

Customer Value and Scope

The chosen scope of the application under development including the priority of features and for whom you are creating value.

- A. Applikationen går ut på att skapa ett stöd för inköp åt funktionstödsavdelningen för Göteborgs stad. Stödet är i form av en hemsida och används vid beställningar av inköp till externa leverantörer (som IKEA eller Mio). Denna process görs just nu helt manuellt på papper och det blir oftast fel och är mycket omständlig.

Klienten var från början tydlig med följande kriterier

- Enkel design som är lättförståelig för alla användare
- Hjälpmedel för att minska att fel produkter eller att fel produktinformation fylls i.
- Möjligheten att skicka PDF direkt till inköpsavdelningen

För att uppnå dessa krav försökte vi jobba med hjälp av Scrums-ramverkets horisontella struktur. Som användare krävs främst att man kan välja ut produkter från leverantörer och dess antal, skriva in nödvändig information (för beställare och leverantör). En PDF med nödvändig information skapades för att sedan skickas till inköpsavdelningen. Viktigt är att kunna skapa PDF:er från enskilda leverantörer då det är det formatet som önskades. För att hantera vilka produkter som var tillgängliga läses csv-filer in med upphandlade produkter, dessa filer säkerställer att endast godkända produkter kan väljas. Csv-filerna som tillhandahålls från leverantörerna säkerställer även att rätt produktinformation samt pris matas in.

- B. Ytterligare funktionalitet som skulle skapat värde för kunden skulle kunna vara
- automatiskt sparande av kontaktinformation alternativt rena användare där relevant information sparas och statistik över köp kan göras
 - Användning av databaser som möjliggör adminfunktionalitet

Att automatiskt spara kontaktinformationen som fylls i till beställningen emellan genereringen av PDF för att minska upprepning av input. Denna funktionalitet skulle även kunna öka säkerheten för felinformation. Genom att implementera användare kan all denna information sparas samt relevant statistik som kan användas för exempelvis vilka produkter som beställs oftast. En databas skulle kunna implementeras för att spara de inlästa listorna för tillgängliga varor för beställning. Detta skulle kunna medföra en typ av admin funktionalitet, där endast vissa personer (chefer/de som får listorna) kan lägga till leverantörer och dess tillgängliga produkter.

A → B

För att vi skulle ha hunnit med de ytterligare funktionaliteter som nämnts i B skulle vi bland annat ha, för det första utmanat oss mer i varje sprint (något vi diskuterar i KPI punkten), för det andra mer effektiv planering inför varje sprint (något vi diskuterar i nästa punkt).

Gruppens kontakt med stakeholders har varit kontinuerlig och stakeholdern är nöjd med resultatet. Men samtidigt kunde detta utnyttjas mer vilket troligtvis hade lett till att fler funktioner som skapat värde för stakeholdern önskats.

Sedan var det nya tekniker som gruppen i helhet var ovana vid, vilket gjorde att utvecklingsprocessen tog längre tid. Speciellt i början var detta ett problem som gjorde att alla user stories tog längre tid än planerat. Detta medför att framtida projekt inte kommer kräva samma startsträcka och på så sätt kunna tillföra en större mängd värde till stakeholdern.

The success criteria for the team in terms of what you want to achieve within the project (this can include the application, but also your learning outcomes, your teamwork, or your effort)

- A. Våra kriterier för att vårt projekt är lyckat i form av läromål är att i stort sett lära oss att arbeta enligt Scrum. Vi vill mer specifikt lära oss om hur man gör korrekta och horisontella user stories, något som var ett stort problem i början. Våra user stories är bättre i alla "invest" kriterier, men har förbättringspotential i nästan alla delar. Exempelvis så är user stories inte alltid så breda att vi behöver prata emellan user stories för att lösa dem, men i vissa fall så blir man tvungen eftersom vi inte planerat tillräckligt bra att kunna lösa den helt ensam.

Rent tekniskt ville gruppen även utveckla sina kunskaper inom programmeringsspråk, detta avsåg både möjligheten till att lära sig nya samt fördjupa sig i redan bekanta språk.

