



# Project Wood

---

## Post Mortem Report

2013-10-27

Grupp 10

Andreas Trakossas  
Gustav Simonsson  
Henrik Merlander  
Lucas Wiman  
Markus Johansson  
Patrik Larsson



## Innehållsförteckning

	sid
1. Tillvägagångssätt .....	2
2. Spenderad tid .....	3
3. Samarbetet .....	3

## 1. Tillvägagångssätt

### Scrum

Scrum gav oss ett enkelt sätt att planera och arbeta med veckobaserade versioner av vårt program. Det gjorde det lätt att se vad för något som behövde göras och när det behövde göras. Vi har inte hittat några egentliga nackdelar med Scrum, förutom kanske att vi inte kunnat följa vissa delar fullt ut då vi inte haft endast detta projektet och gruppmedlemmarna har haft olika scheman att passa.

Scrum är en effektiv metod som ger utdelning för tiden den tar att sätta sig in i. Vi hade använt Scrum i liknande projekt men där kanske projektet var det enda som gruppen jobbade på samtidigt, till exempel på ett företag. Vi hade inte använt tekniken om projektet var ännu mindre och gruppen bestod av färre medlemmar.

### Sprintplanering

Sprintplanering var en stor fördel i vår process, det gjorde att vi visste exakt vad som skulle vara gjort under en specifik vecka och vem som skulle göra vad. Nackdelen med det var att alla kanske inte alltid fick arbete som krävde lika mycket tid och jobb. Vi tror att vi sparade mycket tid på att ha sprintplaneringar i början av varje vecka, då vi annars inte hade haft särskilt bra koll på vem som gör vad, och när saker blir klara. Detta är en teknik vi antagligen skulle använda nästan oavsett vilket sorts projekt det är, möjligtvis inte om kraven av någon anledning förändras snabbt under en sprint.

### Daglig scrum

Här kunde vi inte följa Scrum fullt ut, utan istället för att träffas dagligen i korta möten ordnade vi Skypesamtal ett par gånger i veckan och hade en Skypegrupp där vi, när vi ville, kunde ställa frågor och stämma av med andra i gruppen om hur vi låg till. Det gav oss både fördelar och nackdelar jämfört med att träffas och prata. Fördelen var att vi inte behövde specifika tider utan kunde anpassa oss efter att vi hade olika scheman och möjligheter att träffas. Tyvärr fick vi inte samma kommunikation som ett möte i verkliga världen hade gett oss, men vi kan ändå tänka oss att ersätta dagliga scrums med onlinemöten i framtiden.

### Sprintgranskning

Sprintgranskning gjorde att vi kunde se hur vi låg till i förhållande till vad vi hade planerat att få gjort under en specifik sprint. Vi känner att det är en nödvändighet för att se till att alla har gjort det de skulle göra samt att alla features kommer med. Detta sparade oss mycket tid eftersom vi alltid visste vilka funktioner som var klara och inte behövde få det som en tråkig överraskning några dagar senare.

### Parprogrammering

Ibland använde vi oss av parprogrammering, framförallt just efter vi hade haft möten och ville komma igång med veckans uppgifter direkt. Fördelen med det var att vi ofta gjorde framsteg snabbt och vi kunde sedan övergå till mer självständigt arbete. Vi hade nog haft ännu mer nytta av parprogrammering i ett större projekt där varje del inte var möjlig eller för tidskrävande att arbeta på för en ensam person.

## 2. Spenderad tid

Vi har haft ca 20 timmar per vecka och person till vårt förfogande under detta projekt. Vissa veckor har det antalet timmar inte uppnåtts, medan under vissa har det överskridits. I dessa 20 timmar ingår förutom effektiv programmeringstid föreläsningar, handledarmöten, våra möten samt diskussioner på Skype m.m.

Även om vi inte skrivit ner exakt hur mycket tid vi lagt ner på projektet varje vecka så uppskattar vi ändå att vi lagt runt 20 timmar per vecka och person.

