



# Reflektionsrapport

DAT255 – Software engineering project

## **Team Turtle**

Daniel Ahlqvist

Alfred Björk

Henrik Boström

Rasmus Jemth

David Söderberg

Eric Wennerberg

18-26 oktober 2016

# Innehållsförteckning

<b>1. Applicering av Scrum</b>	<b>3</b>
1.1. Roller, lagarbete och socialt kontrakt	3
1.2. Använda strategier	3
1.3. Tidsfördelning	4
1.4. Effort, velocity och uppgiftsnedbrytning	4
<b>2. Reflektion på sprintretrospektiv</b>	<b>6</b>
<b>3. Dokumentation av sprintretrospektiv</b>	<b>7</b>
3.1. Sprint 1 (19 till 22 september)	7
3.2. Sprint 4 (6 till 13 oktober)	7
3.3. Sprint 5 (13 oktober)	8
<b>4. Reflektion av sprintåterblickar</b>	<b>9</b>
<b>5. Hantering av nya verktyg och tekniker</b>	<b>10</b>
<b>6. Reflektion av förhållandet mellan prototyp, process och produktägarvärde</b>	<b>11</b>
<b>7. Gruppens process i förhållande till litteratur och gästföreläsningar</b>	<b>12</b>
<b>8. Utvärdering av D1A och D2</b>	<b>13</b>
8.1. D1A – Legobygge	13
8.2. D2 – Halvtidsåterblick	13
<b>9. Burn-downdiagram</b>	<b>14</b>

# 1. Applicering av Scrum

## 1.1. Roller, lagarbete och socialt kontrakt

Gruppen har inte haft några officiellt definierade roller, men grupperingar efter önskade uppgifter har gjorts. Exempelvis har Eric, Henrik, David och Rasmus haft störst fokus på själva kodandet, medan Daniel och Alfred har sysslat mer med användarupplevelsen, i form av grafik, bandesign och ljud.

Följande sociala kontrakt tecknades i början av projektet:

- Varje gruppmedlem förväntas utföra 10 timmars arbete per vecka.
- Ett fast möte i slutet av varje vecka (onsdag-fredag) för sprintplanering och sprintåterblick.
- Dagliga avstämningar via Slack<sup>1</sup> (dagligt scrummöte)
- Följa Scrum och uppdatera boards på Github.

Första punkten har följts väl av samtliga gruppens medlemmar. Medlemmarna anser alla att de lagt någonstans mellan 10 och 15 timmar per vecka, och i vissa fall aningen mer. Även andra punkten har följts, då ett sådant möte hållits varje torsdag eftermiddag.

Avstämningar i Slack har inte följts på den nivå som diskuteras i det sociala kontraktet, men gruppen har ändå samtalat där om vad de gjort, samt att Githubrepot är kopplat till en särskild kanal, där notiser kommer varje gång någon gör en ändring. Därför går det lätt att se vad som gjorts under dagen. Gruppen anser att det inte varit nödvändigt att ha dagliga avstämningar på det sätt som först diskuterades. Detta eftersom projektet inte drivits på heltid och samtliga medlemmar har inte arbetat med projektet varje dag.

Följa Scrum och uppdatera kort har följts helt och hållet. Enda skillnaden mot kontraktet är att gruppen bestämde sig för att göra detta i Trello istället för på Github, då Trello har fler funktioner och är mer lättanvänt.

## 1.2. Använda strategier

Utvecklingen har skett relativt individuellt. Varje torsdag har gruppen haft ett sprintmöte, där föregående sprint sammanfattats och testats, samt kommande sprint planerats. Utöver detta har gruppen, åtminstone delar av den, ofta suttit tillsammans och programmerat samt diskuterat problem och lösningar.

Saker som diskuterades när delar av gruppen jobbade tillsammans missades ofta att förmedlas till resterande delar av gruppen. Genom att dokumentera det som diskuterades och även vad som kom ut av diskussionerna hade gruppen kunnat undvika denna miss i kommunikation.

---

<sup>1</sup> Meddelandetjänst som gruppen använt som kommunikationsmedium.

### 1.3. Tidsfördelning

Tidsanvändningen har inte loggats av någon i gruppen. Därför är det svårt att veta exakt hur mycket tid varje person lagt. Dock anser sig alla, som tidigare nämnt, ha lagt mellan 10 och 15 timmar per vecka på sina uppgifter, men underlag för detta saknas.

