

Final Team Reflection

DAT257/DIT257

AXE-N

Ahmad Al-Aref

Oscar Arvidson

Sara Galligani Vardheim

Tilda Grönlund

Sam Salek

Erik Wetter

Malte Åkvist

Handledare:

Alderic Wong

1. Introduktion

Vi har skapat en produkt för AB Lejonstaden, ett fastighetsutvecklingsföretag i Göteborg. För att utföra sina kalkyler inför ett nytt projekt använder de sig i dagsläget av en Excel-fil som de själva programmerat. Den är inte användarvänlig, inte effektiv att dela med intressenter och de behöver skapa ett nytt dokument för varje nytt projekt. Med bakgrund i detta bad de om en desktopapplikation som kan hantera flera projekt och utföra de nödvändiga beräkningarna på ett mer användarvänligt sätt. Dessutom ville de att resultatet av beräkningarna skulle kunna exporteras till en pdf-fil, vilken de skulle kunna dela med sina intressenter och kollegor.

Projektet sträckte sig över sex sprints, där Sprint 1 startade i läsvecka 4 och Sprint 6, den sista sprinten, startade under läsvecka 9. Varje sprint gick från måndag-måndag och bröts av det veckovisa mötet med vår Product Owner (PO). Projektets upplägg planerades i en business model canvas under läsvecka 1 (se figur 1).

<u>Key Partners</u> AB Lejonstaden, fastighetsbolag. Bidrar med kunskapsbasen för projektet.	<u>Key Activities</u> Statusmöten med kund. Interna möten. Sprints.	<u>Value Propositions</u> Effektivisering av lönsamhetsberäkning.	<u>Customer Relationships</u> Co-creation. Strävar efter möten en gång i veckan.	<u>Customer Segments</u> Finns endast en kund.
	<u>Key Resources</u> Företagskontakt, samt kunskapsbas för projektet. Handledning kring projektet, och vad som eftersträvas. Programmeringsmjukvara.		<u>Channels</u> Telefonkontakt. Genomgång och återkoppling möten via Zoom. Trello används för product backlog.	
<u>Cost Structure</u> Inga utgifter.			<u>Revenue Streams</u> Inga inkomster.	

Figur 1: Business model canvas skapad under läsvecka 1.

2. Customer Value and Scope

2.1 The chosen scope of the application

Nuläge:

Som tidigare nämnt var syftet med projektet att skapa en desktopapplikation för AB Lejonstaden för att effektivisera beräkningarna i deras kalkyler. Övriga viktiga aspekter av projektet var användarvänlighet och delbarhet. För att uppnå dessa kriterier konstruerades fem Epics: GUI Design, Beräkningar, Sparande, Exportera till PDF, och Användarvänlighet. Det bestämdes i samråd med PO att vår minimum viable product (MVP) skulle uppfylla alla dessa Epics. Projektets scope var i stort sett detsamma under samtliga sprints, med vissa designförslag från PO under de veckovisa mötena. Inför sprint 3 framkom dock en förfrågan om att kunna löpande bestämma vilka kostnadsposter som var momspliktiga, vilket lämpligtvis skulle lösas med en checkbox. Detta inkluderades dock inte i vår MVP, men uppnåddes ändå.

Önskvärt läge:

I en av de sista sprintarna framkom en förfrågan om att möjliggöra för import av Excel-dokument i programmet. Detta skulle förenkla användningen av applikationen då användarna inte skulle behöva mata in gamla projekt manuellt, utan enkelt föra över dem i det nya programmet. Eftersom det kom upp så pass sent under arbetets gång beslutades det att inte implementera denna funktion då det inte skulle hinnas med. Utöver detta uppnåddes MVP samt ytterligare funktionalitet utöver MVP.

Hur vi når önskvärt läge:

För att implementera import av Excel-dokument skulle framförallt mer tid krävas. Med mer tid hade nödvändig kunskap kunna samlas in och lämpliga åtgärder i koden göras. Den största anledningen till att denna funktionalitet inte hanns med var dock att frågan kom upp så pass sent under arbetets gång, vilket är något vi tar med oss in i framtida projekt. Med mer ingående inledande möten kring projektets scope hade denna förfrågan kunnat fångas upp tidigare och därigenom hinnas med under projektet, eftersom vi då hade kunnat disponera tiden annorlunda.

2.2 The success criteria for the team

Nuläge:

Inledningsvis var målet med projektet att skapa en desktopapplikation som vår PO kunde använda som ett alternativ till deras nuvarande system, vilket innebar att programmet behövde ha samma funktionalitet som Excel-filen. Ett annat mål för applikationen var att det skulle vara möjligt att skapa en PDF av innehållet i applikationen. Det var en del jobb att få till den funktionaliteten och ett tag uteslöt vi det från vårt success criteria. Efter ett par veckor insåg vi dock att det faktiskt var möjligt att implementera inom tidsramen för projektet och därigenom fick det återigen bli ett success criteria. Avslutningsvis var ett mål med projektet att ta tillfället i akt att vidga våra kunskaper genom att ta del av varandras erfarenheter för att erhålla kunskaper att applicera på framtida projekt.

