

Reflektion

Sammanfattning

Att arbeta i projekt kan vara en utmaning på flera sätt. Ibland kommer man inte överens inom projektgruppen, ibland så saknas kompetens och styrning och ibland så är det riktigt roligt trots att man inte är överens och att man saknar kompetens. Detta projekt som vi nu genomfört har varit både roligt, frustrerande och lärorikt.

För de som saknat förkunskaper inom Android och Git så har det varit tillfälligtvis stor frustration över att informationen över hur och vad som bör göras till en början är mycket svårtillgänglig.

För de med förkunskaper har det varit en utmaning att på bästa sätt försöka informera om de nya systemen på ett pedagogiskt och korrekt sätt och samtidigt koordinera arbetet så att de olika delarna av applikationen kan passa ihop på ett bra sätt.

På det hela så har alla i gruppen kunna hitta utmaningar och lärdomar i projektet vilket är det man vill uppnå med sin skoltid. Frågan är dock om det man lärt sig är det som står i kursplanen?

Förkunskaper

I gruppen hade vi relativt stora skillnader i kunskap kring de områden som kursen berör. Alla har åtminstone 15hp Java i ryggen och även dryga 15hp i C-programmering och algoritmer och datastrukturer. Det var dock endast en person som hanterat Git och Android sedan tidigare. Man kan fråga sig om det verkligen är nödvändigt att introducera två helt nya system till studenterna. Båda systemen är mycket viktiga och givande för en student att kunna och ha arbetat med men eftersom introduktionen till dessa system var allt annat än komplett så har inläringen av systemen bidragit med mer besvär än nytta för att uppnå kursmålen.

Git

Eftersom att 4 av 5 saknade kunskap om Git sedan tidigare så var det en mycket stor utmaning att komma igång med arbetet med versionshanteringen. Att skilja på add/remove/commit/push/pull/fetch/merge/branch/tag/label är mycket svårt om man aldrig jobbat med versionshantering tidigare. Att 3 av 5 i gruppen dessutom aldrig jobbat i terminalen eller kommandoprompten gjorde att inläringen av Git blev ännu svårare.

När väl Git-kunskapen började sjunka in så började det så småningom dyka upp Git-konflikter, både lätta och svåra. Att lära sig hantera Git och lösa de problem som uppstått har tagit oerhört mycket av projektiden i anspråk.

Versionshantering som är en fundamental del av arbetet i en projektgrupp hör absolut hemma i en sådan här kurs men för att studenterna ska få en ärlig chans att lära sig hantera systemet så borde man delat upp föreläsningen om Git i två delar, en enkel och en svårare. Man borde även ha infört en laboration i Git för alla studenter där de med tydliga och pedagogiska uppgifter skulle kunnat genomföra och lära sig de mest grundläggande delarna.

Android

Samtidigt som 4 av 5 fick slåss med Git så skulle samma 4 personer samtidigt lära sig programmera för Android. Att programmera för ett mobilt system är mycket speciellt och har man inte erfarenhet kring det sen tidigare så är inlärningsströskeln mycket hög. Android är ett mycket komplext system med ett helt annat API i stora delar än vad Java har. Att sätta sig in i hur Android fungerar har även det varit mycket frustrerande för de som sedan tidigare saknade kunskap om hur systemet är uppbyggt och vilka klasser i API't som är de vanligaste.

Efter några långa veckor av plågsam inläring om hur systemet fungerar så går det naturligtvis bättre även om det fortfarande dyker upp relativt enkla problem som är svåra att hitta lösningar till om man inte vet precis hur systemet och API't fungerar. Dokumentationen är bitvis mycket bristfällig och många exempel innehåller direkta fel.

Man kan ifrågasätta nödvändigheten av att bygga ett Androidprojekt i en kurs där man enligt kursmålen skall lära sig om;

- hur mjukvarutveckling går till
- hur tester och krav används
- hur man arbetar med scrum/vattenfall eller andra projektmetodologier och verktyg.

Att sätta sig in i Android API't och programmera för ett specifikt system som de flesta i gruppen aldrig använt känns mer hemma i en programmeringskurs, t.ex. "Androidutveckling". Tyvärr så har inläringen av Android hindrat inläring av de kursmål som nämns på kurshemsidan.

Scrum

Att arbeta i Scrum var nytt för alla utom en som stött på det i liten skala sedan tidigare. Tyvärr så var det väldigt svårt att få arbetet att flyta på ett bra sätt eftersom att det gick åt mycket tid till att lära sig hantera Git och Android. Det ledde också till att det var i princip omöjligt att göra rimliga uppskattningar under planeringspokern. Scrum har många bra poänger i hur man bör arbeta som utvecklarteam och det har varit mycket nyttigt att få denna erfarenhet. Sprintar på 1 vecka där man bara kan jobba 20-30 timmar per vecka (där en betydande del av den tiden tas i anspråk av möten, handledning och föreläsningar) blir svårt att hantera. Framförallt eftersom att de två första veckorna var bortkastade på föreläsningar och annan information som vi haft mycket liten nytta av i arbetet. När man sedan kunde komma igång med att planera sprintar så blev sprintarna mycket tunna eftersom det gick åt så mycket tid till att lära sig Android.

