

Final Reflection

Jonathan Eksberg, Jesper Larsson, Simon Torell, Jonathan Heinin,
David Lidevi, Martin Lillö, David Arvidsson

May 2020



Innehåll

1 Kundvärde och omfattning	4
1.1 Vald omfattning och prioritet av funktionalitet	4
1.1.1 Sammanfattning av projektet och aktuell status	4
1.1.2 Hur kommer det se ut nästa projekt?	4
1.1.3 Hur åstadkommer vi detta?	5
1.2 Mål med projektet och lagarbete	5
1.2.1 Sammanfattning av projektet och aktuell status	5
1.2.2 Hur kommer det se ut nästa projekt?	5
1.2.3 Hur åstadkommer vi detta?	6
1.3 User Stories, acceptanskriterium och värden	6
1.3.1 Sammanfattning av projektet och aktuell status	6
1.3.2 Hur kommer det se ut nästa projekt?	6
1.3.3 Hur åstadkommer vi detta?	7
1.4 Acceptanstest	7
1.4.1 Sammanfattning av projektet och aktuell status	7
1.4.2 Hur kommer det se ut nästa projekt?	7
1.4.3 Hur åstadkommer vi detta?	7
1.5 KPI:er och dess användning	8
1.5.1 Sammanfattning av projektet och aktuell status	8
1.5.2 Hur kommer det se ut nästa projekt?	8
1.5.3 Hur åstadkommer vi detta?	9
2 Socialt kontrakt och prestation	9
2.1 Socialt kontrakt	9
2.1.1 Sammanfattning av projektet och aktuell status	9
2.1.2 Hur kommer det se ut nästa projekt?	9
2.1.3 Hur åstadkommer vi detta?	9
2.2 Nedlagd tid på kursen och det relaterar till leverabler	9
2.2.1 Sammanfattning av projektet och aktuell status	9
2.2.2 Hur kommer det se ut nästa projekt?	10
2.2.3 Hur åstadkommer vi detta?	10
3 Design beslut och produkt struktur	10
3.1 Design beslut och kundvärde	10
3.1.1 Sammanfattning av projektet och aktuell status	10
3.1.2 Hur kommer det se ut nästa projekt?	10
3.1.3 Hur åstadkommer vi detta?	11
3.2 Vilken teknisk dokumentation använder ni och varför?	11
3.2.1 Sammanfattning av projektet och aktuell status	11
3.2.2 Hur kommer det se ut nästa projekt?	11
3.2.3 Hur åstadkommer vi detta?	12
3.3 Hur ni använder och uppdaterar eran dokumentation under sprintarna?	12
3.3.1 Sammanfattning av projektet och aktuell status	12
3.3.2 Hur kommer det se ut nästa projekt?	12
3.3.3 Hur åstadkommer vi detta?	12
3.4 Kodkvalitet och kodstandarder	12
3.4.1 Sammanfattning av projektet och aktuell status	12
3.4.2 Hur kommer det se ut nästa projekt?	13
3.4.3 Hur åstadkommer vi detta?	13
4 Användning av Scrum	13
4.1 Roller och dess påverkan	13
4.1.1 Sammanfattning av projektet och aktuell status	13
4.1.2 Hur kommer det se ut nästa projekt?	13
4.1.3 Hur åstadkommer vi detta?	14

4.2	De agila metoderna och dess påverkan	14
4.2.1	Sammanfattning av projektet och aktuell status	14
4.2.2	Hur kommer det se ut nästa projekt?	14
4.2.3	Hur åstadkommer vi detta?	15
4.3	Sprintreflektion och dess relation till omfattning och kundvärde	15
4.3.1	Sammanfattning av projektet och aktuell status	15
4.3.2	Hur kommer det se ut nästa projekt?	15
4.3.3	Hur åstadkommer vi detta?	16
4.4	Best practice för inläring vid användandet av nya verktyg	16
4.4.1	Sammanfattning av projektet och aktuell status	16
4.4.2	Hur kommer det se ut nästa projekt?	16
4.4.3	Hur åstadkommer vi detta?	16
4.5	Relation till litteratur och gästföreläsningar	17
4.5.1	Sammanfattning av projektet och aktuell status	17
4.5.2	Hur kommer det se ut nästa projekt?	17
4.5.3	Hur åstadkommer vi detta?	17

1 Kundvärde och omfattning

1.1 Vald omfattning och prioritet av funktionalitet

1.1.1 Sammanfattning av projektet och aktuell status

Vi fastslog vår MVP redan under första veckan och konstruerade en Epic User story utifrån detta via samtal med vår stakeholder. Resterande epic User Stories utformades under samma möte och alla dessa User Stories sammanfattade vad vi skulle göra i projektet.

I slutet på sprint två såg vi att några User Stories inte gav arbetet ett värde. Exempelvis togs knappar fram i UI:t som inte hade någon tillägnad funktionalitet. Vi hade alltså inte slice:at "kakan" korrekt då inget värde hade skapats.

Som ett resultat av otydliga User Stories fick vi en hel del buggar i applikationen under utvecklandet som uppdagades vid merge av flera branches. Flera andra buggar blev även till i följd av få erfarenheter i Flutter vid utvecklandet av avancerade komponenter i UI:t. Mycket experimenterande fram och tillbaka resulterade i en del buggar. Under mitten av projektet infördes även ett "mittmöte" med vår stakeholder som drev arbetet effektivare framåt eftersom vi upptäckte korrigeringar tidigare i sprinten, vilket hindrade att User Stories ej blev klara i tid.

I sprint fyra arbetade hela laget på delar som var beroende av varandra vilket ledde till en hel del buggar som behövde lösas sprinten därpå. Det resulterade i att hälften av sprintens User Stories inte kunde anses som klara och vår velocity blev inte särskilt representativ för det faktiska arbetet vi utförde.

Vi insåg inför våra sista sprintar att den epic som var lägst prioriterad av vår stakeholder inte skulle hinna utföras innan projektets slut. Därför hölls en diskussion med stakeholdern om hur vi skulle prioritera arbetet de sista två sprinterna. Det resulterade i att ny funktionalitet i "huvudappen" prioriterades över "kundversionen".

Sista sprinten spenderades mycket tid på att se till att hela appen var homogen t.ex. att knappar och vyer såg likadana ut och var användarvänliga.

1.1.2 Hur kommer det se ut nästa projekt?

Vid utförande av ett liknande projekt i framtiden kommer vi ta med oss att fastslå en MVP tidigt i processen då det gav alla i utvecklingsteamet tidigt en klar bild av hur vi skulle sätta igång.

Det kommer läggas en större vikt att skriva tydligare User Stories och se till att de täcker upp för den förväntade funktionalitet User storyn ska uppnå. Detta är för att kunna leverera det man lovat samt alltid skapa värde i varje sprint.

