Sprint No.2. Detta gjorde att gruppen undvek att gå miste om samtliga poäng då en liten del av en *user story* inte blev klar. Även i Sprint No.4 och Sprint No.5 borde *User Stories* blivit uppdelade ännu mer. Det har varit en utmaning för gruppen att estimerar *User Stories effort points* och de *User Stories* som inte genomfördes eller är halvfärdiga togs inte med i KPI:n.



Figur 12. Stapeldiagram över velocity i procent under fem sprintar.

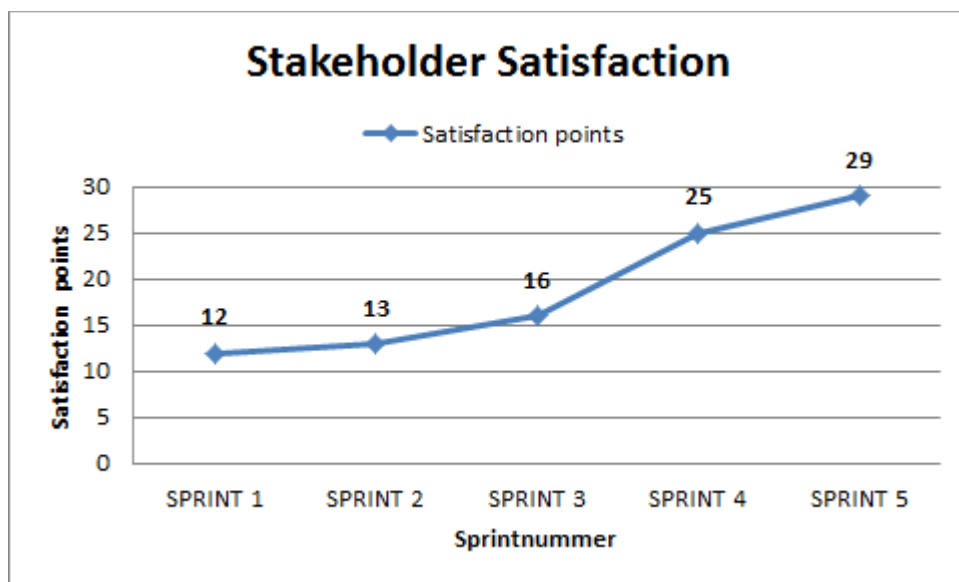
## 9.3 KPI No. 3 - Stakeholder Satisfaction

Nedan visas en tabell över Stakeholder Satisfaction och dess tre beståndsdelar. De tre delarna summeras varje sprint till antal satisfaction points där 30 poäng per sprint är max antal, då varje beståndsdel kan få mellan 1-10 poäng.

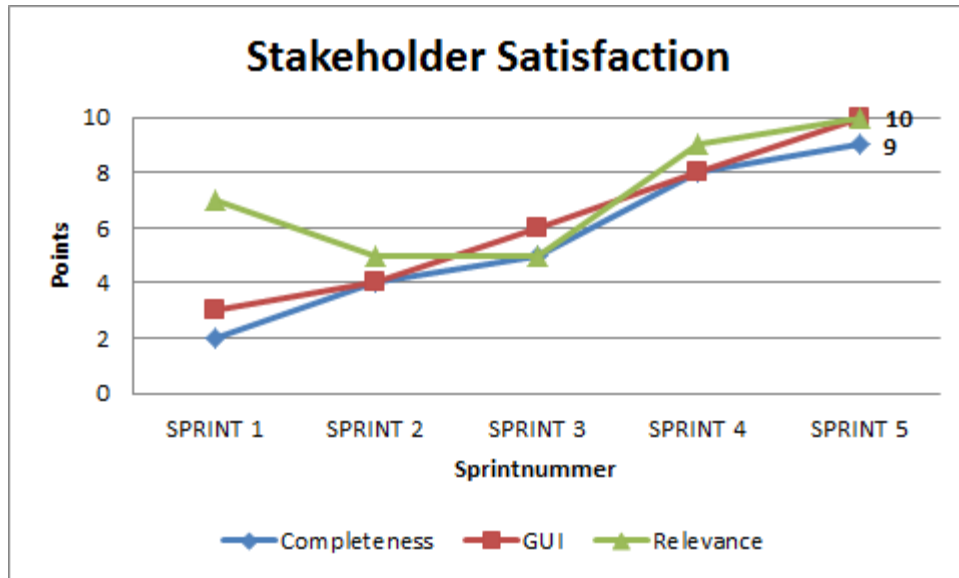
Stakeholder Satisfaction	SPRINT 1	SPRINT 2	SPRINT 3	SPRINT 4	SPRINT 5
Completeness	2	4	5	8	9
GUI	3	4	6	8	10
Relevance	7	5	5	9	10
Satisfaction points	12	13	16	25	29

Tabell 2. Stakeholder Satisfaction med de tre mätetalen completeness, GUI och relevance under fem sprintar.

Trots att poängen växlade hade kundnöjdheten en positiv utveckling och under sista sprinten var resultatet 29 poäng av 30 möjliga, se figur 13.



Figur 13. Ett linjediagram som summerat poängen från completeness, GUI och relevance och således utgör Stakeholder Satisfaction KPI:n under fem sprintar.



Figur 14. Ett linjediagram över completeness, GUI och relevance som tillsammans utgör Stakeholder Satisfaction KPI:t.

## 10. Acceptanstest och interaktion med intressenter

Acceptanstester har hållits veckovis med representanter från *Product Owner*. Dessa har skett i form av att *Mock-GUI* samt front-end har demonstrerats. Anledningen till att båda har visats upp är att front-end demonstrerar var projektet är idag medans *Mock-GUI* visar var projektet är påväg. På så vis får kunden både en uppfattning om hur långt på vägen gruppen kommit samt hur gruppens vision av den slutgiltiga produkten ser ut. Uppvisningen av front-end möjliggör även en mer levande och realistisk känsla av produktens funktionalitet, något som ett *Mock-GUI* inte kan ge. Utifrån denna demonstrationen har kunden gett feedback på befintlig funktionalitet samt ytterligare funktionalitet som kunden vill ha men även svar på de KPI:er som varit aktuella. Detta gjorde att feedback kunde utvärderas och implementeras samt gav information om att gruppen var på rätt spår och hur långt ifrån klara kunden ansåg att applikationen var.