Önskvärt läge:

Vi uppnådde våra success criteria eftersom funktionaliteten på applikationen överensstämmer med Excel-filen och det är möjligt att exportera innehållet till en PDF. Det har även skett ett stort kunskapsutbyte mellan gruppmedlemmar där alla har varit måna om att både lära sig och hjälpa andra. Däremot hade det varit önskvärt att kriteriet med PDF:en aldrig prioriterades bort till följd av tidsbrist, även om det löste sig och blev implementerat.

Hur vi når önskvärt läge:

För att se till att samtliga success criteria uppfylls och att inget prioriteras bort är ett alternativ att planera tiden på ett sätt som möjliggör uppfyllande av samtliga kriterier. Ett annat alternativ hade varit att noggrant formulera samtliga kriterier baserat på projektets tidsomfattning och resurser och därigenom avgöra vad som är möjligt.

2.3 User stories

Nuläge:

Projektet är som tidigare nämnt uppdelat i fem epics. Dessa epics omfattar projektets stora delar och skapar tillsammans en helhetsbild av projektet. Våra user stories är kopplade till respektive epic och ger således en tydlig koppling till området som behandlas. Våra user stories innehåller beskrivningar av vad PO önskar, en checklista för dess acceptance criteria med de tasks som ska uppfyllas samt en checklista för vår definition of done. Baserat på detta analyserade vi som grupp respektive user story och bedömde den förväntade

svårighetsgraden. Varje svårighetsgrad visualiserades med en färg - grönt, gult eller rött - för att enkelt avgöra hur stor arbetsbelastning vi kunde förvänta oss inför en kommande sprint.

Efter varje avklarad user story reflekterade vi över svårighetsgraden och jämförde detta mot de ingående förväntningarna. Respektive user story tog hänsyn till respons från PO och skapade således värde för dem. Detta skapade även värde för oss, då vi i små etapper kunde validera projektet och sammanslå detta till en välfungerande och efterfrågad produkt. Genom att dela upp arbetet i user stories, med väldefinierade acceptance stories, abstraherades arbetet och skapade en tydligare förståelse och en tydlig arbetsprocess.

Önskvärt läge:

Under veckomötena med vår PO har flertalet ytterligare aspekter inom varje user story lagts till, vilket ytterligare har utvecklat produkten efter kundens önskan. Hursomhelst hade det i framtida projekt varit gynnsamt att försöka etablera en proaktiv lösning, där flertalet av de omfattande ändringarna från PO hade varit berörda under en tidigare fas av respektive user story. Då respektive user stories svårighetsgrad evaluerades på förhand föranledde det, i kombination med de ytterligare adderade aspekterna från PO under respektive user stories, i att evalueringen i vissa fall ändrades och blev mer omfattande än förväntat.

Hur vi når önskvärt läge:

Genom att tydligt fånga POs åsikter och detaljer kring programmet kan en del av de tillagda punkterna under respektive user story under arbetets gång kunnat hanteras tidigare. Således hade evalueringen av svårighetsgraden också under ett tidigare stadie varit mer rättvis, och mer tid hade kunnat allokerats åt user stories som ansågs vara mer omfattande.

2.4 Acceptance tests

Nuläge:

Inledningsvis var planen för acceptance tests att diskutera de mest relevanta aspekterna med vår PO och att diskutera våra interna acceptanskrav (*definition of done, DoD*) internt. Vår definition of done var genomgående för hela projektet definierat enligt tre aspekter: godkänd av PO, testning (visuell och debug) och dokumentering genom JavaDocs. Efter vårt handledningsmöte vecka 6 adderades även code review. Godkännande av PO hanterades under våra veckoliga sprint reviews genom att vi förklarade och/eller visade för PO vad vi

hade gjort och vad det innebar för applikationen. PO fick därefter komma med synpunkter. Resultatet blev antingen att user storyn blev godkänd, eller att vi fick iterera över den under nästkommande sprint. Många av synpunkterna från PO fick oss att bättre förstå projektets riktning och PO:s önskemål och att invänta godkännande av PO var naturligtvis en mycket viktig del av att acceptera user stories. Även några övriga acceptance tests, utöver definition of done, utfördes, exempelvis körning av programmet med värdena från Excel-filen från vår PO. Detta tillät oss att se huruvida vår applikation faktiskt hade samma funktionalitet som Excel-filen.