I framtida projekt kommer kunskapsnivån i Flutter vara högre vilket kommer resultera i bättre tidsestimeringar för varje User story. Nu kunde man bli tilldelad en User story som visade sig vara för stor. Vi kommer också börja med att estimerar alla User story i början av projektet för att lättare kunna planera kommande sprintar. Mittmötet som vi införde kommer fortsätta användas då de gav värdefull feedback så att man kan korrigera saker i tid.

Det kommer även läggas stor vikt i att se till att alla User Stories i en sprint följer INVEST-kriterierna men framför allt att de är oberoende av varandra. Detta kommer förebygga buggar som skapas när olika delar av utvecklingslaget programmerar på samma funktionalitet.

Alla komponenter som återkommer på flera ställen i applikationen kommer tydliggöras tidigt i processen så dess kod kan återanvändas genomgående. Detta ger mindre duplicerad kod samt att alla komponenter följer samma design.

1.1.3 Hur åstadkommer vi detta?

I detta projekt hade vi turen att vår stakeholder var en gruppmedlems far och därmed kunde kommunikationen med denna ske smärtfritt och mer eller mindre när som helst. Detta medförde att feedback-loopen var mycket kort och vi kunde fastslå vår MVP snabbt. I framtiden kommer troligen inte tillgängligheten till vår stakeholder vara lika stor eller frekvent. Detta kommer sätta press på att tiden spenderat med stakeholderna är effektiv och att vi verkligen ser till att förstå dennes önskingar och problem när tillfälle ges.

För att kunna utforma bättre User Stories med tydliga acceptanskriterier krävs mer tid under sprint planeringen. Vidare, att skriva bra och tydliga acceptanskriterier som är tillräckliga för att se till att en User story levererar rätt funktionalitet är något som vi tror kommer komma med tiden. Detta då erfarenhet av att skriva dem samt kontinuerliga diskussioner om dem med kollegor kommer göra att man lär sig hur de utformas optimalt och vilka potentiella missar man kan göra.

Ju mer vi utvecklar i Flutter desto bättre kommer vi bli på att veta vad vi hinner med. Vi vet redan nu hur man kunde planerat utvecklingen bättre eftersom vi har bättre koll på vad just Flutter har för begränsningar och möjligheter, till exempel vilka widgets som finns och hur de används. För att kunna ha ett möte krävs det en överenskommelse med stakeholdern tidigt i projektet samt argumentera för värdet detta möte kan skapa för båda parterna.

För att nå mer oberoende User Stories i varje sprint bör vi lägga mer tid i planeringsfasen för att se till att alla User Stories i en sprint kan utvecklas helt oberoende av varandra för att undvika buggar. Vi bör inte följa en prioriteringslista slaviskt då det är bättre att flytta fram en User story till en senare sprint så att den aktuella sprintens User Stories fortfarande är oberoende.

Vi borde ha planerat upp tidigare i processen att prioritera komponenter som är återkommande i designen tidigt i processen för att kunna återanvända mycket kod. När man delar upp User Stories i gruppen kan det vara värdefullt att hålla koll på vad andra ska utveckla under sprintarna så att man kan se om man kommer utveckla något liknande.

1.2 Mål med projektet och lagarbete

1.2.1 Sammanfattning av projektet och aktuell status

Målet som vi hade i början av projektet var att lära oss hur man arbetar agilt. Vi fick samtidigt lära oss hur man arbetar distribuerat. Vi ville lära oss hur man skriver lagom stora User Stories och tasks samt lägga upp lagom stora sprintar. Ett mål var att kunna dela upp arbetet jämnt mellan varandra och att alltid leverera det vi lovat.

Fortsättningsvis ville vi bli bättre på att utveckla i utvecklingsmiljön så att vi kunde arbeta oberoende av varandra. Vi kände inledningsvis att det inte gick att arbeta så effektivt som vi önskade då så fort vi fastnade var vi tvungna att vänta på den medlem med mycket tidigare erfarenhet i Flutter. Detta resulterade i ett ineffektivt arbetsflöde. Vi försökte åtgärda detta genom att rotera mellan vem som programmerade i alla subgrupper. Vi hoppades att detta skulle ge fler bättre kunskaper inom ramverket.

I projektet delade vi upp gruppen i mindre subgrupper som bestod av 2-3 personer. Tanken var att vi skulle dela upp arbetet under sprintarna mellan subgrupperna så att vi kunde få mer gjort varje sprint. Under projektet hade vi alltid samma subgrupper. En diskussion om att byta personer i subgrupperna uppstod men vi tog valet att inte göra det. Den mest erfarna medlemmen bytte från att vara med i en subgrupp till att arbeta självständigt samt vara en flytande hjälpare mellan subgrupperna för att effektivisera arbetet. Det slutade med att vi hade fyra subgrupper.

1.2.2 Hur kommer det se ut nästa projekt?

Till nästa projekt kommer vi bättre kunna tillämpa det agila arbetssättet. Till exempel genom att effektivt konstruera User Stories samt kunna estimerat dem på ett bra sätt. Dock anser vi oss inte

fullärda inom det agila arbetssättet utan behöver bygga upp en erfarenhet över flera projekt.

Om vi programmerar i samma utvecklingsmiljö, Flutter, i kommande projekt kommer vi att ha mer grundläggande kunskap och kommer därmed kunna arbeta mer självständigt från början.

Valet att ha samma personer i subgrupperna kan ändras till att grupperna förändras under projektets gång. Detta eftersom att alla subgrupper lärde sig något annorlunda beroende på vilken User story de fick tilldelat och kan då dela med sig av sina kunskaper, speciellt när det gäller ett nytt språk.

1.2.3 Hur åstadkommer vi detta?

Att effektivt jobba agilt och skapa lagom stora User Stories etc. är något som man lär sig genom att arbeta agilt. Alltså genom erfarenhet. I framtiden kommer vi också ha bättre koll på hur lång tid saker tar att utveckla. Därför kan mer fokus ligga på att effektivisera arbetet ännu mera.

Om vi inte arbetar i Flutter eller något annat ramverk vi är familjära med så är det viktigt att sätta sig in i det redan från början. Detta för att inte bli beroende av någon annan och effektivt kunna se till att arbetet fortskrider smidigt.

Om vi hade bytt personer hade man enklare kunnat dela kunskaper med varandra och hitta lösningar.

1.3 User Stories, acceptanskriterium och värden

1.3.1 Sammanfattning av projektet och aktuell status

Vi var tidiga med att prioritera våra User Stories beroende på stakeholderns krav. Detta var för att mer effektivt kunna leverera en MVP. Vi skrev inte estimering eller acceptanskriterier eftersom vi inte hade kunskap till hur man gjorde det. Senare använde vi tid som estimering.