När det kommer till gruppdynamik så har det mestadels gått bra, men vi har haft lite problem. Vi hade ett socialt kontrakt, men som inte alltid fullföljdes. Detta kontrakt var dock något vi inte hade jobbat aktivt med, så när något problem dök upp så hade vi inte något tydligt sätt att hantera det. Ansträngningen eller engagemanget tycker vi har varit på en bra nivå, där alla har gjort sin del av arbete.

- B. Vi hade önskat att bli bättre på user stories, där vår kunskap av dom är så pass bra att vi effektivt och utan mycket krångel kan planera sprinten. Detta har tagit mycket tid under planeringsfasen som i framtiden kan läggas på att skapa värde för kunden.

När det gäller gruppdynamiken skulle vi kunna vara tydligare utifrån vårt social contract (vilka regler som gäller osv) och använda detta på ett mer utförligt sätt när problematik uppstår. Detta möjliggör att gruppen tillsammans arbeta mot ett gemensamt mål effektivare.

A→B

För att bli bättre på user stories hade vi behövt jobba mer med scrum processen inom gruppen och generellt lära oss mer om det. Detta är något som kommer med erfarenhet vilket gör att vi kommer behöva jobba med ramverket även i framtiden för att utveckla förmågan.

Sedan hade vi velat vara tydligare i planeringsfasen av varje sprint, så vi specificerar otydliga delar mer och diskuterar hur alla user stories interagerar med varandra efteråt. Detta medför effektivare jobb under själva sprinten då vi inte behöver lösa dessa typer av problem halvvägs genom user stories.

För gruppdynamiken hade vi velat jobba mer med det sociala kontraktet istället för att se genom fingrarna och strunta i det när något känns fel. Att se genom fingrarna är en kortsiktig lösning som varken gagnar gruppen långsiktigt eller personen/personerna som valt att avvika från kontraktet.

Your user stories in terms of using a standard pattern, acceptance criteria, task breakdown and effort estimation and how this influenced the way you worked and created value.

- A. Under projektets gång har user stories utvecklats i takt med att vi blivit bättre på att applicera Scrums ramverk. User storiesarna är relativt oberoende, men det finns (som sagt) fortfarande problem i planeringsfasen där vi inte bestämt alla delar och hur de jobbar ihop. Detta gör att vissa user stories inte blir helt oberoende. Vi har även inte jobbat hårt med att göra dem negotiable, där vi ibland har gjort user stories som tydligt beskriver en intention utan implementation, men det har mestadels uppstått av slump. Att göra varje user story värdefull är något som gruppen jobbar mycket med, eftersom att ramverket från början var nytt för hela gruppen har detta utvecklats med tiden. Dock har detta inte alltid skett på grund av att metodiken är ny, ett exempel är en user story som ville refaktorisera kod. Detta ger inget tydligt värde till kund, och är egentligen något som skulle gjorts i samband med något annat som ger värde. När det kommer till uppskattning av user stories så är vi relativt nöjda. Till en början var det svårt, men allt eftersom så visste vi hur mycket vi klarade av och gjorde user stories som vi ganska enkelt kunde uppskatta. Att göra user stories små lyckades vi fort med, och nästan varje har varit tillräckligt liten för att klaras av i en sprint.

Testning har inte varit en hög prioritering, vilket kan ha lett till några buggar. Vi har märkt att olika sprints user stories ibland kan ha påverkat varandra negativt som, om vi hade haft tester, kunde ha undvikits.

När det kommer till tasks i user stories så var detta något vi hade problem med i början. De var väldigt breda, ospecifika och omätbara. Just nu är dom bättre, men ibland skriver medlemmar som inte är tillräckligt insatta i området som user storyn behandlar, vilket resulterar i att tasks inte kan bli specifika nog.

- B.** Vi hade önskat att våra user stories blev mer oberoende, mer övergripande gällande implementation (men med specifika tasks), fortsatt värdefulla (utan user stories som t.ex “refactor”), fortsatt uppskattningsbara, fortsatt små. Eftersom vi inte hade några särskilda krav på testning så var de tillräckligt testbara nu, men om mer testning implementeras så ska de även bli mer testbara. Dock så hade vi önskat en högre kvalitet på lösningarna på user stories. Därför hade vi önskat mer test krav. Tasks önskas även bli mer specifika, små och mätbara.