Önskvärt läge:

Vi hade flertalet sätt att kolla huruvida vår kod fungerade som avsett och därigenom avgöra om den kunde accepteras, varav de flesta följdes mycket väl. I ett önskvärt läge hade samtliga acceptanskrav följts på ett enhetligt och konsekvent sätt. Exempelvis kunde code review ha följts bättre. Eftersom det tillfördes som ett acceptanskrav senare än exempelvis JavaDocs, och inte var en lika naturlig del av processen som exempelvis godkännande av PO, blev det något bortglömt. Som följd fick det inte heller den positiva effekt som det hade potential för, även om det arbetades med till viss del. Eftersom det här var första gången vi arbetade enligt Scrum var det mycket värdefullt att få tips från vår handledare, exempelvis om att använda oss av code review. Ett önskvärt läge hade varit att självmant kunna bedöma vilka acceptanskrav som är lämpliga och tillför värde.

Hur vi når önskvärt läge:

För att uppnå önskvärt läge hade vi kunnat vara mer noggranna och räkna in tiden för samtliga acceptanskrav i den uppskattade tiden för en user story. Vi hade även kunnat bestämma samtliga acceptanskrav vid samma tillfälle, eftersom alla krav därigenom hade blivit en lika naturlig del av processen. För att uppnå en djupare förståelse och därigenom självmant kunna bedöma vad som är lämpliga acceptanskrav är det bästa alternativet att öva.

2.5 KPIs

Nuläge:

De tre KPIer vi har använt oss av är stress, förståelse och produktivitet. Stress och förståelse mättes genom att varje gruppmedlem själv avgjorde hur stressad hen varit under den senaste sprinten respektive hur mycket hen förstått av programmet under den senaste sprinten. Båda

KPIer bedömdes på en skala 1-10. Produktiviteten mättes med hjälp av svårighetsgraderna vi satt på respektive utförd user story. Genom att se hur många user stories som avklarats under sprintens gång, och vilken svårighetsgrad dessa haft, kunde vi avgöra hur produktiva vi varit under sprinten. Produktivitet och stress kompletterade varandra väl, eftersom en lägre produktivitet ofta kunde förklaras av en förhöjd stressnivå. Vidare kunde en lägre produktivitet vara att föredra, för att förhindra att stressnivån höjdes alltför mycket. Om vi till exempel visste att vi skulle ha mycket att göra i andra kurser under en kommande sprint, så planerades färre user stories utföras under den veckan.

Önskvärt läge:

Eftersom två av KPIerna var väldigt individuella och subjektiva blev de ofta beroende av vad som hände i gruppmedlemmarnas andra kurser och övriga liv. Dessutom kunde de ibland bli lite feljusterade om någon glömde fylla i dem i tid och därför hade en annorlunda bild av sin stressnivå och förståelse under den gångna veckan. Därför hade ett önskvärt läge kunna vara att använda KPIer som beror mindre på yttre omständigheter. Ett annat möjlig önskvärt läge skulle kunna vara att hålla hårdare på att fylla i KPIerna i tid, för att minska variationerna i noggrannheten.

Hur vi når önskvärt läge:

För att minska beroenden på yttre omständigheter skulle vi i nästa projekt kunna använda oss av mer objektiva KPIer, som går att mäta i mer än varje individs upplevelse av den gångna sprinten. Det andra önskvärda läget, att hålla hårdare på att fylla i KPIerna, hade kunnat uppnås genom att inkludera en klausul kring detta i ett social contract. På så vis hade vi haft explicit underlag för att vi kommit överens om att fylla i KPIerna direkt efter avslutad sprint, med praktiska påföljder om det inte uppfylls.

3. Social Contract and Effort

3.1 Social Contract

Nuläge:

Under första veckan i kursen upprättade vi ett gruppkontrakt och lämnade in på canvas. Vecka 4, då projektet började dra igång på riktigt, justerade vi kontraktet något för att definiera hur konflikter skulle hanteras inom gruppen. Även vecka 5 justerades kontraktet då

vi lade till våra roller för att kontraktet skulle ge en bättre överblick över vem som gör vad i projektet. Bägge dessa justeringar skedde till följd av de tips vi fick under handledningsmötena. Efter vecka 5 justerades kontraktet inte ytterligare, men vi fortsatte att jobba utefter det.

Önskvärt läge:

Gruppkontraktet hade i ett önskvärt läge även kunnat utvecklas med vad respektive roll innebär. Samtliga gruppmedlemmar tilldelades en roll, vilket vecka 5 adderades till gruppkontraktet, men ingen beskrivning av vad detta innebar adderades. Det hade gjort kontraktet ännu tydligare, men vi valde att inte bearbeta gruppkontraktet ytterligare.

