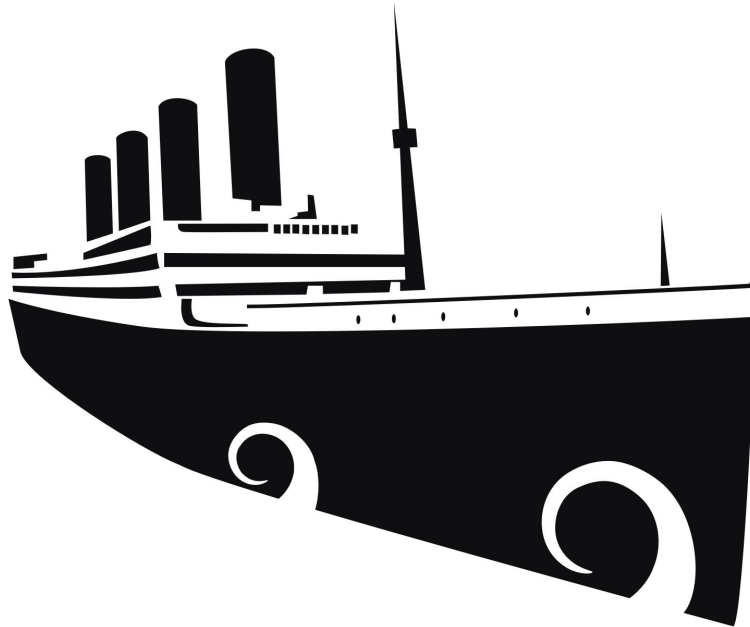


Reflektionsrapport

Software Engineering Project - DAT255

Project Titanic



Elin Forsberg
Jakob Larsson
Max Lundström
Oscar Sands
Tobias Stenshagen
Oskar Wahlbäck

Introduktion

Javaapplikationen Titanic utvecklades i kursen Software Engineering Project (DAT255) vid Chalmers tekniska högskola. Applikationen är en del i ett nätverk av applikationer som tillsammans syftar till att förbättra och underlätta informationsdelning och meddelandehantering i Göteborgs hamn. Titanic är en del i denna utveckling men behandlar endast VTS:s meddelandehantering och information. Applikationen utvecklades genom att använda sig av scrum som är en metodik för agil utveckling. Denna rapport är ett resultat av gruppens reflektioner över arbetssättet under projektets gång.

Introduktion	2
Tillämpning av scrum	4
Roller	4
Socialt kontrakt	5
Parprogrammering	5
Stand-up meetings	6
Time distribution	6
Effort, velocity and task breakdown	7
Reflection on the sprint retrospectives	8
Reflection on the sprint reviews	8
Best practices for using new tools and technologies	9
Eclipse SourceTree / IntelliJ, Maven och GIT	9
Slack	10
Trello	10
Latex/Google docs/drive	10
Reflection on the relationship between prototype, process and stakeholder value	11
Utvärdering av D1 Legodokumentet	13
Utvärdering av D2 KPI-dokumentet	14
Utvärdering av D3 Backlog & Business model canvas	15
Backlog	15
Business Model Canvas (BMC)	15
Utvärdering av D4 Halvtidsutvärdering	17
Appendix A - Documentation of sprint retrospectives	18
Appendix B - Documentation of sprint reviews	20
Appendix C - KPI-charts	24

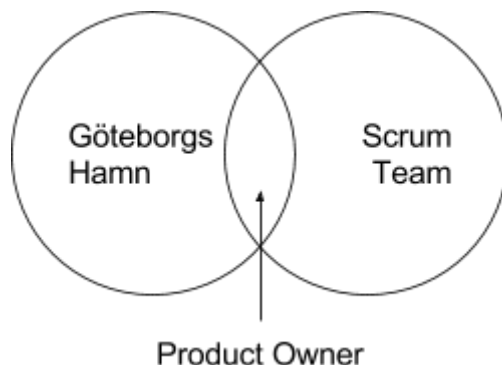
Tillämpning av scrum

Detta kapitel förklarar hur scrum applicerats under projektets gång och gruppens reflektioner kring det redovisas.

Roller

Inom projektet har vi använt SCRUMs fördefinierade roller Scrum-master, Product owner och Development team. Dessa roller har tilldelats personer som sedan haft denna roll under hela projektets gång för att få en ökad kontinuitet och effektivitet. Dock hade Scrum mastern ett flertal uppgifter, så dessa valdes att delas upp mellan flera individer i gruppen. En av gruppmedlemmarna har varit ansvarig för den huvudsakliga kontakten med de övriga grupperna och att gå på scrum of scrums. Vidare har det varit en annan gruppmedlems uppgift att uppdatera backlogen i Scrum boarden, så att User stories läggs till och flyttas. En tredje gruppmedlem har varit ansvarig för att hålla sprint retrospectives. En annan uppgift som Kniberg (2016) nämner är Scrum masters är att se till att tider hålls för möten. I denna grupp valdes att lägga det ansvaret på samtliga medlemmar då det föll sig mest naturligt. Det hade kunnat vara fördelaktigt om person hade varit Scrum master och varit ansvarig för alla uppgifter eftersom den personen hade kunnat ha en tydligare överblick över hela arbetet ur fler aspekter. Trots detta anser gruppen att fördelningen har fungerat väl tack vare att gruppen i stor utsträckning arbetat tillsammans. Att personer i gruppen har varit ansvariga för olika delar har inneburit att alla uppgifter uträttats.

Under detta projekt var Product Owner inte definierad och under mötena med Göteborgs Hamn deltog alla gruppmedlemmar och klargjorde vad VTS önskade ha för funktionalitet. Detta för att få en så bra bild och förståelse som möjligt för hur produkten kunde skapa värde. Sedan definierade Scrumteamet User stories då Göteborgs Hamn inte hade möjligheten att skapa dessa utifrån Scrum metodiken. Figuren nedan visar en grafisk representation av hur gruppen uppfattade att Product Owner var en roll i kombination mellan beställare och utförare, dvs. Göteborgs Hamn och Scrum teamet. Gruppen tror dock det vore fördelaktigt att ha en specifik Product owner eftersom det skulle bidra till tydligare prioritering. Under projektets gång träffade gruppen olika representanter från hamnen vilken innebar en viss variation i efterfrågan av features.