A→B

För att skriva mer oberoende user stories bör vi vara mer effektiva alternativt lägga mer tid på planeringen av sprinten. Detta hade gjort att vi kunde definiera tydligare gränser och tasks så att vi hade märkt och förbättrat dom ifall överlapp hittades. Sedan borde alla vara med på vilka tasks samt vad de beskrev, särskilt för user stories där man har varit delaktig i det relevanta området. På detta vis hade det snabbare kommit fram till mer specifika och relevanta lösningar på problem. Vi hade även fokuserat på att göra övergripande user stories, något vi inte funderade mycket på innan, samt undvika vertikala slices (som exempelvis refactor eller design user stories).

Testning är något som vi även hade velat implementera för att stärka kvalitén på vår user stories. Eftersom vi hade låga krav på testning (t.ex unit tester) kan våra user stories skapa problem i senare sprints, om man inte upptäcker någon bug. Alltså kan de tillåta en lösning som inte är tillräckligt komplett, något vi skulle vilja undvika med hjälp av ett ökat test krav.

Your acceptance tests, such as how they were performed, with whom, and which value they provided for you and the other stakeholders.

- A.** De enda test som krävs för acceptering är att kompilera koden samt användartest (att klicka runt o se att produkten fungerar som den ska). Sedan hade vi några enstaka unit tests, men dessa var inget krav. Detta gav det minsta behövliga värde, att produkten fungerar.

För att ens kod ska accepteras in i main branschen så måste en gruppmedlem som

inte jobbat på user storyn acceptera en pull-request. På detta sätt får personen mer förståelse för koden, samt ser de koden med ett nytt perspektiv och kan upptäcka fel.

- B.** Eftersom de krav vi haft endast gav de minst behövliga värdet så hade mer testning gett värde till både oss och kunden eftersom buggar och andra fel hade upptäckts tidigare och inte lett till felaktiga lösningar längre fram. Det hade alltså i långa loppet kunnat göra utvecklingen snabbare, men mestadels säkrare, så att allt fungerar som förväntat.

A→B

För att nå målet kan vi använda oss av exempelvis unit tester då den ger en bra överblick av vilka delar av koden anses vara säker, och vad som behöver undersökas mer. Dessa skulle kunna köras som en del av DoD så man säkert vet att koden som lanseras till kunden håller måttet.

The three KPIs you use for monitoring your progress and how you use them to improve your process.

- A.** Vi har inte följt upp våra KPIer på ett bra sätt, vi gjorde det endast de två första veckorna fullständigt. Vi har inte testat mycket av koden, vilket gör en av våra KPIer (code coverage) oanvändbar. När det kommer till vår andra KPI (antal commits) så går det hitta förbättringar. Vi har bland annat få commits som är av den större storleken, något vi ville undvika. Sedan har vi lite ojämnt antal commits per vecka, där den sista sprinten hade väldigt många commits. När det kommer till nästa KPI, velocity (predicted och actual), så har vi i princip alltid klarat av alla user stories som vi förväntade oss, förutom de första. Detta kan dock betyda att vi har siktat för lågt i vissa fall, och att vi kunde klara att arbeta mer.
- B.** Vi vill framöver ha KPIer som skulle kunna användas bättre och beskriver oss bättre. Code coverage hade varit en bra KPI om vi testade mer, sedan kan den även ha varit lite för självklar eftersom man ofta vill ha 100% innan man skickar till kunden. Att kolla sin predicted och actual velocity är en KPI som vi fortfarande känner ger värde till processen då den kan beskriva om man kan ge mer eller om man siktar för högt. Vår KPI som beskrev antal commits skulle vi säga var mindre användbar då man måste ha mycket annan information för att tydligt tyda om det är ett bra eller dåligt värde (exempelvis om en vecka är halv vecka). Därför hade vi velat byta ut den mot en mer värdefull KPI. Sedan skulle vi velat reflektera över våra KPIer och använda oss av dem mer, så att vi under processen kunde förbättra oss. Exempelvis när det kommer till velocity så

hade vi velat fundera mer över om det faktiskt fanns förbättringspotential i vårt arbetssätt, vilket vi i efterhand såg att det fanns, även om siffrorna såg bra ut.

Även KPIer över “mjuka” värden hade varit relevanta att använda sig av. Exempelvis skulle KPIer över upplevd stress eller ansträngning kunnat användas. Dessa hade medfört en annan typ av värde som hade kunnat beskriva gruppens arbetsbelastning och hur fördelningen i gruppen sett ut över sprinten.