Hur vi når önskvärt läge:

För att nå ett önskvärt läge där gruppkontraktet hade varit så tydligt och översiktligt som möjligt hade gruppkontraktet kunnat bearbetas mer noggrant, både innan projektets start och kontinuerligt under projektets gång. Det är viktigt med ett väl genomarbetat gruppkontrakt och för att upprätta ett sådant krävs att tid allokeras åt bearbetning av det.

3.2 Time spent on the course

Nuläge:

I genomsnitt lade vi sammanlagt omkring 47 timmar per vecka på att arbeta med projektet. Som minst spenderade vi 22 timmar på projektet under den första sprinten, vecka 4, och maximalt 60 timmar vecka 6 och 8. Tiden inkluderar tid som spenderats på grupp- och handledningsmöten samt möten med PO. Eftersom anteckningarna fördes under vårt veckoliga torsdagsmöte inkluderades inte tiden som spenderades mellan torsdagen och måndagen under någon sprint. Gruppmöten brukade också räknas som endast 1 timme, trots att samtliga gruppmedlemmar närvarade, vilket även det kan ge en något missvisande bild om hur mycket tid som lagts under veckan.

Önskvärt läge:

Eftersom vi nått vår MVP kan det anses att tiden som lades på kursen var tillräcklig. Å andra sidan motsvarar kursen 50% arbetsbelastning, det vill säga 20 timmar per vecka per person, vilket borde ha resulterat i sammanlagd 140 timmar per vecka för gruppen. Ett önskvärt läge hade varit att all tid som är tänkt att spenderas på projektet hade utnyttjas, då det hade resulterat i uppfyllandet av många tidigare nämnda önskvärda lägen. Det hade även varit

bättre om all tid som spenderades på projektet under varje sprint noterades, istället för bara måndag till torsdag samt gruppens gemensamma tid.

Hur vi når önskvärt läge:

För att under framtida projekt uppnå önskvärt läge hade all tillgänglig tid kunnat utnyttjas till att förbättra projektet. Detta kan uppnås genom att se över vilken tid som finns tillgänglig och genom att samtliga medlemmar ständigt letar efter fler uppgifter att utföra än enbart de som allokerats eller valts vid början av sprinten. Vi hade även kunnat notera mer kontinuerligt hur mycket tid som spenderats på projektet under sprinten i början eller slutet av varje sprint, istället för att göra det under torsdagarna. På så vis hade all tid som spenderas på projektet antecknats, istället för att bara inkludera tiden från vissa dagar. Dessutom hade vi vid tillfällen där flera gruppmedlemmar närvarade kunnat räkna mantimmar, istället för gemensam tid. Detta hade givit en mer representativ bild av tiden som lagts på projektet.

4. Design decisions and product structure

4.1 Design Decisions

Nuläge:

Våra designval var en dynamisk process som förändrades mycket under projektets gång. Inledningsvis valde vi att använda oss av designmönstret *Model View Controller*, vilket skapar kundvärde genom att bidra till skapandet av en användarvänlig applikation då den delar upp logik och användargränssnitt. Vi använde oss även av *Maven* för att hantera alla API:s och plugin, som JavaFX, redan från start. Vecka 5 började vi använda oss av *Singleton*-klasser och implementerade designmönstret *Observer*, då det ser till att inte skapa onödiga beroenden mellan klasser. Det bidrar till kundvärde eftersom det blir enklare att förstå och skriva ny kod. Vecka 7 valde vi att ta bort *Observer*-mönstret från applikationen, eftersom vi upplevde, genom vår code review, att mönstret inte var lämpligt för användningsområdet. Istället implementerade vi en variant av designmönstret *Facade* för att gömma komplex logik av vyerna till en mer lättanvänd och lättläslig klass. Under vecka 8 fokuserade vi på designen på användargränssnittet och tillförde olika designrelaterade önskemål från PO. Vecka 9 valde vi att implementera designmönstret *Factory* för att kunna uppdatera vyerna på alla lägenhetstyper samtidigt utan att ladda om de igen, vilket resulterade i att gränssnittet för lägenhetstyper blev mycket snabbare och mer effektivt.

Önskvärt läge:

Våra designbeslut var väldigt dynamiska och förändrades under projektets gång, vilket tillät oss att både testa oss fram och lära oss mer genom trial-and-error. Det var positivt för projektet eftersom det tillät oss att anpassa oss och hitta den design som var mest lämplig för just vår applikation, till skillnad från att bestämma oss för en designplan i förväg och strikt följa den. Ett önskvärt läge hade däremot kunnat vara att mer noggrant planera och bearbeta olika designval och designmönster i början av projektet. Även om det var bra att låta designvalen vara dynamiska hade den initiala planen kunnat vara tydligare.

