

Post-Mortem Report

Projekt Tendu:

Markus Andersson Norén, Johanna Hartman, Roy Nard, John Petersson, Christian Meijner, Ludwig Lindberg, Hampus Forsvall

Chalmers Tekniska Högskola, IT 2, grupp 6

Software engineering project, hösten 2013

1. Inledning

2. Process och tillvägagångssätt

2.1 Tidsåtgång och planering

2.2 Verktyg

2.2.1 Parprogrammering

2.2.2 Sprint-planering

2.2.3 Sprintutvärdering

2.2.4 Stå-upp-möten

2.2.5 Trello

3. Diskussion

3.1 Kommunikation i gruppen

4. Slutsats

1. Inledning

Spel till mobilen är något som har funnits länge, men man har i de flesta fall varit begränsad till att spela ensam, eller över internet. Dagens smartphones har flera tekniker för att spela spel mellan telefoner; bluetooth, wifi, eller över internet. Vi har implementerat de två förstnämnda.

Vårt spel, Tendu, fungerar så att man jobbar i lag för att lösa små utmaningar tillsammans. I dagsläget finns två mindre utmaningar att spela, var och ett med många nivåer, och det är enkelt att implementera fler småspel.

Att utveckla spel som involverar nätverk kan vara en stor utmaning, inte bara rent tekniskt utan även för att hålla hela gruppen uppdaterade så att även de som inte arbetar med nätverket faktiskt kan använda sig av nätverkstjänsterna som utvecklats. Den här rapporten behandlar för- och nackdelar med vår arbetsprocess, och ger tips till de som vill utveckla liknande applikationer.

2. Process och tillvägagångssätt

I utvecklandet av Tendu har vi använt oss av olika tekniker för att få en bra struktur på arbetet och för att kunna dela upp arbetsuppgifter på ett smidigt sätt. Då metoder som *Scrum* (*SCRUM.org, 2013*) eller *Extreme programming* (*Don Wells, 2013*) är anpassade för större projekt, på stora företag med en kund i fokus, kände vi att de metoderna inte var helt anpassade för vår situation. Därför har vi snarare gjort vår egna sammansättning av metoder som passar till vårt projekt.

De flesta metoder vi har använt oss av är direkt hämtade från SCRUM; Projektet är uppdelat i så kallade *Sprints* som varar i en vecka. En Sprint är en mindre period av ett SCRUM-projekt för vilket man sätter delmål för hela projektet. I början av varje sprint har vi en *Sprint-planering*, där vi sätter mål för veckan och fördelar arbetet. Varje dag har vi haft ett kort möte - *Daily scrum*, där varje medlem

kort berättar vad den håller på med så att alla håller sig uppdaterade.

Vi har även testat på *Parprogrammering*, som egentligen tillhör metoden extreme programming.

2.1 Tidsåtgång och planering

Eftersom vårt projekt är av relativt komplex natur, och alla i gruppen är oerfarna när det kommer till nätverk och grafik, har projektet redan från start krävt att vi jobbat tillsammans med både planering, möten och implementation. Det har gjort att gruppen hela vägen hållit sig på samma spår och att besluten har blivit gemensamma.

På en vecka har vi haft flera olika sorters möten. Våra dagliga möten har vi försökt hålla till ca 15 minuter medan vi lagt ungefär en timma eller två på våra längre möten. Alltså ungefär 4 timmar/gruppmedlem vilket blir totalt 28 timmars mötestid för hela gruppen/vecka. Mer tekniska diskussioner har vi i största möjliga mån försökt hålla utanför våra möten.

I arbetet med implementationen av vår applikation har vi använt oss av parprogrammering. I en grupp på sju personer känns det naturligt att dela upp arbetsuppgifter i mindre grupper. I snitt har alla gruppmedlemmar lagt ungefär 5 timmar/dag på implementation där hälften av den tiden lagts på att programmera tillsammans med någon/några i gruppen. Totalt alltså 25 timmar per vecka och person, eller $25 \cdot 7 = 175$ timmar per vecka för hela projektgruppen.

2.2 Verktyg

2.2.1 Parprogrammering

Vi testade att använda parprogrammering som egentligen är ett moment tillhörande metoden *extreme programming*.

Parprogrammering innebär helt enkelt att man sitter i par och programmerar på en dator. En fördel med detta var att många slarvfel direkt eliminerades snarare än att hänga med till testningfasen, vilket hade lett till större tidsåtgång till att felsöka och identifiera

problemen.