Detta är något som definitivt kunde gjorts bättre. Gruppen borde ha hållit bättre kontroll på den tid som lagts. Ett förslag är att gruppen borde haft ett kalkylark där varje arbetstillfälle antecknats, i form av den tid som lagts och vad som gjorts. På så vis hade det gått att se vilka medlemmar som lämpat sig bäst för specifika uppgifter och därför hade arbetet kunnat effektiviseras genom att uppgifter tilldelas den person som är mest lämplig, baserat på föregående prestationer inom samma område.

### 1.4. Effort, velocity och uppgiftsnedbrytning

Gruppens velocity bestämdes till 60, baserat på att alla gruppmedlemmar tillsammans förväntades jobba 60 timmar i veckan. Således motsvarade en effort point en timmes arbete.

För att uppskatta effort för varje user story användes Planning Poker<sup>2</sup> och utifrån dessa svar användes ett medianvärde. Inför varje sprint plockades user stories med en sammanlagd effort på 60 ut från product backloggen och lades i sprint backloggen. När user stories hade valts ut, delades dessa upp mellan alla gruppmedlemmar. Det var sedan upp till varje gruppmedlem att skriva *requirements* för de user stories de var tilldelade.

Sprint	Effort
1	44
2	48
3	47
4	50
5	66
<b>Totalt</b>	<b>255</b>

*Sammanlagd effort för varje sprint.*

---

<sup>2</sup> Cohn, Mike. Planning Poker. <https://www.mountaingoatsoftware.com/agile/planning-poker> (Hämtad: 2016-10-25).

I den första sprinten var gruppens user stories breda, och inte tydligt definierade. Detta innebar att det var svårt att veta när den var utförd. Efter första sprint reflektionen kom gruppen överens om att lägga mer tid på att skriva löskopplade user stories som kunde utföras parallellt.

<b>Som användare vill jag ha en bana där jag står stilla.</b>	<b>Som användare vill jag få tydlig feedback när jag klarat eller misslyckats med ett uppdrag.</b>
<b>Requirements:</b> <i>Inga krav specificerade.</i>	<b>Requirements:</b> <ul style="list-style-type: none"><li>• Om spelaren tar fel ord på ett uppdrag, ska tydlig negativ feedback ges, men ordet fortfarande spelas upp.</li><li>• Om spelaren tar rätt ord på ett uppdrag, ska tydlig positiv feedback ges. Ordet ska även spelas upp.</li><li>• När spelaren tar ett ord på ett uppdrag, byter uppdragsindikatorn automatiskt till nästa uppdrag.</li></ul>

*En user story från sprint 1 (till vänster) och en från sprint 4 (till höger). Att lämna kravspecificeringen till varje enskild gruppmedlem ledde ofta till att det inte blev specificerat.*

Även vilken ordning gruppen utförde sprint planning mötena ändrades mot slutet av projektet. Istället för att tilldela en user story effort och sedan skriva requirements, skrevs nu requirements först, tillsammans i gruppen och sedan genomfördes Planning Poker. Detta ledde till att gruppens requirements blev tydligt definierade och kraven var accepterade i hela gruppen.

Eftersom gruppen efter varje reflektion implementerade förändringar i arbetssättet utifrån vad som gick fel finns det ingenting gruppen nu i efterhand kan säga gick fel. De punkterna som kunde gjorts annorlunda, gjordes annorlunda i nästa sprint.

## 2. Reflektion på sprintretrospektiv

Under gruppens sprintretrospektiv diskuterades punkter som en eller flera gruppmedlemmar tyckt kunnat förbättras. Ofta så ledde dessa diskussioner till att någonting förändrades inför nästa sprint. Några exempel på sådana punkter är velocity, kodarkitektur och nedbrytning av user stories.

Retrospektiven gjordes ofta muntligt, och skrevs endast ner efter sprint 1, 4 och 5. Gruppen kände att dessa diskussioner var oerhört givande eftersom det var lätt att implementera förändringar till nästa sprint. Därför hade det varit önskvärt att dokumentera alla retrospektiv för att kunna stämma av om det nya arbetssättet faktiskt löste problemet som belysts under föregående retrospektiv.

