



POST MORTEM RAPPORT

KATLA

Anton Freudenthaler, Joel Gustafsson, Mattias Nilsen, Lisa Lipkin,
Erik Norlander, Anna Nylander
Chalmer Tekniska Högskola – DAT255 HT2014

Innehållsförteckning

Introduktion av projektet.....	2
Metod och tillvägagångssätt.....	2
Scrum och arbetsprocess	2
Tekniska hjälpmedel.....	3
Tidsdistribuering.....	3
Processanalys.....	4
Scrum.....	4
Veckomöten	5
Daily Standup	5
Sprint planering.....	5
Workshop	6
Programmering	6
Tekniska hjälpmedel.....	6
Git	6
Waffle.io	6
Gruppdynamik	7
AGA.....	7
Slutsats	7

Introduktion av projektet

Projektet har gått ut på att producera en bättre sms-applikation för att förenkla skickandet av sms och reducera faran med detta för lastbilschaufförer. Studier har visat att den farligaste distraktionen för bilförare är sms:ande. Därför var målet med denna applikation, härnäst refererad till som OBAMA (On Board Auxiliary Messaging Application), att skapa en så säker miljö som möjligt för bilförare att skicka och ta emot sms. Genom att förbättra och uppmuntra säkrare funktioner och minimera interaktionen med telefonen håller vi förarens ögon och koncentration på vägen.

Huvudskillnaden mellan OBAMA och andra sms-applikationer är att det läggs mindre fokus på olika funktionaliteter, då säker användning är största prioritet. Andra sms-applikationer lägger väldigt mycket fokus på att trycka in så mycket funktionalitet och information på en skärm som möjligt. OBAMA strävar efter att förstärka de viktigaste funktionaliteterna för bilföraren och inte bombardera denna med information. Samtidigt ska effektiviteten bibehållas vid användning även utanför bilmiljön. Idén på utformningen av applikationen är dock baserad på en ytlig analys byggd mycket på antaganden, men p.g.a. den korta tidsramen så fanns inte tid till en djupare utredning av vad som behövts för att effektivare uppnå tidigare nämnda mål. OBAMA är en färdig produkt i enlighet med visionen även om mycket funktionalitet saknas.

Metod och tillvägagångssätt

Scrum och arbetsprocess

Arbetsprocessen har utgått ifrån Scrum, som är ett ramverk för hur man kan utveckla mjukvara genom en iterativ och agil process. I Scrum utgår arbetet från så kallade Sprints som är ett bestämt tidsintervall där varje gruppmedlem har en eller flera uppgifter att utföra, den vanligaste längden på dessa Sprints är mellan 7 och 30 dagar. Med tanke på det begränsade antalet veckor har varje Sprint varit ungefär en vecka lång. Varje Sprint utgår från ett planeringsmöte som hålls i början av varje Sprint. Under det mötet har arbetet för Sprinten delats upp mellan gruppmedlemmarna och möten samt workshops för veckan planerats. Utöver detta har ett möte med handledare legat på tisdagar, varefter ett möte hållits för att

kunna diskutera feedback från handledaren men även för att ha större diskussioner till exempel om hur en viss del av användargränssnittet ska se ut.

Nästa del i Scrum är så kallade Daily Standups, dessa är tänkt att utföras varje dag och ta väldigt kort tid, ungefär 15-20 minuter. Under dessa så ska alla svara på tre frågor: Vad har jag gjort? Har jag några problem? Vad ska jag göra? Dessa Standups har hållits 2-3 gånger per vecka och har varit ett sätt att kontrollera hur de andra gruppmedlemmarna ligger till och dessutom varit ett forum för att diskutera problem som uppkommit. Utöver de möten som ingått i Scrum processen så har det även hållits i workshops. Dessa har varit till för att se till så att alla medlemmar i gruppen förstår de externa biblioteken och de utvecklingsverktyg använts, som till exempel versionshanteringssystemet Git.

Tekniska hjälpmedel

För att underlätta arbetet har ett antal digitala verktyg utnyttjats, bland annat Github och Waffle.io. Github är en webbplats som ger användare en överblick över det arbetet som gjorts med hjälp av versionshanteringssystemet Git. För varje uppgift har en specifik branch i Git skapats där arbetet sker tills implementationen är färdig. Då sammanfogas denna med utvecklings-branchen som sedan vid utgivning/programsläpp sammanfogas med master-branchen.

Hemsidan <http://waffle.io> har varit till stor hjälp då det gäller att organisera gruppens Scrum Board. Waffle.io tillhandahåller en virtuell anslagstavla där alla notiser är direkt kopplade till issues på Github.com.

Google drive har använts för att synkronisera alla nödvändiga dokument och filer som till exempel mötesprotokoll mellan datorer och gruppmedlemmar.

Tidsdistribuering

Målet för gruppen har varit att lägga 20 timmar per vecka och person på projektet. Av dessa har cirka 4 varit på möten med gruppen och 4 varit på föreläsningar. Detta har lämnat 12 timmar i veckan per person att utföra de uppgifter som specificerats under mötena. Under olika veckor har gruppen hållit i workshops på allt från 2 till 11 timmar per vecka. Utöver detta

uppgger sig Erik ha lagt mellan 5 och 7 timmar per vecka, Mattias ha lagt mellan 5 och 10, Joel mellan 5 och 10, Lisa mellan 5 och 7, Anna mellan 5 och 7 och Anton 8.

