

# Post-mortem rapport

Åke - din resehjälper i vardagen

## Introduktion

Vi är gruppen El Bús och vi består av fem medlemmar: Tobias Hallberg, Thomas Högberg, Anton Josefsson, Henry Ottervad och Mikael Ragnhult. Vi har tillsammans utvecklat en Androidapp som heter Åke. Den här rapporten beskriver vår utvecklingsprocess, samt reflektioner kring utvecklandet.

# Ordlista

## **Scrum**

Utvecklingsprocess. Man har en *backlog* (lista med uppgifter i prioritetsordning). Uppgifterna har uppskattad *kostnad* (arbetsbörda). Utvecklingslaget har en uppskattad *velocity* (arbetshastighet). Välj uppgifter med kostnad som tillsammans är lika med velocity. Valda uppgifter jobbas på under *sprintar* (perioder av utveckling).

## **Trello**

Webbaserad projekthanteringsapplikation. Virtuellt anslagstavla med post-it lappar som representerar uppgifter. Dessa kan tilldelas användare.

## **REST**

Representational State Transfer är en modell för utveckling av mjukvara avsedd för online-tjänster. REST har den egenskapen att servern inte har någon kännedom om klientens tillstånd och vice versa utan svarar endast med data vid HTTP-anrop. Det är upp till klienten att välja hur denna datan sedan ska hanteras.

## **Slack**

Kommunikationsplattform speciellt designad för grupper.

## Arbetssätt och processer

Arbetet delades upp i olika uppgifter varje vecka med hjälp av Scrum. Olika uppgifter hade olika *velocity* vilket representerade hur stor arbetsbörda uppgiften förväntades kräva.

Uppgifterna sorterades upp efter prioritet och de som ansågs viktigast valdes först.

Vi använde Trello för att lägga upp detta arbetet och varje uppgift representerades där som ett kort. Olika kort tilldelades olika personer och när de ansågs färdiga flyttades de till en hög för klara kort. Vi parprogrammerade under stora uppgifter där vi ansåg att det behövdes och när vi skulle integrera stora komponenter.

Ungefär 15 timmar per person och vecka spenderades på projektet. Där räknas tid in för att lära sig Android-programmering, föreläsningar, möten samt programmerandet av själva applikationen.

Den samlade tiden uppskattar vi fördelad till följande:

Front-end:	15%
Back-end:	20%
Planering (möten):	20%
Buggfixande:	15%
Tävling:	15%
Dokumentation:	15%

Uppskattad tidsuppdelning för varje enskild gruppmedlem:

	Mikael	Tobias	Henry	Anton	Thomas
Front-end	5%	25%	20%	5%	20%
Back-end	35%	15%	10%	30%	40%
Planering	20%	20%	20%	20%	10%
Buggfixande	10%	10%	10%	15%	15%
Dokumentation	20%	20%	20%	15%	5%
Tävling	10%	10%	20%	15%	10%

## Scrum

Scrum hjälpte oss att organisera och planera veckorna på ett strukturerat sätt. Utan ett bestämt sätt att fördela arbetsuppgifter kändes det svårt att hålla reda på vem som skall göra vad utan att göra jobb i onödan. Genom att dela upp projektet i mindre delar får man dessutom en klarare bild av projektet både i helhet och i detalj.

Sättet vi använde Scrum på gjorde också att det var lätt att sätta kortsiktiga mål. I början så bestämde vi få saker som skulle göras till nästa vecka och bara det. Detta skapade problem i de fall vi hade överskattat tiden det skulle ta att göra färdigt en uppgift. Om någon i gruppen blev färdig efter halva tiden så hade personen inget att fortsätta med under resten av dagarna.

Vi åtgärdade detta problem genom att hålla en backlog med saker som skulle göras och att det var fritt fram att ta en ny uppgift ifall den redan tilldelade uppgiften blev färdig för tidigt. Då vi tog till oss nya uppgifter la vi till uppgiften i Trello (vår anslagstavla) så att inte två personer arbetade på uppgiften samtidigt.

Med Scrum hade vi ett långt (~2h) möte varje vecka där vi gick igenom vad som har gjorts, vad som skulle göras, samt framtida planering. Om vi hade tid/ork över efter mötet så arbetade vi tillsammans. Vi hade även ett dagligt kort obligatoriskt möte (*stand up meeting*) där var och en förklarade vad hen arbetade med och om hen stött på problem. Vi tyckte att det var ett bra sätt för att uppdatera alla i gruppen hur vi låg till i den aktuella sprinten, även om det innebar att varje gruppmedlem behövde passa en tid varje dag. Om gruppen inte hade kommit bra överens hade det eventuellt kunnat vara obehagligt att säga att man inte arbetat sen senaste mötet, men annars ser vi inga nackdelar till att ha dessa möten. Vi hade gärna använt dem i liknande projekt i framtiden. De dagar vi ej hade möjlighet att vara fysiskt närvarande vid ett sådant möte hölls det via Slack. Tiden vi spenderade i "möten" varje vecka var runt 2,5 timmar, resten spenderade vi på de specifika uppgifter vi var tilldelade. Detta tyckte vi var mycket tidseffektivt.