Projektplanering

I vår grupp hade vi planerat att ha en färdig applikation en vecka innan inlämning. Sista veckan skulle då kunna ägnas till dokumentation, finslipning och eventuella bugfixar. Tyvärr slogs planeringen i bitar när vi 10 dagar innan inlämning fick veta att betygskriterierna ändrats, 3 dagar innan inlämning så skickades det ut ett mail om att vi skulle skapa ett "artefaktdokument" där vi skulle poängsätta varandras insatser och på något sätt markera kod efter person. Dessa ändringar av betygskriterier så nära inlämningen är varken schysst eller seriös. Vi hade planerat vårt projekt noga efter de kriterier som satts upp och hela den planeringen gick åt skogen med en sådan sen ändring. Det var inte bra.

Arbetsgång

När man jobbar i Scrum och alla jobbar på varsitt håll utan att ha tillgång till gemensamma ytor att arbeta på så uppstår många problem. Att det inte har varit möjligt att använda skolans datorer för projektet har verkligen ställt till det. Det har lett till att alla sitter vid sina datorer hemma och programmerar och försöker kommunicera så gott det går via Skype. Tyvärr så tar det lång tid och det är svårt att kommunicera på ett bra sätt med textkommunikation så därför uppstår det många missförstånd som leder till mycket onödigt arbete.

Det är också svårt att lösa gemensamma problem som har med design och modellering att göra när man inte kan sitta tillsammans. Vi har märkt av detta, framförallt i klassen `RouteActivity` som växt åt flera håll samtidigt och som vi har fått refaktorera ett flertal gånger vilket kostade oss mycket tid. En insikt som vi fått är att man kanske bör betrakta varje sprint som ett litet vattenfallsprojekt där man i början på varje sprint gör en design och modell för det man skall skapa under sprinten. Med tanke på de korta sprintar och den begränsade tid vi haft på oss så har det dock varit svårt men det är absolut något vi tar med oss in i framtiden. Nu när vi har bättre koll på Git och Android skulle det vara oändligt mycket enklare att genomföra ett scrum-projekt där vi faktiskt kan jobba på ett bra sätt tillsammans.

Utveckling

Under utvecklingen av vår applikation så har vi noga försökt följa de kriterier som sattes upp i början på kursen. Det har gett oss en bra bild av vad som behöver göras. När vi känt att vi legat bra till med kriterierna så har vi undersökt andra saker som skulle varit relevanta för vår applikation om vi haft tid att implementera dem.

Vi har bland annat kikat på Google App Engine och Google Cloud SQL för att undersöka möjligheten att implementera en synkronisering av applikationsdata över flera olika enheter. Efter en hel del undersökning och läsande kunde vi dock konstatera att vi inte kommer att hinna implementera en sådan lösning och att det dessutom kostar pengar att använda Google Cloud SQL.

Vi kikade då lite på möjligheten att använda Google Drive och/eller Dropbox för synkronisering av data. Tyvärr så har Google Drive API't för Android inte släppts ännu utan man måste fortfarande använda det gamla Google Docs API't som inte är helt enkelt att implementera. Vi har tyvärr inte hunnit färdigställa Dropbox integrationen men vi har experimenterat lite med det och vi har skrivit en klass som kan hantera Dropbox kommunikation. Det som återstår för att få en fungerande synkronisering är dock att vi behöver hämta all relevant data från databasen, skriva den till en fil som vi sedan laddar upp i Dropbox. Vi skulle även behöva ladda upp musiken från användarens enhet till Dropbox för att kunna synkronisera den. Man kan även tänka sig en inställningsskärm där användaren kan välja vad denna skulle vilja synkronisera så man inte behöver synka allt.

Allt eftersom vi vant oss vid att hantera Git så har vi mer och mer försökt få ett bra spår, så kallad "traceability", mellan alla våra bugfixar och branches. Tyvärr så fick vi inte stänga branches för att de som betygsätter skulle kunna se hur många branches vi haft och hur vi jobbat med dem. Hade vi fått stänga dem så hade GitHub självt kunna stänga den berörda issuen och lägga in en koppling mellan issue och branch eftersom de skulle vara döpta med samma namn. Denna koppling finns även mellan commits och issue-nummer, vilket vi försummade att utnyttja under större delen av projektet men började använda mot slutet. Issue-nummer #123 och #124 är demonstrationer på hur vi i fortsättningen kommer att jobba med Git och GitHub för att koppla commits till issues där det behövs.

Förändringar av kursen

Det finns några förändringar vi skulle föreslå som skulle förbättra denna kurs till nästa år. De är som följer:

- Använd 2-3 första veckorna till föreläsningar och laborationer för studenterna där de kan lära sig hantera de verktyg som de behöver för att genomföra projektet. Git, Android, Tester, Jenkins, Ant m.m. är exempel på sådant som har varit mycket förvirrande i onödan.
- Efter den första inlärningsperioden så kan projekten börja. Studenterna bör då ha en klar idé om vad de vill göra så de kan börja programmera direkt. Detta ger 5 veckor till utveckling utan föreläsningar så att studenterna kan fokusera på att genomföra projektet.
- Betygskriterierna måste vara tydligare och satta i sten när projektet börjar. Det dokument som vi haft att arbeta efter har varit fullständigt obegripligt till stora delar. De sena ändringarna av kriterierna är heller inte bra.