

Final Report - Orup

DAT257

Agile software project management

Oscar Adamsson

Fredrik Ahlqvist

Joel Båtsman Hilmersson

Elin Forsberg

Elin Hagman

Erik Juliusson

Daniel Persson

Agnes Rosendal

Customer Value and Scope

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

A: Scopet har varit flera minispel som ingår i en applikation lämplig som ett pedagogiskt verktyg för en förskola. Stakeholdern har varit en blivande förskollärare. Uppdelning av arbetet (features) har skett på så sätt att varje gruppmedlem tagit lite större ansvar för ett visst spel. Funktionalitet prioriterades framför visuella aspekter, som var tänkta att genomföras senare under arbetets gång. User stories var till en början för stora, men blev mindre efter reflektion kring problemet. De var ofta horisontella och inte vertikala.

B: Scopet har varit bra och något som vi hade velat behålla. Stakeholdern, och dess roll, hade kunnat inkludera mer feedback från slutanvändaren (förskolebarn). Arbetet hade kunnat delas upp mer mot att bli klar med ett spel i taget, för att snabbare ha en MVP till stakeholdern. Detta hade kunnat vara på bekostnad av den effektivitet som det parallella arbetet ändå medfört. Ett större fokus på vertikala user stories hade i så fall också varit önskvärt.

A → B: Ha ett närmare samtal med stakeholdern kring hur denna bäst arbetar mot slutanvändaren (förskolebarn). Lägga upp arbetet, user stories, mot att prioritera ett spel i taget. För att det ska vara möjligt bör user stories bli mer vertikala men också mindre, för att minska risken att få problem med parallella user stories. Helt enkelt minimera beroende mellan user stories.

2 - 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: En MVP, fungerande applikation, där vi har utvecklat ett antal minispel som ingår i den. Lära sig/bli bättre på tillvägagångssättet scrum. Lära sig /utveckla sina förmågor inom Unity (en spelmotor) och hur den kan användas för att skapa applikationer. En viktig del var också vårt teamwork. Vi ville ha en bra stämning, lagom arbetsbelastning och öppen kultur.

B: Större fokus på att applicera scrum, kanske utmana oss själva med att utforma scopet på ett sådant sätt att scrum vore ännu mer kritiskt att använda för att lyckas. Lägga större fokus på varje gruppmedlems individuella mål, testa att integrera dessa mer i hur arbetet utformas varje sprint.

A → B: Lägga om scopet till ett mer vertikalt flöde och arbeta med varje spel för sig själv, för att sätta scrum arbetet "på prov". Tydligare belysa varje individs specifika mål under gruppmöten och följa upp dessa.

3 - 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: Vi har använt en tydlig mall för user stories, med krav att det finns acceptance criteria för respektive user story och uppföljning när någon missat detta. Ibland har det varit något för långa/stora user stories men det blev generellt bättre med arbetets gång. Arbete med user stories inom olika minispel parallellt ökade mängden arbete som kunde åstadkommas per vecka, då det ofta krävdes stakeholder feedback innan nya user stories inom ett spel kunde skapas. Svårt att uppskatta velocity på user stories, testade flera olika tillvägagångssätt för detta. Skapandet av user stories kunde ganska ineffektivt, som en följd av gemensamma diskussioner kring varje ny user story. Mer effektiva tillvägagångssätt gav istället negativa konsekvenser för estimeringen av velocity.

B: Vi hade velat ha mindre, oberoende, user stories. Maximera effektivitet och precision av estimering av user stories. Implementera sätt att ha mer löpande feedback från stakeholder för att kunna åstadkomma fler user stories per minispel per vecka.

A → B: Ramverk för hur user stories ska utformas för att vara små och oberoende. Mäta tid och precision av velocity estimering för olika metoder för att bättre fastställa ett optimalt tillvägagångssätt. Undersöka möjligheter för mer direkt/frekvent feedback från stakeholder.

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

A: User stories arbetas med i nya branches vilket framgår av branchens namn. Vid det att en/flera user stories anses klara gjordes en pull request för att få in branchen i main. Vid konflikter mergades main in till branchen först, för att sedan återigen göra en pull request. Valfri gruppmedlem kan kontrollera denna och ansvarar då för att testa/pricka av så att varje acceptance criteria i user storyn överensstämmer med de commits som skett. Detta har skapat värde hos stakeholdern eftersom denne kan vara mer säker på att det som har kommit in i den färdiga delen av produkten är fri från buggar och andra potentiella problem.

Följdaktligen skulle också Definition of done kontrolleras, men det följdes inte till så stor grad. Om gruppmedlemmen hittar fel görs en beskrivande kommentar i pull requesten och notis skickas till den ansvariga för branchen/user storyn. Om allt stämmer så mergas den in i main. Vissa gruppmedlemmar var betydligt mer bekväma/rutinerade med att hantera pull requests och fördelningen var därmed skev i antal genomförda "kontroller" per gruppmedlem.