A→B

För det första ska vi använda oss av KPIerna i större utsträckning genom hela projektet. Detta innebär att reflektera över dem, möjligtvis varje vecka, för att sedan faktiskt använda oss av informationen vi får och då förbättra oss i det området. Sedan skulle vi bytt KPIn antal commits, och kanske code coverage, mot mer representativa och givande KPIer. Exempel på nya hade varit “flow efficiency” för att effektivisera den aktiva tiden som spenderas på arbete, eller “code churn” för att se hur stabil koden vi skriver är. Att reflektera över dem hur lätta de är att få fram och hur lättlästa de är också relevant. KPIer som är svåra att ta fram och förstå ger inte någon nytta och ökar sannolikheten att man slutar använda dem helt.

Social Contract and Effort

Socialt kontrakt

- A.** Vår arbete för en väl fungerande gruppdynamik inleddes med att skapa ett socialt kontrakt varpå gruppen listade önskade kategorier, hur de skulle hanteras samt vad som skulle ske vid en eventuell misskötsel av de uppsatta reglerna. Dokumentet signerades av samtliga medlemmar som ett signum för att man läst och godkänt samtliga punkter av kontraktet. I kontraktet ingick hur vi tänkt hantera möten, arbetsfördelning, beslutstaganden, dokument- och filhantering. Vi behöll samma gruppkontrakt under hela arbetsprocessen och följde hur vi skrev att vi skulle hantera olika områden relativt väl. Vi skrev under det sociala kontraktet i projektets början vilket dock ledde till att vi inte kunde förutspå exakt hur vi skulle hantera de olika punkterna i kontraktet. Efter ett par sprints kändes några punkter lite utdaterade. Under kolumnen “Åtgärd vid misskötsel” hade vi valt att hantera misskötsel på samma sätt för alla olika områden, vilket inte riktigt var optimalt då misskötsel inom olika områden inte alltid börs hanteras på samma sätt.
- B.** I framtiden hade det varit bra att lägga ned mer tid på kontraktet eftersom det är något som börs följa under hela projektets tid. Hur områden skulle hanteras borde

innehållit mer omfattande information som även tar hänsyn till specifika fall som t.ex. om en gruppmedlem är sjuk och inte har möjlighet att arbeta lika mycket som alla andra, räknas det då som misskötsel? Eller vad som skulle hända om under ett beslutstagande rösterna blev splittrade 50/50? Åtgärden för misskötsel borde också vara mer specifik beroende på vilket område det handlar om. Det är inte alltid det räcker att uppmärksamma den berörda om problemet fortsätter att ske.

A→B

Under projektet såg vi det sociala kontraktet som något som var spikat i sten, vilket inte var den bästa synvinkeln. För att förbättra grupp dynamiken hade det varit bättre att se det sociala kontraktet ur en mer dynamisk lens. Ett förslag är att under varje sprint retrospective även gå igenom och utvärdera hur väl det sociala kontraktet följts, samt om eller hur väl de olika punkterna håller. Detta hade gett möjligheten och uppmuntrat att mer aktivt komma med förslag om ändringar/tillägg till kontraktet. På detta sätt bör det vara mycket lättare att ha ett kontrakt som håller sig relevant under hela projektets gång istället för ett som förlora sin relevans allt efter tiden går.

Utvärdering av upplevd arbetsprocess

- A.** Våra största indikationer på hur gruppen i helhet och gruppmedlemmar som individer upplevt arbetsprocessen var våra veckoliga grupp reflektioner och individuella reflektioner. Grupp reflektionen gav oss en överblick om hur arbetet för sprinten hade gått, men det är osäkert ifall den sanningsenligt återspeglade hur alla i gruppen upplevde arbetsprocessen eller inte. De individuella reflektionerna gav en viss bild på hur man som individ upplevt arbetsprocessen, medan de var publikt tillgängliga för alla i gruppen var det inget som var uppmuntrat för andra bortsett från respektive skribent att läsa.
- B.** Utvärderingen av upplevd arbetsprocess fungerade väl i detta projekt arbetet, men det finns alltid utrymme för förbättringar till framtida projekt. Även fast vi hade en översiktlig bild på hur processen kändes för gruppen tack vare gruppreflektionerna, var det svårt att noggrant veta hur alla upplevt det. I framtida projekt hade det varit bra att lägga mer vikt på individuella åsikter och sedan sammanfatta dessa för att ge en klarare bild. Här kan exempelvis KPIer över stress och ansträngning vara till hjälp för att överskådligt kunna se hur arbetsprocessen upplevts av gruppen