I detta projekt ingick alla medlemmar även i development team eftersom alla arbetade med utvecklingen av produkten. Utöver de fördefinierade roller inom scrum har gruppen utsett en

deadlineansvarig inför varje inlämning. Denna hade det yttersta ansvaret för att se till att deadline hölls och att de filer som hörde till en inlämning var uppladdade. Gruppen valde detta eftersom tidigare erfarenheter visat att det ibland uppstod brister i kommunikationen som lett till att deadlines missats.

Gruppen anser att rollfördelningen har fungerat väl under arbetets gång och att detta har resulterat i att arbetet fungerat bra. Rollfördelningen har varit förhållandevis lös, men då gruppen mest arbetat i helgrupp eller mindre grupper men ofta vid samma tid och plats har arbetet fungerat väl. Med en lös rollfördelning menas att gruppmedlemmar har varit ansvariga för respektive uppgift men uppgiften i sig har i de flesta fall lösts av hela eller delar av gruppen. Den lösa rollfördelningen och att gruppen arbetat mycket tillsammans har gjort att hela gruppen fått träna på och fått en bredare inblick i Scrum metodiken. Ett alternativ för att uppnå detta hade varit att rotera rollerna inom gruppen men detta valdes bort för kunna ta vara på den kunskap vi erhållit i föregående sprint. Vidare tror gruppen att en mer strukturerad och väldefinierad rollfördelningen skulle vara att föredra om projektet varit mer omfattande och under ett längre tidsspann.

Socialt kontrakt

Gruppen har använt ett socialt kontrakt där det beskrivs vilka förväntningar och åtaganden som finns både inom gruppen och mellan individer. Under diskussion framgick att gruppen tyckte det var viktigt att hålla tider för att undvika irritation och konflikter. Vidare fanns det en vilja att arbetet skulle vara effektivt och givande och gruppen tyckte att transparens och ett öppet klimat där inga frågor är dumma frågor var viktigt. Dessa två punkter är grunden i det sociala kontrakt som gruppen utformade.

Under projektets gång har kontraktet inte använts aktivt men tillsammans med den föreliggande diskussionen om förväntningar inom gruppen har det möjliggjort att samarbetet fungerat effektivt och konfliktfritt. I efterhand tycker gruppen det var värdefullt för projektet att lägga ner tid att diskutera förväntningar och individuella förutsättningar innan arbetet påbörjades. Under hela arbetets gång har samarbetet inom gruppen fungerat väl.

Parprogrammering

Inom gruppen hade individer olika scheman och parprogrammering gjorde det möjligt för olika par att till viss del under sprintens gång individuellt lägga upp arbetet. Individuella scheman hade kunnat hanteras genom att dela upp Tasks mellan enskilda individer men parprogrammering ansågs ha flera fördelar gentemot att programmera individuellt. Vidare hade samtliga gruppmedlemmar använt arbetssättet under tidigare kurser och tycker att det är ett bra sätt att arbeta på.

Gruppen valde i ett tidigt skede att använda sig av parprogrammering eftersom det på ett naturligt sätt involverade hela gruppen i arbetet. Metodiken möjliggjorde att gruppen lärde sig från varandra genom att dra nytta av varandras styrkor och tidigare erfarenheter. Detta effektiviserade arbetet eftersom den tid som gruppmedlemmar allokerade till att lära sig delar där tidigare erfarenheter saknades eller var bristfälliga kunde minimeras. Exempelvis märktes detta tydligt vid arbetet med verktyg som Trello, Api:er och Swing-paketet där delar av gruppen hade använt verktygen i stor utsträckning medan det var helt nytt för andra.

Parprogrammering visade sig också vara motiverande. Gruppen upplevde att vid tidpunkter då problem uppstått var det både enklare och roligare att hantera detta tillsammans gentemot det skulle vara att göra det individuellt. Genom att arbeta i par tror gruppen också att lösningarna blev bättre eftersom individer inom paren kompletterade varandra och fler infallsvinklar beaktades.

För att säkerställa att hela gruppen arbetade mot samma mål trots arbete vid olika tid och på olika platser krävdes att User stories bröts ner i små väldefinierade Tasks och gruppen använde Git och Github. I början hade gruppen problem att använda git och github vilket innebar svårigheter att dela kod och att arbeta franskt. Detta hanterades genom att gruppen under första sprinten frångick konceptet parprogrammering och istället arbetade i helgrupp för att tillsammans skapa en tydlig målbild med projektet. Med tidens gång blev gruppen dock bättre på att bryta ner User stories i tydliga Tasks och lyckades också koppla upp sig mot git och github, vilket resulterade i att parprogrammeringen fungerade bättre över tid.

Stand-up meetings

Stand-up meetings hålls enligt scrum metodiken dagligen för att vid korta möten diskutera 3 frågor.

1. Vad gjorde du igår?
2. Vad kommer du göra idag?
3. Finns det några hinder?

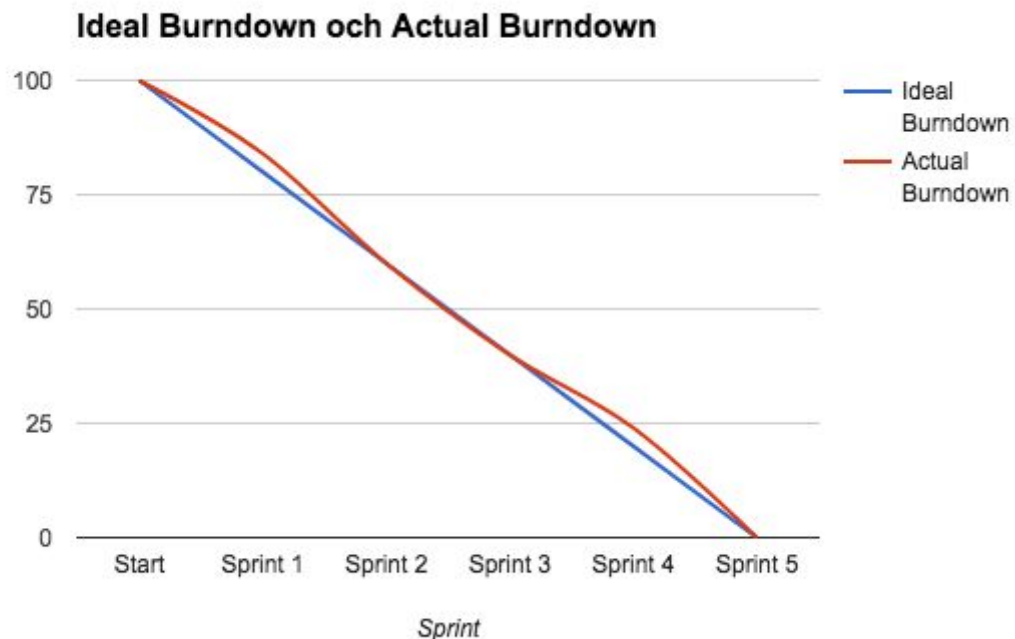
Mötet är ett sätt att för gruppen att förstå vad som har åstadkommit och vad som är planerat i den närmsta tiden för att kunna skapa en bild av vad som är kvar att göra på projektet. Mötet är inte ett möte för att deklarerat projektets status utan snarare ett möte där deltagarna gör åtaganden gentemot varandra.