Vi hade ingen motsvarighet till Unit tester inom projektet då det ansågs inte ge tillräckligt mycket nytta då många av sakerna i Unity är kopplade till grafiska objekt som är svåra att testa i sig. Detta anses inte påverka värdet av produkten eftersom testerna inte hade bidragit med tillräckligt mycket då man enbart hade kunnat testa en liten del av all kod.

B: Definition of done skulle kunna ha tagits mer hänsyn till vid att man accepterar pull requests. Det hade också varit bra om fler kände sig bekväma med att kolla igenom pull requests under ett

tidigare stadie av projektet, för att jämnar ut arbetsbördan inom den typen av uppgift men också för att få en bättre samförståelse kring vad som skett under sprinten. Det hade gett värde till oss som en grupp då arbetsbördan av att acceptera pull requests jämnas ut.

Det skulle också vara bra om det var lättare att se vad varje pullrequest hade för några acceptanskriterier då man i nuläget behövde navigera till trello och kolla upp userstoryn för att se detta.

A → B: Genom att tidigt gå igenom hur pull requests godkänns med alla i gruppen hade fler vågat känna sig bekväma med att göra det redan från början. För att öka användandet av Definition of done skulle vi som grupp kunnat ha pushat varandra mer till att inte slarva med det. Ytterligare en lösning hade varit att lägga till Definition of Done samt acceptanskriterier i pull request så den som kollar igenom en pull request lätt kan bocka av detta innan den accepterar den. Det hade tvingat den som kollar igenom pull requesten att kolla igenom samt acceptera dessa innan det går in i main.

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

A: Vi använde följande KPI:er: Stressnivå, antal user stories som är kvar från sprinten och nöjdhet med projektet. Tanken var att använda stressnivå för att bättre uppskatta vilken velocity nästa sprint ska ha, givet att vi skulle se samband mellan stressnivå och velocity. Antal user stories från sprinten var tänkt att användas för att förstå hur mycket vi hunnit med under sprinten. Nöjdhet var till för att förstå hur nöjda vi var med projektet. Under projektet ändrades stressnivå till två separata stressnivåer (stress inom och utom projektet) och antal user stories kvar från sprinten togs bort.

B: Separat KPI för stress inom och utom projektet från början av projektet. Inte ha antal user stories som är kvar i sprinten som en KPI, då det inte är en individuell KPI och vi inte kunde använda den för att dra förbättra våra processer på ett bra sätt. Generellt hade vi också velat använda våra KPI:er bättre.

A → B: Lägga till separat KPI för stress inom och utom projektet. Implementera tydligare metoder för hur KPI:erna ska användas för att förbättra arbetet, hur dom skapar värde. Exempelvis ha ett tydligt syfte med resultaten och ha tid avsatt för att ev. förändra saker som konsekvens av dom.

Social Contract and Effort

6 - Your social contract i.e., the rules that define how you work together as a team, how it influenced your work, and how it evolved during the project

Social contract

A: Det sociala kontraktet har generellt följts väl, men vissa punkter skulle kunna ha uppfyllts bättre. Framförallt har det varit svårt att följa punkten att sitta tillsammans i arbeta i någon större utsträckning på grund av skillnader i schema och att I-studenterna har haft kandidatarbete som har krävt mycket tid. Dessa kandidatarbeten har inneburit många intervjuer som planerades in efter respondenternas önskemål, vilket har gjort att även om möten med IT-projektet har schemalagts i förväg har det inte alltid varit möjligt att arbeta tillsammans i önskad utsträckning.

Under projektets gång fördes diskussioner kring hur vi arbetade och förhållningsreglerna som hade satts. Dessa förhållningsregler förändrades i och med dessa diskussioner, vilket resulterade i fyra skriftliga förändringar av kontraktet.

En förändring av det sociala kontraktet som skedde skriftligt var punkten rörande förberedelse inför möten. Mötesprotokoll fördes från start av projektet, men scrum masterns uppgifter var inte helt specificerade. Detta ändrades efter att gruppen insåg att mer konkreta arbetsuppgifter vore bra för mötenas effektivitet och gruppmedlemmar uppmanades skriva in diskussionspunkter i förväg för att inte hamna på sidospår under mötet

Vidare implementerades standup-möten en bit in i projektet och skrevs in i det sociala kontraktet, vilket gav god insikt i hur gruppens medlemmar låg till i sitt arbete med user stories och vilka problem som hade dykt upp. Dessa möten hölls på måndagar, onsdagar och fredagar. På måndagarna stängdes varje sprint, men vi hade standup i början av måndagsmötet för att samla upp vad som hade gjort under helgen.

En annan förändring som skedde i kontraktet handlade om hur Github skulle användas. Från början var det sagt att man alltid skulle merge in main i sin branch för att undvika konflikter. Detta ändrades till att endast vara nödvändigt om konflikter faktiskt uppstod.

