

Segway Simulator

En studie av agil mjukvaruutveckling i projektform

Pontus Doverstav
Viktor Karlsson
Christoffer Medin
Pontus Malm
Anton Myrholm
Victor Nilsson

Chalmers Tekniska Högskola
DAT255
2013-10-25

Sammandrag

Följande rapport behandlar processen i utvecklandet av en Android-applikation; mer specifikt kommer fokus ligga på utvecklingsmetoden Scrum, som arbetsgruppen för projektet "Segway Simulator" har använt sig av. Rapporten kommer att beskriva Scrum och på vilket sätt gruppen har använt sig av den, samt analysera vad i metodiken som har bidragit till en effektiv arbetsprocess och vad som skulle kunna förbättras. Sammanfattningsvis kan det sägas att arbetsgruppen har funnit att Scrum inbegriper element som gör det till en mycket effektiv metod för mjukvaruutveckling, men att det är både omständigt och onödigt att använda metoden fullt ut om inte alla kriterier är uppfyllda.

Innehållsförteckning

Sammandrag

1: Introduktion

1.1: [Bakgrund](#)

1.2: [Syfte](#)

2: [Teori](#)

[Mjukvaruutveckling och agil utveckling](#)

[Scrum](#)

3: [Metod](#)

[3.1: Planering](#)

[3.2: Realisering](#)

[3.2.1: Arbetsmetoden](#)

4: [Resultat](#)

[4.1: Applikation](#)

[4.2: Dokument](#)

5: [Diskussion](#)

[5.1: Arbetsmetod](#)

[5.2: Gruppdynamik](#)

[5.3: Val av verktyg och design](#)

[5.4: Slutsats](#)

1: Introduktion

1.1: Bakgrund

Att arbeta med mjukvaruutveckling är en ständigt föränderlig process. Allt eftersom en produkt närmar sig färdigställande kan kriterier komma att ändras; exempelvis kan din plattform komma att uppdateras eller ersättas av något annat, det kan finnas fel programmet som inte upptäcks förrän projektet gått vidare till ett senare arbetsstadie, och så vidare. Det är också svårt att hålla ett redan existerande program vid liv, särskilt om sagda program används i stor skala. Hur rigorös testning man som utvecklare än försöker uppnå, kan man inte jämföra det med alla möjliga situationer som kan tänkas uppstå när över en miljon användare använder programmet. I projektet "Segway Simulator", ett småskaligt projekt med mål att skapa ett spel för Android-telefoner, används Scrum, en agil metodik för mjukvaruutveckling. Följande rapport kommer att behandla vilken roll Scrum har under utvecklingsprocessen, samt vilka för- och nackdelar som fanns med metoden.

1.2: Syfte

Syftet med projektet "Segway Simulator" är att ta fram en fungerande applikation med hjälp av metoden Scrum. Att kunna presentera en fungerande applikation i sig är bara ett av målen; arbetsprocessen är en lika stor del i resultatet som produkten. I projektets slutskede ska det alltså vara möjligt att utvärdera arbetsgången i samma mån som spelet i fråga. Huruvida applikationen har färdigställts avgörs av dess funktionalitet, baserat på de specifikationer som gruppen satte i början av projektet (se bilaga 1). Arbetsprocessen utvärderas å sin sida i följande rapport av gruppen i fråga samt av handledare.

