

Post-Mortem Rapport

Projektgrupp DevIT - kurs DAT255

Arbetsgång

Översikt

När kursen drogs igång blev det snabbt uppenbart att den handlade om att lära sig att självständigt hantera ett mjukvaruprojekt gentemot riktiga kunder, baserat på Scrumprinciper. Vi var på workshopen där vi fick lära oss hur Scrum fungerar. Först gick vi igenom det teoretiskt på en föreläsning, och därefter fick vi testa att omsätta det i praktiken genom tillämpningar av Scrum-metoder i projektet för dagen, som gick ut på att tillsammans bygga en stad i LEGO. Vår tanke var att fortsätta använda oss av scrumprinciper, och vi lade upp planer för det, exempelvis genom att skapa en virtuell scrumanslagstavla i Trello. Till viss del använde vi oss av Scrum under hela arbetsprocessen, men det var mest i början och slutet som vi verkligen använde oss av user stories och "poäng".

Innan vi började programmera lades mycket tid på att skissa applikationens olika vyer på papper och planera vad vi ville ha för funktionalitet, och vilka delar som skulle prioriteras. Vi diskuterade detta mycket och såg till att alla i gruppen hade gemensamma mål vad gällde hur applikationen skulle se ut och fungera. När vi väl hade format en idé, pratade vi med människor i olika åldrar och från olika städer om vad de trodde om den. Då fick vi bekräftat att det fanns önskan om och ett behov av de funktioner vi ville bygga applikationen på, vilket gjorde oss säkra på att vi ville utveckla den. Vissa av oss läste på om Android Studio innan vi började programmera, eftersom en del av oss inte hade jobbat med det eller IntelliJ innan.

Vi hade även gruppmöten två gånger i veckan då vi gick igenom vad vi hade gjort sedan det senaste mötet, planerade vad vi skulle göra den närmaste tiden och löste tekniska problem så som versionshanteringskonflikter.

Under hela arbetet delades olika arbetsuppgifter ut till gruppmedlemmarna, så var och en hade sitt eget ansvarsområde. Detta betydde inte att personen behövde göra

hela det arbetet själv, utan snarare att personen såg till att arbetet blev gjort. I början delade vi upp arbetet så att tre personer gjorde skisser och funderade på designen medan två personer började med backenden.

Under kodningsprocessen delades snarare specifika klasser och funktionalitet upp, oftast utifrån de user stories som vi skrev ned under planeringsarbetet. Vi varierade hur vi arbetade och med vem, det blev mycket individuellt arbete hemifrån men vi tillämpade också så kallad pair programming och ibland satt vi flera från gruppen och arbetade individuellt men tillsammans, så att man kunde ta hjälp av varandra om det skulle behövas.

Delmoment

Scrum:

Fördelen med Scrum är att man vet vad man ska göra och har en god översikt över projektet och hur det ligger till tidsmässigt, både sett till specifika delar samt projektet i helhet. Scrum ger en god uppfattning över vad som behöver göras, både på kort och lång sikt. De konstanta mötena innebar att problem snabbt kunde åtgärdas och hanteras av gruppen utan att felet hindrade arbetet nämnvärt. Eftersom vi hade våra user stories var det enkelt att plocka ut specifika arbetsuppgifter och fördela dem till gruppens medlemmar.

Nackdelen var att ingen tog ansvar för att se till att Scrumprinciperna följdes, och tog på sig rollen som Scrum-master och product owner, vilket troligtvis berodde på att vi inte har arbetat med Scrum tidigare. Detta ledde till att vi bara använde Scrum i början och slutet av projektet. Backloggen gjorde vissa i gruppen stressade, då de fick en konkret bild över hur mycket arbete som behövde göras.

Under perioden vi använde Scrum märkte vi tydligt hur effektivt det var. I slutskedet av projektet när det var väldigt stressigt och mycket att göra hjälpte Scrum oss att prioritera funktionerna och distribuera arbetsuppgifter på ett tidseffektivt sätt. I framtiden skulle vi använda det i projekt som framför allt handlar om mjukvara och som sker i grupper som är av ungefär samma storlek som i det här projektet. Vi skulle inte använda det i stora projekt som inte går att dela upp i mindre user stories.