Hur vi når önskvärt läge:

Även om vi diskuterade strukturen vid projektets start så varken dokumenterade eller bearbetade vi det särskilt noga. För att nå önskvärt läge hade vi kunnat vid projektets start tydligt strukturera upp en första plan för programmets struktur, exempelvis göra ett initialt UML-diagram att utgå från. Andra tillvägagångssätt för att nå önskvärt läge hade kunnat vara att använda sig av flowcharts, klassdiagram eller textdokumentation.

4.2 Technical documentation

Nuläge:

Den tekniska dokumentationen utvecklades under de första tre veckorna. Inledningsvis använde vi oss enbart av JavaDoc. Vi ansåg att det skulle hjälpa både oss och vår PO att förstå vad vi har gjort samt underlätta för oss att senare upprätta eventuella andra dokumentationsformer. Tidigt i arbetet gjorde vi även en mockup med hjälp av verktyget Figma. Detta valet grundades i gruppmedlemmars vana vid att använda det, samt dess generella användarvänlighet. Vi använde oss genom hela projektet av Zoom och Messenger för att kommunicera med varandra, skrev koden i IntelliJ och lagrade allting på GitHub. Under varje sprint använde vi oss av Trello för att lagra information kring den aktuella sprinten och våra acceptance criterias. Här fördes även vår backlog. Vi använde oss av Google Docs för att dokumentera våra KPI:er, gruppreflektioner och anteckningar från möten. Även anteckningar från mötena med vår PO fördes i Google Docs. Dessa program valdes för den tekniska dokumentationen eftersom de är intuitiva och underlättar arbetet för samtliga gruppmedlemmar. Programmen möjliggör även fjärrarbete, vilket var nödvändigt till följd av Covid-19. Vecka 6 bestämde vi oss för att addera code review till den tekniska

dokumentationen, där vi bland annat undersöker huruvida klasserna följer diverse principer och korrekt formatering.

Önskvärt läge:

Ett önskvärt läge hade varit att använda ytterligare dokumentation för själva programmet, exempelvis upprätta ett UML-diagram och kontinuerligt uppdatera det allteftersom koden förändras. Vi hade kunnat lägga mer tid på domain model och allmän grundläggande dokumentation av hur programmet var avsett att se ut och fungera.

Hur vi når önskvärt läge:

Genom noggrann planering och granskning av olika alternativ för teknisk dokumentation hade vi tydligt kunnat avgöra vid projektets början vilka hjälpmedel som hade varit mest lämpliga. De vi använde oss av fungerade mycket bra, men ytterligare initial planering hade kunnat resultera i tillförande av exempelvis UML-diagram vilket kunde haft mycket positiva effekter.

4.3 Ensuring code quality and enforcing coding standards

Nuläge:

Vi försäkrade oss om kodens kvalitet och standard genom att parprogrammera och vid tillfälle ha större programmeringsmöten där vi satt flera och hjälptes åt. Eftersom vi använde oss av zoom kunde vi dela vår skärm och visa hur koden såg ut om vi behövde ett extra öga på något som inte fungerade. Genom att hjälpas åt fyllde vi kunskapsgap hos olika individer och försäkrade om så hög kvalitet som möjligt på koden. Javas kodstandarder upprätthölls genom projektet genom att tillämpa kunskap erhållen från tidigare kurser, och om någon gruppmedlem märkte att viss kod inte uppfyllde kodstandarder så såg den medlemmen till att ändra och förbättra koden. Vecka 6 införde vi, som tidigare nämnt, även code review vilket ytterligare bidrog till att försäkra kodkvaliteten.

Önskvärt läge:

Vårt arbetssätt fungerade bra för samtliga medlemmar och resulterade i en slutprodukt där kvaliteten är relativt hög. I ett önskvärt läge hade koden varit ännu bättre, vad gäller kvalitet och standarder.

Hur vi når önskvärt läge:

Genom att tillämpa kunskap som vi erhållit från varandra i och med detta projekt på framtida projekt kan vi förhoppningsvis både skriva och känna igen kod av högre kvalitet. Code review kan utvecklas till att gruppmedlemmen som utför den meddelar personen som skrivit koden vilka brister som finns och hur de kan åtgärdas, istället för att endast åtgärda det på egen hand. På så sätt hade samtliga gruppmedlemmar utvecklats ännu mer under projektet, vilket hade resulterat i ännu högre kvalitet i framtida projekt.

4.4 Updating documentation throughout the sprints

Nuläge:

Redan under första veckan av sprinten etablerade vi stående möten inom gruppen, samt med PO för projektet. Under de interna mötena använde vi oss av dokumentation som inkluderade de KPI:er som vi använde oss av, för att få en bättre insikt i de faktorer som kan påverka arbetet. Under möten med PO dokumenterades dennes åsikter, vilket ytterligare diskuterades under de interna mötena. Vidare etablerades även vidare dokumentation vid respektive handledningsmöte, som gav oss insikt i hur projektet fortsatt kan utvecklas.