A→B

För att få en klarare bild på hur alla i gruppen upplevde arbetsprocessen hade vi kunnat utnyttja kursens [veckoliga utvärdering](#). Den hade gett oss möjlighet att sammanställa individuella åsikter anonymt och sedan granska resultatet. Då får alla en chans att uttrycka hur de upplevt arbetsprocessen utan att oroa sig om vad andra kommer att tänka. Utvärderingen hade även gett oss möjlighet att jämföra hur väl vi ligger till i förhållande med andra grupper, vilket kunde vara en bra indikator på hur väl det fungerar i vår grupp.

KPIerna som nämns under **B** hade hjälp till och kunnat ge indikatorer samtidigt som dessa är väldigt enkla att genomföra vilket ökar sannolikheten att de genomförs.

Prestation

- A.** Vi var tidigt eniga om att försöka fördela arbetet jämnt över samtliga gruppmedlemmar, men samtidigt lämna bra med utrymme för att ha överseende vid de tillfällen när man inte hade möjlighet att bidra lika mycket under en viss sprint. Detta har skötts relativt väl av samtliga i gruppen. Vi anammade konceptet med att låta var och en välja själv vilken user story man kände passade bäst för en själv sett till ens kunskaper och tid. Detta gjorde att mer eller mindre samtliga i gruppen lyckades leverera förväntade resultat och presentera dessa under vår sprint retrospective som vi använde oss av under fredagarna. Vi stötte aldrig på problemet att vi inte lyckades bli färdiga med en sprint i tid, vilket är en direkt konsekvens av ett väl fördelat arbete. Vi har fått en liten kvantitativ insikt vad det gäller varje gruppmedlems kodning i projektet genom 'Git Insights'. Vi har dock varit väl införstådda med att det inte enbart är kvantitet av kod som förtäljer prestation utan vi har sett till olika aspekter såsom hur aktiv man är under möten, vilka idéer man kommer med till projektet, hur hjälpsam man är mot resterande gruppmedlemmar och dylikt.
- B.** Att dokumentera är något som alltid kan bli bättre. Vi borde använt oss mer av tillgängliga verktyg såsom KPI dokumentation och uppföljning. "mjukare" KPI:er hade även gjort det möjligt att mäta gruppmedlemmarnas prestationer utan att direkt koppla det till mängd kod. Båda dessa förslag i kombination hade gett en bredare förståelse för hur varje enskilda person i gruppen presterat.

A→B

För att förbättra våra prestationer och för att få bättre översikt kring prestationer hade vi behövt använda oss av våra KPI:er betydligt bättre. Hitta mätbara punkter som kan appliceras på varje gruppmedlem och utvärdera dessa vecka för vecka. Samtidigt hade en ökad och tydligare kontakt med stakeholdern lett till att vi tvingats prestera bättre. I takt med att vi blev mer bekanta med projektet skulle vi frågat efter vad som kunde skapa mervärde för stakeholdern. I dagsläget låg fokus på att genomföra det som exakts efterfrågades och inte ta på oss mer vilket i efterhand skulle varit fördelaktigt för stakeholdern.

Design decisions and product structure

How your design decisions (e.g., choice of APIs, architecture patterns, behaviour) support customer value