Gruppmöten:

Med hjälp av två gruppmöten i veckan fick alla gruppmedlemmar en bra bild av vad som skulle göras och vem som hade ansvar för vad. Det var också lätt att hantera olika situationer som dök upp under arbetets gång, exempelvis sjukdomar. Vi valde att ha två möten i veckan för att vi inte var säkra på att vi skulle hinna få nämnvärda förändringar gjorda under kortare intervall samt på grund av svårigheten med att inte ha fasta lokaler där vi alltid träffades och jobbade. Med fasta lokaler hade det varit mer naturligt att ha ett möte varje dag. En nackdel med möten var att vi lätt tappade fokus och började prata om annat, vilket gjorde att de inte alltid blev speciellt tidseffektiva. Men trots att mötena ibland blev lite ofokuserade ledde det aldrig till att vi inte visste vad vi skulle göra.

Mötena har bidragit till en bra effektivitet och vi kommer definitivt att använda det i alla projekt framöver.

Förberedandet (skisser, planering, etc):

De stora förberedelser vi gjorde i projektets inledningsskede innebar att vi alltid visste vad som skulle göras, hur allt skulle se ut samt gav oss en enhetlig struktur. Ett misstag vi gjorde var att vi missbedömde vårt tidsspann och var för noggranna med förberedelserna så att realiseringen fördröjdes. En annan bidragande faktor till att vi inte började programmera direkt var att vi inte hade tillgång till API:erna. Den omfattande förberedelsen och planeringen gjorde att resten av arbetet blev mer effektivt då vi alltid visste hur applikationen och dess funktioner skulle se ut och fungera samt vad som skulle göras när. Vi anser att vi alltid ska se till att förbereda oss väl och skulle använda denna metod i alla projekt, dock får innehållet i förberedelsen ändras beroende på projektet.

Parprogrammering:

Fördelarna med att programmera i par är att man har olika tankar och idéer om hur ett problem ska lösas, och är det något den ena inte förstår, kan den andra hjälpa till och förklara. Däremot kan det bli mindre tidseffektivt om två personer jobbar med samma problem istället för med varsitt. En annan nackdel med att programmera i par är att man lätt börjar prata om annat än programmeringen. Vi tycker det är bra att programmera i par när det handlar om nya saker som man inte har sett förut. Det är

även bra ifall någon skulle bli sjuk, då den andra personen i paret är väl insatt i koden och kan fortsätta på egen hand. Vi skulle inte använda parprogrammering för enkla programmeringsuppgifter, då är det mer tidseffektivt om två personer arbetar med varsin uppgift.

Hjälpmedel

Vi använde oss av Västtrafiks Rest-API då det föll sig naturligt eftersom vi skulle göra en reseplanerare för Göteborgs kollektivtrafik och behövde därför hämta tidtabeller och reseinformation.

Ett annat hjälpmedel vi använde var GSON, som är Googles bibliotek för hantering av JSON-filer, då det är bättre än det inbyggda stödet i Android för JSON. Vid sökningar på Google hittade vi oftast lösningar som sa att man skulle använda GSON istället för det inbyggda stödet.

I början av projektet bestämde vi att det var viktigt att träffas ofta för att få en bättre översikt över arbetsgången, vad som gjorts och vad som fortfarande behövdes göras så vi planerade in två stycken obligatoriska lunchmöten varje vecka. Att vi planerade in mötena under lunch var bra, vi lade vårt fokus på diskussion och satt inte och kodade under tiden. Hade vi några tekniska problem med exempelvis Git, eller om något problem dykt upp så diskuterade vi det på mötet och försökte lösa problemen efter mötet.

För att underlätta kommunikationen mellan gruppmedlemmarna och planeringen av projektet använde vi oss av Trello, Facebook, Google docs och Skype. Trello, som är en virtuell anslagstavla, underlättade för oss framförallt i början av projektet då vi i grova drag strukturerade åtta veckors arbete. Vi hade "Att Göra-listor", en kalender, listor med applikationens tänkta funktioner och liknande anslag. Vi skapade också en grupp på Facebook där vi lade upp dokument, relevanta länkar samt alla mejl som Marcus fick från tävlingsledningen. I den tillhörande chatten diskuterade vi projektet när vi jobbade på olika håll.