Under projektets gång använde vi oss även av Trello för att dokumentera hur långt vi kommit i respektive user story, samt om våra acceptance criterias var nådda. Vidare användes även Javadoc och kommentarer i koden för att underlätta för gruppmedlemmar att förstå funktionaliteten av koden som skrivits.

Önskvärt läge:

JavaDoc och kommentarerna i koden gav tydlig inblick för gruppmedlemmarna att tydligt förstå tanken och funktionaliteten av respektive kodblock. Hursomhelst användes detta inte i den utsträckning som kan tänkas vara önskvärt, vilket kan ha påverkat förståelsen inom vissa delar och user stories. Vidare hade ett eventuellt användande av UML/klassdiagram skapat en kontinuerlig uppdatering av projektet, samtidigt som det hade förbättrat förståelsen av den faktiska strukturen av applikationen.

Hur vi når önskvärt läge:

Genom att tidigt etablera ett ramverk för kommentarer, JavaDoc och användning av UML/klassdiagram under projektets gång hade denna aspekt kunnat utveckla arbetet. Genom

ytterligare dokumentation och genomgång av våra KPI:er kring förståelse av koden hade det även kunnat kopplas till denna faktor, genom att granska den strategi som valts för att underlätta för medlemmar i gruppen att förstå koden. Genom att dokumentera tydligt i respektive kodblock, ihop med användandet av UML/klassdiagram, hade det bidragit till ett ökat samarbete inom gruppen där varje individ känner att de har full koll på de delar som skrivits.

5. Application of Scrum

5.1 Roles within the team

Nuläge:

Under vecka 5 fick vi under ett handledningsmöte ett tips kring att etablera roller inom gruppen för att utveckla och skapa en bättre uppdelning av arbetet. Då projektet bestod av en tydlig frontend och backend skapades roller inom gruppen för att gynna dessa aspekter. Frontend teamet arbetade till stora delar med skapandet av en välfungerande och intuitiv GUI, vilket var av stort värde för PO som ville utveckla den tidigare excelfilen till att eventuellt vara ett program som flertalet kollegor inom företaget kan nyttja. Backend teamet arbetade till stora delar med den logik och funktionalitet som programmet förväntades bistå med. Genom uppdelningen av arbetet riskerades en eventuell förlust av förståelse för helheten, vilket motverkades genom valet av en scrummaster inom gruppen. Vidare hade gruppen även en vice-PO som till stora delar var vår närmsta kontaktperson till PO utifall eventuella frågor dök upp. Vice-PO:n framhävde även åsikter från PO, då denne hade bäst insikt och kontakt med personer, och var vidare även med inom backend teamet.

Scrummastern hade koll på projektet som helhet, och att uppdatera gruppmedlemmarna och de uppdateringar som gjorts inom respektive område. Scrummastern har varit en grundpelare i projektet då denne tydligt gav information och ramverk för att fortsatt utveckla projektet i den riktning som var önskvärd. Genom en tydlig överblick och god insyn har scrummastern skapat en tydligare arbetsfölj för resterande medlemmar.

Backend teamet bestod av en “repo manager & development lead”, samt tre “developers & testers”. Dessa hade som arbete att utveckla programmets logik och funktionalitet för att möta de mål som satts upp för projektet.

Frontend teamet bestod av en “design lead” och fick ytterligare hjälp av “repo manager & development lead”:en. Frontend teamets arbete grundade sig i att skapa en intuitiv GUI som skulle vara anpassat efter kundens behov. Ytterligare skapade frontend teamet värde genom att vidare utveckla programmets design för att skapa en modern och attraktiv produkt.

Önskvärt läge:

Rollfördelningen var gynnsam för gruppen, då det tydligare gav en indikation på vilka områden som respektive medlem hade ansvar över. Denna rollfördelning hade kunnat etableras i ett tidigare stadie, då arbetet i vissa fall var skiftande för vissa medlemmar. Mot slutet av projektet blev indelningen av roller inte lika aktuellt, då frontend teamet hade fler aspekter som behövde tas hänsyn till än backend teamet. Således hade det varit gynnsamt för arbetet om rollfördelningen varit annorlunda mot slutet.

Hur vi når önskvärt läge:

Genom att tidigt etablera en tydlig rollfördelning för gruppmedlemmarna, samt skapa en timeline över förväntad arbetsbörda inom respektive grupp. Således skapas en tydligare bild av respektive grupps arbetsbörda under början och slutet av arbetet, som möjliggör ett eventuellt skifte i rollfördelningen för att anpassa gruppens omfattning efter arbetsbelastningen inom respektive team. Vidare hade det varit önskvärt att ha en bättre insikt i hur rollfördelningen etableras inom mjukvaruutveckling, för att skapa de bästa förutsättningarna för rollfördelningen att skapa värde.