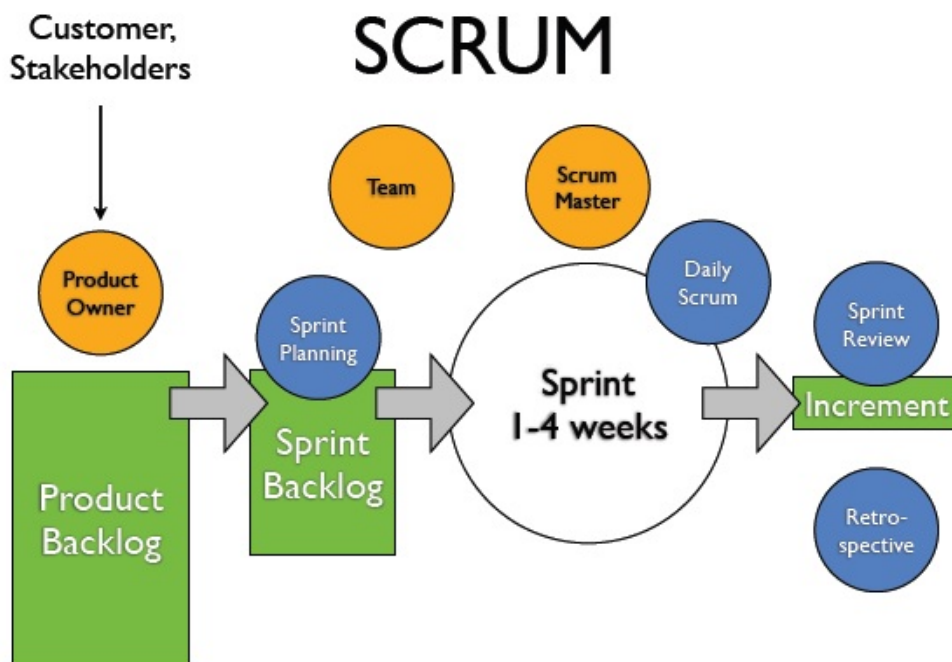
2: Teori

Mjukvaruutveckling och agil utveckling

Mjukvaruutveckling kan ta sig många olika uttryck när det kommer till arbetsstruktur och resursfördelning. Agil mjukvaruutveckling fokuserar på att arbetsprocessen ska vara flexibel och öppen för förändring, till exempel ska nya önskemål och kriterier från uppdragsgivaren kunna tillgodoses snabbt och smidigt i stället för att spika en plan från början och sedan hålla sig till denna (Manifesto for Agile Software Development, 2013). Att arbeta under en mer traditionell projektform skulle innebära att kunden vet vilken produkt han vill ha innan han ens ser den, att specifikationerna inte ändras under projektets gång, och så vidare. Agil utveckling frånkommer de uppenbara problemen med denna modell, och kan komma att förhindra missnöje från uppdragsgivaren i och med att de har större möjlighet att påverka projektet under arbetsgången. Det finns självklart flera olika modeller som faller under agil mjukvaruutveckling; scrum är en av dessa.

Scrum

Scrum faller som tidigare nämnt under kategorin agil mjukvaruutveckling. Metoden bygger på att ett relativt litet team utvecklare delar upp arbetsprocessen i förhållandevis korta "Sprints" och att man under dessa har korta, dagliga möten för att stämma av vad som gjorts och vad som ska göras till dagen efter. Scrum är förstås mer omfattande än så, och kan förklaras i figur 1.



Figur 1: Arbetsmetodiken Scrum (Luvö, 2013)

Arbete som utförs enligt Scrum utgörs främst av ett självorganiserande och multifunktionellt team på 3-9 medlemmar - arbetslaget ska kunna klara sig själva vad gäller organisation och medlemmarna ska inom gruppen ha all kompetens som är nödvändig för projektet.

Uppdragsgivaren betecknas här som produktägaren (Product Owner) - denna person har ansvar för att bestämma vilka funktioner som har vilken prioritet. Funktionerna som produkten ska ha definieras i sin tur i en backlog (Product Backlog) och är i princip en lista på arbetsuppgifter för gruppen. Gruppens kontakt med sin produktägaren är projektets Scrum Master - den här personen ansvarar i princip för att arbetet ska flyta på genom att hålla kontakt med uppdragsgivaren. Denna "gruppleddare" informerar produktägaren om utfört arbete och redovisar loggboken (Luvö, 2013).

I början av projektet specificeras alltså produktens funktionalitet av produktägaren via en backlog, som "gruppleddaren" (Scrum Master) hanterar tillsammans med sagda produktägare. Arbetslaget inleder därefter en så kallad sprint - en period där gruppen utgår från ett första sprint-möte och därefter börjar arbeta. Under sprintperioden, som i regel sträcker sig över 1-4 veckor, möts arbetsgruppen varje dag i korta möten för att stämma av vad som gjorts och vad som bör göras. Efter en avslutad sprint utvärderas det utförda arbetet, och resultatet, ett så kallat inkrement, bör vid det här laget vara något konkret som kan visas upp för uppdragsgivaren.

3: Metod

Projektets utvecklingsprocess kan betecknas som agil, men har inte följt mönstret Scrum hela vägen. Vilken typ av applikation, dess funktioner och utvecklandeprocessen bestämdes relativt snabbt, varpå produkten tog form och fler funktioner tillkom under projektets gång. Gruppen har varje vecka möten med en handledare som ger förslag på funktioner, input på vad som kan göras, tips på arbetsfördelning, och så vidare.

3.1: Planering

Projektets planerades under de första veckorna, varpå det bestämdes att ett spel som drar nytta av egenskaper hos en mobiltelefon (exempelvis accelerometer eller mikrofon) skulle vara intressant att utforska närmare. Accelerometern ansågs i slutändan vara mest relevant för gruppens intresse, varpå gruppen gemensamt bestämde applikationens uppbyggnad och funktioner. Projektet kräver inte många externa bibliotek, vilket gjorde arbetet med exempelvis AndEngine, som utgör spelmotorn, förhållandevis enkelt. Under denna fas planerades också hur användargränssnittet skulle utformas. Planeringen pågick främst under de första två till tre veckorna av projektets totala tid.

