

Team reflection lv7

Översikt

En stor förbättring av denna sprint var planering, samarbete, testning/kommentarer och daily scrum. Planeringen har gått bra på grund av alla kunde koda effektiva och blev tydligt tilldelade stories samt övriga uppgifter som fick saker gjorda. Sedan erhöll gruppen en mycket tydlig diskussion med och hjälpte varandra i kodande samt utpeka det viktiga i koden och försöka anpassa det efter vad som är viktigt för kunder. Detta har sedan påverkat gruppens daily scrum där vi kunnat hitta ett bra beroende och man ständigt uppdaterar varandra förhållandevis till hur man arbetat med appen. Testning och kodande fick en tydlig struktur och detta är för det mesta för att vi började bli mer engagerade i test code och code review samt hittat ett flöde i hur man arbetar med detta.

Det som tycks kan förbättras är att skriva nya stories. Gruppen känner att man approximativt har samma stories sedan 3 veckor tillbaka och vill gärna uppdatera dem. Delvis för att vi fått mycket feedback från kunden men också för att gruppen blivit mera självmedveten angående kodande. Stories förväntas uppdateras inför nästa sprint. gruppen förväntar sig också att ha ett fortsatt arbete med tester/kommentarer.dw och skapa en UML-diagram för klasserna.

Scope

Det har skett en förbättring när det kommer till att samarbeta i gruppen. Kommunikationssvårigheter lyftes upp i början av sprinten för att kunna underlätta och förbättra arbetsmiljön. Resultatet visar på en ganska bra förändring till att gruppens arbetseffektivitet blev rätt så bra.

Tester

Vi har kommit fram till att det blir svårt för code testers att skriva alla tester och vi bestämde att alla skriver tester på sin egna kod, men att code testers har en lite mer administrativ uppgift inom detta istället, code testers kan därmed agera som hjälpmedel till andra gruppmedlemmar och se till att alla tester är bra gjorda. Anledningen till detta skrev vi lite om i förra team reflections, och det är att den som skrev koden känner till den bäst, och det är därmed enklast för den som skrivit koden att testa den.

Angående widget tester så ser det krångligt ut i dagsläget och vi kommer att titta vidare på det för att hitta en lösning.

Teknisk dokumentation

Vi har under veckan fixat en lathund för hur vi ska komma igång med Dart Doc vilket är darts standard för att dokumentera kod. Det kommer hjälpa oss att förstå koden vi skrivit bättre eftersom vi även på egenskrivna klasser kan hovra över namnen på klasser, metoder osv för att få reda på information om dessa. Då slipper vi gå in i

klassen och behöva läsa hela och lista ut vad den gör. Målet är att få till en liknande lathund för UML-diagram för att få ytterligare förståelse för hur appen är uppbyggd. Det kommer göra att vi lättare kan se var man kan komma åt olika funktioner.

Uppdatering av dokumentation

Vi har en tendens att bestämma nya arbetssätt som t.ex. att använda UML-diagram, Dartdoc eller olika system för tester. Men rutiner och konkreta arbetssätt finns ofta inte på plats förrän i mitten av veckan efter. Detta leder ofta till att flera lagmedlemmar inte börjar jobba riktigt så förrän veckan därpå. Det leder fortsättningsvis till långa ledtider för förändringar och att det kan ta ytterligare ännu längre innan man ser resultatet av en förbättring. Dessutom kan det också få vissa förändringar att felaktigt framstå som onödiga eller krångliga och kanske rentav skrotas då alla inte hunnit anamma arbetssättet. Detta händer bland annat med vårt system för peer-review som dog i sin vagga. I fallet dokumentation har ovanstående fått effekten att vi har en dokumentationsskuld vi kommer behöva hantera nästa vecka.