5.2 Agile practices

Nuläge:

Under hela arbetets gång har vi använt oss av flertalet agila tillvägagångssätt för att utveckla och förbättra samarbetet inom projektet. Genom indelningen av projektet i epics, för att sedan skapa user stories inom respektive epic, gav oss tydliga ramar för hur projektet skulle prioriteras och vilka delar som ingår i skapandet av den produkt som vi hade för avsikt att skapa.

Kommunikationen under projektets gång har också varit ett agilt tillvägagångssätt som vi använt oss av. Vi har haft sprint-starter på måndagar, och har även utfört handledarsamtal och genomgång med PO under samma dag. Då PO kunde ge tydliga åsikter kring vad som producerats kunde vi veta om någon user story behöver förbättras från tidigare veckor. Varje torsdag hade vi även ett mid-week möte för att gå igenom eventuella svårigheter som någon user story medför, och för att få insikt i hur projektet utvecklas. Upplägget var gynnsamt för samtliga medlemmar, då man tydligt fick åsikter kring tidigare sprints från PO, samt att man under mid-week mötet kunde be om eventuell hjälp kring user stories från andra medlemmar. För att etablera tydliga kommunikationskanaler där medlemmar snabbt kunde få svar använde vi oss av zoom samt messenger. Dessa två möjliggjorde för respektive medlem att snabbt kunna ta kontakt med andra i gruppen, och således förbättra arbetsprocessen.

Vi har även använt ytterligare agila tillvägagångssätt genom en product backlog, där vi använde oss av Trello för att dokumentera respektive sprint och de tasks som de inkluderade. Backloggen användes under hela projektets gång, och gav en tydlig överblick över projektets alla delar. Vi var även öppna för att förändra tidigare utförda user stories och gå tillbaka om dessa inte passade in i projektet, och backloggen möjliggjorde att denna process kunde gå smidigt till.

Vi har även använt oss av KPI:er som omfattade stress, förståelse och produktivitet. Genom användandet av KPI:er har vi kunnat få en bättre förståelse för projektets utveckling, samt hur respektive medlem ligger till. Stress och förståelsen kring projektet gav insyn i faktorer som kan påverka produktiviteten, och gav även oss bra insyn i hur varje person mår under och inför respektive sprint.

Önskvärt läge:

Backloggen på Trello innehöll alla user stories och tasks som kopplades till respektive epic. Det gav en tydlig överblick över projektets olika ingående delar, samt hur dessa kopplas till varandra. Hursomhelst var dessa inte prioriterade i någon speciell ordning, vilket hade kunde underlättat för teamet inför respektive sprint.

Hur vi når önskvärt läge:

Genom att tidigt applicera en prioritetsordning av arbetet hade backloggen ytterligare kunnat skapa värde. Vår indelning för svårighetsgraden som respektive user story delades in i kan

användas för att ytterligare skapa en rimligt estimerad prioriteringsordning av arbetet som behöver utföras.

5.3 Sprint review

Nuläge:

Från projektstarten har vi haft genomgång med vår PO varje vecka för att få åsikter kring det arbete som har utförts under den gångna sprinten. Upplägget har gynnat vårt arbete då PO tidigt i sprinten kan kommentera de eventuella ändringar som behöver göras, och möjliggjort det för oss att snabbt kunna prioritera arbetet för att uppfylla kundens önskningar. Genom PO:s åsikter kring föregående sprint använde vi oss av backloggen för att utforma nästkommande sprint, och arbeta kring de delar som behövde utvecklas.

Mötena med PO gjorde i flertalet fall att vissa user stories fick fler tasks, eller att vissa tasks som ansågs vara klara behövde göras om. Baserat på omfattningen av de eventuella utvecklingsområdena kunde vi prioritera vårt arbete inför kommande sprint. Varje möte med PO var även del av vår DoD, då PO hade bäst koll på vad som behövde implementeras och förändras inom projektet. Under dessa möten, som utfördes via Zoom, kunde vi utnyttja remote-control funktionen vilket kunde möjliggöra för PO att direkt interagera med programmet. Således fick vi vidare insikt i dennes tankar vid direkt användning av programmet.

Önskvärt läge:

En förbättringsmöjlighet inom sprint reviewerna hade varit att ha ett stående internt möte efter respektive sprint. Där hade vi kunnat presentera det arbete som utförts under den gångna sprinten, samt vilka eventuella svårigheter som man stött på under sprintens gång. Vidare hade passet kunnat användas för att tydligt presentera de delar som man arbetat på. Genom att använda sig av ovanstående möten hade det eventuellt skapats en tydligare helhetsbild och ökad motivation kring arbetet hos medlemmarna.

