

TDT4140 - Retrospektiv 2
Sprint 2 (kodenavn *Black Hawk*)

Gruppe 29

Innhold

1 Introduksjon	2
2 Hva vi gjorde bra	2
3 Hva vi gjorde mindre bra	3
4 Hva vi skal forbedre	3
4.1 Pushe kode oftere til Master og Heroku	3
4.2 Bruke testing aktivt i utviklingsprosessen	3
4.3 Mestre tidssonefunksjonaliteten til rammeverkene våre	4
Bibliografi	4

1 Introduksjon

Gruppen møttes på kafé for å holde retrospektiv for sprint to. Det ble utdelt 3 lapper til hvert gruppemedlem. På hver av lappene skrev vi om det som ble gjort bra, mindre bra samt konkrete aksjonspunkter til forbedring basert på sprint 2. Deretter diskuterte vi hvert punkt for å få dypere innsikt i hva som var vellykket og forbedringsmulighetene fremover.

2 Hva vi gjorde bra

Vi har fortsatt å holde oss til Scrum-rollene vi fordelte i gruppekontrakten, slik at ansvarsområdene var like tydelige. Rutinene rundt Daily Scrum ble fulgt strengere, noe som førte til større klarhet overfor hva som var forventet av oss, samt hva som måtte gjøres. Også rundt brukerhistoriene har vi forbedret oss. Vi satte oss et konkret mål om å bryte ned brukerhistoriene til enda mindre underoppgaver, som gjorde at vi fikk en bedre forståelse av når brukerhistoriene var helt ferdig. En annen positiv virkning med flere underoppgaver (tasks) var at vi lettere kunne fordele dem jevnt innad i gruppen. Det gjorde at vi individuelt fikk innblikk i flere forskjellige brukerhistorier. Videre er gruppen fornøyd med godt oppmøte til arbeidsøker, og at vi har jobbet svært mye sammen i henhold til gruppekontrakten. Dette har gjort det enkelt å kommunisere om problemer som oppstod underveis i sprinten, og har skapt gode muligheter for parprogrammering som var et annet aksjonspunkt vi ønsket å jobbe mer med.

Release-planen vi ferdigstilte i første sprint, ble endret før sprint to. Etter samtale med produkteier gjorde vi en omprioritering av brukerhistoriene, og endte til slutt opp med færre brukerhistorier enn vi først hadde planlagt. Dette hjalp oss å stille forberedt på uforutsigbare krav fra produkteier, og ga oss bedre utgangspunkt til å fokusere på kjernefunksjonaliteten. Det at release-planen var gjort enklere gjorde at vi fikk mer tid til å kvalitetssikre hver brukerhistorie. Vi hadde jevn progresjon etter vi kom i gang, og vi ble ferdige til planlagt tid.

Det ble nevnt i det første retrospektivet at vi er glade for at vi i gruppekontrakten satte oss klare mål for karakteren gruppen vil oppnå. Dette har bidratt til å skape en konsensus rundt hvor mye arbeid som må legges ned hver uke for at vi skal nå våre respektive mål. Det gode samarbeidet vi opplevde under forrige sprint har fortsatt under denne sprinten. Til forskjell fra tidligere føler vi at vi har tatt samarbeidet et steg videre ved å bli flinkere til å involvere hverandre i egen kode. Gjennom diskusjoner og deling av erfaringer har vi kommet frem til mer effektive måter å implementere ny funksjonalitet på.

Samtlige medlemmer av gruppa er fornøyd med produktet vi nå leverer. Vi føler at vi har kommet produktkravene i møte på en tilfredsstillende måte, og at den viktigste funksjonaliteten er i god stand. Dette kan ha kommet av en gjennomgående godt fungerende gruppedynamikk, og at gruppemedlemmene har vært flinke til å samarbeide om deling av kunnskap og ferdigheter.

3 Hva vi gjorde mindre bra

For det første har gruppen for lite kunnskap om Git. Dette har ført til unødvendig tidsbruk og flere tilfeller av merge-conflicts. Selv om vi ble flinkere til å benytte oss av Git igjennom sprinten, slurvet vi med å pushe ny kode til master-branchen så ofte som vi planla.

I forrige retrospektiv ble vi enige om å bli flittigere rundt testing av kode. Dette er noe vi har forbedret i denne sprinten, men det er fortsatt rom for forbedring. Det ble skrevet mer testkode i denne sprinten, men vi var ikke flinke nok til å bruke testene vi hadde skrevet. Vi kunne også skrevet grundigere tester som omfattet flere scenarier. Grunnen til at vi ikke så en stor forbedring fra forrige sprint kan skyldes et sterkt ønske om å bli ferdig og en holdning om at "manuelle" tester er godt nok. Dette kunne muligens vært unngått med "Test-driven development".

I tillegg har gruppen opplevd "bugs" som har utfordret vår tekniske kompetanse. Et eksempel er en tidssone-bug, der serverens tidssone har ligget en time før vår egen. Det er gått mye tid til å fikse dette, og vi ikke har klart det. Vi har dog ikke prøvd alt - vi kunne for eksempel oppsøkt andre med liknende erfaringer på nettforum eller hos andre studenter. Dette gjelder også andre mindre bugs vi har mislyktes i å fikse.

Til tross for at vi har vært flinkere til å holde Daily Scrums under møtene, har vi ikke nådd vårt mål om å også gjennomføre dem over Slack når vi har jobbet individuelt.

4 Hva vi skal forbedre

Basert på diskusjonen over har gruppen valgt å fokusere på følgende tre aksjonspunkter.

4.1 Pushe kode oftere til Master og Heroku

For å opprettholde en konstant funksjonell nettside hadde vi planlagt å benytte oss av lokal server for å teste ny kode, for deretter å overføre velfungerende kode til vår master-branch og Heroku-nettsiden. Til tross for at vi til en viss grad har benyttet oss av dette, anser vi det som et viktig aksjonspunkt å forbedre bruken av denne metodikken. Dette vil være hensiktsmessig fordi det tvinger kvalitetssikring av kode underveis i prosessen, i tillegg til at vi gradvis får oversikt over produktets fungerende elementer. Det vil trolig dermed bli mindre behov for debugging og følgefeil underveis i utviklingen, og gruppen vil spare tid og frustrasjon mot slutten av neste sprint.

4.2 Bruke testing aktivt i utviklingsprosessen

Et aksjonspunkt vi ønsker å innføre er aktiv bruk av testkode i programvareutviklingen. Fremover blir det et krav om at eksisterende tester må kjøres før ny kode blir integrert i develop- og master-branchen. Gruppen skal prøve ut "Test-driven development" for å "tvinge" samtlige gruppemedlemmer til å lære mer om og skrive bedre tester.

4.3 Mestre tidssonefunksjonaliteten til rammeverkene våre

Underveis i sprinten ble vi i flere tilfeller møtt av problemer med tidssoner som ikke samsvarte mellom Django, Heroku-nettsiden, og vår egen kode. Overgangen til sommertid kompliserte problemet ytterligere. Med tanke på at vårt produkt er en reservasjonsside, er dette en utfordring som bør håndteres på en skikkelig måte. I stedet for å forsøke å fikse feilene fortløpende skal gruppen på forhånd bruke tid på å sette seg inn i de ulike rammeverkenes implementasjoner av tidssonefunksjonalitet. På denne måten kan vi sikre at koden er kontinuerlig fungerende underveis i sprinten.