3.2: Realisering

Merparten av arbetet utfördes gemensamt i grupp med så många närvarande som möjligt. Arbetslaget delade aktivt och dynamiskt upp arbetet allt eftersom projektet framskred; nya funktioner och förslag kunde behandlas väldigt snabbt i och med att hela gruppen fanns på plats. Även problem som uppstod under arbetsgången kunde behandlas av hela gruppen utan att behöva vänta till senare. Ett normalt arbetstillfälle innebar i stora drag att gruppen gemensamt satte sig ner och diskuterade vad som borde göras med utgångspunkt i produktens dåvarande funktionalitet, i och med att grundfunktionaliteten uppfylldes förhållandevis snabbt. Testning utfördes sporadiskt under dessa tillfällen med syfte att leta buggar i applikationen av samtliga gruppmedlemmar. Programmeringen i fråga var inte helt jämnt fördelad - vissa gruppmedlemmar utförde merparten av kodningsarbetet, vilket tillät de övriga att lägga mer tid på dokumentation, testning och planering.

3.2.1: Arbetsmetoden

Scrum var en utgångspunkt för arbetsmetodiken, men frångicks ganska snabbt när det framgick att gruppen utförde merparten av arbetet gemensamt med alla medlemmar närvarande. Planering och implementering utfördes parallellt, genom att gruppen dynamiskt och spontant delade upp arbetet mellan medlemmarna, och fortsatte att diskutera förbättringsområden under arbetets gång.

4: Resultat

Efter åtta veckors tid har projektet resulterat i en fullt fungerande applikation. Applikationen SegwaySimulator är ett plattformsspel i två dimensioner som styrs främst med telefonens accelerometer. Utöver detta finns en utvärdering av gruppens arbete i form av denna rapport, samt dokumentation som använts under planering och testning. .

4.1: Applikation

SegwaySimulator är ett spel där användaren styr en Segway över en hinderbana; målet är helt enkelt att nå till slutet av sagda hinderbana. Användaren har möjlighet att uppgradera sin Segway för att öka hastigheten, höjden som fordonet kan hoppa, och så vidare.

Applikationen kan främst ses som ett koncept - alla funktioner som definierades i början av projektets gång är implementerade, men det finns fortfarande utrymme för att utöka omfattning och funktionalitet. Ett exempel på vad SegwaySimulator visar är meriten av ett enkelt interface på ett spel - inga knappar används, utan i stället används telefonens accelerometer för att förflytta karaktären, högra sidan av skärmen för att hoppa, och vänstra sidan av telefonen för en "boost", det vill säga att öka hastigheten markant.

4.2: Dokument

Projektet har också genererat relevant dokumentation - detta är värt att notera som en del av resultatet då projektets syfte inte enbart är en faktiskt produkt. En utvärdering av arbetsprocessen är en lika stor del som spelet i fråga, varför produktens dokumentation gällande testning, designval, funktioner och liknande också ska tillskrivas som ett resultat av projektet. Notera att en stor del av dessa dokument endast användes av utvecklarna och kan därmed innehålla material som främst är avsett för användning under utvecklandeprocessen. Dessa dokument finns att tillgå på följande webbplats:

https://github.com/Meddan/SegwaySimulator/wiki/_pages

5: Diskussion

Följande avsnitt behandlar projektets arbetsmetodik. Avsnittet innefattar gruppens analys av metodiken och arbetsgången som helhet. Gruppens arbete har till stor del drivits framåt av "trial and error", då arbetsmetoden tillät hela gruppen att snabbt reagera på och lösa problem som uppstod.

Diskussionen behandlar för- och nackdelar med projektets arbetsmetod, gruppdynamiken och valet av verktyg och design för produkten. Med detta avsnitt avslutas och rapporten med en sammanfattning av gruppens tankar om projektet som helhet.

5.1: Arbetsmetod

Projektets arbetsgång utgick till en början från scrum, men övergavs snabbt då det blev uppenbart att strukturen med dagliga möten, behovet av en scrum master och specifika inkrement inte var av stor vikt för gruppen. I stället implementerades funktioner kontinuerligt av olika medlemmar; planering och implementation flöt samman. Nya funktioner kunde diskuteras fram och implementeras direkt, om konsensus i gruppen var att sagda funktion var relevant för projektet.

För ett projekt i den här skalan fungerade den här arbetsmetodiken bra. Mycket på grund av att alla medlemmar hade en gemensam vision av vad målet var, men sannolikt också eftersom det inte fanns någon produktägare som dikterade villkor för produkten. Utvecklingsprocessen var dock fortfarande väldigt dynamisk; arbetsgruppen kunde snabbt ta gemensamma beslut om designen och implementera dessa. Modellen tillät också samtliga medlemmar att testa applikationen och leta buggar, om utrymme fanns. Kommunikationen fungerade överlag felfritt i och med att alla gruppmedlemmar var på plats större delen av tiden då applikationen implementerades.