Då projektet genomfördes på deltid vid sidan av andra kurser blev det problematiskt att hålla dagliga stand-up meetings. Under de flesta sprintar avsatte gruppen gemensamma dagar för att ses och jobba på samma plats. Tack vare att gruppen arbetade nära varandra och att den var relativt liten fungerade kommunikationen inom gruppen bra. Detta gjorde att alla visste vad alla gjorde och ett faktiskt möte med genomgång av frågorna blev överflödigt då detta skedde kontinuerligt under arbetets gång per automatik. Gruppen tror dock att det skulle vara ett betydligt värdefullare verktyg vid större utvecklingsprojekt eller om arbetet delats upp mer individuellt. Då skulle en kontinuerlig uppdatering under arbetets gång inte vara möjlig i samma utsträckning som vid detta projekt.

Time distribution

Gruppen anser att tidsfördelningen har varit jämnt fördelad på alla medlemmar under projektets fortskridning. Dock har gruppen även insett att det inte är tiden som varje enskild medlem lägger som är den viktigaste aspekten, eftersom vissa individer är snabbare än andra. Vidare har det varierat hur mycket utdelning gruppen erhållit av den lagda tiden varje enskild vecka. De första veckorna innehöll mycket inläsning av litteraturen, API förståelse och att skapa en överblick, vilket inte direkt skapade värde för kunden, utan snarare utvecklade gruppens interna processer. Dessa uppgifter kallas enligt

Scrum metodiken för Tech Stories. De sista veckorna gav större direkt värde till kund på tack vare att gruppen under de första veckorna lagt tid på att förstå kund och metodik. Detta illustreras i grafen nedan där burndownen av produktutvecklingen visas i förhållande till en ideal burndown. Vidare har vissa tekniska komplikationer såsom svårigheterna med att komma in i portCDM legat till grund för gruppen fått lägga mer tid de sista veckorna, och mindre tid andra, men i genomsnitt 20 timmar/vecka per person.



1. Illustration över Ideal Burndown och Actual Burndown, i grafen ses att actual burndown är på noll, detta för att även om inte produkten blev helt klar avslutades kursen och alla dessa moment var klara.

Effort, velocity and task breakdown

Scrum-metodiken beskriver att gruppen tillsammans ska estimerar velocity och antal points för varje User story. I detta projekt har vi gjort det med vad Kniberg (2016) beskriver som “Gut feel”, alltså magkänsla, vilket i början ledde till något felaktiga estimeringar. Dessa estimeringar har med tiden blivit bättre då gruppen har erhållit en större insikt i hur många points eller hur mycket tid en User story eller Task faktiskt innefattar. Vad gruppen kunde gjort annorlunda inom projektet är att använda metoden “time estimating using planning poker” som Kniberg (2016) beskriver och som gjordes under legoövningen, för att få bättre estimeringar tidigare i processen. Å andra sidan finns stora likheter mellan dessa metoder eftersom båda resulterar i en gemensam diskussion om hur många points en User story eller Task innefattar. Velocityn eller hastigheten gruppen hade begränsades i sin helhet till 20 timmar gånger 6 personer per vecka, vilket resulterar i 120 “mantimmar” eller 3 “mandays” enligt Kniberg. Gruppen valde dock att använda 10 storypoints per sprint att fördela på de

User stories. I efterhand insåg gruppen att det hade kunnat vara fördelaktigt att ha fler poäng eftersom det skulle underlätta att sprida ut poängen till fler små tasks. Då projektets tidsram var kort valde gruppen att inte använda fokusfaktorn, som är ett effektivitetsmått baserat på föregående sprints hastighet och resurser. Vid ett större projekt över en längre tid tror dock gruppen att fokusfaktorn hade varit intressant för att förbättra sin estimerade velocity över tid.

För att konkret kunna arbeta med en User story och skapa funktionaliteten den beskrev delade gruppen in den i mindre tasks. När User stories skapades utifrån diskussioner med Göteborgs hamn, delades de samtidigt in i mindre mer distinkta delar. Genom uppdelningen av User stories kunde dubbelarbete undvikas och hela gruppen kunde arbeta simultant. I början fungerade uppdelningen mindre bra eftersom de User stories som togs fram var för stora och abstrakta. Detta grundades dels i otillräckliga erfarenheter inom gruppen om arbetssättet men också i bristande förståelse i vad projektet skulle resultera i. Senare i projektet förbättrades detta dock avsevärt tack vare att gruppen vände sig vid arbetssättet. Gruppen upptäckte och håller med Kniberg (2016) om att User story breakdowns till task fungerade väl till att identifiera beroenden mellan de olika funktionerna och vad som behövde klargöras med kunden.

Reflection on the sprint retrospectives

Under projektets gång har retrospectives genomförts varje onsdag, alltså i slutet av varje sprint. Det har varit ett bra komplement och underlag till planeringen för nästkommande sprint. Kniberg (2016) beskriver att retrospectives kan utformas med tre kolumner, två som behandlar sprinten som varit och en som ser framåt. Gruppen utformade ett exceldokument med tre kolumner med frågorna “What went well during the sprint cycle?”, “What went wrong during the sprint cycle?” och “What could we do differently to improve?”. Dokumentation från gruppens retrospectives återfinns i Appendix A.