Hur vi når önskvärt läge:

Genom att etablera ett stående internt möte dedikerat till presentation av respektive medlems arbete under den gångna veckan, kan en tydlig struktur skapas inom projektet.

Presentationerna kan ytterligare genomföras via zoom/på plats, där man presenterar arbetet visuellt för resterande i teamet.

5.4 Best practices for learning and using new tools and technologies

Nuläge:

Vi har under projektet använt IntelliJ som IDE, som inkluderar ett GUI för GitHub vilket underlättade för de gruppmedlemmar som var nya till att använda Git. Övriga verktyg som vi använt oss av är SceneBuilder och Trello. För att underlätta hanteringen av utseendet på användargränssnittet använde vi oss av CSS, vilket sedan kopplades samman med SceneBuilder-projektet.

Vi har som grupp varit väldigt måna om att samtliga gruppmedlemmar ska ha förståelse för och vara bekväma med de verktyg vi använt. Av den anledningen har vi till exempel varit noga med att alla använder samma IDE, så att vi får ett likformigt arbete och underlättar fyllnad av kunskapsgap. En “best practice” för att lära sig ett nytt verktyg är att ta hjälp om man fastnar istället för att lägga tid på att traggla med samma sak. Under projektets gång har de gruppmedlemmar som behövt det fått hjälp genom Zoom-möten med andra gruppmedlemmar eller genom att söka upp informationen på internet, på t.ex. StackOverflow.

Önskvärt läge:

Det är önskvärt att alla i gruppen är öppna inför nya verktyg och teknologier. I den första sprinten använde vissa gruppmedlemmar sig av Eclipse istället för IntelliJ, eftersom det var det de vara vana vid sedan tidigare. Denna skillnad i bruk av IDE skapade svårigheter vid användning av Git, eftersom kopplingen till GitHub fungerar lite olika mellan de två IDEerna. Det är alltså även önskvärt att samtliga gruppmedlemmar använder sig av samma programvara.

Hur vi når önskvärt läge:

Genom att fastslå en standardiserad programvara för samtliga gruppmedlemmar vid projektets start kan vi undvika denna typ av skillnad i framtida projekt. Då kan även förståelsen för det verktyget bli större inom hela gruppen, eftersom vi får mer tid på oss att lära oss när vi börjar tidigare.

5.5 Relation to literature and guest lectures

Nuläge:

Vi har utformat majoriteten av vårt arbetssätt utifrån föreläsningarna i början av kursen. Även övningarna var hjälpsamma för att skapa en förståelse kring hur ett agilt arbetssätt ser ut. Utöver detta har vi under kursens gång haft veckovisa handledningsmöten med vår handledare, Alderic Wong. Enligt hans förslag implementerade vi team roles och användningen av code review, vilket underlättade uppdelningen av vårt arbete och säkerställde god kodkvalitet. Handledningsmötena har även givit oss värdefull kunskap kring hur ett projektbaserat arbete av denna typ bör utföras. Code review:n utformades med stöd i EvokeTechnologies (2015) för att styrka vårt tillvägagångssätt. Slutligen har Final Team Reflection utformats utifrån Smith (2001), eftersom det rekommenderats av kursansvariga.

Önskvärt läge:

I framtida projekt hade arbetet kunnat utgå från gästföreläsningar i större utsträckning. Nu tog vi främst hjälp från ordinarie föreläsningar, viss litteratur och handledningsmöten. Det hade varit väldigt lärorikt att se hur agilt projektarbete fungerar i praktiken, till exempel i näringslivet när större program utvecklas. På så vis hade vi kunnat få praktiska tips och tankesätt som ligger utanför det akademiska. Vidare hade samtliga veckovisa team reflections kunnat utformas med hjälp av Smith (2001). Detta tillvägagångssätt hade gett tydligare riktlinjer för nästkommande sprint.

Hur vi når önskvärt läge:

Kursen har erbjudit många bra resurser, men i ett framtida arbete skulle vi kunna leta upp fler egna artiklar och föreläsningar. Ett annat alternativ skulle kunna vara att intervjua någon som programmerar professionellt och på så sätt uppnå ett bättre arbetssätt. Dessutom hade vi som tidigare nämnt kunnat använda oss av Smith (2001) i större utsträckning.

6. Referenslista

EvokedTechnologies (2015), *Code Review Checklist - To Perform Effective Code Reviews*.

Hämtat 2021-04-26 från:

<https://www.evokedtechnologies.com/blog/code-review-checklist-perform-effective-code-reviews/>

Smith, R. (2001). *Formative Evaluation and the Scholarship of Teaching and Learning*, *New Directions for Teaching and Learning*, vol. 88, s. 51-62.