Parprogrammering kan anses vara mindre tidseffektivt än att programmera individuellt då varje “vanlig” timme motsvarar två mantimmar då två personer jobbar på ett problem, men vi upplevde snarare att det sparade tid, då parprogrammering ger upphov till bättre och mer dynamiska lösningar då problem attackeras ur mer än en synvinkel direkt. Det blir lätt att bolla idéer med varandra och lättare att undvika konstiga lösningar, som senare genererar problem.

Denna metod visade sig vara överraskande effektiv. Efter att, fram till detta projekt, ha jobbat näst intill uteslutande individuellt när det kommer till direkt kodande ansågs uppoffringen klart för stor vid första anblick. Detta motbevisades dock till viss del då många problem kunde lösas direkt eller i vart fall vid ett väldigt tidigt skede. Trots detta så finns det klara fördelar med att jobba individuellt också och en hälsosam blandning av de båda metoderna är att föredra.

Parprogrammering verkar passa för att lösa abstrakta och specificerade problem där en uppenbar lösning saknas. Vid enklare typer av problem, där varje person kan lösa uppgiften lika snabbt som ett par, kanske parprogrammering kan vara överflödigt.

2.2.2 Sprint-planering

Varje vecka började med en sprint-planering, där hela gruppen fick tillfälle att yttra sig i frågor rörande projektet. Alla hade möjligheter att kontinuerligt medverka och på så sätt hålla sig uppdaterade i alla frågor.

Genom sprint-planeringen får man varje vecka ett antal klara mål som varje medlem jobbar mot. Det blir lättare att bryta ner större uppgifter och bidrar till att det blir enklare att fördela arbetet. Gruppen agerade mer som en enhet och alla kände sig mer delaktiga.

En nackdel är att sprint-planeringen tar tid. Varje vecka lades 2 timmar på sprint-planering vilket motsvarar 14 timmar för hela gruppen.

Någon form av uppdelning av ett större projekt tror vi är nödvändig. Det är svårt att planera ett projekt till fullo i början, därför behöver

man sätta delmål för varje sprint i projektet, och för det fungerar sprint-planeringen bra. När man jobbar agilt passar sprint-planeringen extra bra eftersom väldigt lite är specificerat från början. Vi använde oss av sprint-planering under hela projektiden.

2.2.3 Sprintutvärdering

I slutet på varje sprint reflekterade vi över veckan som gått i en så kallad sprintutvärdering. Här fastställde vi vad som hade gjorts, vad som kunde ha gjorts bättre och samlade stoff för nästa sprint-planering.

Genom sprintutvärdering ökade arbetsprocessens kvalitet allt eftersom projektet fortgick, eftersom vi varje vecka försöker trycka på punkter som kunde förbättras.

Som alla möten, så tar Sprintutvärdering upp tid som annars kan läggas på implementation. Varje vecka lades 1 timma på Sprint-utvärdering, vilket motsvarar 7 timmar för hela gruppen, men allt eftersom tiden går bör sprintutvärderingen göra att man jobbar mer strukturerat, och därmed sparar tid.

Sprintutvärdering passar de allra flesta former av projekt, speciellt längre projekt, men passar nog sämre för korta och väldefinierade projekt där kunskap om uppgiftens alla delar redan finns från projektets start.

2.2.4 Stå-upp-möten

Varje dag, eller efter behov har vi haft så kallade stå-upp-möten, där alla gruppmedlemmar har berättat vad de gör, vad de planerar att fortsätta med och får en chans att lyfta fram problem som hindrar att projektet fortgår. Det är ett snabbt sätt att hålla alla medlemmar uppdaterade på vad som sker i projektet och framförallt vem som behöver hjälp med olika problem. Här upptäcks enkelt stora

problem, till följd av detta kan resurser enkelt omfördelas.

Att ha stå-upp-möten dagligen tar visserligen tid, och alla i gruppen behöver samlas på samma ställe, dessutom kanske man eventuellt behöver avbryta sitt arbete för stunden, men det är bättre än alternativet där gruppen har dålig koll på vad som behöver göras.

Vi tror att stå-upp-möten passar bäst för projektet med många medlemmar och stora delar som utvecklas parallellt, men för mindre projekt med färre inblandade personer, där alla i gruppen ändå kommunicerar på daglig basis, kanske det är överflödigt.