Den första frågan tar upp de delar gruppen skulle göra igen om samma sprint skulle göras om. Detta blev såväl ett sätt att berömma oss själva som en bekräftelse på vilka delar inom arbetssättet vi skulle fortsätta med. Den andra frågan berör de delar som fungerat mindre bra under sprinten och som skulle förändras om sprinten skulle göras om igen. De punkter som togs upp här blev också en del i planeringen framåt för att förhindra att samma misstag gjordes om. Svaren på den sista frågan var i många fall starkt relaterad till frågan som behandlade det som gått mindre bra under sprinten. Under flera retrospectives blev den tredje kolumnen gruppens lösning på problem som tagits upp i den andra kolumnen. Boken förespråkar att välja ett fåtal förbättringsåtgärder att fokusera på under nästkommande sprint. Under detta projekt upplevde gruppen inte att de problem som uppstod varken var för många eller krävde för stora förändringar för att inte alla kunna åtgärdas direkt under nästkommande sprint.

Reflection on the sprint reviews

Sprint reviews eller Sprint Demos som Kniberg (2016) kallar dem är en viktig del av Scrum processen där teamet får en möjlighet att visa upp vad de åstadkommit under sprinten. Vidare beskrivs att det finns fyra stora fördelar till varför en sprint review är viktig:

- Andra personer inom projektet får veta vad ditt team gör
- “Demon” kan få värdefull feedback från intressenter
- En möjlighet att socialisera och diskutera arbetet, få tips och råd
- Det förhindrar att delar lämnas halvfärdiga då de måste kunna visa att de fungerar

Vid Sprint review utgick gruppen från vad som åstadkommits i Trellon under sprinten. De User stories som hade valts för sprinten utvärderas i hur väl de var uppfyllda. Då gruppens definition of done var bristfällig i början blev det diskussionen kring om en story eller task var klar eller ej. Detta tog gruppen med sig och var i nästa sprint därför noggrannare med att utforma en DoD som alla kunde stå bakom. Detta möjliggjorde att gruppen bara inte hade någonting att visa upp utan även också hur detta kunde återkopplas till sprintens mål, vilket Kniberg (2016) också nämner är viktigt. Alla i gruppen har välkomnat detta inslag och ansett att det varit värdefullt att lägga fram vad som faktiskt åstadkommit under sprinten och momentet har synliggjort gruppens framsteg. Dessutom har det underlättat att fastställa målen för nästa sprint.

Best practices for using new tools and technologies

Gruppen har under arbetets gång använt ett flertal verktyg, metoder och tekniker vid utvecklingen. Nedan beskrivs mer utförligt vad gruppen sett för fördelar och nackdelar med respektive verktyg.

Eclipse SourceTree / IntelliJ, Maven och GIT

I början av projektet fick gruppen inte GIT att fungera, detta löste sig dock med tiden. Ett problem var att gruppen använde Eclipse som IDE och SourceTree som GUI för att hantera GIT, detta visade sig dock innehålla svårigheter och då några av de andra grupperna hade fått det att fungera med IntelliJ och Maven valde gruppen att byta till dessa. Fördelar med GIT är att det går att använda som en avancerad Google Drive med versionshantering, vilket gör att det går att “pull” ner kod och “push” upp kod till Github. Detta medför en förenkling av versionshantering, att alla kan arbeta samtidigt på olika delar av koden och att alla har tillgång till den senaste versionen av koden. Gruppen arbetade mycket tillsammans varvid vikten av att använda GIT kan diskuteras vid detta projekt. Det tog lång tid att implementera och lära sig använda verktyget som istället skulle kunna användas till att utveckla produkten, men det visade sig ändå vara en värd investering i slutet. Gruppen använde sig av Stack Overflow och andra forum som hittats via googling av problemen/användning av verktyget. De flesta problem har redan lösts av någon annan utanför gruppen sedan tidigare, vilken har kunnat utnyttjats. Detta har varit ett bra och effektivt sätt att hantera problem och nya lösningar. Fördelar med IntelliJ är att GIT och Maven fungerar mycket bättre än i Eclipse samt att användarvänligheten är bättre.

Nackdelen var att alla i gruppen enbart arbetat i Eclipse, vilket gjorde att gruppen fick lägga tid på att lära sig IntelliJ. När gruppen väl lärt sig IntelliJ och arbetet blev kompatibelt med GIT och Github underlättades arbetet och kunde fortgå mer effektivt. Detta eftersom arbetet nu kunde delas upp och ske parallellt. Dessutom kunde parprogrammering tillämpas i större utsträckning vilket gjorde att schemakrockar inte blev ett stort hinder för gruppen.

Slack

Gruppen har inte använt Slack för intern kommunikation utan exklusivt för kommunikation med andra grupper och handledare för att kunna fråga andra hur de löst liknande problem. Gruppen valde att arbeta mest i helgrupp och att verktygen för intern gruppkommunikation inte av så stor vikt. Fördelar med Slack är enligt gruppen att det blir en särskiljning från det "privata".

Trello

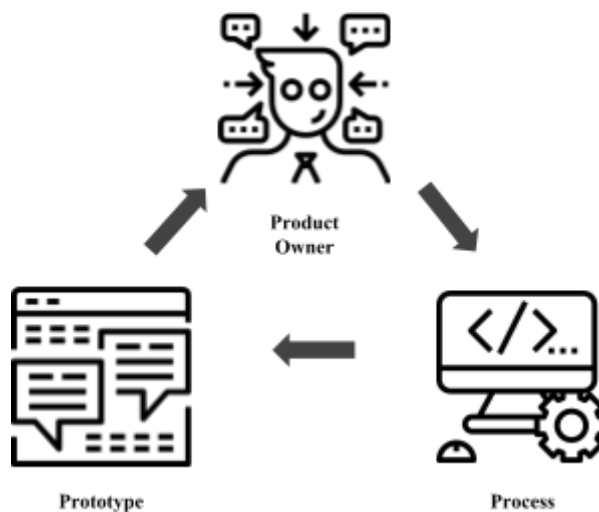
De flesta i gruppen var nya användare av Scrum och verktyg som Trello. Med tiden blev alla införstådda i strukturen, hur det skulle användas och den positiva inverkan av ha en god struktur. I början var alla gemensamt ansvariga för att hålla Trello uppdaterad men detta gick senare över till att en person var ansvarig och stämde av med ansvariga för hur User Stories fortgick. Trello gjorde arbetet väldigt strukturerat och att alla som arbetar med det får direkt feedback och ser hur korten flyttas mellan de olika listorna.

Latex/Google docs/drive

Gruppen planerade i början att använda Sharelatex för ordbehandling kontinuerligt, då det nyligen har fått nya funktioner likt de som återfinns i Google Docs. Detta övergavs dock inom ganska kort tid då det visade sig att värdet av att använda det inte översteg tiden för alla att lära sig verktygen, plus att det i Google Drive ingår fler funktioner såsom Google Spreadsheet.