Tävlingsspecifika moment

Den första workshop som anordnades var konceptstugan, där vi jobbade med vårt koncept. Vi skickade två representanter från gruppen, då de andra inte kunde eller var sjuka. En tredje gruppmedlem var senare med via skype när vi satt och bollade idéer med vår tilldelade mentor.

Vi kände att informationen om konceptstugans syfte kunde ha förmedlats bättre, då deadlinen för första inlämningen till tävlingen var kl 13 dagen efter konceptstugan och vi redan hade kommit väldigt långt i inlämningsprocessen. Konceptfilmen var klar och konceptbeskrivningen samt realiseringsplanen behövde endast finslipas, och vi tänkte att vi under konceptstugan skulle få hjälp med det sista inför inlämningen, medan vår mentor hade uppfattningen att det var under själva konceptstugan vår idé skulle ta form och utvecklas till ett koncept. Vi fick lära oss många smarta knep för att ta fram idéer som vi definitivt kommer att ha nytta av i framtida projekt, men vi hade fått ut mycket mer av konceptstugan om den legat en vecka tidigare i planeringen.

Vår mentor tyckte vår idé var bra, och sa att han själv gärna skulle använda vår applikation. Däremot tyckte han inte att konceptet var särskilt innovativt, varför vi satte oss och funderade på funktioner som om de inkorporerades skulle göra applikationen mer nyskapande och unik. Han fick oss att se på vår idé utifrån, och även då detta gjorde oss osäkra på konceptet vi valt, blev vi också mer medvetna och funderade på utvecklingsmöjligheter kontinuerligt under processens gång, vilket i sig var både bra och dåligt, då det ibland tog tid från realiseringen av det vi redan bestämt.

Vi gick inte på några andra workshops, framförallt för att det krockade med våra scheman eller att vi inte hade tid. Nu i efterhand känns det som att detta varit värt att lägga tid på, framförallt "innovation platform" och "From concept to business". Vi hade genom att prata mer med branschfolk eller handledare kunnat utveckla vår idé vidare och bli säkrare på den.

Tidsfördelning

Redan i början av projektet bestämde vi oss för att dela upp de olika delarna av projektet mellan gruppmedlemmarna, detta för att vi insåg att det skulle bli det mest

effektiva arbetssättet. Effektivt för att alla har varsitt ansvarsområde att utveckla och arbeta på. Då var och en inte behöver förstå i detalj hur allt fungerar, slipper alla lägga tid på att fördjupa sig inom samtliga delar av projektet. Projektet delades upp mellan gruppmedlemmarna dels utifrån vad man var bra på och dels vad man hade intresse att lära sig. Vissa delar utfördes dock av hela gruppen, såsom möten och en hel del av det förberedande arbetet, exempelvis konceptutvecklingen och skissande.

Möten: 50 timmar totalt, genomsnitt 4 personer och 2h per möte i veckan

Förberedelse (skiss, konceptutveckling, brainstorming, research etc.): 200 timmar totalt, genomsnitt 40h/person fördelat på tre veckor

Backenden: 150 timmar totalt under hela projektet (två personer: 30h +120h)

Grafik (xml) : 86 timmar totalt under två veckor (tre personer: 30h+30h+26h)

Koppla samman/implementering: 180 timmar totalt under 4 veckor (3 personer 80h + 80h + 20h)

Rapport och annan dokumentation: 75 timmar totalt, (fem personer: 20h + 20h + 15h + 10h + 10h)

Tävlingspecifika moment: 43 timmar totalt, (fem personer: 18h + 10h + 6h + 5h + 4h)

Hela projektet: 784 timmar.

Reflektion

Grupparbetet

Vi hade ett mycket givande samarbete och bra kommunikation. Alla ville bidra med sin kunskap och sina tankar, både i fråga om kodens utformning, och de andra delarna av projektet såsom skisser och dokumentation. Överlag hade gruppen ett driv från första början, som höll i sig även i de motgångar som uppstod. Om vi hade använt oss mer av Scrum hade vi nog utnyttjat drivet bättre, vilket hade resulterat i att vi hade fått mer gjort på projektet.