## 3. Dokumentation av sprintretrospektiv

### 3.1. Sprint 1 (19 till 22 september)

Positivt:

- Bestämd velocity hölls på ett ungefär.
- En körbar applikation producerades.

Negativt:

- Koden var inte refaktorerad i slutet av sprinten. Frågeställning: Ska detta göras i slutet av varje sprint eller mellan sprintarna?
- Svårt att dela upp uppgifterna mellan olika personer, eftersom gruppen inte riktigt hade koll på hur LibGDX fungerar.
- Inga unit-/integrationtester.

Fattade beslut:

- Tydligare struktur och skiss av hur arkitekturen ska se ut innan varje sprint påbörjas. Detta eftersom att gruppen vill slippa skriva om i princip allt, vilket vid detta tillfälle var tvunget.
- Försöka löskoppla user stories mer, för att de ska bli mindre beroende av varandra

### 3.2. Sprint 4 (6 till 13 oktober)

Positivt:

- Gruppens velocity ligger på en bra nivå.
- Bättre git workflow. Fler merges, och oftare, alltså färre merge conflicts än tidigare.
- User stories är mer löskopplade, och lättare att jobba på parallellt.

Negativt:

- Uppdelningen mellan content och dev är lite obalanserad. Content är lite svårare att uppskatta, både vad som kan anses vara klart och hur lång tid det tar att genomföra. Det leder till diffusa user stories.
- Alla user stories delas upp i sprintbackloggen under sprintplaneringsmötet, vilket leder till att vissa personer blir "färdiga" med sitt, trots att det finns user stories kvar att göra.

Fattade beslut:

- Endast dela ut de 6 första (eftersom vi är 6 personer) user storiesen och sedan plockar man det nästa prioriterade om man blir klar med sin.
- Konkreta krav på innehållet, specifikt banorna. I denna sprinten sa vi endast, gör banor, men det var inte nerbrutet i delmål.

### 3.3. Sprint 5 (13 oktober)

Fördelar:

- Endast de sex högst prioriterade user storierna delades ut under under sprintplaneringsmötet. Varje individ fick sedan plocka den översta storien när denne var klar med sitt föregående.
- Fortsatt bra velocity och effort.
- Små konkreta user stories. Lätta att uppskatta effort.

Nackdelar:

- Om en pull request på Github är godkänd av alla, kan vem som helst mergea den och inte endast den som skapade förfrågan.
- Fler gruppmedlemmar bör godkänna eller neka en pull request, så det går snabbare från det att förfrågan skapas till den kan mergas.
- Fortsatt inga tester.



## 4. Reflektion av sprintåterblickar

Sprintåterblickarna (*sprint reviews*) har fungerat mycket bra. Under sprintplaneringsmötena, som hölls varje torsdag, gick föregående sprints user stories igenom en efter en. Först lästes alla krav igenom på storyn och därefter testades det så att allt fungerade. Punkterna som fungerade markerades i checklistan i Trello. När alla punkter i checklistan var ikryssade, flyttades user storien till "avklarad" ("*done*"). Detta repeterades för alla user stories. Om någon story inte var uppfylld, behölls den under "pågående" ("*doing*") och togs med till nästa sprint. Detta hände dock bara med några enstaka stories under projektets gång.

## 5. Hantering av nya verktyg och tekniker

Applikationen har utvecklats med hjälp av spelramverket *libGDX*. När projektet påbörjades hade endast en av gruppmedlemmarna erfarenhet av *libGDX* sedan tidigare. Detta löstes genom att övriga gruppmedlemmar kollade på diverse instruktionsserier på YouTube för att sätta sig in i det innan utvecklingen av projektet hade påbörjats. Den första sprinten inkluderade grundläggande implementering och användning av *libGDX* och under denna sprint så arbetade alla gruppmedlemmar tillsammans fysisk för att snabbt kunna hjälpa varandra runt eventuella problem.

För att konstruktion av banor har gruppen använt programmet Tiled. Även detta var ett ny teknik för majoriteten av gruppen, och även här kom YouTube till undsättning. En annan metod för att tackla detta problemet var att alla gruppmedlemmar konstruerade en bana var och under nästa möte så diskuterades dessa banorna. Då hittade gruppen inspiration från varandras banor, och kunde även diskutera lösningar på vanliga problem.