Metoden är dock inte utan nackdelar. Att planera på lång sikt är betydligt svårare, liksom resursfördelningen. I och med att gruppen frångick en scrum master och inte hade någon produktägare att arbeta mot är det av stor vikt att alla tar sitt personliga ansvar om inte arbetsfördelningen ska bli för obalanserad. Att projektet var relativt litet i skala bidrog också till att metodiken fungerade - det fanns svårigheter med att prioritera de olika funktionerna, något som skulle kunna innebära problem för större arbeten.

Trots detta är det en metod som fungerar väl, men det kräver antagligen att samtliga medlemmar i arbetsgruppen har ett väl fungerande samarbete och att gruppen inte blir för stor. En annan begränsande faktor är också gruppens förutsättningar för att kunna sätta sig på samma plats och arbeta rent fysiskt. Det har dock fungerat att kommunicera på liknande sätt, exempelvis genom Skype, under kortare perioder. Det är dock svårt att dra några slutsatser om hur det skulle fungera under längre tid. Vad som främst talar för metoden är dock att hela gruppen har varit bekväma med arbetsgången och därmed motiverade att arbeta; att projektet går framåt och kan visa resultat är ett tecken på att det är en modell som fungerar.

5.2: Gruppdynamik

Gruppens kommunikation och förmåga att arbeta tillsammans har i stora drag fungerat väldigt bra. I allmänhet har de flesta i arbetsgruppen bidragit till implementationen och designen av produkten. Det har funnits problem med arbetsfördelningen, men i de fall där en gruppmedlem har utfört mindre arbete med koden försöker sagda medlem arbeta på exempelvis dokumentation i stället för implementation - strukturering av arbetet har alltså varit en kollektiv insats av hela gruppen. I och med att arbetsfördelningen fungerat bra och produktens funktionalitet var väldefinierad redan tidigt under planeringen fanns inget behov av en gruppleddare.

Kunskaperna och erfarenheterna hos gruppen är väl fördelade, men gruppen har saknat kompetens vad gäller grafisk formgivning - men i och med att produkten mer är ett uppvisande av koncept med mycket funktionalitet har inte det varit något högprioriterat problem. Inte heller har det funnits något problem med arbetsvilja - att arbeta på det sätt som känns naturligt för gruppen verkar tvärtom för att produkten utvecklas naturligt enligt gruppens vision. Möjligheten att arbeta under sådana omständigheter kan ha varit en av de viktigaste faktorerna i att arbetet konstant gick framåt.

5.3: Val av verktyg och design

För att göra ett spel med fysik är möjligheterna i Javas standard-API begränsade, givet den korta tiden. SegwaySimulator använder sig av biblioteket AndEngine (en spelmotor), samt ett API som utökar AndEngine med en fysikmotor (AndEnginePhysicsBox2DExtension). Valet av AndEngine var ingen lång process - gruppen hade fått rekommendationer om att använda just detta bibliotek, och med tanke på att gruppens vision inte var en överdrivet komplex applikation ansågs valet av spelmotor spela mindre roll.

Applikationen i sig är designad för att vara intuitiv och lättanvänd. Själva spelet frångår knappar för att ge användaren möjlighet att se skärmen, men också för att förenkla spelbarheten - det är lättare att kunna interagera med halva skärmen, snarare än en liten knapp.

5.4: Slutsats

Mjukvaruutveckling kan ta sig många olika uttryck. En agil utvecklingsprocess kräver att arbetslaget kommunicerar, då förändringar i specifikation såväl som implementation kan uppstå utan förvarning. Likväl behövs en viss form av struktur, något som arbetet med SegwaySimulator åsidosatte, mycket på grund av att det inte fanns någon extern part utöver handledare att få input och förslag från. Detta innebar att projektet låg helt i utvecklarnas händer, och därmed implementerades de funktioner som ansågs vara intressanta ibland innan hänsyn togs till specifikationerna som skrevs i början av projektet.

Därmed inte sagt att produkten tog någon skada av detta, då bristen på struktur vägdes upp av gruppens motivation och intresse, något som vi anser vitalt för en given uppgift. Den nyttjade metoden skulle antagligen inte fungera lika bra i ett projekt där det finns en uppdragsgivare med i bilden, men att samtliga medverkande känner en vilja att lägga tid på ett projekt och att det finns ett genomgående intresse tycks vara det viktigaste för att ett mjukvaruprojekt ska falla väl ut.

Källförteckning

Luvö, Thomas. (2013) *Scrum*, Föreläsning i kurs DAT255 Chalmers Tekniska Högskola, 2013-09-09

Agile Alliance. (2013) *Manifesto for Agile Software Engineering*
<http://www.agilemanifesto.org/> (2013-10-22)