Chalmers Tekniska Högskola

“Post Mortem”-rapport

Grupp 5

Martin Calleberg
Vidar Eriksson
Linus Hagvall
Martin Hermansson
Jonatan Magnusson
Lucas Persson

Läsperiod 1, höstterminen 2013
Software Engineering Project, DAT255

Innehållsförteckning

1. Vilka arbetsprocesser och metoder användes i projektet?

Vi har använt oss av metoder liknande SCRUM, stå-upp-möten och parprogrammering. Dessa har vi dock anpassat till hur vi känner att de passar in på vårt arbetssätt, som är att sitta tillsammans hela gruppen och programmera. Vi använde grunderna i SCRUM men hoppade över användandet av “storypoints”, då det enligt oss var svårt att skapa bra user-stories för ett spel. Innan man kan börja färdigställa user-stories behöver man en grundläggande spellogik, vilket inte ger några storypoints.

Stå-upp-mötena skedde kontinuerligt under arbetsprocessen och även om vi ofta blev sittande under mötena så kom vi ändå igenom mötena i någorlunda god tid. Vi delade upp oss i par och varje par jobbade med ett område, paret hjälpte sedan varandra vid behov men det var inte ren parprogrammering.

2. Ungefär hur mycket tid spenderades (både totalt och av varje gruppmedlem) på de olika stegen/aktiviteterna och på projektet i helhet?

Då vi tänkte arbeta mest tillsammans på campus så schemalade vi ett antal tillfällen per vecka. Den schemalagda tiden var ca 23 timmar per vecka, något mindre i början, och vi jobbade sällan mycket hemifrån. Dock var inte alla med på alla arbetstillfällen, och därför är detta den ungefärliga tiden varje person lagt på projektet:

- Martin Calleberg: ca 140 h
- Vidar Eriksson: ca 100 h
- Linus Hagvall: ca 140 h
- Martin Hermansson: ca 140 h
- Jonatan Magnusson: ca 100 h
- Lucas Persson: ca 140 h

Totalt: 760 h

Den första en och en halv veckan av projektet arbetade vi endast med specifikationen och GUI. Alla projektmedlemmar gjorde sig bekanta med Android och utvecklingsmiljön. Därför var alla ganska förberedda när vi de efterkommande veckorna började programmera. Buggletande skedde till viss del kontinuerligt genom hela projektet, den sista veckan (LV 7) var dock helt dedikerad till buggar.

3. Arbetstekniker

Vi har till olika utsträckningar använt oss av de olika metoderna parprogrammering, SCRUM och dess stå-upp-möten. Dock har vi inte använt oss av rena versioner av dessa utan till stor del fritt modifierat dem för att de på ett bättre sätt ska gå att applicera på det projekt vi arbetat med.

3.1 Parprogrammering

Då projektet gick lätt att dela upp i tre större delar (modell, server, vy) gjorde vi som så att vi satte två personer på varje del. Därför skapades goda förutsättningar till parprogrammering eftersom det alltid fanns två personer som var insatta i en viss del av koden. Dock var det sällan vi gjorde ren parprogrammering då en skrev och en stod bakom, men det förekom om paret skulle implementera någon större feature inom sin del.

3.1.1 Vad var fördelarna baserat på erfarenheter från denna uppgift?

Ett exempel på när det var fördelaktigt att använda sig av parprogrammering var när olika delar av programmet skulle sättas samman i en helhet. Då behövde man kunskaper från båda personer som skrivit två olika delar eftersom en person var mer insatt i en del och en annan person i en annan del. Då skrev en koden medan den andre kontrollerade och berättade vad som skulle skrivas eller rättas till.

3.1.2 Vad var nackdelarna baserat på erfarenheter från denna uppgift?

Det går långsammare att parprogrammera om det är ganska simpel eller lätt kod att skriva. Detta eftersom det är mindre risk att göra fel och man då inte behöver ha två som kontrollerar koden. På grund av detta användes inte parprogrammering under sådana omständigheter där koden var någorlunda simpel.