Den sista förändringen som skedde skriftligt var bytet av KPI:er. Till en början var en KPI hur stressnivån för gruppmedlemmarna såg ut. Efter diskussion insåg gruppen att stressnivån var hög inom andra kurser, men låg inom projektet, vilket gjorde att KPI:erna omformulerades för att se stressnivån inom projektet och utom projektet separat.

Under projektet hade vi ingen uttalad regel kring hur koden skulle se ut, med undantag av kommentarer på engelska, vilket gjorde att metodnamn med stor bokstav etc. inte stod med i kontraktet. Vi hade inte heller definierat hur kodkommentarer skulle skrivas förutom att de skulle vara på engelska.

B: Det sociala kontraktet anses ha varit bra, men har inte använts i den utsträckning som det hade kunnat och hade därför kunnat skapa mer värde för gruppen. Standup-möten hölls inte från projektstart, men hade varit bra att hålla direkt vid projektstart då osäkerheterna kring arbetet var större i början av projektet.

Det hade varit bra att ha det konstaterat i kontraktet att metodnamn skulle börja med stor bokstav och ha definierat hur kommentarerna skulle skrivas

A → B: Genom att ha en bättre uppföljning kring hur väl kontraktet följs hade det sociala kontraktet kunnat bidra med mer värde för gruppen. Vidare hade det varit bra att direkt skriva in standup-möten i kontraktet för att underlätta arbetet i projektets tidigare faser när medlemmarna inte hade vant sig vid Trello och programmeringsprojekt i grupp. Att även tydliggöra scrum masterns ansvar i det sociala kontraktet hade varit bra för att få mer givande möten direkt från projektstart.

Genom att skriva in hur koden och kodkommentarer skulle se ut i kontraktet hade koden och kommentarerna blivit mer enhetliga och inga ändringar hade behövt ske i efterhand.

7 - The time you have spent on the course and how it relates to what you delivered (so keep track of your hours so you can describe the current situation)

A: Under arbetet har antalet timmar som varje gruppmedlem lagt ner på projektet skrivits in i en tidslogg. Tidsrapporteringen har inte alltid skett direkt efter nedlagd tid, utan har i många fall rapporterats senare och således inte blivit helt exakt. Detta beror även på att det skiljer sig mellan effektiv och total tid. Den nedlagda tiden har även skiljt sig mellan gruppmedlemmar där IT-studenterna har rapporterat och arbetat fler timmar än I-studenterna. Detta dels eftersom majoriteten av IT-studenterna var bekväma med att arbeta i Unity och hade tidigare erfarenhet av att programmera spel, medan framförallt I-studenterna inte hade denna erfarenhet och därför inte kunde bidra lika mycket. Vidare lades även mycket av I-studenternas tid på kandidatarbete. Detta var något som gruppen hade diskuterat innan start och var införstådda med. Det framgick även att majoriteten IT-studenterna var mer vana vid att föra en tidslogg, vilket I-studenterna inte hade gjort tidigare och detta kan även spela viss, om än liten, roll i tidsrapporteringen. Även om arbetsfördelningen inte har varit helt jämn mellan gruppmedlemmar har gruppen kunnat se att de individuella stressnivåerna inom projektet inte har varit särskilt höga tack vare de KPI:er som har använts.

Oavsett skillnaderna i nedlagd tid har samtliga gruppmedlemmar bidragit väl till arbetets slutförande. Utifrån den totala nedlagda tiden är gruppen även väldigt nöjd med slutprodukten vilket antyder att tiden har spenderats väl.

B: En mer jämn fördelning av nedlagd tid hade varit att föredra då det dels innebär att känslan av att någon i gruppen inte har bidragit lika mycket och att man själv har lagt ner onödig tid inte kan framkomma. Det innebär även att de som i detta projekt inte har lagt ner lika mycket tid inte heller känner sig som sämre gruppmedlemmar på grund av deras lägre antal timmar, samt sämre kunskap och erfarenhet. För framtida projekt vore det därför fördelaktigt om samtliga

medlemmar hade ungefär samma förutsättningar att arbeta och bidra. Om alla gruppmedlemmar hade mer kunskap om Unity hade även effektiviteten varit högre.

För framtida projekt vore det även bra om timmarna som är inrapporterade i tidsloggen är mer exakta och rättvisande för att ge tidsloggen mer precision och relevans.

A → B: Genom att bli bättre på att föra in timmar i tidsloggen direkt efter arbete på projektet blir tidsloggen mer exakt. Vidare kan ett framtida projekt fördelaktigt väljas utifrån de erfarenheter och den kunskap som samtliga gruppmedlemmar besitter, även om skillnader också kan vara något positivt.

Under första sprinten hade vi ett arbetspass där de som tidigare inte hade arbetat i Unity fick lära sig lite om hur man gjorde GameObjects osv. En mer omfattande genomgång hade kunnat göra att alla kände sig bekväma med att ta på sig lite svårare user stories och även arbeta mer effektivt.