Vi upptäckte under projektets gång att många av våra User Stories var tvetydiga och kunde tolkas olika om vad det faktiskt innebar att implementera dem. Det resulterade i buggar samt att viss funktionalitet uteblev vid sprintens slut. Under handledning i kursen fick vi tips att ta fram tydligare acceptanskriterium för att se till att all förväntad funktionalitet implementeras av utvecklingslaget.

För att förbättra "läsbarheten" ytterligare i våra User Stories bröts dem även ner till mindre User Stories för att lättare estimeras och veta vad som skulle göras. Vi märkte även här att använda tid som estimering inte var optimalt och vi bytte därför sätt att estimeras till Small, Medium, Large. Detta fungerade halvdant då de olika kategorierna inte sade så mycket för User Stories. Vi valde därför att i senare sprintar även använda ett poängsystem i kombination med detta.

Det framkom att våra User Stories fortfarande inte var kompletta, speciellt när det gällde storleken på dem samt deras acceptanskriterier. Vi bröt upp några User Stories och skrev tydligare acceptanskriterier t.ex. att alla knappar skulle fungera samt att de skulle vara buggfria. Vi provade även att ändra på sättet vi skrev tasks på och byggde upp de mer steg-för-steg som skulle göra att det blev lättare att sätta sig in i arbetet. Det skulle också underlätta för andra subgrupper, om de ville hjälpa till att veta vad de kunde göra.

I sprint fyra kunde vi inte undkomma att vi alla arbetar på samma funktionalitet vilket skapade tekniska konflikter. Dessa konflikter kunde dock undvikas om bättre planering hade utförts.

Vi implementerade inget nytt i sista sprinten vilket gjorde att vi inte skapade nya User Stories utom massa små tasks istället.

1.3.2 Hur kommer det se ut nästa projekt?

I framtiden kommer vi fortsätta vara tidiga med att prioritera User Stories eftersom att det funkade bra. Vi kommer också att estimeras User Stories innan vi börjar med sprintarna. Vi kommer använda oss av Planning Poker som strategi för att estimeras. Acceptanskriterier kommer också att läggas till

för varje User story.

I kommande projekt kommer det även läggas stor vikt i att definiera vilket värde som förväntas levereras varje sprint och se till att User Stories med tillhörande acceptanskriterium inte lämnar någon plats för feltolkning.

Våra User Stories kommer även att hållas i en skala motsvarande Small eller Medium då för stora User Stories i Large-skalan upplevts som svårare att förstå omfattningen av. Small, Medium och Large kommer också bara användas som en kategorisering av User Stories och ett poängsystem, så som Planning Poker, kommer användas som estimat.

Alla tasks kommer vara så enkla som möjligt och stegvis vägleda utvecklaren till vad denne ska göra och underlätta för andra utvecklare i laget att hoppa in och hjälpa till.

1.3.3 Hur åstadkommer vi detta?

Vi tar med oss värdet i att använda någon typ av effort-poäng istället för tid som estimat eftersom tid är relativt till vem som utför User storyn och dennes tidigare erfarenheter. Detta kommer också ge bättre uppfattning om hur mycket man hinner med i varje sprint.

För att skapa User Stories som inte lämnar plats för feltolkning bör utvecklingslaget tillsammans lägga mer tid i planeringsstadiet och se till att alla är med på noterna. I vår definition of done kunde vi även lägga till att fler än en person skulle kontrollera alla "färdiga" User Stories.

Vi tar med oss den erfarenhet vi fått kring estimering på ett effektivt sätt men känner att det är mer erfarenheter som gäller för att bli bättre på det samt göra rimliga tasks. Testa, misslyckas och prova igen. Dessa erfarenheter kommer vi erhålla genom att utföra liknande projekt i framtiden.

1.4 Acceptanstest

1.4.1 Sammanfattning av projektet och aktuell status

Under projektets gång utfördes en demo av applikationen för intressenten i slutet av varje sprint. Under demon fick intressenten möjlighet att poängtera om vi är på rätt spår och uppfyller de värden han är ute efter. Detta gjordes på en skala 1-5 där 5 motsvarade att allt vi gjort är funktionalitet intressenten var ute efter. Under mötet räknades även antalet korrekitioner som påpekades att det bör förändras. Att utföra den räkningen hade syftet att driva på kontakten under sprintens gång och korrigera innan sprintens slut.

Eftersom vi upptäckte att vi ofta lämnades med ett par korrekitioner efter sprint-demon infördes ett mitt-möte för att fånga upp dessa tidigare. Under dessa mittmöten kunde intressenten poängtera saker som sänkte antalet korrekitioner i sprintens slut. Exempelvis hur objektens storlek anges i appen.

1.4.2 Hur kommer det se ut nästa projekt?

Till framtida projekt är det bra att ha en fortsatt frekvent kontakt med intressenten. Detta är för att smidigare kunna leverera exakt det intressenten efterfrågar vilket skapar ett stort värde för båda parterna.

1.4.3 Hur åstadkommer vi detta?

Vi bygger upp en god relation med vår intressent för att skapa en smidig kontakt och kontinuerligt följa upp på feedbacken som ges.

1.5 KPI:er och dess användning

1.5.1 Sammanfattning av projektet och aktuell status

Vi valde tre KPI:er: "Hand of Five", "Velocity" och en kundenkät.

Anledningen till att vi valde Hand of Five som KPI var för att hålla koll på välmåendet i gruppen. Detta var för att ändra på hur vi arbetade om någon eller några inte var nöjda.

Med Hand of Five svarade alla medlemmar alltid efter varje programmeringstillfälle på olika frågor. Till en början hade vi bara en fråga som löd "Hur mår du?" vilket man svarade med en skala från 1-5. Vi märkte dock att den frågan inte gav så mycket så vi bytte istället till tre nya frågor som skulle berätta mer. Dessa var: "Hur känns arbetsbelastningen?", "Hur mycket känner jag att jag bidrar?" och "Tycker jag att det här är kul?". Dessa tre frågor skulle helt enkelt berätta mer om läget i gruppen. KPI:n fungerade bra men resultatet användes aldrig.

Till en början missuppfattade vi i gruppen hur vi skulle tillämpa "Velocity" vid utförande av projektet. Vi tänkte att vi skulle estimerar hur lång tid en User story skulle ta att utföra och sedan jämföra det med hur lång tid det faktiskt tog. Detta krävde att tidsestimeringen var tvungen att utföras efter det att uppgifterna delats ut på utvecklare i laget eftersom tiden är relativ till vem som utför det.

Vi valde därför att testa på att använda klädstorlekar istället, alltså Small, Medium eller Large för att estimerar en User story. Detta i hopp om att kunna estimerar hur många Small, Medium och Large respektive utvecklare skulle hinna med och då hela laget som helhet på en sprint. Då kan även estimeringen ske innan de tilldelats en specifik utvecklare.