3.1.3 Hur effektiv var tekniken i förhållande till hur lång tid den tog att använda?

Då tekniken enbart användes då något större eller mer komplext skulle implementeras så har vi inte en hel bild av hur det är att arbeta helt med parprogrammering. Dock tycker vi att den var väldigt effektiv i de fall den användes och många buggar har säkerligen blivit förhindrade tack vare denna metod.

3.1.4 I vilka situationer skulle ni använda den här tekniken i ett framtida projekt?

Om det som ska skrivas är komplext eller inte får bli fel är det lämpligt att använda sig av parprogrammering. I situationer där olika delar av ett program ska vävas samman kan det även vara tidseffektivt att hela tiden ha båda delarnas kompetens på plats för att ha snabb tillgång till så mycket kunskap som möjligt.

3.1.5 I vilka situationer skulle ni inte använda den här tekniken i ett framtida projekt?

Som ovan nämnt känns tekniken endast användbar vid mer komplexa fall. Därför skulle vi inte vilja använda metoden då en stor mängd relativt simpel kod ska skrivas. Dessutom om det behövs skrivas mycket dokumentation, såsom javadoc, kan det vara överflödigt att sitta två personer. I sådana lägen kan det vara en nackdel då, till exempel, en person märker ord.

3.1.6 Om ni hade använt metoden/tekniken i en del av projektet och inte helaprojektet, hur var det då att använda det jämfört med att inte använda det?

Parprogrammering användes främst under specifika förhållanden, då koden var komplex. Dock var det väldigt stor skillnad på det som skulle implementeras gång till gång då vi använde parprogrammering. Detta har gjort att det för oss varit svårt att jämföra dessa gånger mot de gånger vi ej använt metoden. En faktor inom parprogrammering som varit positiv i alla lägen den har använts är att man har haft mindre chans att göra fel då det varit lättare att dubbelkolla med sin partner.

3.2 Modifierad SCRUM

Som nämnt tidigare har vi använt oss av en modifierad version av SCRUM. Med “modifierad” menas bland annat att vi inte har haft någon extern kund och därmed inte någon enskild “Product Owner” som kommunicerat med denna. Istället har vi som grupp satt upp krav under projektets gång och valt att prioritera vissa saker högre vid vissa tillfällen. Vi har haft planeringsmöten varje vecka där vi evaluerat föregående sprint och även planerat vad vi vill ha gjort under veckan som legat framför. Till en början följdes detta väl, men då vi i slutet av projektet fick sätta nya krav under näst sista veckan av olika omständigheter handlade det mest om att bli färdiga. Då vi alla varit någon typ av “Product Owner” har detta fungerat och beslut har lätt kunnat tas i kritiska situationer.

3.2.1 Vad var fördelarna baserat på erfarenheter från denna uppgift?

Att jobba i sprinter ger en bra välstrukturerad planering där det blir lättare att se hur man ligger till tidsmässigt. Dagliga SCRUM-möten ger en snabb och bra överblick över hur utvecklingen ligger till i förhållande till planeringen.

3.2.2 Vad var nackdelarna baserat på erfarenheter från denna uppgift?

Vi upplevde att det var svårt att applicera SCRUM på ett spel eftersom det kan vara svårt att bryta ner ett spel i väldefinierade user stories. Som tidigare nämnt anser vi att det ofta blir svårt att färdigställa enskilda användarfall tidigt i ett spel då mycket underliggande spellogik krävs innan grundläggande sådana kan implementeras.

3.2.3 Hur effektiv var tekniken i förhållande till hur lång tid den tog att använda?

Om SCRUM hade använts fullt ut tror vi att metoden hade varit förhållandevis ineffektiv i vårt fall. Dock slutade vi använda allt inom SCRUM då vi snabbt insåg att det inte passade in på vårt projekt. Vår modifierade SCRUM-variant kändes däremot effektiv.

3.2.4 I vilka situationer skulle ni använda den här tekniken i ett framtida projekt?

Fullständig SCRUM skulle vara mer användbart i projekt där de olika projektmedlemmarna arbetade på olika platser. Dessutom vore det mer användbart om man arbetade gentemot en kund. Då hade en person kunnat vara mer ansvarig när det gäller kontakt med denna samt för kravsättning. Detta hade krävt att man skulle ha haft mer bestämda roller i projektet. I detta projekt har alla deltagit i själva utvecklandet och det vore orättvist om en person enbart hade sagt vad de andra skulle göra och inte gjort något själv.