Till att börja med lades de två första veckorna på att ta fram idén till den applikation som skulle göras. Här skrevs visionen och en första specifikation för applikationen togs fram. De två nästkommande veckorna spenderades på att förstå androidutveckling samt AGA. Där inräknat den tid som uppsättning av projektet med AGA samt Android tagit.

Fortsättningsvis har ungefär 20 timmar spenderats på att ta fram det grafiska gränssnittet till applikationen. Implementationen av gränssnittet har tagit cirka 30 timmar, där är inte bakomliggande funktionalitet och implementation medräknat. De resterande 250 timmarna har spenderats på implementation av funktionalitet i applikationen. Dessa timmar är svåra att bryta ner i mindre delar då ingen tidsrapportering har förts under projektets gång.

Processanalys

Scrum

Användandet av Scrum hade en del fördelar. Till att börja med gjorde det gruppen uppmärksam på behovet av att kontinuerligt uppdatera mål och utvärdera arbetet. Tack vare de olika momenten i Scrum var alla införstådda i det pågående arbetet, trots att bara de som jobbade aktivt med ett område hade full insikt i hur det fungerade.

En annan nackdel var att gruppen valde att inte ha någon fast Scrum master, då denna roll istället fick gå runt i gruppen. Detta ihop med för lite kunskap i Scrum har gjort att gruppen inte kunnat hantera de situationer och problem som uppstod. Vidare ledde detta till att projektet gled ifrån Scrum metodiken.

Att hela gruppen agerade produktägare medförde vissa problem. Det var svårt att hålla både rollen som utvecklare och kund samtidigt. Kunden behöver tänka på hur applikationen skall användas. Utvecklaren behöver istället tänka på hur kundens önskemål skall realiseras. Dessa två skilda tankesätt påverkar varandra starkt och gör arbetet ineffektivt.

Veckomöten

Veckomötena fungerade mycket bra. De var organiserade och gav alla en bra inblick i hur utvecklingen fortskred. De var också bra tillfällen att ta upp problem som berörde hela gruppen samt revidera planen för fortskridandet av projektet. På grund av tidsbegränsning blev mötena effektiva. En negativ aspekt med tidsbegränsningen var dock att det var ont om tid att diskutera och behandla de problem som uppkommit.

Att mötena endast hölls en gång i veckan gjorde att det tog för lång tid för problem att behandlas. Mycket diskussion fick skjutas upp och fick sällan en ordentlig uppföljning.

Daily Standup

Som tidigare nämnts användes inte Daily Standup dagligen. Den korta uppdateringen på hur det gick för alla gruppmedlemmar möjliggjorde för alla att ha insikt i hur projektet fortskred. Vid vissa tillfällen uppkom diskussioner och något långgrandiga förklaringar. Detta tog tid ifrån andra planerade aktiviteter.

Den spenderade tiden på projektet var inte konstant eller jämt fördelad över veckorna, samt att alla gruppmedlemmar inte hade samma möjlighet att arbeta under alla tillfällen. Avsaknaden av rutin i tiderna gjorde således att Daily Standups ibland blev onödiga. Det fanns inte heller någon rutin på hur de problem som gruppmedlemmarna tog upp skulle hanteras. En stor del i att det var svårt att hantera problemen berodde på att uppgifterna var såpass uppdelade, vilket resulterade i att varje person ofta fick lösa sina problem själv.

Sprint planering

Sprintplaneringarna möjliggjorde en kontinuerlig utvärdering av hur projektet fortskred samt tydliga tillfällen där när nya uppgifter och funktioner kunde läggas till. Det var även enkelt att fördela uppgifter inom gruppen, så att arbetsbördan inte blev ojämn.

Gruppen saknade kunskap om hur en väl fungerande Sprint skulle vara utformad och hur stora uppgifter var möjliga att slutföra under en Sprint. Detta resulterade i att det blev svårt att planera Sprints. Även den vaga gränsen mellan "userstory" och "task" försvårade uppgiften. Det krävdes hela tiden återhopp till tidigare Sprints då många uppgifter inte blev klara. Den

bristande kunskapen i Android gjorde att tidsuppskattningen för olika uppgifter ofta blev inkorrekt. Mycket tid gick åt till att förstå Android och problem med detta istället för att lösa de uppgifter som ingick i Sprinten.

Workshop

De workshops som gruppen hade var mycket givande då alla fick möjlighet att ta del av viktiga kunskaper samt gav tillfällen för att strukturerat driva projektet framåt i en form där diskussion var framstående. De problem som gruppen stötte på under workshops var till största del att arbetet blev ofokuserat och onödigt tidskonsumerande.