Vi hade tjänat på att arbeta mer som grupp, genom att exempelvis ha en bestämd dag i veckan då vi satt tillsammans och programmerade. Även om vi arbetade med olika saker skulle vi ändå ha någon i närheten att ställa frågor till. Vi använde, som tidigare nämnts, Facebook för att kommunicera med varandra, och det fungerade bra. Problemet med det var dock att det kunde ta tid innan man fick svar på en fråga,

vilket ofta resulterade i att man redan hade löst problemet innan man fick svar. Därför är det oftast bättre att sitta tillsammans och arbeta.

Vad fungerade bra?

Ibland när man arbetar med projekt inträffar oförutsägbara händelser som exempelvis sjukdom. Detta tycker vi att vi hanterade bra genom att kalla samman till extrainsatta möten med de som kunde arbeta. Då planerade vi noggrant vad vi var tvungna att göra och delade upp arbetet mellan de närvarande. Vi förutsatte att personerna som inte kunde närvara inte heller skulle göra så mycket mer under en tid, utan utgick ifrån att de som var där var de enda som kunde arbeta. Detta fick oss att samla krafterna och organisera oss bättre. Gruppens öppna arbetsklimat där vi har kunnat säga vad vi tycker om lösningar och planering etc. har underlättat hanteringen av oväntade händelser samt arbetet i gruppen överlag. Grupsamarbetet har trots sjukdom fungerat väl då vi har kunnat nå varandra dygnet runt via Facebook.

Vad fungerade mindre bra?

Vi insåg efter ett par veckor att vi tagit oss vatten över huvudet. Vi räknade inte med att fyra av fem medlemmar skulle bli sjuka och inte kunna jobba så mycket under den tiden. Vi kunde ha valt att göra en del av vår applikation istället för att ta på oss så mycket arbete som vi gjorde. Då hade den specifika delen kunnat bli ännu bättre genomtänkt. Koden i sig hade kunnat bli mer fokuserad, effektiv och optimal. Det hade nog även gjort vår applikation mer attraktiv för innovationstävlingen. Vi funderade redan i början på att en reseplanerare kanske kunde bli alldeles för stor och komplicerad, och funderade ett tag på att göra något annat. Till slut var det ändå en reseplanerare vi bestämde oss för, eftersom hela gruppen hade idéer om hur den skulle fungera och se ut, och vi inte kunde släppa tanken. Vi prioriterade vår vilja och vårt engagemang över allt annat. En idé som alla är intresserade av att jobba med och som känns nyskapande kommer man långt med, och gör att man är villig att lägga ner mer av sin tid. Trots att projektet blev för stort för oss att hinna med, anser vi ändå att vi fick ihop mycket funktionalitet som gör att applikationen faktiskt går att använda.

Vi använde nog för mycket tid till att få själva sökfunktionen att fungera. Den blev absolut bra, men var kanske den minst innovativa delen av projektet. Att vi lade så mycket tid på API:er och backenden som vi gjorde var lärorikt, men i slutändan stödde applikationen mycket mer funktionalitet än vi hann implementera. Allt detta gick inte att visa upp på finaldagen, vilket var väldigt synd. Om vi hade hårdkodat mer hade vi kunnat fokusera på att få applikationens mer innovativa delar funktionella.

Vad kunde ha gjorts annorlunda?

Vi kunde ha ansträngt oss ytterligare för att komma på något mer innovativt att arbeta med, fast omgivningen och vi var mer intresserade av en reseplanerare. Tävlingen handlade ändå om innovation, och vi skulle nog ha lagt mer vikt vid det ordet.

Vi hade även kunnat se till att alla dokumenterade sin kod bättre, bland annat med JavaDoc. Det skulle ha underlättat mycket, då en hel del tid gick åt till att läsa och sätta sig in i någon annans kod.

Vi borde även ha varit hårdare med att följa Scrum-principer. De tillfällen då vi använde oss av det hjälpte oss verkligen. Det hade gett oss en bättre översikt och på så sätt hade vi fått bättre struktur på vår arbetsprocess.

I början av projektet skulle vi sett till att lägga in de grundläggande funktionerna först, exempelvis navigationen, så vi inte hade behövt göra halvbra lösningar på grund av tidsbrist.