## 3. Samarbetet

Det som fungerade bra med vårt arbetssätt var att vi träffades varje vecka och diskuterade om hur föregående vecka varit, vad som skulle göras denna vecka samt hur vi skulle fördela arbetet. Vi delade upp arbetet så att spelet byggdes upp iterativt genom att börja med grunderna och sen bygga på det därifrån. Vid de tillfällena vi träffades arbetade vi parvis med de uppgifter vi delat ut, vilket bidrog till att vi snabbt kunde implementera saker eller lösa de problem som fanns eller dök upp på vägen. Målet var från början att träffas flera gånger i veckan och utöva parprogrammering, men då flera av gruppens medlemmar inte läst samma kurser denna läsperiod var det svårt att hitta tider mer än en gång i veckan då alla kunde.

Förutom mötena i skolan hade vi även en Skypegrupp där vi hade inplanerade möten en till två gånger i veckan. Vi hade där även en öppen chatt där man kunde diskutera allt som hade med projektet att göra, vilket skapade bra och regelbunden kommunikation i gruppen. Alla kunde berätta om eventuella problem, idéer och vad man lyckats göra.

Det som fungerade mindre bra var effektiviteten vid mötena i skolan. Även om vi fick gjort det vi ville så tog mötena ibland onödigt lång tid. Större delen av tiden spenderades på diskussioner som i sammanhanget var irrelevanta. En annan sak vi borde gjort är någon form av UML-skiss tidigt i projektet, men eftersom vi inte gjorde det så ledde det till att när vi skulle implementera saker så visste vi inte alltid vart det borde implementeras. Detta ledde i sin tur till att projektets struktur inte blev så bra som vi tänkt från början och vi var i läsvecka tre tvungna att strukturera om koden för att förbättra strukturen.

Att alla gruppens medlemmar inte läst samma kurser under denna läsperiod har ytterligare försvårat arbetet, då alla haft olika mycket tid till detta projekt. Därför har det varit svårt att fördela arbetet jämnt. Det har även varit svårt att uppskatta hur omfattande och tidskrävande en viss del varit, och utefter det dela upp arbetet jämnt.

Vi anser ändå att vi överlag jobbat mycket bra som grupp. Trots att ingen tog på sig någon direkt ledarroll så har arbetet fördelats relativt bra och vi har lyckats åstadkomma vad vi velat på utsatt tid.

### Vad vi kunde gjort annorlunda

Eftersom vi var helt nya till Android-utveckling och utveckling med libGDX-biblioteket när vi påbörjade det här projektet så visste vi inte riktigt hur allt skulle gå till från början. Men nu har vi mycket större kunskap om hur man utvecklar projekt av denna typ för vår målplattform och det kommer att göra det enklare för oss att slutföra ett liknande projekt, eftersom vi inte kommer att behöva lära oss allt från grunden igen. Med vår nyvunna kunskap kommer vi även kunna planera projektet bättre eftersom vi nu har ett bättre grepp om vad ett libGDX-spel består av och kan på grund av detta även dela ut arbete under projektet i en mer effektiv manér.

Nästa gång vi gör ett liknande projekt så kommer vi förmodligen att försöka skapa UML och flödesscheman av huvudkomponenter i spelet/applikationen vid ett tidigare skede än vad vi gjorde i detta projekt.

Eftersom de flesta av oss även var nya på Git när projektet startade så bromsade det ner våra framsteg i början då vi fick lägga ner mycket tid på Git istället för kodning. Nästa gång kommer vi även ha denna kunskapen att tillgodose vilket kommer att göra att man kan köra igång utan dessa problem.

Vi skulle nog även lägga mer tid på att modularisera olika komponenter av projektet för att minska kopplingen. Och även försöka att se till att klasser och metoder har mer väldefinierade syften och undvika att ge vissa metoder multipla syften.