Small, Medium och Large i sig blev för stora grupper för att enkelt kunna estimerar med och ett poängsystem infördes inom varje "klädstorlekkategori" och vi avslutade projektet med en korrekt användning av velocity, dock endast under sista sprinten vilket gjorde att vi inte effektivt kunde använda oss av det.

Kundenkäten användes för att se till att vår intressent var nöjd med vad vi producerat under den senaste sprinten. Frågorna som ställdes var: "Hur mycket värde anser du att veckans sprint gett?" samt "Hur många korrekationer behöver göras?". Den första fick kunden gradera från 1-5. Den andra frågan sattes beroende på hur många korrekationer kunden önskade göra nästa sprint. Anledningen till detta var för att se om vi gör rätt saker samt få en indikation på om vi förstod vår intressent korrekt när vi pratat med denne sprinten innan.

1.5.2 Hur kommer det se ut nästa projekt?

Hand of Five fungerade bra men vi använde oss inte av den som vi borde gjort. I nästa projekt kollar Hand of Five innan varje tillfälle och ändrar efter behov.

Om i ett senare projekt vi lämnas med en intressent som inte är lika nöjd som i detta projekt hade Hand of Five varit en bra tillgång för att kontrollera om det har någon korrelation med gruppens välmående.

Till nästa projekt kommer vi börja estimeringen med någon typ av abstrakt estimat, likt de effort-poäng vi använde i slutet av detta projekt. Detta skulle resultera i en mer korrekt användning av "velocity" redan från start och möjlighet att nyttja dess fördelar genomgående i projektet.

I framtida projekt kommer vi troligen behöva interagera och hantera krav från mer än en stakeholder. Detta kommer sätta krav på de frågor som vi ställer till dem samt ett generera ett nytt behov av att ställa fler och mer detaljerade frågor för att få en rättvisande bild av vad klienten önskar. Vidare kommer vi behöva balansera krav som ställs från dem för att kunna leverera en produkt de alla är nöjda med.

1.5.3 Hur åstadkommer vi detta?

Det hade varit bra att se över den innan varje sprint och leta efter någon avvikelse hos någon eller några gruppmedlemmar. Detta hade eventuellt gjort att man ändrat på arbetssättet.

Vi använder oss också från start en väl beprövad estimeringsmetod, exempelvis "Planning Poker", för att få en korrekt "velocity" att utgå från. Vi skulle även behöva ett antal sprintar med liknande arbetstid för att kunna kalibrera hur mycket man hinner med på en sprint. Med hjälp av effort-estimeringen kommer man kunna ha ett tal att utgå ifrån om en sprint skulle variera i arbetstid i förhållande till en sprint med normal längd.

Det allra viktigaste för att uppnå god kommunikation med alla de stakeholders vi kommer behöva att hantera i framtiden är att belysa vikten av god kommunikation med dem. Detta för att få dem förstå vikten av att de tar tid ur sin dag att hjälpa oss kommer hjälpa dem i längden på grund av ett smidigt projekt.

2 Socialt kontrakt och prestation

2.1 Socialt kontrakt

2.1.1 Sammanfattning av projektet och aktuell status

Innan första sprinten började tog vi fram ett socialt kontrakt vilket innefattade punkter relaterade till ambitioner och prestationer. Sedan uppdaterade vi det sociala kontraktet allt eftersom arbetet i sprintarna fortlöed. Till exempel uppdaterades kontraktet efter första sprinten med en punkt gällande kvalitetssäkring.

Det sociala kontraktet har varit vägledande genomgående i projektet och hade en väldigt positiv inverkan på arbetet, speciellt i den inledande delen av projektet. Att gemensamt skriva det första utkastet av det sociala kontraktet gjorde det möjligt att tillsammans reflektera över vilka mål vi har med projektet och hur de ska uppfyllas. Det tydliggjorde vilka förväntningar gruppmedlemmarna hade på projektet och gjorde det väldigt tydligt för alla vad som förväntades av dem.

Det var framförallt i de första sprintarna vi arbetade med att uppdatera det sociala kontraktet och genomgående upplevs att det sociala kontraktet har följts och använts på ett väldigt bra sätt.

2.1.2 Hur kommer det se ut nästa projekt?

Samtliga medlemmar upplever att det sociala kontraktet har fyllt sin funktion väldigt väl under projektets gång, och i kommande projekt kommer vi försöka arbeta med sociala kontrakt på ungefär samma sätt som vi gjorde detta projektet. Projektet har tydliggjort gruppmedlemmarna hur viktigt och värdefullt ett socialt kontrakt kan vara, vilket vi kommer bära med oss till framtida projekt.

2.1.3 Hur åstadkommer vi detta?

I många projekt har ett socialt kontrakt inte en självklar roll, till exempel hade många av gruppmedlemmarna i det här projektet inte använt sig av sociala kontrakt i tidigare arbeten. I framtida projekt måste vi därför tydligt kommunicera värdet av ett socialt kontrakt till övriga gruppmedlemmar som kanske inte arbetat med det tidigare. För att utforma ett väl fungerande socialt kontrakt kan vi dessutom till stor del utgå från det som vi använde i den här kursen, vilket är en stor fördel.

2.2 Nedlagd tid på kursen och det relaterar till leverabler

2.2.1 Sammanfattning av projektet och aktuell status

Den genomsnittliga nedlagda tiden under projektets gång har varit ungefär 20 timmar i veckan för var och en av gruppmedlemmarna. Detta har följts i så stor utsträckning som möjligt då det under

vissa sprintar var mindre tid respektive mer tid som behövdes till följd av buggfixar samt deadlines i parallella kurser. Gruppen kom fram till vid utformningen av det sociala kontraktet att alla eftersträvade en hanterbar arbetsbelastning och att ingen ska känna varken press eller förväntningar att lägga mer tid än överenskommet. Denna överenskommelse gällde under samtliga sprintar och välkomnade även för de studenter som ville lägga mer tid än 20 timmar i veckan just för att de tyckte det var roligt. Sammanfattningsvis levererade vi nästan alltid det vi lovade intressenten genom att ta hänsyn till andra kursers deadline vid planering av sprinter då vissa gruppmedlemmar inte kunde lägga samma tid.

2.2.2 Hur kommer det se ut nästa projekt?

Till nästa projekt bör hela gruppen ha en uppfattning om vilka veckor medlemmarna har deadlines i andra kurser eller är upptagna med annat för att få en bra översikt över projektets period och därmed planera utefter det. I det här projektet uppstod det inte några konflikter vad gällande ojämn arbetsfördelning då samtliga höll sig till det förväntade tiden och hade förståelse för varandras inställning till projektet. Till nästa projekt är det önskvärt att få samma gruppdynamik, samstämmighet och arbetsmiljön då det är ytterst viktigt för ett framgångsrikt projekt.