Programmering

Programmeringen under projektets gång hade två olika utformningar. Det ena var individuell programmering och det andra var programmering i samlad grupp. När gruppen möttes för att programmera hade man fördelarna av att man lätt kunde fatta beslut och hålla varandra uppdaterade. En annan fördel var att gruppens samlade kunskap fanns tillgänglig. Detta underlättade till viss del möjligheten att få hjälp inom gruppen. Tyvärr var arbetet så pass uppdelat, och den allmänna kunskapen om Android så pass knapp att det kunde vara svårt att på ett smidigt sätt sätta sig in i alla gruppmedlemmars olika problem. De flesta gruppmedlemmarna fick mer gjort och var mer effektiva när de satt självständigt med sina uppgifter.

Tekniska hjälpmedel

Git

Tack vare en tidig workshop i de grundläggande kommandona i Git har användandet av det fungerat mycket smidigt. Att jobba på separata brancher för separata uppgifter och delar i programmet minimerades konflikter som kan uppstå när flera personer redigerar samma fil. De konflikter som ändå uppstod var små och lätta att hantera.

Waffle.io

Användandet av Waffle.io gav en klar inblick i vad som skulle göras, vad som behandlades för tillfället och vad det var som var klart. Att man även såg precis vem som höll på med en viss uppgift var fördelaktigt. Det gav en god inblick i alla gruppmedlemmars arbete.

Gruppdynamik

Gruppdynamiken fungerade överlag bra, alla tog mötestid och arbetsuppgifter seriöst och det har inte uppstått några större dispyter eller problem med kommunikationen inom gruppen. Att gruppmedlemmarna kände varandra sedan tidigare bidrog till det goda klimatet i gruppen.

Dock hade det behövs en tydligare ledare, samt att arbetet inte delades upp lika hårt som det gjorde då det blev svårt att diskutera problem.

Det största problemet med gruppdynamiken har varit att gruppen har svårt att hålla sig koncentrerade och fokuserade vid workshops. Det har ofta hänt att det diskuterats saker som varit irrelevanta för tillfället istället för att jobba.

AGA

Problematiken, med att skriva ett sms samtidigt som det är säkert att köra, är svår. Därför gjordes en tydlig begränsning av applikationen som gjorde uppgiften mindre omfattande och lättare att lösa. Visionen hade med fördel kunnat innehålla användning av AGA i större utsträckning. Den är hårt kopplad till att köra lastbil men inte fullt så kopplad till AGA.

Slutsats

I ett framtida projekt hade gruppen utifrån dessa erfarenheter haft en liknande men något modifierad arbetsprocess. Här under följer en slutsats och utvärdering av de olika metoder och verktyg som använts.

Till att börja med har de digitala verktyg som använts fungerat bra och varit effektiva. Git har ökat gruppens produktivitet och effektivitet och är ett bra fungerande versionshanteringssystem. Waffle.io har även det fungerat smidigt och försett gruppen med en bra överblick över det pågående arbetet.

Fortsättningsvis har Scrummetodiken fungerat mindre önskvärt. Detta beror framförallt på de bristande förutsättningarna för metoden, bristande förståelse för Scrum, samt att Scrum blev introducerat sent i processen. Att Android sedan tidigare var okänt för gruppen gjorde det även svårt att få väldefinierade Sprints då det var svårt att veta hur lång tid olika uppgifter skulle ta

och vad de exakt behövde innehålla. De sprints som fanns i projektet var väldigt korta och med tanke på gruppens ovana att använda Scrum var det inte optimalt för projektet. Däremot har det varit mycket som fungerat bra med Scrum. Arbetet för varje Sprint fördelades på ett sätt som möjliggjorde att varje person kunde sitta individuellt och arbeta med sin samling uppgifter. Innan Scrum infördes var det svårt att redogöra för arbetet sedan senaste mötet och även komplicerat att behandla problem som uppstått. De tydliga riktlinjerna för Daily Standup gjorde att denna process förenklades.

Vidare var workshops något som stundvis fungerade bra och stundvis fungerade dåligt. De fungerade bra när specifika moment diskuterades under en workshop eller när gruppen hade ett gemensamt problem som behövde behandlas. Däremot fungerade de dåligt när en workshop endast gick ut på att sitta samlat och programmera. Gruppen har vid dessa tillfällen haft svårt att koncentrera sig. Dessa tillfällen hade med eventuell fördel kunnat bytas ut mot någon sorts parprogrammering. Detta hade gjort att två personer var insatta i varandras problem och varje problem skulle då på ett effektivare sätt lösas.

Utöver detta har gruppdynamiken fungerat bra. Det har inte uppstått några större dispyter som på något sätt stört arbetet. Det har rått en öppen diskussionsmiljö som gynnat projektet. De stora problemen med arbetet har inte berott på gruppen utan snarare på bristande kunskap om AGA, Android och Scrum vilka genererade tidsödslande problematik.

Slutligen hade gruppens tillvägagångssätt vid ett framtida projekt liknat det som använts i detta projekt. Med mer kunskap om Scrum och Android hade projektet flutit på bättre. Dessutom hade själva arbetet fungerat bättre med hjälp av parprogrammering och undvikande av onödiga gruppsammankomster. De digitala verktygen hade även de använts igen.