Utöver detta kan bättre planering av möten vara positivt för att göra det lättare för gruppen att arbeta tillsammans. Detta gäller inte nödvändigtvis möten inom detta arbete då planeringen av möten skedde väl, men framförallt planeringen av arbete i andra kurser så som kandidatarbeten. Detta är dock komplicerat eftersom samtliga I-studenter arbetade i olika kandidatgrupper och en gruppmedlem inte heller skrev kandidatuppsats på I-sektionen vilket gjorde schemat ännu svårare att få ihop.

Design decisions and product structure

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

A: Genom vårt val med flera minispel underlättade vi för att kunna skapa userstories som inte byggde på varandra. Vi har kunnat jobba parallellt på olika saker samtidigt utan större konflikter, vilket gjort att vi kunnat leverera ny funktionalitet i samtliga spel under varje sprint. Detta har i sin tur bidragit till kundvärde, då vi kunnat åstadkomma framsteg i flera delar av spelet under varje sprint. Vi valde att använda en API för att hämta dagens väder i väderspelet, vilket bidrar till kundvärde genom att barnen kan öva på vädret som är just den dagen.

B: Vid ett större projekt går det att tänka sig att man kunnat utnyttja API:er i även de andra minispelen, såsom musik-spelet eller memory-spelet, så länge det stödjer stakeholdern och målgruppens önsknings. Ett exempel skulle vara att utveckla memory-spelet till att inkludera en mycket större mängd djur med hjälp av API:er om det var eftersökt hos kunden. Vi hade också velat testa produkten på målgruppen tidigare, alltså låta den testas av förskolebarn för att på så sätt hitta problem och saker som kan förbättras.

A → B: Vid ett större projekt, eller för ett program som skulle fortsätta utvecklas hade vi kunnat ha bättre kontakt med stakeholdern för att se om implementationer av flera API:er skulle förbättra produkten vi skapar. Genom att låta spelet testas på målgruppen (förskolebarn) tidigt hade våra designval och dess relation till kundvärde tidigare kunnat bekräftas eller dementeras. Detta tillvägagångssätt stöds av den agila principen *fail fast* (Salimi, u.å.).

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

A: Den som skriver koden har haft huvudansvar för att koden dokumenterats. Däremot, med vårt upplägg att koden alltid reviewas av en annan person innan den mergas in till main, har också den person som tillåter en merge möjligheten att flagga ifall dokumentationen tycks vara otillräcklig.

Programmet har byggts med Unity, och vi valde att enbart föra dokumentation genom kommentarer i koden. Det eftersom att t.ex UML diagram skulle ta upp tid som vi hellre ville lägga på själva produkten. Detta gjorde att projektet utvecklades sprint för sprint, och bilden om hur slutprodukten skulle se ut växte fram successivt i takt med feedback från stakeholder etc. Det fanns ett större självförtroende hos IT-studenterna, som gjort spel tidigare, kring att veta vad vi inom tidsramen skulle hinna åstadkomma, samt hur det skulle göras. Det var t.ex. enklare för dem att ta initiativ till vilken funktionalitet som behövdes och inte, medan I-studenterna (åtminstone till en början) hade svårare att se framför sig exakt vad som skulle behöva göras för att nå slutprodukten.

B: Det hade varit bra att i början av projektet ha en tydligare plan för vad som ska göras i projektet totalt, t.ex. vilken sprint som skall fokusera på vilken del av produkten. Det hade gjort det enklare att skapa fler user stories till product backlogen, då en förståelse för hur slutprodukten skulle se ut redan var tydlig och definierad.

Det hade även varit fördelaktigt att till ett kommande projekt enklare kunna se vem som skrivit vilken kod, för att veta vem man kan fråga ifall ändringar behövs. Detta hade kunnat göras genom användning av teknisk dokumentation. Ett exempel på teknisk dokumentation skulle också kunna vara att införa ett dokument som förklarar vad en viss klass gör. Detta hade gjort det lättare för personer att sätta sig in i koden och förstå kopplingar bättre vilket gjort det mer framtidssäkert för andra att ta projektet vidare.

A → B: Det hade gått att uppnå en gemensam bild av slutprodukten genom att göra någon sorts schema/flödesdiagram/visuell styrning som gjort det tydligt för vilken del av produkten som ämnas bli klart under varje sprint. Det hade också bidragit till värde för stakeholdern, då stakeholdern kunnat se vad vi ämnat bli klara med under respektive sprint i ett tidigare skede.

För ett större projekt hade det kunnat införas någon form av teknisk dokumentation över hur saker hör ihop, vilket t.ex. hade gjort det enklare för oss att hjälpa till på flera av minispelen istället för att främst jobba på sitt "huvudspel".