2.2.3 Hur åstadkommer vi detta?

För att åstadkomma det förstnämnda vore det bra om man arrangerade ett möte väldigt tidigt, helst innan själv mjukvaruutvecklingen drar igång, där alla får dela med sig av sina uppbokade tider under projektets period. Vidare hade detta kunna realiseras med att skapa en gemensam kalender, förslagsvis, Google Calendar där alla kan lägga in när de kan respektive inte kan för lättare kunna planera utefter det. För att få samma upplevelse gällande arbetsmiljö och gruppdynamik vid nästa projekt bygger det mycket på att initialt ha en dialog om var och ens förväntningar på kursen i termer om prestation, komplexitet och omfattning. Detta just för att åstadkomma det vi gjorde.

3 Design beslut och produkt struktur

3.1 Design beslut och kundvärde

3.1.1 Sammanfattning av projektet och aktuell status

Vi valde att använda oss av Flutter för att utveckla applikationen då det möjliggör stöd för både Android och iOS. Intressenten uttryckte egentligen inte ett behov för stöd i Android men det möjliggör en större frihet för intressenten och kan visa sig vara värdefullt i framtiden. Ytterligare en anledning till att Flutter möjliggjorde kundvärde var att en av gruppens medlemmar hade god kunskap om det sedan tidigare. Vi använde oss av ett flertal API:s och bibliotek, bland annat listener för att lyssna efter användarinput, samt färdigutvecklade funktioner för integration mot kamera, telefon, mail, scanna dokument med mera. Det möjliggjorde att vi kunde utveckla fler funktioner som intressenten efterfrågade och värdesatte, alltså bidrog det med en stor mängd kundvärde.

En designprincip som vi tillämpade var Separation of Concern, vilket ledde till ett smidigare och tydligare arbete som i slutändan kan leda till en bättre produkt för kunden. Ett problem som uppstod under vissa sprintar var att klasser blev allt för stora och svåra att överskåda. För att hantera det refaktorerade vi i efterhand koden. Refaktorisering skedde även i efterhand vid vissa tillfällen när vi upptäckte att funktioner var användbara på fler än ett ställe.

3.1.2 Hur kommer det se ut nästa projekt?

Genomgående så ansågs Flutter som ett väldigt användbart verktyg vilket möjliggjorde värdeskapande för samtliga medlemmar, vid kommande projekt för utveckling av applikationer ses alltså Flutter som ett lämpligt verktyg. Även implementeringen av bibliotek och API:s fungerade väldigt väl och ansågs effektivt, därför kommer det användas även i framtida projekt. I framtida projekt är det önskvärt att undvika att refaktorisera koden först efter att hela klasser är utvecklade.

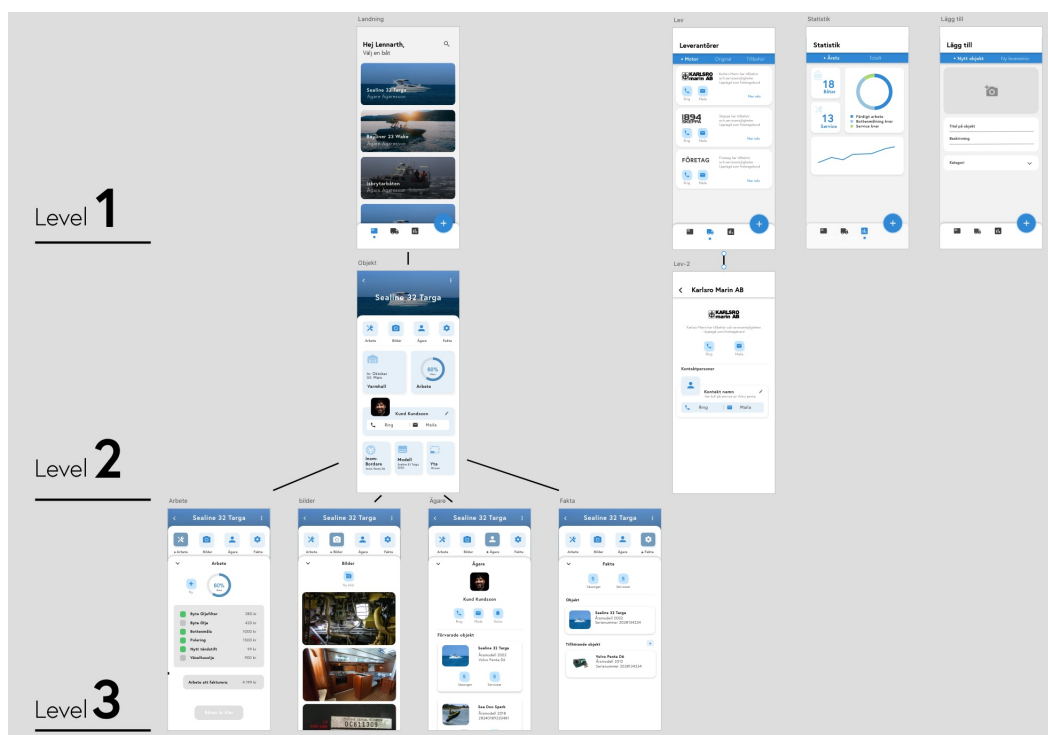
3.1.3 Hur åstadkommer vi detta?

För att slippa refaktorisera koden i efterhand krävs det bättre planering. Vid planeringen så är det lätt att förutse vilka funktioner som kommer återanvändas och då kan dessa direkt läggas som fristående kod och implementeras där de behövs. Även för stora klasser kan till viss mån undvikas med hjälp av planering. I projektet var den tekniska dokumentationen endast kopplad till utseendet av själva applikationen, alltså ingenting om hur klasser skulle vara uppbyggda och hur dessa skulle relatera till varandra. I framtida projekt kan vi använda oss av till exempel klassdiagram för att i ett tidigare skede upptäcka klasser som troligtvis blir för stora. Innan första sprinten genomförs kan man även gå igenom applikationens mockup och leta efter återkommande funktioner, så kan dessa implementeras i fristående klasser direkt, istället för i efterhand vilket ofta var fallet under det här projektet.

3.2 Vilken teknisk dokumentation använder ni och varför?

3.2.1 Sammanfattning av projektet och aktuell status

Under projektet har vi framförallt använt oss och utgått från mockups vi gjorde under de första två veckorna i kursen. Dessa mockups, se urklipp nedan, har hjälpt oss vid utformandet vid appen då man hela tiden vet vad resultatet ska vara dock har det inte gett någon teknisk beskrivning kodmässigt. Vi har använt oss av mockup för vägleda arbetet och för att säkerställa att samtliga medlemmar har en enhällig bild över hur produkten ska se ut. Vi har även använt mockupen för att kommunicera med intressenten och sett till att produkten vi utvecklar är i linje med det som han önskar. Genom mockupen blir det tydlig vad han kan förvänta sig och han kunde även komma med synpunkter kring specifika funktioner, utseende med mera. Genom att iterativt ta fram mockupen tillsammans med intressenten säkerställde vi att vi utvecklade en produkt som var värdeskapande för honom och i linje med det som han förväntade sig. Utöver mockups har vi kommenterat koden för att underlätta förståelsen i de mer komplicerade kodavsnitten. Mockupsen har även bidragit till att generera idéer för hur olika användargränssnitt kan implementeras.



