

Team Reflection

Tester

Vi har upptäckt att det är svårt att kunna skriva tester för någon annans kod. Det kan alternativt beror på att vi måste skaffa en rimlig kodstruktur för projektet genom att vi exempelvis tydligt kommenterar koden så att man lättare kan förstå hur olika klasser interagerar och så vidare. Ett ytterligare alternativ kan vara att vi delegerar rollen som testare till alla gruppmedlemmar så att alla skriver sina egna tester efter man har jobbat klar med sin del. Ett till problem som vi upptäckte är att det inte verkar effektivt att skriva Widgets tester (i dagsläget) eftersom att vi inte har bestämt hur appen kommer se ut designmässigt.

En lösningsförslag kan vara att först skapa en Integration tests eller Continuous integration services så att vi får en global bild på hur bra appen fungerar därmed starta med Widgets tester när vi har väl kommit igång med designdelen.

Kod design

Vi har märkt att det arkitekturella mönstret som vi designar koden på är relativt osmidigt. detta märktes tydligt när vi delade upp arbetet i ansvarsområden och blev insnöade på specifika delar vilket gjorde att man fick dålig koll på hur resten av kodandet fungerar. Dessutom har vi märkt att det behövs en bättre teknisk dokumentation i form av exempelvis klassdiagram eller komponentdiagram. Det kan bli väldigt ineffektivt och rörigt utan något som styr upp.

User stories

Vi tog upp förra veckan att vi skulle börja använda fibonacciföljden för att estimerar hur mycket tid en storie förväntas ta. Den här veckan använde vi oss av det och vi upplever att det fungerade bättre. Även om vi beräknade fel på vissa punkter så fick man en bättre översyn på hur mycket tid som skulle gå in denna veckan när vi inte längre hade så breda estimates. Totala beräkningen blev dock lite missvisande eftersom de ofärdiga user stories som skulle med till denna sprinten hoppade över med sina fulla poäng vilket gjorde att vi fick högre total än vad vi tror att vi kan prestera per vecka. Det är något att ta med

sig att kanske beräkna om dem eller att inte lägga in för många nya så att vi inte beräknar att jobba över vår förmåga.