Teknisk dokumentation över alla klasser skulle kunna göras genom att använda ett verktyg som genererar ett html dokument över alla klasser baserat på XML kommentarer i koden. Det hade gjort att man lätt hade kunnat få ett enhetlig dokument med alla klass och metoder i varje script.

Man skulle även kunna tänka sig att man i DoD skriver med att det ska finnas kommentarer över vem som skrivit vilka scripts för att underlätta i denna aspekten.

10 - *How you use and update your documentation throughout the sprints*

A: Vi har använt oss av mötesprotokoll för måndag- och fredagsmöten, vilket gör att händelseutvecklingen sprint för sprint tydligt går att ta del av. Dessutom har våra sprint review, sprint retrospective och team reflections dokumenterats vilket bidragit till möjligheten att kunna följa hur projektet utvecklats.

Eftersom vi hela tiden har gjort nya mötesprotokoll / andra dokument har det sällan varit i behov att gå tillbaka och uppdatera tidigare dokument. Avvikande från detta har varit vårt social contract som har blivit uppdaterat med nya punkter allt eftersom vi känner att det har behövts ändras.

Efter någon vecka i projektet bestämde vi också att scrummastern skulle lägga ut mötesprotokollen inför ett möte, så att varje individ i gruppen hade en chans att gå in och lägga in punkter för nästa möte istället för att göra det direkt i början av varje möte.

B: Vi hade kunnat införa något system för att också dokumentera vad som sades under standup möten på onsdagar. Det hade gjort det ännu tydligare hur arbetet under sprintens gång såg ut. Denna punkt hänger ihop med idén om att ha en tydligare bild om vad som sker, sprint för sprint. Vi hade sällan explicita mål för sprinten, vilket vi hade kunnat införa för att tydligare kunna följa utvecklingen mellan sprinten. Genom att dokumentera målen för sprinten hade det också varit enklare att utvärdera hur framgångsrik sprinten var, i termer av hur väl vi uppnådde målet.

Det hade också varit hjälpsamt att göra återkopplingar från förra mötet för att kunna följa upp vad som var planerat att göra tills mötet som var efter. Vi hade en punkt om detta i vårt mötesprotokoll men den används väldigt sällan.

A → B: Vi hade också kunnat införa mötesprotokoll (om än korta och inte lika detaljrika) även under onsdagar, vilket hade gjort att man kunnat följa utvecklingen bättre under sprintens gång.

Explicita sprint mål hade kunna skrivas upp i mötesprotokollet där vi startar upp den nya sprinten. Det hade gjort så att man hade kunnat gå tillbaka och kolla vad målet för sprinten är om man är osäker. Det hade också varit en bra sak att ha när man sedan gör sprint retrospective eftersom man lättare ser vad som faktiskt var planerat så man kan jämföra det med vad som hunnits med.

Återkopplingar till föregående möte hade kunnat genomföras genom att gå tillbaka till föregående möte och kolla vad vi skrev upp på "till nästa möte" punkten. Samt vara noga på att skriva upp saker som vi tänker att vi ska ta upp och hantera på nästa möte.

11 - *How you ensure code quality and enforce coding standards*

A: På grund av projektets storlek och tidslängd valde vi att inte ha särskilt mycket explicita kriterier på kodkvaliteten. Viss kod skulle därför kunna ha gjorts på ett bättre, eller mer optimalt sätt, men det har varit ett medvetet val under projektets gång att prioritera att det fungerar istället för att det ska vara perfekt. Ju mer kännedom vi fick kring Unity har vi också kunnat utnyttja inbyggd funktionalitet i programmet mer. Vi har också alltid sett till att en person som inte gjort userstoryn godkänner koden innan den mergas, vilket har höjt kvaliteten. Personen som godkänner userstoryn kollar nämligen igenom hur koden ser ut, och om det visat sig att koden inte är optimerad kunde man ge förslag på förbättringar innan den mergades in i main. I början av arbetet var inte alla bekväma med att godkänna kod, men det blev fler som vågade ta för sig i det senare under arbetet. Pågrund av projektets utformning (ett spel i Unity), skrevs inga tester explicit, istället testades funktionaliteten manuellt genom att spela det, för att på så sätt upptäcka buggar. Vidare skapades ingen standardiserad filstruktur för filer och bilder, vilket gjort att det är lite svårare att navigera runt och hitta bilder/filer. De flesta resurserna var grupperade efter spel men det fanns några resurser som låg utspritt.

B: Till ett annat projekt, speciellt om det är större eller mer långtgående, skulle det behövas fler sätt att säkerställa kodkvaliteten. Dessutom skulle det vara bra med mer enhetliga namn på metoder, variabler, filnamn, etc. för att enklare kunna förstå koden för någon som inte skrivit den. Ett organiserat system för filer/bilder/etc i Unity hade också föredragits i syfte att göra det enklare att navigera och hitta saker. Någon form av mer etablerad testning hade också kunnat göras. Det hade också varit bra att låta alla ta ansvar för att godkänna pull request tidigare i projektet,

A → B: Genom att bestämma att metodnamn skulle börja med stor bokstav och definiera hur kommentarer ska skrivas i koden hade kvaliteten höjts. Genom att skapa en filstruktur innan arbetet påbörjades hade det varit enklare att navigera i programmets uppbyggnad. Genom att genomföra mer etablerad testning hade kvaliteten också höjts. Genom att gå igenom gemensamt i gruppen hur man godkänner pull request, och låta alla testa att godkänna en under de första sprintarna, hade det ansvaret blivit mer jämnt fördelat.