Sättet vi arbetade på fungerade mycket bra så länge en stor del av medlemmarna i gruppen kunde vara med på möte en gång i veckan och sedan också deltog i de dagliga mötena. Vi tror att det hade varit svårt för alla att vara med på mötena om gruppen hade varit större eftersom tiden för möte måste passa för alla.

I framtiden skulle vi kunna tänka oss att använda denna teknik även i andra projekt, om projektet tillåter ett så fritt upplägg som Scrum. Det kan till exempel finnas situationer då medlemmarna i förväg inte vet hur mycket tid de kan lägga på projektet. I så fall kan planeringen haverera om sprintar måste vara klara vid specifika tider för att projektet ska kunna fortgå. Även i projekt som är svåra att dela upp i distinkta deluppgifter eller flexibla projekt där backloggen måste förändras mitt i veckan tror vi inte att denna teknik passar in.

## Reflektioner

Genom att dela ut uppgifter till gruppmedlemmarna varje vecka så var det tydligt vem som höll på med vilken del av applikationen. Med hjälp av Trello kunde gruppmedlemmarna se vilken gruppmedlem som arbetade på vilken uppgift. Via Git kunde medlemmarna också se i vilka specifika delar av koden som var gruppmedlem arbetade på. Med hjälp av kommunikationsverktyget Slack var det enkelt att kommunicera med övriga gruppmedlemmar. Om någon i gruppen körde fast, fanns det alltid någon som snabbt kunde hjälpa.

Tidsplaneringen fungerade bra med tanke på att många i gruppen inte hade någon erfarenhet av Androidprogrammering sedan tidigare. Genom att redan tidigt gå igenom vårt mål med projektet, samt hur vi önskade att vår app skulle fungera och se ut, hade vi ett tydligt mål framför oss. Detta gjorde det enklare för oss de gånger vi utvecklade självständigt och var något vi återbesökte flera gånger under utvecklingsperioden allt eftersom att vår förståelse för problemdomänen ökade.

Vi har under utvecklingens gång även kommit bra överens, vilket inte bör underskattas. Vi tror att detta även hänger samman med att vi redan från början hade en tydlig bild av vårt koncept. Vi slapp således konflikter och missförstånd senare i utvecklingsprocessen. Genom Scrum så hade även varje gruppmedlem en tydlig uppgift med ett tydligt mål i varje sprint, vilket gjorde att gruppens medlemmar visste vad de själva, samt andra arbetade med. Varje person kunde därför koncentrera sig på sin egna specifika uppgift och var inte beroende av någon annan i gruppen (förutom under de få gånger två personer arbetade på samma uppgift).

I tidsberäkningen för uppgifterna så tog vi inte med att oförutsedda svårigheter kunde uppstå. Detta resulterade i att vissa uppgifter som var planlagda för en vecka, tog längre tid än så. Endast en gruppmedlem arbetade på API:et, och eftersom detta tog längre tid än beräknat så sköts utvecklingen mot back-end upp. Här kunde vi valt att två gruppmedlemmar skulle arbeta tillsammans mot API:et.

På grund av tidsbrist hann vi inte med så mycket testning av koden som vi hade önskat. Vi hade också planer på att var gruppmedlems kod skulle kollas igenom av en annan medlem, men på grund av tidsbrist så skedde inte alltid detta. Vi lät aldrig heller testa applikationen på den tänkta målgruppen. Detta betyder att applikationen kan innehålla moment som känns svåra eller krångliga.

Det fanns en hel del arbete som inte stod i backloggen, t.ex. tävlingsspecifika uppgifter och sammanfogande av koden. Saker som inte stod i backloggen hamnade då i stort sett i händerna på den som tog ansvar för det. Dessa uppgifter hade vi inte kollat så noga på eller tagit reda på ungefär hur lång tid de kunde ta. Det innebär att arbetsbördan vissa veckor inte var proportionerligt uppdelad mellan gruppmedlemmarna.

Delvis relaterat till ovanstående, ju längre in i projektet vi kom desto mindre noggranna var vi när det gällde att följa vår utvecklingsprocess. Detta var egentligen inte ett jättestort problem då alla hade samma ambitioner och en klar bild över vad som behövdes göras. Visst arbete, som buggfixande och små förändringar, fördelades utan tydlig kommunikation. Detta resulterade i att vissa gruppmedlemmar fick mer arbete än andra. Vid slutet mot utvecklingen var Scrum i princip fullständigt borta, vi höll dock fortfarande minst ett långt möte varje vecka.

## Reflektion över icke-processspecifika beslut

För att hantera vår backlog använde vi oss av Trello. Med hjälp av listor av kort för saker som skulle göras, saker som var klara och saker som var i arbete kunde var gruppmedlem

se vad övriga medlemmar arbetade på, samt se vad som var klart och vad som skulle göras. Vid osäkerhet av vad som skulle göras härnäst eller om någon i gruppen ville veta vem som gjorde vad så kunde alltså alltid den informationen snabbt hittas. Vid längre projekt med en stor grupp (som vi var i detta fall) så är ett sådant verktyg något vi kan rekommendera, och gärna använt i framtiden. Hade vi varit en mindre grupp och arbetat under kortare tid så hade det förmodligen varit överflödigt.