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

Genom diskussion med hamnen identifieras värdefulla framtida egenskaper i prototypen. Dessa integreras genom gruppens processer och en prototyp tas fram. Genom att utveckla gruppens interna processer blir arbetet med att förbättra prototypen mer effektivt. Efter varje sprint har prototypen visats för intressenten och vidareutvecklingen har planlagts. Denna iterativa process illustreras i figur 1.



Figur 1. Relation mellan prototyp, process och värde för product owner

Redan under legoövningen (D1) blev det tydligt hur viktigt det var att förstå vilket värde som skapades till product owner. Detta värde skapas genom prototyper som senare blir produkter vilka byggs med hjälp av våra processer. Att förstå behov och värde som skapas till product owner gör det möjligt att omförhandla och bygga en prototyp som uppfyller kraven men samtidigt går att bygga med hjälp av de tillgängliga verktygen, vilka product owner inte alltid har insikt i.

Under första sprinten bokades ett möte med VTS Göteborg och hela gruppen åkte dit för att förstå hur systemet fungerar idag och hur produkten skulle kunna förbättra dagens arbete. Under mötet presenterades den plan som tagits fram för prototypens utformning och användningsområde vilken till hög grad inte motsvarade VTS behov. Vidare diskuterades hur planen kunde omarbetas för att skapa värde för VTS och mötet gav en insikt i hur arbetet faktiskt fungerade idag, vilket kunde användas vidare i projektet. Det som kom fram under detta mötet visade sig inte ligga i linje med vad hamnen hade tänkt sig, vilket gjorde uppgiften än mer komplex. Under projektets gång har prototypen diskuterats med intressenterna en gång i veckan. För att ha ett ständigt fokus på att prototypen ska skapa värde har en KPI varit kundnöjdhet och detta har utvärderats på en skala 1-5 vid varje möte. Vid dessa tillfällen har prototypen demonstrerats och planen för nästkommande sprint har diskuterats och fastställts. Trots detta har gruppen haft svårigheter att fokusera på att varje sprint ska addera värde som är synligt för intressenten.

En insikt gruppen kommit till i efterhand är att mer tid borde lagt på planering, utformning av specifika Tasks och User stories samt ett aktivt arbete med tunna vertikala delar. Jan-Philipp Steghöfer förklarade under sin föreläsning (L7) hur lösningar kan delas upp i smala vertikala delar, vilket innebär att en lösning delas in i små delar som omfattar allt från back-end till front-end och användargränssnitt. Genom denna metod skulle arbetet fokuserats på att skapa värde synligt för intressenten och efter varje sprint hade prototypen kunnat demonstreras och visa upp ny funktionalitet. Denna insikten kom till gruppen ganska sent under projektets gång men gruppen tror att om denna metod hade använts mer aktivt hade det haft stor betydelse för projektet. Dels genom att minska gapet mellan intressentens förväntningar och development teamets produkt men också för att ha ett tydligt fokus och skapa förutsättningar för ett mer effektivt arbetssätt. Att denna metod inte användes fullt ut innebar att back-end och front-end delvis utvecklades separat och när dessa senare skulle kopplas samman uppstod viss problematik kopplat till kompatibilitet som gruppen inte förutspått.

Trots att gruppen aktivt arbetat med att förstå hur prototypen ska utvecklas för att skapa värde har det emellanåt varit problematiskt eftersom representanten för intressenten varit olika från vecka till vecka. Detta har inneburit låg kontinuitet och med det har synen på hur produkten kan skapa värde varierat. I utgångspunkten vore det optimalt att träffa samma representant varje vecka för att få tydliga gemensamma mål för produkten. Då detta inte varit möjligt under detta projekt har gruppen hanterat problematiken genom att diskutera fluktuationerna i förväntningarna på prototypen med respektive representant. Dessutom har flera interna diskussioner haft för att tolka och prioritera egenskaper som av olika representanter framhållits som viktiga. Å andra sidan kan gruppen inse att det finns fördelar med att träffa flera olika representanter från intressenten för att få ett bredare perspektiv och en bättre slutprodukt som tar hänsyn till helheten. Sammanfattningsvis kan det sägas att detta komplicerat arbetet för gruppen men för intressenten adderat värde.

Gruppens arbetssätt har under projektets gång varit under ständig utveckling. Genom att utveckla dessa har arbetet för att ta fram en prototyp underlättats. Att förbättra processerna skapar indirekt värde för intressenterna eftersom förbättringsarbetet möjliggör effektivare arbetsmetoder vilket innebär att mer värde hinner skapas. Detta förbättringsarbete har främst gjorts genom att aktivt använda de insikter som framkommit vid Sprint Retrospectives. Då arbetssättet var nytt för hela gruppen tog det mycket tid i början av projektet för att lägga grunden och förstå verktyget som används för att förbättra processerna. Efter ett par sprintar blev retrospectives en naturlig del i arbetet och fokus kunde läggas på att faktiskt förbättra processerna och således kunde arbetet fortgå mer effektivt.

Utvärdering av D1 Legodokumentet

Efter legoövningen utformade gruppen tre strategier som skulle användas för att underlätta det kommande projektet. Den första strategin, *Aktivt verka för samarbete mellan grupper och inom gruppen*, beskriver hur detta ska främjas genom transparens mellan och inom gruppen och bra kommunikation. Under projektet har detta varit en viktig del och tack vare Scrum of scrums och trådar i Slack har olika gruppers lösningar kunnat återanvändas, t.ex. hämtningen och sändningen av meddelanden av gruppen Boat McBoaty. Detta har resulterat i ett effektivare arbete. Vidare hade gruppen problem med Git och Github och även då fanns möjlighet att ta hjälp av en grupp som stött på liknande problem. I strategin beskrivs att gruppen aktivt ska arbeta med att dela ut mindre arbetsuppgifter inom gruppen för att kunna arbeta på flera fronter men genom tydlig kommunikation ändå sträva mot samma mål. Detta har främst hanterats genom att arbeta på samma plats vid samma tid, men även till viss del i Trello. Sådär i efterhand tycker gruppen att strategin var bra och en viktig del för att säkerställa ett effektivt arbetssätt mellan och inom gruppen. Det skulle kunna förbättras genom att arbeta i närheten av fler grupper eftersom direkt kommunikation är enklare att förstå och förmedla. Dock finns det svårigheter att ordna det i praktiken på grund av att projektet är på deltid och alla har olika scheman.