Application of scrum

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

A: scrum master: Förutom Sprint 1 så hade vi en växlande scrum master som byttes i början av varje Sprint start-up så att i slutändan fick fem av åtta medlemmar vara scrum master. Ansvarsområden som scrum master var att skapa ett mötesprotokoll inför varje möte och sedan

leda mötet genom punkterna. Vi valde att låta scrum mastern bestämma hur möten skulle genomföras och hur de skulle dokumenteras för att prova olika metoder och hitta de som passade våran grupp bäst. Ett tidigt problem vi hade var att möten kunde dras ut och inte vara fokuserade vilket vi löste genom att ge scrum master mer makt angående tidsestimering av mötespunkter. Valet gjorde så att möten blev kortare och mer effektiva och vi hade en scrum master som använde en timer för vissa moment för att få laget att inte fastna på en punkt.

Product Owner: Vi hade en konstant Product Owner under projektets gång som fick bli gruppmedlemmen som hade bäst kontakt med våran Stakeholder. Den här rollen fick aldrig bli en scrum master för att inte ge för mycket ansvar till en person. Rollens ansvar var främst att ha kontakt med Stakeholder och utföra en sprint review med Stakeholder efter varje sprintavslut. Stakeholdern fick sedan komma med feedback på arbetet som hade gjorts under sprinten samt komma med förslag på framtida utvecklingen och förbättringar. Den här feedbacken var det första vi gick igenom på varje möte och diskuterades under sprint retrospective. All feedback som skulle implementeras blev user stories och blev prioriterade under nästa sprintuppstart. I början ville vi avsluta en sprint innan sprint review med Stakeholder genomfördes, men insåg att feedbacken kom för sent så istället tidigare lade vi mötet med Stakeholdern så att vi fick feedback innan sprinten var slut så att vi kunde förbereda det inför nästa sprint. Det var också Product Owners ansvar att ställa frågor från gruppen till Stakeholder och svaren gavs samtidigt som feedbacken.

Programmerare: De resterande sex personer utan roller blev utvecklare som hade lika mycket ansvar och kunde fritt välja vilka user-stories de skulle jobba på under sprinten utan några mer specifika roller. Både Product Owner och scrum master hade också ansvar som utvecklare men deras andra roller hade prioritet. Under möten utsågs en gruppmedlem till sekreterare och skrev ner allt som blev sagt.

B: Alla rollerna förbättrades kontinuerligt efter varje sprint ju mer vi insåg vad som funkade för oss men det fanns även mer utrymme för förbättring. Idéen att scrum mastern hade kontroll över hur möten skulle hållas var en bra start innan vi visste mer om vad som passade oss. Med det sagt, så hade en mer bestämd mötesstruktur gynnat laget om vi visste exakt hur möten skulle hållas, vilket skulle kunna innebära en konstant scrum master.

Product Owner rollen passade bra för projektet och det hjälpte att rollen redan hade bra kontakt med Stakeholder. I ett framtida projekt så är den som får rollen kanske inte lika uppenbart och skulle kräva ett beslut internt om vem som ska få bli Product Owner. Owner rollen är svår att förbättra med den erfarenhet vi har haft dock så hade snabbare feedback tidigt i projektet varit gynnsamt.

Den sista förbättringen är att bestämma roller inom Programmerar-rollen så att de med mest förkunskaper inom specifika områden kan fokusera på user-stories som passar deras kompetenser bättre och kortar ner utvecklingstiden.

A → B: För att scrum master rollen ska vara förutbestämd så borde den gruppmedlem som majoriteten anser är mest passande för rollen bestämmas i början av första sprinten. Då kan

scrum mastern skapa en struktur för varje möte och ha en mycket bättre tidsestimering för varje punkt vilket skulle effektivisera gruppens arbetssätt.

Product Owner rollen skulle förbättras genom att ha flera möten med Stakeholder eller mail-kontakt så att frågor kan få svar direkt under en sprint istället för i slutet.

Programmerar-rollen förbättras genom att alla medlemmar får lista deras kompetenser så kan roller delas ut baserat på vad som behövs för projektet, samt att man kan dela upp user-stories baserat på de kompetenserna.