- A. Grundvalen av plattform och leverans är grundade i kontraktets behov. Ett smidigt sätt för anställda att kunna skapa inköps PDF:er. För detta bestämdes en hemsida som leveranssätt på grund av dess portabla format. Detta möjliggör även användandet av HTML & CSS för grafisk design vilket ökar utvecklingstakten. Denna idé om en hög utvecklingstakt genomsyrade även valet av API:er för den java baserade servern. Skapning av PDF:er och hantering av HTTP servern sköttes av de simplaste API:erna vi kunde hitta. Varje sida var uppdelat i ett eget "context" vilket separat skapade och returnerade en komplett sida. Detta gjorde varje del av kodbasen mer modulär och öppnade upp portarna för att parallellt arbeta med separata "context". Dessa faktorer bidrar till en simplare och effektivare scrumcykel där mer kan uppnås på samma tid och den tiden kan delas i fler user stories.
- B. Dessa val gjordes till viss del kollektivt Ad Hoc när de behövdes. Eller ännu värre utav en enda individ som under sin tilldelade uppgift behövde funktionaliteten. Detta gjorde så att uppföljning och vidareutveckling i vissa delar av kodbasen kändes mer avgränsade då resten av laget inte alltid var på samma sida om hur eller ens varför ett visst API skulle användas. Alla i gruppen bör ha en tydlig bild över vilka API:er som används och varför.

A→B

Dessa förbättringar skulle kunna uppnås genom en mer tydlig kommunikation under startmötet under måndagen. Vilka API:er som ska läggas till/ utvecklas under scrumcykelns gång, samt vilka som introducerades förra veckan för att uppmuntra användningen av den nya förmågan.

Which technical documentation you use and why (e.g. use cases, interaction diagrams, class diagrams, domain models or component diagrams, text documents)

- A. Under utvecklingen har koden för det mesta behövt tala för sig själv. Den enda tekniska dokumentationen vi planerade att göra var beskrivningar av alla klassers gränssnitt enligt Javadoc-formatet, men även detta gjordes ofta inte då flera av oss glömde att kommentera sin kod och vi därför fick skriva mycket av dokumentationen efter att vi var färdiga med programmeringen. Vi har inte använt några diagram då vi inte alls planerade strukturen på programmet i förväg - istället lade varje medlem helt enkelt till funktioner på det vis de själva bedömde lämpligt, vilket kändes tidseffektivt i stunden men ledde till jobbiga konflikter då olika lösningar som utvecklades parallellt var inkompatibla med varandra och gjorde

koden svår att förstå och modifiera då liknande problem inkonsekvent löstes på olika vis.

- B.** Vi borde ha en plan för programmets grova struktur innan vi börjar skriva kod, i formen av standardiserade diagram som utvecklarna kan känna igen från andra projekt. Alla klasser och metoder borde komma med kommentarer som beskriver deras funktion så att andra utvecklare snabbt kan förstå hur de ska användas.

A→B

För att uppnå detta behöver vi tillsammans komma överens om vilka diagram som behövs för att tillräckligt väl beskriva den planerade strukturen av programmet och skapa och uppdatera dem enligt nästa del. Vi måste kräva av oss själva och andra att det vi levererar uppfyller gruppens specifikation för dokumentation. Inför varje sprint bör varje user storie ha tydliga tasks för hur dokumentation bör se ut så att alla vet vad som förväntas av den specifika user storyn. För att upprätta hålla detta kan en fast punkt i retrospective införas som utvärderar gruppens dokumentation.

How you use and update your documentation throughout the sprints

- A.** Vi har som sagt knappt haft någon dokumentation att använda och uppdatera.
- B.** Om vi har dokumentationen som beskrivet i förra delen bör den alltid stämma, och all kod som skrivs bör ta den i akt och följa dess specifikationer.

A→B

Vi måste se till att, när vi ändrar någon kod, vi antingen ser till att den fortfarande betar sig som dess kommentarer antyder, eller om syftet är att ändra dess beteende, även ändra kommentarerna för att beskriva det nya beteendet. Vi måste kräva att all kod som skrivs följer strukturen i våra diagram och avslå commits som bryter mot den. Om planen ändras ska dokumentationen ändras innan koden skrivs, och detta får inte ske under en sprint - lämplig tid för att ändra strukturdiagrammen är under måndagsmötet då alla kan väga in och anpassa sin plan för sprinten efter diagrammen eller vice versa.

How you ensure code quality and enforce coding standards

- A.** Kodkvalité och formateringsstandarder uppehölls mest genom sociala tryck och inte strukturella. Eftersom grupper under de tidiga veckorna var mer en vänskapsgrupp och mindre en proper arbetsgrupp. Under projektets gång så övergick trycket till ett

strukturellt genom större granskning under pull requests. Därav förändrades vår DOD "Accepterad av någon annan i gruppen", inte i definition utan i utförande. Den sociala aspekten fångade fortfarande det som inte pull request granskningen gjorde. Dessa frågor togs sedan upp under något möte: måndag, onsdag, eller fredag.