3.2.5 I vilka situationer skulle ni inte använda den här tekniken i ett framtida projekt?

Om man skulle göra ett projekt på fritiden utan tidspress, eller ett liknande skolrelaterat projekt, skulle vi antagligen avstå från att använda SCRUM i full utstäckning. Detta med tanke på erfarenheterna vi erhållit om att det kommer kännas mer tillgjort än vad det är användbart. SCRUM som metod passar bättre då inte alla projektmedlemmar arbetar i samma rum alternativt om projektet är mycket större.

3.2.6 Om ni hade använt metoden/tekniken i en del av projektet och inte hela projektet, hur var det då att använda det jämfört med att inte använda det?

Precis som tidigare fastställts så använde vi SCRUM mest i början av projektet. Senare var det i huvudsak uppdelningen i olika sprintar (d.v.s. att utveckla iterativt) som var kvar. Den senare versionen av SCRUM kändes bättre anpassad till projektet och var därför mer lönsam att använda.

3.2.7 Stå-upp-möten

Stå-upp-möten har varit effektiva för att snabbt få en överblick över de senaste förändringar inom projektet men har stundtals blivit överflödiga då nästan all utveckling har skett gemensamt och mycket berättas och/eller frågas om när det kommer upp snarare än på nästa möte.

4. Vad fungerade bra med hur ni jobbade i projektet

Applikationen har en naturlig uppdelning i tre delar, vy, modell och server. Dessa delar skall programmeras så fristående som möjligt från varandra. Under arbetsprocessen avsattes två personer till varje område, en uppdelning som delvis fungerade väldigt bra. Det innebar att vi i stor utsträckning undvek problem där en person blev en flaskhals. Det har också gjort det väldigt enkelt att programmera fristående delar då en enskild person i de flesta fall inte vet hur andra delar av applikationen ser ut förutom de bestämda kommunikationsmetoderna. Dock var detta också något negativt, vilket vi återkommer till under punkt 5 nedan.

Som tidigare nämnt har nästan allt arbete skett gemensamt så det har alltid funnits hjälp till den som behöver. Detta var framförallt väldigt användbart när det handlade om att förstå hur andra delar av applikationen var skrivna.

5. Vad fungerade dåligt med hur ni jobbade i projektet

Vi var dåliga på att kontrollera varför inte alla sprintmål uppfylldes, vilket innebar att befintliga problem blev liggande för länge innan de togs tag i. Vi var dåliga på att trycka på om någon gruppmedlem inte lyckades uppfylla sina mål för sprinten vilket gjorde att vissa features blev liggandes över flera sprintar.

Ett problem som kunde uppstå på grund av vår uppdelning är att det blir oerhört begränsad insyn i ett område förutom för de två ansvariga personerna. Detta har gjort att ingen i projektet har haft särskilt god uppfattning över hur alla delar ligger till jämfört med tidsplanen. Detta möjliggjorde att ett område kunde hamna långt efter innan resten av gruppen fick reda på det.

6. Reflektion över val som inte haft med arbetsprocessen att göra

Till en början hade vi svårt att veta vad vi ville utveckla för applikation. Vi hade tankar om att göra en bluetooth-walkie talkie eller ett surround system där man parar ihop flera telefoner via nätverk och sedan kan få "hemma bio"-känslan lite mer portabel alternativt att göra en applikation för att göra det lättare att spela "Assassin"[1]. Vi insåg dock att det första av förslagen var en ytterst onödig produkt och det nästkommande svårt att implementera med tanke på den fördröjning som lätt uppstår då man skickar saker över nätverk samt att olika enheter har olika ljudkort som därmed spelar upp ljud med olika fördröjningar. Detta gjorde att det skulle bli extremt svårt att synkronisera ljudet på ett bra sätt och få det att låta bra. Dessutom insåg vi att den sista applikationen gav för lite fördelar gentemot att spela på traditionellt sätt.