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

A: Under projektets gång så följde vi flera olika *agile practices* och blev kontinuerligt mer och mer bekväma med hur vi skulle applicera det till vårt projekt. I början definierade vi ett scope för vårt projekt och följde det. Vi satte upp en scrum board direkt som vi uppdaterade konstant och kunde separera våra user stories inom olika Epics och ge varje user story en velocity. Alla user stories följde även en mall som nämnts tidigare och började alltid i en product backlog för att sen flyttas till sprint backlog. Vi hade även sprint retrospectives såväl som sprint reviews, både inom gruppen och med vår stakeholder varje vecka med tydliga roller inom gruppen. I sprint 6 introducerade vi även prioritering på user stories med delar som vi ansåg var viktiga att implementera in i slut produkten.

Andra agile practices som nämns i *the agile manifesto* (Lynn, 2022) är att vi levererade kontinuerligt utveckling av vår produkt med kunden och använda all feedback vi fick för att göra ändringar i framtiden, in i sista sprinten. Vi gav mycket frihet till alla att komma med egna idéer som sen kunde tas upp med vår stakeholder så att alla skulle vara engagerade med det som gjordes. Vi försökte att ha alla möten på plats men tillät även digital anslutning. Vårt arbetssätt var även hållbart och vi höll ett tempo som skulle kunna hållits i flera sprints till. Sist så satsade vi på att ha så många möten som möjligt per vecka med en sprint review och start up på måndagar, stand up på måndagar, onsdagar och fredagar med mer detaljer på fredagar.

B: *Agile practices* som vi skulle kunna ha lagt till eller förbättrat är att variera sprint längder beroende på hur det går. Det är bättre att avsluta sprints tidigare om vi har fått ny utveckling och visa det för kunden, men i vårt projekt hade vi bara kundkontakt på helger och med bara tre möten per vecka så fanns det inte mycket tid att stänga och starta sprints förutom på måndagar. En annan förbättring hade varit att ha prioriteringar på user stories tidigt och inte lämna viktig utveckling till slutet. Det hade gjort så att user stories som var beroende på andra inte låg i sprint backlog innan man kunde börja på dem.

A → B: För att åstadkomma flera chanser att diskutera hur det går i projektet och eventuellt starta sprints på andra dagar så hade flera möten behövts. Stand-up möten varje dag hade gjort att gruppen fick en bättre bild över hur mycket som var kvar och givit chansen att stänga sprints tidigare. Sist så skulle vi ha introducerat prioritets labels på Trello för stakeholder feedback user stories, samt user stories som skulle byggas vidare på. Då skulle vi alltid ta med hög prioritet user stories i varje sprint och inte lämna dem till nästa vecka.

14 - 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: Valet av product owner blev Daniel Persson då han hade bäst kontakt med vår stakeholder. Vi stängde och startade upp nya sprintar på måndagar, och till en början gjordes därför sprint reviewsen med en försening, då stakeholder och product owner genomförde sprint review över helger. Det gjorde att vi inte drog nytta av återkopplingen från stakeholder särskilt effektivt. Vi beslöt därför att det var bättre att visa stakeholder produkten på helgen när sprinten fortfarande var igång, då det sällan gjordes stora uppdateringar mellan fredagar och måndagar ändå.

Sprint reviewsen gav i första hand idéer på vad för nya user stories som behövdes, snarare än att det gjorde att vi omprioriterade user stories. De user stories som skapades utifrån feedback blev alltid prioriterade inför nästa sprint för att göra kunden nöjd. Det var däremot en konsekvens av att vi oftast som grupp skapade nya user stories inför varje sprint, snarare än att det alltid låg en mängd user stories i product backlogen. Det i sin tur blev så, troligen på grund av att vi i början valde att rotera ansvaret för att vara scrum master. Hade det varit en och samma person, hade det möjligen blivit mer kontinuitet, och möjlighet för någon att leda arbetet framåt. Däremot var det vettigt att välja att rotera ansvaret, då vi är studenter i en projektgrupp och ville lära oss så mycket som möjligt gemensamt.

Vår DoD innefattade ett krav som var definierat "Varje sprint ska skapa ett värde hos stakeholdern". Vid varje sprint försökte vi alltså åstadkomma förändringar som skapade ett värde hos stakeholdern. Feedbacken från stakeholdern var speciellt bra i form av att anpassa spelet till den tänkta målgruppen, nämligen förskolebarn ihop med pedagoger. Feedback såsom att inte inkludera poängräkning (för att undvika tävlingsmoment), eller att ha text i menyerna (för att underlätta för pedagogerna), förändrade alltså hur spelet utvecklades.

B: Nästa projekt hade med fördel kunnat lägga upp sprint uppstart/stängning och sprint reviews på ett mer effektivt sätt. Dessutom hade product owner kunnat inkludera stakeholdern i prioritering av user stories, med förutsättningen att stakeholdern är någorlunda införstådd i arbetssättet. Det hade också varit positivt att på ett tydligt sätt avgöra om varje sprint ger värde till stakeholdern (DoD).