- B.** Konkreta kontroller för att bibehålla kvalitén i repo:t behövs för att behålla läsbarheten och möjliggöra vidareutveckling. Detta kan enkelt uppnås genom en standardisering av vårt nuvarande system av pull request granskningar där en lista finns av saker som tillägget ska uppnå finns. En annan process som skulle hjälpa är att ha en gemensam struktur för var i projektet filer ska vara, såsom html, css, och javascript filer.

A→B

En tydligare struktur för granskning av ny kod skapas i början av projektet och uppdateras allt eftersom ifall brister upptäcks. För att underlätta denna granskningen och förhindra att arbetare antar saker så borde filstruktur och liknande planeras i förväg där tydliga förvaringsställen för specifika filtyper finns.

Under sprintplanning bör veckans struktur diskuteras så att alla vet vad som preliminärt kommer tillkomma och på vilket sätt. Detta underlättar i slutet av veckan när en sammanställning ska göras. Under retrospective bör vi utvärdera det som gjort noggrannare och analysera om det som lagts till upprätthåller den önskade kvaliteten.

Application of Scrum

The roles you have used within the team and their impact on your work

- A. Vi har under kursen gång rullat på positionen scrummaster. Detta gjordes för att alla skulle ha möjlighet att få testa på dess uppgifter. På grund av att detta var gruppens första kurs i agilt ledarskap kändes detta som ett rimligt beslut. Nackdelen har varit att vi inte haft så tydliga roller vilket gjort det lite rörigt i vissa moment. Vi har även rullat runt på positionerna sekreterare och developers. En person har varit PO under hela projektet för att effektivisera kontakten med stakeholdern
- B. I framtida projekt kommer tydligare roller att vara betydligt mer användbart för gruppens medlemmar. På så sätt kan projektet dra nytta av alla gruppmedlemmars styrkor något som vi inte till fullo gjorde i detta projekt. Vi hade endast en PO vilket gjorde att kontakten kunde ske effektivt. Dock blev PO en mellanhand eftersom vi endast utnyttjade honom som kontaktperson, detta ledde till en långsammare informationsspridning.

A→B

Genom att från början sätta tydliga roller kan B uppnås. Detta anser vi inte behöver betyda att en enhälligt behöver vara scrummaster men att dess uppgifter tydligt delas upp på medlemmarna som ska ta dess ansvar. Detta gör att personerna i fråga kan lära sig och utvecklas med uppgifterna något vi under projektets gång hade svårt med eftersom positionen flyttades runt. Att endast ha en PO tycker vi var bra och något vi i fortsättningen kommer använda oss av i enlighet med Scrum men försöka ha gemensamma möten med stakeholdern med jämna mellanrum.

The agile practices you have used and their impact on your work

- A. Vi har använt oss av det agila arbetssättet Scrum, eftersom vi fick en introduktion till dess utformning i början av kursen. Appliceringen av Scrum var till en början något trevande då vi hade svårt att vet hur user stories skulle utformas, vad de olika rollerna innebar och vad alla begreppen kopplat till Scrum betydde. Efterhand ökade arbetets effektivitet och vi skrev user stories varje måndag, arbetade med dem under veckan och gjorde Sprint Reflection på fredagar. De user stories vi gjorde var relativt stora och det blev oftast runt fem user stories per sprint för gruppen totalt. Vi använde oss av standups på onsdagar för att gå igenom problem som gruppen kan ha stött på. För att vara effektivare på möten ville vi använda timeboxing. Detta blev dock aldrig verklighet på grund av att vi inte var helt säkra på hur det fungerade.

- B.** Önskvärt hade varit att skapa user stories i större utsträckning och gör dem mindre och mer kvalitativa. Effekten hade blivit att medlemmarna i gruppen hade till större utsträckning arbetat självständigt och anpassa hur mycket som görs varje vecka beroende på hur mycket tid som medlemmarna upplever sig ha. Effektiva möten tre gånger i veckan hade varit önskvärt där måndagen går ut på att skriva user stories och onsdagen för att lösa problem. Fredagens syfte blir att summera veckan och ta med oss lärdomar.