Andra insikten som erhöles under legoövningen var värdet av att *förstå problemet som skulle lösas*. Detta skulle göras genom att ställa "varför"-frågor men det visade sig vara betydligt svårare än under legoövningen. Under första veckan besökte gruppen VTS och fick en förståelse för deras behov, något som inte helt stämde överens med PortCDMs krav. Genom att klargöra vad VTS'en gör idag förstod gruppen att det fanns en del av kommunikationen som kunde standardiseras genom meddelanden. Under detta projektet blev det svårt att tyda vilket problem som egentligen skulle lösas eftersom olika personer hade olika syn på vad som egentligen var problemet. Dessutom fanns det relativt lite tid att diskutera med olika intressenter och gemensamt skapa en målbild. Gruppen arbetade aktivt för att skapa en förståelse för problemet. I början av varje sprint diskuterade gruppen med intressent för vart fokus under nästkommande sprint skulle vara. Gruppen anser att detta är en essentiell del för att kunna skapa en värdefull produkt och tycker att det är något som kunde arbetats med mer aktivt, genom fler och längre dialoger med PortCDM.

Den tredje strategin som utformades efter legoövningen, *ta tid för kontinuerlig reflektion*, skapades för att identifiera förbättringar och ständigt utveckla de interna processerna. Detta ämnas göras genom att starta varje schemalagt möte med att diskutera saker som funkar bra och saker som kan utvecklas. Detta ligger dock inte i linje med scrum och innan projektet började omformulerades denna strategi. Istället för att starta varje möte med reflektion planlades retrospectives efter varje sprint, där gruppen diskuterade vad som gått bra respektive sämre under den gångna sprinten. Sammanfattningsvis är gruppen nöjda med sina insikter och strategier utformade efter legoövningen och anser att dessa är en bra avvägning för att skapa värde för såväl intressenter, prototyp och Development team. Dessa har visat sig vara mer komplexa i verkligheten men ändå varit värdefulla för projektet.

Utvärdering av D2 KPI-dokumentet

I början av projektet valdes 6 KPIer för att mäta och säkerställa såväl kundnöjdhet och en bra produkt som processförbättringar och trivsel inom gruppen. En illustration över de KPI:erna som valdes ut och mättes återfinns i appendix C. För att säkerställa kundnöjdhet fick representanten från hamnen ratea den utveckling som skett under den senaste sprinten mellan 1 och 5. På grund av att hamnen hade olika representanter vid de olika mötena uppstod viss problematik med denna KPI:n. En feature som diskuterats och anses skapa värde för en representant var lågt prioriterad av en annan. Vidare blev det svårt för representanten att utvärdera vad som hänt den senaste sprinten då personen i fråga inte sett hur det såg ut innan sprinten. Dessutom berodde detta mätetal till stor del av hur gruppen presenterade lösningen. Det är svårt att utvärdera kundnöjdhet helt objektivt, men då gruppen anser att det är en viktig del i produktutvecklingen anses KPI:n vara värdefull och underlätta för gruppen att förstå vad som skapar värde för kunden.

Gruppens trivsel internt har utvärderats varje vecka. Det har dock varit svårt att utvärdera hur trivseln i enbart detta projekt har varit under sprinten eftersom andra aspekter såsom andra projekt, vädret och andra utomstående faktorer påverkar ratingen. Det är svårt att undvika det. Under projektets gång har trivseln varit hög (över 4 på en skala 1-5) och gruppen tror att om det funnits problematik knuten till trivsel hade detta kunnat lyftas vid utvärdering av denna KPI:n, som skedde varje vecka. Detta har dock skett i helgrupp, varvid det kan varit svårt för individer att lyfta problematik om alla andra tycker sprinten varit trivsamt. Gruppen känner varandra väl och tror dock inte att detta varit ett problem.

För att säkerställa hög kvalitet har också antal buggar och antal buggar som lösts under en sprint mätts. Då gruppen på förhand inte definierat buggar har det varit svårt att avgöra hur många buggar som faktiskt dykt upp och lösts under en sprint. Därmed har detta mätetal snarare blivit en approximation gjord av gruppen än en väldefinierad KPI. Denna KPI hade kunnat och kanske till och med borde utvecklas vidare. De buggar som ändå identifierats under projektet har också lösts under samma sprint, varvid dessa KPI:erna är helt lika under detta projekt. Slutligen har velocity och estimated work remaining utvärderats efter varje sprint för att synliggöra framgång och hur mycket som är kvar att göras. Det har varit aningen komplext att avgöra hur mycket som är kvar att göra eftersom slutmålet inte varit helt bestämt på förhand utan snarare korrigerats från vecka till vecka. Då KPI:et illustreras syns att i början av projektet skapades relativt lite värde på grund av att inlärningsprocess medan i slutet skapades relativt sett mycket värde på kort tid.

Det har varit intressant att aktivt arbeta med KPI:er och mäta hur projektet utvecklar sig ur olika aspekter. Det har varit engagerande för gruppen eftersom prestation kontinuerligt har utvärderats och framgångar synliggjorts. Dock anser gruppen att det hade kunnat vara värdefullt att vidareutveckla KPI:erna och definiera de lite tydligare för att bättre kunna mäta dessa. Eftersom detta projekt endast varade under 5 sprinter erhöles endast 5 mätpunkter. Eftersom mätetalen främst är intressanta för att se hur något utvecklar sig vore det intressant med fler mätpunkter, då skulle förändringar i arbetssätt kunna kopplas till utvecklingen av KPI:er. Sammanfattningsvis tycker gruppen att de utvalda KPI:erna omfattar de mest intressanta aspekterna i projektet men anser att hur vissa av KPI:erna bedöms kunde utvecklats.

Utvärdering av D3 Backlog & Business model canvas

Backlog

De ursprungliga PBI (product backlog item) var User stories men var inte representativa "tårtslices" utan ändrades väldigt mycket innan första sprinten. De beskrev funktionaliteten i formen "produkten måste kunna hantera..." och inte en historia som användaren skulle kunna säga "Som VTS behöver jag kunna kommunicera...". Gruppen anser att detta var en värdefull övning som gav insikt i hur viktigt det är med riktiga User stories samt hur de är utformade. Vidare märktes också att detta kan användas för att fånga värde i andra kontexter. Att utforma detta efter kundernas User stories skulle nog resultera i en lite annorlunda utformning baserad på kundens önskan istället för, som ofta idag, baserad på gamla antaganden.