A → B: Genom att ha sprintreview på samma dag som sprinten stängs, hade det underlättat för att sedan kunna få feedback och återkoppling från stakeholder innan man hinner för långt på nästa sprint. Genom att bygga upp en större mängd user stories i början i product backlogen hade man kunnat utnyttja stakeholderns åsikter mer. Då hade product owner med input från stakeholder kunnat prioritera user stories på ett sätt som bättre matchar DoD, alltså att ge värde för stakeholdern under varje sprint. För att tydligare veta om DoD uppfyllts hade vi kunnat ha en KPI som var att fråga stakeholdern om sprinten skapat värde.

15 - 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)

A: I vårt projekt använde vi Trello som scrum board, Visual Studio som vår IDE, samt Unity som spelmotor för utveckling av spelet, Discord för kommunikation, GitHub för kod och dokument och Google Drive under möten för anteckning och våran tidslogg. Förutom Discord och Google Drive så var resten av verktygen nya för vissa medlemmar i gruppen, iallafall så vi använde dem under projektet. Vårt scrum board användning förbättrades med tiden när vi blev mer bekväma med verktyget. Vi började även använda Butler för att automatisera processer vilket kortade ner mötestider. Visual Studio var alla bekväma med att använda, men språket C#, var nytt för vissa och krävde bara tid att lära sig. Unity var nytt för majoriteten av gruppen men tack vare ett tre timmars workshop med de i gruppen som kunde Unity blev alla snabbt vana med programmet vilket gav projektet en bra start. Vi implementerade även Unity Remote vilket effektiviserade arbetet eftersom det gjorde det lättare att testa appen på mobilen. Github var alla bekväma med förutom att göra pull requests till main. Efter ett möte så sa vi att alla skulle börja acceptera pull requests mer vilket förbättrade arbetsfördelningen inom gruppen. Vår tidslogg började bra med uppdateringar men alla var inte lika bra på att uppdatera under projektets gång vilket gjorde att dokumentet inte var lika värdefullt.

B: Val av verktyg var bra för projektet och behöver inte förbättras. Det som kan förbättras är användningen av verktygen. Organisering av dokument och tidsloggar, kod standard, Trello organisering och kommunikationsansvar hade varit gynnsamt för projektet så att allt följer en standard och inte blir rörigt eller lämnas utan att bli uppdaterat. Gruppens kunskap inom verktygen hade också kunnat förbättras genom att lägga mer tid på upplärning som till exempel Unity och Github.

A → B: Ett sätt att förbättra användningen av verktygen är att sätta en ansvarig för varje verktyg som också blir "expert" på verktyget. Det säkerställer att till exempel Trello används rätt från början och är organiserad. Då kan den ansvariga också kontrollera att alla fyller på tidsloggar eller att branches tas bort från GitHub. Det hade också varit bra att veta alla i gruppens kompetenser med de valda verktygen för att veta vad som hade behövts mer tid att lära upp. Ett exempel hade varit om vi visste från början att många i gruppen inte var bekväma med att genomföra pull requests, så hade vi tidigt kunna gått igenom det med alla.

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

A: Eftersom scrum and *agile practices* var nytt för många i gruppen så var vi tvungna att referera till litteratur och föreläsningar för att lära oss hur det ska användas. User stories, scrum roller och team reflection punkter togs från föreläsningar. Gruppkontrakt tog inspiration av exempel kontraktet som hittades på Medium (Drill, 2019). Våra KPI värden togs från Sealights (2022).

B: Om möjligheten presenterar sig själv så hade det varit bra att ta del av alla gästföreläsningar då det hade hjälpt att ha mer resurser i början av sprint 1. Det hade också hjälpt att läsa mer om optimala agile workflows innan så att man hade bättre grund istället för att lära sig mer vecka för

vecka. Det hade också hjälpt ifall kursen hade innehållit någon gästföreläsning men eftersom detta var utom vår kontroll är detta svårt att uppnå. Det hade gjort att man hade kunnat få en inblick i hur scrum appliceras i verkligheten.

A → B: Det enda sättet att genomföra målet är att söka efter mer information tidigt in projektet. I nästa kurstillfälle borde någon gästföreläsning hållas.

Referenser

11 scrum metrics and their value to scrum teams. Sealigns. (15 oktober, 2019).

<https://www.sealigns.io/software-development-metrics/11-scrum-metrics-and-their-value-to-scrum-teams/>

Drill, M. (5 maj, 2020). *Agile ways of working- social contracts*. Medium.

<https://medium.com/swlh/agile-ways-of-working-social-contracts-ab6b8c93429f>

Lynn, R. (21 mars, 2022). *Agile best practices for more effective teams*. Planview.

<https://www.planview.com/resources/guide/agile-methodologies-a-beginners-guide/agile-best-practices-effective-teams/>

Salimi, S. (u.å.). *Fail fast*. Agile Academy.

<https://www.agile-academy.com/en/agile-dictionary/fail-fast/>