3.2.2 Hur kommer det se ut nästa projekt?

Till nästa projekt hade det varit bra om vi hade fler mockups att utgå från så att man inte hamnar i den situationen de sista sprintarna som vi gjorde. Vidare skulle det vara viktigt att ha uppdaterade mockups under projektets gång i samband med att kundens specifikationer kan ändra sig. Här blir

det dock en avvägning mellan att lägga tid på att uppdatera och/eller skapa nya mockups och bara koda utan mockups. Denna avvägning är emellertid viktig att göra då olika uppfattning om hur koden ska skrivas kan påverka leverabler till följd av buggfixar.

3.2.3 Hur åstadkommer vi detta?

Den åtgärd som krävs för att uppnå detta vore att löpande uppdatera mockupsen samt våga lägga tid på att skapa fler mockups när dessa saknas. En annan åtgärd till samma problem är att ha en tydlig kommunikation genom avstämningsmöten där man dagligen checkar av för att undvika att utvecklarna får olika uppfattningar om hur appens features ska utformas. Ett annat förslag på att uppnå en gemensam uppfattning om hur appen ska se ut vore att skapa ett klassdiagram för att få en bättre helhetsbild av applikationen men här hamnar man i samma avvägning huruvida det är värt att lägga den tiden mot hur mycket det ger.

3.3 Hur ni använder och uppdaterar eran dokumentation under sprintarna?

3.3.1 Sammanfattning av projektet och aktuell status

Allt eftersom vi utvecklade produkten och sprintarna fortlöd som uppkom det i viss mån förändring i produkten och i det uttryckta behovet från intressenten. På grund av det uppdaterade vi mockupen så att den i så stor mån som möjligt var aktuell och följde det som tillkom. Dock var det mot slutet vissa förändringar som inte gjordes i mockupen, vilket gjorde att i de sista två sprintar hade vi dock inte någon mockup att utgå från vilket gjorde att utvecklingslaget fick ta egna beslut vilket kan ha lett till att buggar uppstod samt att layouten/design skiljde sig åt mellan de olika skärmarna. Sammanfattningsvis har vi kommenterat all kod för att varje subgrupp under sprintsen ska ha lättare för sig att navigera i klasser. Detta har även varit mycket användbart de sprinter då vi har samarbetat mellan olika subgrupper. Det har även fungerat som inspiration de tillfällen då en subgrupp har fastnat på en task och sedan tittat på tidigare skriven kod för att kunna fortsätta. Givetvis hade det varit önskvärt att dokumentera ännu mer i koden men där har en avvägning gjorts vilket ofta resulterar i en inte alltför utförlig kommentering av kod.

3.3.2 Hur kommer det se ut nästa projekt?

I nästa projekt kommer vi att fortsatt arbeta utefter en mockup, för att samtliga medlemmar ska ha samma bild av produkten. Samt att det förenklar och förtydligar kommunikationen ut mot intressenten. Dock är det viktigt att vid framtida projekt kontinuerligt uppdatera mockupen.

3.3.3 Hur åstadkommer vi detta?

För att se till att mockupen och övrig teknisk dokumentation uppdateras under projektets gång kan det implementeras på ett tydligare sätt i gruppens arbete. Förslagsvis kan det läggas till som en punkt på sprint review, så att den tekniska dokumentationen alltid är uppdaterad inför en ny sprint.

3.4 Kodkvalitet och kodstandarder

3.4.1 Sammanfattning av projektet och aktuell status

I den första sprinten hade vi inte tydliggjort vilken kodstandard som skulle följas och vilka principer som skulle tillämpas. För att säkerställa kodstandarden gick produktägaren tillsammans med en medlem i utvecklingslaget igenom koden efter alla User Stories var klara och "städade upp koden". Det var onödigt tidskrävande och bedömningen av koden var i viss mån godtyckligt då det inte var tydligt definierat vilken kodstandard som skulle följas. Från sprint två och framåt utvecklade vi vår definition of done och tog fram en tydlig kodstandard som skulle följas. För att minimera tiden till code review var det upp till varje enskild subgrupp inom utvecklingslaget att utveckla utefter kodstandarden, därefter var det tänkt att koden dessutom skulle genomgå en code review. De punkter som ingick var:

- Återanvända så mycket kod som möjligt.

- Kommentera metoder och funktioner där det är nödvändigt.
- Inte för stora klasser.
- Ingen duplicerad kod.
- Tydliga metodnamn.

Eftersom Flutter är nytt för de flesta av oss sökte vi oss fram till en lämplig kodstandard. Under vissa sprintar genomfördes ingen code review, då vi inte planerade med detta under sprint planing. Eftersom code review inte har genomförts i stor skala av någon annan än den som utvecklat koden, är det möjligt att olika subgrupper inom utvecklingslaget tolkat kodstandarderna på olika vis och koden överlag inte följer samma standard.

3.4.2 Hur kommer det se ut nästa projekt?

I kommande projekt är det önskvärt att arbeta strukturerat med code review. Fördelaktigt är att någon annan än den som utvecklat går igenom koden och ser till så att den följer korrekt standard. Det kan även vara önskvärt att tydligare definiera kriterierna i code review. T.ex. är "Inte för stora klasser" inte speciellt specifikt och vad som egentligen gör en klass för stor kan tolkas olika av olika personer. Code review kommer även inkluderas i sprint planning för att säkerställa att det genomförs som en självklar del av varje sprint.

3.4.3 Hur åstadkommer vi detta?

I ett liknande projekt med tydliga subgrupper så kan subgrupperna turas om att gå igenom varandras kod för att se till att koden genomgående följer samma standard. För att få tydligare kriterier så kan de i den mån det går kvantifieras, där t.ex. antalet rader kan vara en begränsning angående storlek. Genom att på ett tydligare vis inkludera code review i planeringen kommer det prioriteras högre vilket ökar sannolikheten för att det genomförs efter varje sprint.

4 Användning av Scrum

4.1 Roller och dess påverkan

4.1.1 Sammanfattning av projektet och aktuell status

I början av projektet var vi först inte säkra på att behålla rollerna genomgående, men under första sprinten bestämdes att vi skulle ha fasta roller, det vill säga Product Owner, Scrum Master samt Development Team. Vi bestämde även att vi ville programmera i tre subgrupper under projektet, där varje subgrupp har varit direkt ansvarig för en eller flera user stories.