Business Model Canvas (BMC)

Samtliga i gruppen hade använt business model canvas i tidigare kurser och anser att detta är ett bra sätt att dela upp en företagsidéns olika delar för att sedan testa dessa i en iterativ process under arbetets fortlöpande. BMC'n användes dock mest i början för att klargöra projektet så att gruppen fick en överblick. Det som ansetts vara viktigaste är den högra delen av BMC'n där är Value Proposition som förklarar hur gruppen ska leverera värde till Customer Segment (VTS, PortCDM) samt hur detta kommer ske.

Key Partner

Under avsnittet togs projektets alla olika intressenter upp så som fartygen, VTS, port authority och PortCDMhost. Samtliga är fortfarande aktuella men inom ramen för arbetet har intressenten (PortCDM) varit de som utvecklingen och kravspecifikationer främst diskuterats med. Vid en utveckling av projektet anser gruppen att det är viktigt att samtliga intressenter är med i utvecklingen.

Key Activities

Inom key activities skall beskrivas vilka aktiviteter som genererar värde. I detta avsnitt beskrev vilka key activities VTS har idag. Detta borde istället beskriva i termer av vilka aktiviteter inom gruppen som genererar värde t.ex. utveckling av slutprodukten till kund. Vidare gör gruppen detta genom att arbeta agilt och mer specifikt med scrum-metodiken.

Key Resources

Liknande key activities är denna del beskriven utifrån ett annat perspektiv. Key resources är den samlade kunskap och humankapital som funnits inom gruppen och de verktyg som använts under utvecklingen av produkten såsom Eclipse, GIT, Google Drive, IntelliJ etc.

Value Proposition

Här är beskrivet vilket värde gruppen levererar till slutkunden. De ursprungliga punkterna är relativt abstrakta där exempelvis minskad risk för olyckor är ett indirekt värdeerbjudande då slutprodukten i första hand frigör tid som kan läggas på en mer övergripande övervakning av hamnen, vilket i sin tur kan leda till färre olyckor. Det gruppen lyfter fram som värdeerbjudande är något som kan erbjudas på längre sikt men ingenting som skapar värde direkt. Produktens värdeerbjudande i dagsläget snarare att underlätta och effektivisera det dagliga arbetet för VTS och främst de standardiserade meddelanden som sänds dagligen.

Customer Relationships

Inom avsnittet skall beskrivas vilket typ av förhållande gruppen bör ha till customer segments. Då kunden är VTS och PortCDM bör detta förhållande beskrivas utifrån dessa och inte utifrån fartygen som det är nu. Relationen till VTS och PortCDM bör vara så pass utvecklad att utvecklingen av produkten kan ske så agilt och resurseffektivt som möjligt med snabba iterationer och återkopplingar.

Customer Segments

Kunderna är VTS och PortCDM som skrev från början. Vad som dock märkts är att PortCDM's roll är betydligt större inom ramen för kursen och att utvecklingen i samarbete med VTS inte är relevant enligt PortCDM.

Channels

Channels beskriver hur kontakten med customer segment är tänkt att vara. Här skrev gruppen att den kommer ske i direkt kontakt vilket har överensstämmt väldigt bra med hur det skedde i verkligheten. Det har även förekommit kontakt via Slack och email.

Revenue Streams

Detta avsnitt har inte hanterats då inget monetärt utbyte har skett mellan customer segment och gruppen. Gruppen har haft försörjning i form av CSN.

Cost Structure

Gruppen har inte hanterat detta avsnitt.

Utvärdering av D4 Halvtidsutvärdering

Under halvtidsutvärderingen belystes att gruppen haft problem med att prioritera och förstå vad som skapar värde för intressenten men efter vidare diskussioner med hamnen kunde viktiga features urskiljas. Detta borde hanterats vid att arbeta mer med att dela in lösningen i tunna vertikala skivor, på så sätt hade förståelsen mellan Development team och intressent ökat. Då denna insikt kom efter knappt halva projektet blev det svårt att initiera en sådan förändring. Front-end och back-end hade utvecklats separat till stor del och befann sig på olika utvecklingsstadier, vilket delvis berodde på att gruppen fick byta utvecklingsmiljö och hade svårt att koppla upp sig till sandboxen. Gruppen är övertygad om att använda denna metodik från projektets start hade förbättrat den produkt som togs fram. Det hade underlättat samarbetet med hamnen och förståelsen för varandra. Dessutom hade det underlättat att byta riktning på vilka features som skulle inkorporeras. Då halvtidsutvärderingen utformades hade gruppen börjat förstå scrum metodiken och gjort metodiken till en naturlig del i arbetet. Under projektets slutfas blev det mer intensivt än vad gruppen räknat med och det blev mycket fokus på att skapa fungerande kod. Detta medförde att vissa delar såsom ordentlig planering och genomtänkta retrospectives blev sekundära. I efterhand tror gruppen att om hela projektet hade genomstrukturerats av bättre planering hade denna intensivitet i projektsslutskede kunde undvikits.

Appendix A - Documentation of sprint retrospectives

Sprint 1

What went well during the sprint cycle?	What went wrong during the sprint cycle?	What could we do differently to improve?
Efter första sprinten tycker gruppen att samarbetet fungerat väldigt bra. Projektet har varit ganska oklart och det har krävts att hela gruppen arbetat tillsammans vilket fungerat väldigt bra	Gruppen hade svårigheter att bedöma om en task och en user story var färdig eller ej	Lägga till definitions of done till varje task
Kommunikationen inom gruppen har fungerat mycket bra. Mycket tack vare att vi arbetat tillsammans.		