I kursen *Objektorienterat programmeringsprojekt* gjorde Eric, Alfred, Daniel och David en Androidapplikation tillsammans och därför hade dessa erfarenhet av Android Studio och applikationsutveckling sedan tidigare. Henrik hade lite erfarenhet från annat håll, medan Rasmus aldrig använt det tidigare. Alla kom dock snabbt in i det, då samtliga hjälptes åt att instruera de som inte var lika erfarna.

## 6. Reflektion av förhållandet mellan prototyp, process och produktägarvärde

Det ansågs viktigt att produktägaren tidigt fick ut något värde av applikationen. I detta projekt var produktägarens vision att användaren skulle bli bättre på svenska språket. Därför planerades det tidigt i processen att någon form av utläring skulle finnas mer eller mindre omedelbart. Redan i sprint 1 hade prototypen fått sin första nivå av inläring. Detta i form av att ordet "apple" lärdes ut, genom att en bild på ett äpple visades och en ljudinspelning av ordet "apple" spelades upp så fort användaren hoppade. Detta var naturligtvis inget som behölls till den slutgiltiga prototypen, men det var ett första steg mot den.

Produktägarvärdet har alltid legat i fokus. Utlärningsmomentet har prioriterats och detta har legat till grund för många idéer. Eftersom användarna antas vara dåliga på svenska, har det grafiska gränssnittet varit tvunget att göras extra tydligt. Inga krångliga ord eller meningar. Dessutom har knappar försetts med symboler för att användaren ska kunna lista ut vad knappen gör, utan att behöva förstå texten.

## 7. Gruppens process i förhållande till litteratur och gästföreläsningar

Gruppen anser att kursens gästföreläsningar för det mesta har varit intressanta. De har dock inte påverkat gruppens arbetssätt. Detta anser gruppen beror på att innehållet i föreläsningarna har varit svårt att applicera på kursen. Föreläsningarna har bl.a. handlat en del om stora företags interna strukturer. Det är intressant att få höra hur stora organisationer jobbar sett till hela företaget men sånt är svårt att applicera på en liten grupp som vår. Föreläsningarna innehöll mycket lite konkret information kring agila processer för mindre grupper. Det som hade kunnat vara till hjälp är t.ex. fler konkreta tips på hur man tar fram en bra user story, vad man skall tänka på när man tidsplanerar och sätter effort points för att kunna göra en realistisk plan eller andra tips från erfarna projektledare och liknande. Därmed inte sagt att gästföreläsningarna var ointressanta.

Någon litteratur har inte studerats av någon av gruppens medlemmar. Det är dock inget vi har saknat. Om litteratur studeras kan det ge värdefull input till processen på samma sätt som nämnts i stycket ovan, alltså genom konkreta tips. Gruppen anser dock att det har varit mer värdefullt att under ett första försök till att använda scrum få prova sig fram. Litteratur är något som kan komma till nytta framöver om man vill fördjupa sig i ämnet.

## 8. Utvärdering av D1A och D2

### 8.1. D1A – Legobygge

Under legoövningen lärde sig gruppen mycket. I början av övningen visste de flesta gruppmedlemmar knappt vad Scrum innebar, men i slutet hade gruppen åtminstone en grundläggande förståelse. Reflektionen som skrevs efteråt löd enligt följande:

*Under bygget var det bra samarbete i gruppen, trots svårigheten att aktivera sex personer kring ett legobygge. Det hände att flera personer arbetade på samma sak på olika håll, helt i onödan.*

*Realistiska uppskattningar gjordes, som gruppen lyckades uppfylla alla tre sprintarna. Dock hade gruppen kunnat utmana sig själva mer, och genom att fördela arbetet ändå klara av uppgiften, och på så sätt bli mer effektiva. Det gäller att hitta en balans mellan kvalitet och kvantitet.*

Denna reflektion kom att få betydelse under planeringen. Arbetsuppgifter delades ut till olika medlemmar, för att inga skulle arbeta på samma sak. Problemet med att gruppen inte utmanade sig tillräckligt var något som bestod i första sprinten. Efforten som tilldelades uppgifterna var något hög och klarades i flera fall av på kortare tid än beräknat.

Vad man kan se är att reflektionen av legoövningen behandlade likvärdiga detaljer som i det riktiga projektet, vilket gör att samma reflektion kunnat användas även i detta.

### 8.2. D2 – Halvtidsåterblick

## 9. Burn-down diagram