A→B

För att ta oss dit behöver vi arbeta med att bryta ner user stories så att det är lättare att dela upp arbetet mellan oss och gör oss mer självständiga. Svårigheten blir att hålla dem oberoende från varandra samtidigt som de inte blir för stora men något som får strävas efter. Att aktivt arbeta med timeboxing hade kunnat effektivisera möten och sett till att mängden tid som läggs på en fråga hamnar i proportion till dess betydelse för arbetet.

The sprint review and how it relates to your scope and customer value (Did you have a PO, if yes, who?, if no, how did you carry out the review? Did the review result in a re-prioritisation of user stories? How did the reviews relate to your DoD? Did the feedback change your way of working?)

- A.** Under projektets gång hade vi en i gruppen som var PO och skötte kontakten med stakeholdern. Genom den kontakten kunde vi få utvärdering på projektet och vilka förväntningar som fanns samt funktionalitet som förväntades finnas. På sprint reviewn använde vi oss lite av DoD vilket i efterhand gjorde vår definition av att vara klar med en US något oklar.
- B.** Använda sig flitigt av DoD samt få mer konkreta förslag av stakeholdern hade skapat en tydligare väg genom projektet. Genom en tydlig DoD samt flitigt användande kan en bra struktur byggas.

A→B

Om alla hade varit med i kontakten med stakeholdern under vissa möten hade gruppen kunnat skapa sig en tydligare bild av förväntningar och krav på applikationen. Dock viktigt att vi fortsätter använda en PO för smidigare kommunikation men några gemensamma möten hade vi ansett vara fördelaktigt. Samtidigt skulle vår DoD varit en tydligare del av varje review och för att avgöra ifall en US varit klar eller inte. Detta hade ökat tydligheten i mötena då vi på ett objektivt sätt kunnat avgöra vilka US som varit klara och inte.

Best practices for learning and using new tools and technologies (IDEs, version control, scrum boards etc.; do not only describe which tools you used but focus on how you developed the expertise to use them)

Smidigt att ha allt på samma ställen.

- A. Vi använde oss av valfri IDE för kodutveckling. Som gemensam plattform för koden användes GitHub och det var också genom GitHubs verktyg "projects" som vi hade vår Scrum Board. Det fanns en blandad kompetens i gruppen och erfarenheterna av att använda framförallt GitHub var blandade. Detta medförde en viss ineffektivitet i hur vi mergade brancher efter en sprint och det var många konflikter som fick lösas i slutet av veckan när brancherna skulle slås samman till main.
- B. Det optimala läget var att vi hade använt oss av git flow och istället för att merga samman med en merge commit, använda sig av rebase. Detta hade gjort att vi inte hade haft en massa stora konflikter vid slutet av veckan när alla brancher slogs samman och vi hade på så vis kunnat uppnå en högre effektivitet.

A→B

För att uppnå detta hade alla behövt sätta sig in i hur dessa system fungerar istället för att lära sig under resans gång. Då hade en högre effektivitet kunnat uppnås och mer tid hade lagts på värdeskapande aktiviteter. Exempelvis hade några youtube-videor om hur detta görs effektivt kunnat ses av hela gruppen för att säkerställa att alla arbetar på samma sätt.

Relation to literature and guest lectures (how do your reflections relate to what others have to say?)

- A. Vi har under kursens gång inte använt något kurslitteratur eller några gästföreläsningar. Gruppen har under arbetets gång använt sig av litteratur och andra källor för att lösa de tekniska problemen som uppkommit. Detta gäller främst för att öka förståelse för de programmeringsspråk som används och dess funktioner. Vi har även tagit del av information som angivits på Scrums egna hemsida för att få mer information om arbetsmetodiken kursen använder sig av.
- B. Gruppen tror att en gästföreläsning hade kunnat användas för att ge studenterna en ökad förståelse för arbetsmetodiken och allmänna tips. Vi tycker dock inte kursen behöver någon kurslitteratur utan informationen som ges på Scrums hemsida i kombination med föreläsningar är mer än tillräcklig

A→B

Genom att lägga in en gästföreläsning i mitten av kursen kan en ökad förståelse för metoden fås. Vi tycker samtidigt att de första veckorna av trevande i okänd terräng är bra och bör bibehållas men att få höra från en person ute i arbetslivet som applicerar metodiken dagligen hade kunnat göra stor nytta.