Ju mer tiden gick växte gruppens medlemmar i sina roller och arbetet flöt på smidigare, då till exempel Product Owner kunde etablera en bättre kontakt med intressenten med tiden och Scrum Master lättare förstod sig på, samt applicerade de Scrum-principer som finns.

I projektets sista sprint hade gruppen fortfarande samma roller, vilket kan vara negativt för medlemmarna då alla inte får testa på de olika rollerna. Men för projektets skull gav detta ett gott resultat.

4.1.2 Hur kommer det se ut nästa projekt?

Statiska roller är något som vi anser är att eftersträva i nästa projekt. Detta funkade väldigt bra för vår grupp och är bra för att få en god arbetsstruktur. Önskvärt vore dock om Product Owner och Scrum Master skulle ha stenkoll på vad som gäller för att driva arbetet effektivt och kvalitetssäkrat framåt vilket de inte har erfarenheterna för ännu. Utvecklingslaget skulle även kunna bidra till mer kvalitativ utveckling om alla kan Scrum "utantill".

4.1.3 Hur åstadkommer vi detta?

För att åstadkomma detta till nästa projekt krävs erfarenheter, testa, misslyckas och reflektera mera samt en öppenhet till Scrums principer för alla gruppens medlemmar.

4.2 De agila metoderna och dess påverkan

4.2.1 Sammanfattning av projektet och aktuell status

I sprint ett implementerade bland annat ett Scrum Board med en Product Backlog vilken möjliggör en tydlig gemensam bild av visionen för produkten. Före sprinten planerade vi att använda oss av Stand-up Daily Meetings med hjälp av ett gemensamt excelark, men under sprinten fylldes arket inte i av någon lagmedlem. Under sprintreflektionen diskuterades arkets potentiella värde och inför kommande sprints planeras arket att fyllas i samband med genomfört arbete.

I sprint två fortsatte vi med användandet av Scrum Board som vi tyckte gav en bra översikt av arbetet och vi såg även hur långt resterande gruppmedlemmar har kommit och kan då bidra med hjälp om det behövs. Förra sprinten var vi dåliga på att använda våra KPI:er (Kundenkät, Tidsestimering, Hand of Five) vilket vi har blivit mycket bättre på den här sprinten. Användandet av tidsestimering hjälpte med planeringen då vi visste ungefär hur många timmar vi kunde lägga ner på arbetet. Denna sprinten lade vi även till frågor till Hand of Five som berättar mer detaljerat om hur det går på olika områden. Hand of Five var mindre givande för hela gruppen men mer givande för varje individ då man själv kunde reflektera över hur det gick för en själv.

Sprinten därpå märkte vi att vi behövde bli bättre på att flytta tasks på vårt Scrum Board så fort de ska flyttas. Detta för att underlätta för resten av projektgruppen att se vart de behövs fler resurser att hinna med i sprinten.

I sprint fyra syntes en förbättring med avseende på uppdatering/förflyttning av tasks. Det medförde att andra medlemmar i Development team samt Product Owner hade bättre koll på statusen av sprinten.

För att slutföra projektet i sprint fem behövdes en del fixar göras. Denna sprinten använde vi oss inte av vårt Scrum Board som i tidigare sprinter, utan istället användes en lista på fixar. Detta kändes smidigare då det var många smådetaljer som behövde lösas.

4.2.2 Hur kommer det se ut nästa projekt?

Inför nästa projekt hade vi direkt börjat med att estimerar User Stories med poäng istället för i tid. Detta eftersom tid är relativt från person till person.

KPI:n Hand of Five kände vi inte var särskilt givande för gruppen då den sällan användes. Inför nästa projekt vill vi göra Hand of Five mer användbart genom att till exempel studera om det finns en korrelation med för hög arbetsbelastning och mindre upplevt kundvärde och vice versa. Ett annat sätt att göra det mer användbart vore om intressenten fick mer inblick i hur Hand of Five skiljde sig åt under sprintarna eftersom det kan den kan då ha större förståelse för hur utvecklingslaget upplever projektet.

Alternativt kommer vi i nästa projekt att hitta någon annan, mer givande, KPI.

Ett problem som vi stötte på under projektets gång var att Scrum Board inte var uppdaterat. Inför nästa projekt behöver vi bli bättre på att direkt uppdatera vårt Scrum Board.

I sprint fem valde vi att ha våra tasks i Google Docs i stället för i vårt Scrum Board. Att inte ha allt arbete samlat på samma ställe kan i slutändan ge en sämre överblick över projektet, vilket kan göra att saker missas. Dock hade vi nog, i en liknande situation, gjort likadant i nästa projekt just för att de var dessa små justeringar som det gällde.

4.2.3 Hur åstadkommer vi detta?

Tid var ett bra sätt att börja estimeringen av User Stories med enligt handledare, men det var först i sprint fyra vi började med "riktig" estimering. Genom erfarenheten vi samlat på oss kan vi helt enkelt direkt börja med poängestimering i nästa projekt.

För att göra Hand of Five mer användbart kan vi införa ett mittmöte som ska gå ut på att gemensamt kolla på Hand of Five kalkylarket för att se om det behöver göras omstruktureringar för att förhindra/förebygga risker i form av förseningar och så vidare.

För att bibehålla ett uppdaterat Scrum Board ska vi till exempel direkt flytta en task till "In Progress"-kolumnen då tasken påbörjats. Genom att i projektets tidiga stadie ge gruppen information om att uppdatera Scrum Board, kan det underlätta arbetet mera med en tydlig överblick.

4.3 Sprintreflektion och dess relation till omfattning och kundvärde

4.3.1 Sammanfattning av projektet och aktuell status

I sprintreflektion ett pratade vi om att samtliga User Stories i Sprint-Backloggen blev klara och överlag var konsensus i gruppen att arbetet fungerat väl. Förbättringar till framtida sprinter inkluderade att i sprintplaneringen identifiera eventuella gemensamma komponenter mellan subgrupperna för att undvika att flera grupper arbetar på samma problem och då öka återanvändningen av kod. Även dokumentation av koden löpande under arbetets gång lyftes som ett förbättringsområde för kommande sprinter. Vi hade vid reflektionens tillfälle till stor del uppfyllt Definition of Done. Vi missade dock att kommentera all kod vilket i sig inte påverkade arbetet just då, utan snarare det faktum att vi kom överens om att det skulle göras.

Sprintreflektion två samt kundenkäten fick oss att prioritera annorlunda för att skapa mest värde för kunden. Product Owner fick se vad alla hade gjort och relaterade det till vad kunden ville ha. Den feedback vi fick hjälpte oss att planera framtida sprinter bättre. Vi bestämde även att hålla ett demo-möte för intressenten en gång i mitten av sprinten.

