

Team Reflection - Week 1

Group 3

24 april 2020

1 Customer Value and Scope

Vår användare är en vanlig motionär som gillar att spela golf och söker en användarvänlig applikation för att räkna golfslag. Efter konsultation med slutkund har behovet av en enkel slagräknare belysts. En räknare som kan anpassas efter hur många man vill spela på samma scorekort och efter hur många hål användaren vill gå.

Därför var första prioritet att skapa en grund för applikationen där huvudfunktionen är att räkna slag. Sedan specificerade användaren att det vore bra om man kunde räkna slagen för flera spelare på samma scorekort. Detta blev vår andra utmaning, att anpassa räknaren för att bli dynamisk och anpassa sig efter hur många spelare som ska nyttja ett scorekort. Nästa önskemål som användaren specificerade var att appen skulle ha ett sätt att automatisera uträkningen av extraslag beroende på spelarhandikap och banhandikap. Slutligen uttryckte kunden att det vore bra att kunna lagra gamla rundor, så att användaren kan hitta sin bästa runda golf på sin hemmaklubb t.ex.. Appen kommer att utvecklas i ordningen: enkel slagräknare, anpassat för flera spelare, uträkning av extra slag, lagra gamla golfrundor.

Gruppen har generellt tre mål inom kurser, där där samtliga mål är utan inbördes ordning:

- Alla ska känna sig nöjda med sin insats.
- Vi ska lära oss mer kring agil utveckling och Android utveckling.
- Att skapa en välfungerande applikation som skapar värde för kunden.

1.1 User stories / Job stories

Istället för att använda *user stories* har vi valt att använda *job stories* för att beskriva användarbehov. Dessa är snarlika, men vi anser att *job stories* kan vara mer precisa än *user stories*. Kunden har kommit med information kring vilka behov denne har. Utifrån den informationen har vi sedan skapat *job stories* som tydligt beskriver det kunden efterfrågar.

En uppskattning på vilken ansträngning en *job story* estimeras. Sedan har gruppen gemensamt valt vilka *job stories* som kommer vara med i sprinten. Gruppen har sedan gemensamt brutit ner varje *job story* i uppgifter och delat upp vem som gör vilken uppgift. Detta arbetsätt har gjort att vi har kunnat skapa värde på ett bra sätt då all funktionalitet utgår från kunden.

1.2 Development Process

För att övervaka utvecklingens gång använder vi tre verktyg. Gitinspector, velocity och tester. Gitinspector använder vi för att få detaljerad information kring koden utvecklas.

Vi har försökt estimera velocity med hjälp av trello och poängsättning av tasks. Poängsättningen görs med Fibonaccital och är ett ungefärligt estimat i timmar. Detta för att försöka skapa jämn arbetsfördelning men också se hur mycket, eller lite som görs under en vecka.

Under första veckan av utveckling var det hög velocity där alla tasks som sattes vid sprint-planering utfördes, men även sådana tasks som tillades under sprintens gång. Däremot var det sämre velocity under andra veckan där inte alla tasks har slutförts. Utökningsvis finns det förbättringspotential i arbetsfördelning, då alla tasks inte utförts kan det peka mot att det inte representativt delats ut efter individuell velocity.

Slutligen skriver vi tester på alla klasser i modellen för att säkerställa att koden är fungerande men nu våra test klasser är bara för modellklasser men kommer vi att utveckla det. Det är svårt att utvärdera hur bra verktygen fungerar då vi endast har haft två sprintar. Men vi tror att de kommer hjälpa oss enormt.

2 Design Decisions and Product Structure

Nedan presenteras de strategier och verktyg som har använts hitintills i projektet i syfte att skapa värde för kunden och för att säkerställa kvalitet på den kodbas som tagits fram.

2.1 How Your Design Decisions Support Customer Value

Allt utefter projektets fortlöpande har en tydlig dialog förts med intressenten mot detta projekt, i syfte att redan i tidigt skede lägga en grunddesign som kommer lämpa sig för användaren. Som nämnt under avsnitt 2.2 har intressenten gett förslag på funktioner som hade varit önskvärda i den typ av applikation som detta projekt kretsar kring. En osäkerhet som gruppen av utvecklare tog i beaktning var att en intressent inte alltid lyckas specificera de funktioner hen egentligen vill ha.

För att utröna ovanstående problem var arbetslagets initiala strategi att ta fram och implementera funktioner, utöver de av intressenten specificerade funktionerna, i syfte att underlätta för vederbörande att ge åsikter om vad användare kan tänkas vilja ha. Dock kom gruppen till insikten att detta tillvägagångssätt inte nödvändigtvis är önskvärt eftersom intressenten tvingas ta ställning till funktioner som hen kanske inte hade reflekterat över annars. Därmed skulle strategin kunna leda till att intressenten börjar ge förslag på funktioner utefter den linje som lagts fram av utvecklarna snarare än att utgå ifrån vederbörandes egna erfarenhet. Framöver är gruppens plan att återgå till att fokusera mer på de idéer som läggs fram av intressenten och att genomföra dessa så bra som möjligt med avseende på kundbehov och kvalitet på applikationen. Vad det kommer till det sistnämnda förs vidare resonemang i efterföljande avsnitt.

2.2 How You Ensure Code Quality And Enforce Coding Standards

För att försäkra oss att kodbasen har en bra struktur som stöttar utveckling utan oväntade sidoeffekter har vi som mål att ha 100% av modellen testad. Då testning av grafiska gränssnitt är av okänd svårighet har vi valt att exkludera detta från som mål. Genom att utveckla mjukvaran så att modellen kastar undantag kan vi istället hantera dessa undantag i gränssnittet.

För att försäkra oss om att vi har en etablerad standard har vi valt att nyttja den programmeringsstack och arkitektur som återfinns i Android Jetpack [1]. Jetpack innefattar också en programvaruarkitektur, MVVM (Model-View-ViewController), som vi följer för att bäst stödja utveckling inom teamet men även fortsatt utveckling där eventuellt nya utvecklare introduceras till projektet. Genom att ha en etablerad standard underlättas utvecklingen genom en enklare konfliktlösningsprocess. Genom att ha ett etablerat ramverk som specifikt är byggt för Android-utveckling kan man ha dokumentation som slutord vid meningsskiljaktigheter gällande designbeslut.

Referenser

- [1] Google, *Android Jetpack — Android Developers*, 2020. URL: <https://developer.android.com/jetpack> (hämtad 2020-04-24).