I Slack använde vi oss av olika kanaler för olika syften. Bland annat hade vi en kanal för endast ståupp-möten, och en kanal för kommunikation gällande utveckling. Det var väldigt praktiskt att veta var man skulle skriva, och om man behövde kolla längre bak i chattloggen efter en särskild ska, visste man vart man skulle kolla. Utöver detta var det en stor fördel att faktiskt använda en kommunikationsplattform som vi inte använder privat. Om vi hade använt Skype eller Facebook för kommunikation hade gränsen mellan arbete och privatliv blivit otydligare. Med Slack förväntade vi oss inte att folk alltid skulle vara tillgängliga, för i Slack så pratade vi om arbetet.

För att underlätta kommunikationen med REST-servrarna valdes det att använda ett bibliotek, Retrofit, för att på ett smidigt sätt kunna skicka förfrågningar och hantera svarsdatan. Det krävdes lite extra tid att lära sig hur detta bibliotek fungerade, men koden blev effektivare och lättare att utöka.

Vid tidplanering försökte vi alltid räkna med att varje uppgift skulle ta längre tid än vad vi själva trodde, just för att gruppmedlemmarna även skulle ha tid att lära sig och öva på programmeringen mot Android. I flera fall så ansågs uppgiften till en början väldigt enkel, men kunde visa sig vara svårare när gruppmedlemmen skulle göra utföra den i Android.

Att ha möjlighet att testa sin applikationen på en fysisk telefon märkte vi var en stor fördel gentemot att endast köra applikationen i en *emulator* (mjukvara gjord för att efterlikna annan mjukvara, i detta fall Android). Den inbyggda emulatorn var seg och tog lång tid att starta. Utan en fysisk telefon att testköra appen på hade det varit svårt att säkerställa hög kvalitet av appen. Viss funktionalitet var näst intill omöjlig att testa via emulatorn, tex GPS. Här krävdes det en telefon för att veta att GPSen faktiskt hittade rätt position.

## Samarbete inom gruppen

Det mesta av arbetet skedde individuellt och kommunikationen mellan medlemmarna skedde främst över Slack. Dock ansåg vi det viktigt att ha minst ett riktigt möte varje vecka för att planera nya sprintar, avstämma vad som avklarats under den gångna veckan och sammanfoga det klara arbetet.

I ett framtida men liknande projekt hade vi planerat bättre innan vi börjat utveckla. Det hade varit bättre om vi skapat en lång backlog från början. Detta skulle även kunna bidra till att tidsbehovet uppskattas lättare. Något som blev lidande i detta projekt var utförlig testning både av kod och av appen i sin helhet.

## Tävlingen

Eftersom kursen började ganska långt innan tävlingen hade vi väldigt tidigt ett klart koncept över vad vi ville utveckla, så vi prioriterade konceptstugan ganska lågt. Konceptstugan var dessutom lagd dagen innan sista dag för inlämning av tävlingsbidrag och därför kände vi inte att vi kunde vänta med vårt koncept till denna dag. Eftersom vi inte var säkra på om konceptstugan skulle vara meningsfull eller ej så väntade vi ändå med att börja utveckla, något som visade sig vara fel val.

Bara tre av oss gick på konceptstugan och vi som var där åkte hem under lunchen, vi hade ju redan vårt koncept. Feedback från konceptstugan var inte av någon större hjälp. Ett av förslagen därifrån var bland annat att inte göra en app. Efter den konstiga feedbacken vi fick under konceptstugan nedprioriterade vi utvecklingsverkstäderna rejält. Vi valde helt enkelt att fokusera på att göra vårt koncept till verklighet i stället. Trots detta upplevde vi ändå att tävlingen tog mycket tid från kursen. Vi tror att utan tävlingen så hade vi haft mer tid för att säkra kvalitén av vår app.

Överlag kändes tävlingen dåligt kombinerad med denna kurs. Tävlingen, som egentligen inte hade särskilt mycket med mjukvaru- eller apputveckling att göra, kändes slumpmässigt utvald. Att tidschemat för tävlingen inte gick ihop med kursens schema skapade för oss förvirring och stress.

Den gången vi faktiskt gick på en utvecklingsverkstad lät vi varje gruppmedlem bestämma själv om han ville gå på evenemanget. Vi lät det alltså vara frivilligt för varje gruppmedlem att gå på evenemangen, och kände inget behov av att ha en lagrepresentant på plats. Vi hade troligtvis tjänat på att ha representanter på varje utvecklingsverkstad, för att få en större inblick i vad det faktiskt var som förväntades som slutprodukt, men vårt intryck av tävlingen gav känslan av att det inte skulle vara särskilt givande.

De gånger vi fick relevant feedback på vårt koncept delades detta omgående med övriga medlemmar.

El Bús

Tobias Hallberg

Thomas Högberg

Anton Josefsson

Henry Ottervad

Mikael Ragnhult