Det vi tillslut bestämde vi oss för var att skapa en digital version av ett sällskapsspel. Vi hade många spel som förslag, men tillslut landade vi i att RoboRally var ett spel man skulle kunna ha som mall för utvecklandet. Det kändes bra att ha någonstans att "fråga" när man undrade över regler osv. till spelet. Det finns dock några saker vi valt att inte följa när det gäller reglerna i RoboRally. Exempelvis; i sällskapsspelet ska första spelaren som lagt sina kort vända på ett timglas. När den tiden runnit ut slumpas resterande kort förutom de kort de övriga spelarna redan hunnit lägga. I vårt fall är det svårt att synkronisera sådant över nätverk då man kanske inte spelar samtidigt. Därför valde vi att starta "timglaset"/korttimern i samma ögonblick som en spelare trycker på "Draw cards" i spelvyn.

Det tog lång tid innan vi insåg med säkerhet att det inte skulle bli någon server färdig. Det var för kort tid kvar att både kunna färdigställa servern och även kunna buggtesta allt tillräckligt för att vara säkra på att allt skulle fungera som tänkt i tid. Valet stod mellan att skapa en mindre avancerad AI eller en "dummy server". Det beslutades att försöka få ihop en AI då ingen av oss hade någon erfarenhet av servrar sedan tidigare och tiden var väldigt begränsad. Även om en "dummy server" hade varit närmare målet för projektet så ville vi vara säkra på att ha något att visa upp i slutet av kursen.

Det ska också sägas att både vyn och modellen alltid har utvecklats med målet att vara uppdelade på en server och en telefon. All kommunikation mellan dem är anpassad så att det ska vara så lätt som möjligt att lyfta ut modellen till en server.

[1] http://en.wikipedia.org/wiki/Assassin_(game)

7. Hur fungerade grupparbetet i projektet? Vad fungerade och vad fungerade inte?

Grupparbetet fungerade till en början mycket bra, framförallt när krav skulle fastställas. Allt eftersom projektet fortskred har det delvis blivit sämre på grund av att vissa gruppmedlemmar har uteblivit under de möten och arbetstillfällen vi schemalagt. Grupparbetet mellan övriga gruppmedlemmar på dessa möten och arbetstillfällen har dock fungerat bra, men det har varit svårt att kontrollera vad de uteblivande medlemmarna gjort under samma tid.

Det har varit en bra och tydlig uppdelning av arbetsuppgifter där det oftast har funnits två personer med insyn i varje område vilket har gjort det lättare att undvika eventuella flaskhalsar som kan uppstå om en person ensam sitter på kunskap om ett kritiskt område. På grund av denna uppdelning har det också varit lätt att få en svag koppling mellan de olika delarna.

Något annat som fungerat bra är att det funnits motivation inom gruppen när vi setts. När vi satt oss tillsammans för att arbeta har vi sällan prokrastinerat, vilket lett till att arbetet under arbetstillfällena har varit effektivt. Man har därmed också vid problem lätt kunnat fråga någon annan om hjälp angående en viss del av programmet istället för att på egen hand leta igenom koden efter relaterade kodbitar.

8. Vad skulle ni göra annorlunda i framtida, liknande projekt

Vi skulle vara mer noggranna med att se hur långt olika delar i projektet faktiskt har kommit i ett tidigare stadium. På så sätt hade man smidigare kunnat ändra applikationens slutmål och/eller fördela om arbetsbördan.

Vi skulle också specificera kommunikationsprotokollet för data skickad mellan server och klient tidigare istället för att göra lite åt gången. Det blir då lättare att veta vad man ska implementera och lättare kunna skriva ett mer generellt system som klarar av att lägga till nya funktioner ännu bättre än vad det system vi har nu klarar av. Exempelvis var vi i början av projektet tvungna att på båda sidorna av kommunikationen implementera samma funktion ungefär samtidigt för att det inte skulle krascha. Detta löste vi dock senare i projektet, även om det helt hade kunnat undvikas om vi, som tidigare sagt, hade specificerat protokollet i början av projektet.