I sprint två beslutade vi att hålla ett mitt-möte. Detta gjorde att vi i sprint tre fångade upp eventuella missförstånd innan sprintens slut och förebyggde att behöva flytta över uppgifter till nästa sprint.

I sprint fyra infördes en text-kanal där utvecklingslaget snabbt kan ställa frågor som Product Owner tar med sig till intressenten. I den mån det användes fungerade det väldigt bra och resulterade i en ny definition av hur säsonger anges. Under sprintreflektionen diskuterades att i samband med att de olika subgrupperna mergade sin kod, så uppkom det även denna sprint buggar. Det hade skett två sprintar i rad och medför att User Stories som vi trodde var klara ej var det, och behövde inkluderas i nästa sprint. Vid denna tid hade varje subgrupp en egen branch, vilket kunde vara onödigt.

Sprintreflektion fem gick relativt snabbt i förhållande till tidigare. Detta för att gruppen ansåg att sprinten flutit på bra och att mindre total tid har lagts på sprint fem. Det var inga större förändringar som vi gick igenom, utan vi diskuterade att de fixar vi hade var lösta och att det fortfarande fanns något kvar. I sprint fem kodade alla subgrupper i samma branch vilket fungerade bra eftersom vi samlat alla justeringar till den klass de tillhörde.

4.3.2 Hur kommer det se ut nästa projekt?

Att ha en tät kommunikation med intressenten är något som vi tar med oss till nästa projekt. Det var just det som var idén med ett mitt-möte, och den feedback vi fick därifrån vägledde arbetet i rätt riktning och såg till att minimera risken för missförstånd mellan intressent och projektgrupp.

Textkanalen där utvecklingslaget snabbt kunde ställa frågor till intressenten var ett bra sätt att samla tankar, detaljfrågor och generella frågor som kom upp när man kodade. Detta hade vi börjat med direkt i nästa projekt.

Inför nästa projekt hade vi undvikit att skapa onödigt många branches. Detta för att undvika buggar som uppstod vid merge samt för att underlätta vid återanvändning av kod.

4.3.3 Hur åstadkommer vi detta?

För att åstadkomma detta behöver vi vara tydliga med intressenten att en regelbunden kontakt är essentiell för skapandet av en bra produkt. Därefter är det enkelt att införa ett mittmöte samt skapa textkanalen med generella frågor till intressenten.

För att kunna arbeta i samma branch behöver våra User Stories vara självständiga och oberoende av varandra. Om vi lyckas med detta kan man enkelt arbeta i samma branch och dra nytta av dess fördelar, till exempel återanvändning av kod. Dessutom i ett sådant läge behövs det en tydlig kommunikation i form av frekventa commits med tydlig meddelande, så att alla medlemmar är införstådda i vad som arbetas på.

4.4 Best practice för inläring vid användandet av nya verktyg

4.4.1 Sammanfattning av projektet och aktuell status

Utvecklingsmiljön i form av Visual Studio Code, samt användningen av Flutter och dart var i början helt ny för de flesta lagmedlemmar. Med bakgrund av det hade vi en minisprint före sprint ett som bestod av tasks, där varje medlem skulle få igång utvecklingsmiljön samt följa tutorials som beskrev grunderna i framförallt Flutter. En av lagmedlemmarna har tidigare erfarenhet från verktygen och var till stor hjälp för övriga medlemmars inläring. Parprogrammering använde vi i subgrupperna då dessa bestod av en IT-student och en I-student där IT-studenten skärmdelade och kodade. Vad gäller versionshanteringen använder vi oss av Gitkraken, där vissa medlemmar kunde det sedan tidigare och lärde därför ut hur det fungerar till resten av gruppen.

Under sprint två fick vissa subgrupper svårare User Stories och andra lite lättare, vilket ledde till att de som fick de lätta inte utvecklades lika mycket. Utöver detta flöt arbetet på bättre då alla hade fått mer erfarenhet av utvecklingsmiljön. Vi fann även ett nytt sätt att koda tillsammans, detta genom Zoom där man via skärmdelning kan ta över den andra personens dator och t.ex. skriva kod.

Resterande sprintar fortsatte vi att använda oss av parprogrammering. Inga nya verktyg introducerades och alla hade bra koll på utvecklingsmiljön, versionshanteringen samt Scrum Board.

4.4.2 Hur kommer det se ut nästa projekt?

Om det vid nästa projekt skulle bli aktuellt med att lära sig nya verktyg såsom utvecklingsmiljö eller språk, kommer vi även då innan arbetet startar, att lägga tid på inläring via tutorials med mera. Detta för att när arbetet väl drar igång, ska vi vara beredda på att kunna leverera ett gott resultat för första sprinten.

En idé som kom upp sent under projektets gång var att det kunde ha varit en bra att byta subgrupper som parprogrammerar. Detta hade ökat inläringen då alla subgrupper fick olika perspektiv.

Skärmdelningsfunktionen på Zoom var till stor nytta, därför hade vi nästa projekt direkt börjat med att använda den i stället för att bara prata och skärmdela genom Discord.

4.4.3 Hur åstadkommer vi detta?

Med tydlighet i början av projektet angående icke statiska subgrupper, kan man se till att ett skifte sker regelbundet. Till exempel skulle man kunna vid början av varje sprint byta subgrupper. Detta är något som kan stå med i det sociala kontraktet för att det ska finnas konsensus om det i gruppen. Resterande ändringar vi hade velat göra till nästa projekt är lätta att bara införa.

4.5 Relation till litteratur och gästföreläsningar

4.5.1 Sammanfattning av projektet och aktuell status

Under sprint ett och två följde vi ingen litteratur eller baserade något på gästföreläsningar. Det underlag vi utgick ifrån var mer det material som kursansvariga la fram under sina föreläsningar.

Vi började med att estimerar user stories med tid, men efter rekommendation från handledningstillfällen, valde vi att använda något annat typ av system för att estimerar velocity. Vi provade då i sprint tre med att estimerar med "Small", "Medium" och "Large". Under handledningen lyftes det att detta egentligen endast borde användas i produkt backloggen och mer specifika estimeringar för sprint backloggen. Då vi redan hade påbörjat en sprint när förslaget lyfts väntade vi med att tillämpa det.

Under sprint fyra började vi att använda oss av Planning Poker som effort-poäng. Det fungerade bra, men fler iterationer behövdes för att vi skulle kunna använda det på bästa sätt.

4.5.2 Hur kommer det se ut nästa projekt?

Vi kommer i nästa projekt att försätta ta kunskap från litteratur och andra källor för att lösa diverse dilemman.

4.5.3 Hur åstadkommer vi detta?

Detta åstadkommer vi genom att aktivt söka information direkt då krux dyker upp.