Sprint 2

What went well during the sprint cycle?	What went wrong during the sprint cycle?	What could we do differently to improve?
Till skillnad från förra sprinten har mycket arbete skett individuellt på grund av ledighet och olika scheman, men trots detta har alla lyckats hålla fokus på uppgiften och arbeta på individuellt.	Det vore fördelaktigt för projektet om gruppen hade tagit sig tid att ses och diskutera olika lösningar	Använda GIT och Github så att det fungerar att programmera på samma projekt på olika ställen
DoD fungerade bra		Arbeta med aktivt med parprogrammering

Sprint 3

What went well during the sprint cycle?	What went wrong during the sprint cycle?	What could we do differently to improve?
Även under denna sprint är gruppen väldigt nöjda med samarbetet och kommunikationen internt inom gruppen	Återigen mycket schemakrockar	Arbeta mer aktivt med trello
Under denna sprint tog vårt projekt en lite ny vändning och vi tvingades tänka om en del, men gruppen är väldigt nöjda med hur anpassningarna till de nya förutsättningarna fungerade	Har varit oklarheter kring vem som är ansvarig för respektive task/User Story	Bryta ner User Stories i fler tydligare tasks
	Strukturen kring vilka tasks som finns fungerar inte så bra	Göra individer ansvariga för olika tasks

Sprint 4

What went well during the sprint cycle?	What went wrong during the sprint cycle?	What could we do differently to improve?
Det fungerade väldigt bra med uppdelning av arbetet tack vare de åtgärder som togs efter förra Sprint retrospective	Haft väldigt höga ambitioner och prioriterat features som förvisso anses värdefulla men prioriterad dessa före features som är grundläggande för funktionaliteten	Tätare arbete och dialog med uppdragsgivaren
Arbetet blev mer effektivt eftersom mycket av arbetet kunde ske mer parallellt		Lagt ner mer tid på förarbete och planering

Sprint 5

What went well during the sprint cycle?	What went wrong during the sprint cycle?	What could we do differently to improve?
Haft ett tydligare fokus på vad vi måste göra och vad som snarare är en bonus om vi hinner göra	Lagt mycket tid på projektet för att vi underskattat hur tidskrävande en del moment skulle bli	Arbetat mer aktivt med utforma våra user stories som tunna vertikala tårtbitar

Appendix B - Documentation of sprint reviews

Sprint 1

Tech story	Task	Acceptance criteria	Slutförd	Kommentar
(10) Som grupp måste de för kursen nödvändiga verktyg fungera.	<ul style="list-style-type: none"> • Skapa Trello-konto • Skapa Github-konto • Få GIT att fungera • Inläring av trello • Få ordning på Google Drive konton 	När samtliga tasks är genomförda.	Delvis, GIT fungerar inte.	Gruppen fortsatte att försöka få GIT att fungera och lyckades med detta i senare sprints.

Sprint 2

User story	Task	Acceptance criteria	Slutförd	Kommentar
(10) Som VTS vill jag kunna skriva och se meddelanden (VIA GUI) som sedan ska kunna kommuniceras till andra aktörer.	<ul style="list-style-type: none"> • Besöka VTS • Skapa ett användargränssnitt med ett första utkast till hur meddelanden kan synas och skickas • Utveckla en panel där text kan skrivas in från användaren • Inläring av trello • Skriva en metod som hanterar meddelandet och för att sedan kunna skickas • Utveckla en sprint review-mall för att utvärdera KPI:er samt ytterligare diskussionsområden. 		Ja	Ingen DOD men gruppen beslutade att den var färdig.

Sprint 3

User story	Task	Acceptance criteria	Slutförd	Kommentar
(5) Som VTS vill jag kunna skapa standardiserade meddelanden som jag kan redigera och skicka för att snabbt kunna skriva vanligt förekommande meddelanden	<ul style="list-style-type: none">• Skapa ett sätt att redigera standardiserade meddelanden• Skapa ett sätt att lägga in och ta bort standardiserade meddelanden	När ett standardiserat meddelande skapats och raderats	Ja	-
(5) Som VTS vill jag kunna skicka meddelanden till PortCDM för att andra aktörer senare ska kunna ta del av dem	<ul style="list-style-type: none">• Skapa en lokal PortCDM• Skriva en metod som lägger in en string i ett Port Call Message så att det kan kommuniceras till PortCDM• Skicka meddelanden till lokal PortCDM	-	Ja	Ingen DOD men gruppen beslutade att den var färdig.

Sprint 4

User story	Task	Acceptance criteria	Slutförd	Kommentar
(7) Som VTS vill jag kunna hämta meddelanden från lokal PortCDM (VB) för att kunna se viktig information om fartygen	<ul style="list-style-type: none"> • Det krävs ett sätt att hämta de meddelanden som finns i vår VB • Vi behöver kunna filtrera ut meddelanden relevanta för VTS • Vi behöver en metod som strukturerar upp viktig information i meddelandet så att det är enkelt att förstå vad fartygen vill ha svar på 	DOD: När vi kan hämta rätt meddelande och utläsa ifrån det vad fartygen vill ha svar på	Nej, vissa problem med uppkoppling till VB	Att hämta meddelanden från VB löstes tidigt under sprint 5
(3) Som VTS vill jag kunna se till att rätt aktör nås av rätt meddelande som vi skickar för att veta att den information vi förmedlar kommer fram	<ul style="list-style-type: none"> • Vi behöver kunna sätta tags på meddelanden vi skickar • Som VTS vill jag kunna skicka det första Port_call message, med rätt parametrar, till Vessle kommuniceras med PortCDM • Skicka meddelanden till den lokala PortCDM 	DOD: Ett lyckat test med en annan grupp	Nej, det gick att ställa in rätt parametrar men kunde inte testat eftersom gruppen fortfarande hade problem med sandboxen	Flyttas till nästa sprint

Sprint 5

User story	Task	Acceptance criteria	Slutförd	Kommentar
(5) Som VTS vill jag kunna se till att rätt aktör nås av rätt meddelande som vi skickar för att veta att den information vi förmedlar kommer fram	<ul style="list-style-type: none">• Vi behöver kunna sätta tags på meddelanden vi skickar• Skicka ett Port_call message, med rätt parametrar, till Vessle via PortCDM• Skicka meddelanden till den lokala PortCDM	DOD: Ett lyckat test med en annan grupp	Systemet fungerade mot lokal version men uppkopplingen mot sandboxen misslyckades	Skulle behövt mer tid för att lösa uppkopplingen mot sandboxens version
(5) Som VTS vill jag kunna se till att vi tar emot alla meddelanden som vi behöver svara på, för att inte missa någon viktig information	<ul style="list-style-type: none">• Ta emot ett första Port call message, med rätt parametrar, från Vessle via PortCDM• Ta emot meddelanden från den lokala PortCDM	DOD: Ett lyckat test med en annan grupp	Systemet fungerade mot lokal version men uppkopplingen mot sandboxen misslyckades	Skulle behövt mer tid för att lösa uppkopplingen mot sandboxens version

Appendix C - KPI-charts