2.2.5 Trello

Ett av de bästa besluten vi tog var att använda oss av tjänsten Trello. Trello fungerar lite som digitala post-it-lappar där man kan placera uppgifterna gruppen har att göra i olika fack som "To do", "Doing", eller "Done". Man kan även lägga till medlemmar på en uppgift.

Trello har underlättat lång- och kortsiktig planering, och gör det enklare för varje medlem att hålla sig uppdaterad och gör även att man som gruppmedlem alltid har en uppgift att utföra.

Användningsområdet för Trello är ospecificerat och stort; det kan användas i alla möjliga sammanhang. Vi såg inga nackdelar med användandet av Trello då det var smidigt och inte alls tidskrävande. Ett alternativ kunde ha varit riktiga post-it lappar på en whiteboard-tavla, men eftersom vi inte hade ett kontor med en permanent whiteboard-tavla så var inte detta ett alternativ. Man kunde när som helst gå in på Trello och se vad som behövdes göras utan att behöva sitta hela gruppen tillsammans.

3. Diskussion

De olika teknikerna vi har använt har gjort arbetet lättstrukturerat och man har kunnat få en överblick på hur vi låg till i projektet. Att jobba i sprints gör att projektet blir mer självgående och det blir lätt att se vad man har fått gjort den senaste veckan.

Något som fungerade särskilt bra var arbetet med implementation i par och grupp, *parprogrammeringen*. Det blir betydligt lättare att lösa problem när man diskuterar sig fram och tar del av olika gruppmedlemmars kunskap och perspektiv på det hela.

3.1 Kommunikation i gruppen

Vi beslutade tidigt i projektet att vi jobbar till stor del tillsammans. Att arbeta på olika håll och hålla kontakt via sociala medier är inte optimalt, speciellt inte när så mycket av de tekniska bitarna är nya för många i gruppen.

Genom kontinuerliga möten, och även med hjälp av Trello, har projektets uppgifter kunnat delas upp i mindre delar, och gruppen har kunnat dela upp sig för att jobba effektivt på en viss funktion. Med hjälp av dagliga möten har gruppen dessutom kunnat hållas ihop och inte delas upp för mycket. Det har även varit bra att ha tekniska genomgångar med varandra; Att en grupp presenterar vad de har jobbat med, förklarar hur deras implementationer fungerar och vilka designval som har gjorts, för att alla ska hålla sig uppdaterade.

Trots att det ofta har fungerat väl med uppdelning i mindre grupper där var grupp jobbar med en specifik funktion, så är detta något vi även hade kunnat förbättra. Det är lätt att fokusera på en funktion, men svårt att hänga med i vad alla andra i gruppen gör. När funktioner sedan ska knytas ihop kan det vara svårt om man inte förstår alla delar fullt ut. En lösning på det problemet hade varit att rotera medlemmarna i gruppen varje vecka, så att man får jobba lite med alla funktioner. En nackdel med den lösningen blir att den som har specialiserat sig på ett område inte alltid får jobba med det denne är bäst på.

Ett annat problem har varit att vårt projekt har krävt 4 st Android-enheter för att kunna testas fullt ut, något som även löstes när vi satt i grupper och arbetade.

4. Slutsats

Att skapa ett multiplayer-spel har varit en utmaning; ett nätverk kan strula på fler sätt än man kan tänka sig. Det gör det ännu viktigare att jobba strukturerat och tänka i genom alla valen man gör noggrant. Det är också svårt att få alla i gruppen att förstå programmets olika delar såpass väl att man kan använda funktionerna utifrån, och detta problemet är troligen ännu större ute i näringlivet, där de flesta i ett IT-projekt kommer ha olika bakgrund och olika specialiseringar

Det är ändå roligt att man på sju veckor lyckas etablera en gruppstruktur, sätta mål och uppnå dessa mål. Det är lite tråkigt att projektet tog slut så snabbt, eftersom det är tydligt att arbetsprocessen fungerade bättre för varje vecka som gick.

Referenser:

Scrum.org, Scrum-guide 2013:

<https://www.scrum.org/Portals/0/Documents/Scrum%20Guides/2013/Scrum-Guide.pdf#zoom=100>

Don Wells 2013, Extreme Programming

<http://www.extremeprogramming.org/>

Trello, 2013

<http://help.trello.com/?source=helpdot>