

Start av Oblig4

Disse metodene skulle vi bruke i Oblig 4:

- Scrum
- Parprogrammering
- Burndown chart
- Planning poker
- Skrive tester

Etter Oblig 3 hadde vi bestemt oss for å bruke Scrum som utviklingsmetode der vi skulle ha fast møte 1 gang i uken. Vi skulle sitte to eller tre i lag som jobbet med det samme slik at vi tok i bruk parprogrammering. Alle skulle også skrive tester til sine egne metoder. Vi prøvde å bruke Burndown chart, men vi fant ingen god måte å bruke det på. Burndown chart ble da droppet. Vi ble også enige om å bruke Planning Poker slik at vi kunne fordele arbeidet ganske likt mellom personene i gruppen. Vi skulle fortsette med Slack for å snakke i lag, og vi skulle bruke Trello for å ha Scrum boards. Vi skulle bruke Pipeline i Gitlab, men Håvar holdt fortsatt på med å sette dette opp.

Sprint 1

Vi delte oss inn i par eller tre og tre. Noen skulle jobbe med GUI, andre med generelle klasser og andre med tegning.

Sprint 2

Vi var ikke veldig flinke til å bruke Trello på riktig måte, men det gjorde ikke så mye ettersom Håvar fant ut at det var bedre å bruke Issues på Gitlab. Det var lettere og vi kunne få Scrum-lignende boards. Vi fant også ut at Planning Poker tok litt lang tid i forhold til den tiden vi hadde på denne obligen. Vi endte derfor opp med å bare diskutere hvor vanskelig det var og så fordele oppgavene på de som hadde jobbet med noe lignende i Sprint 1.

Sprint 3

Det var nå organiseringen kom skikkelig på plass og alle hadde blitt vant med hvordan arbeidsflyten vår var. Nå fordelte vi ikke lenger oppgavene på samme måte som tidligere fordi alle oppgavene som skulle bli gjort lå på git. Dette betydde at du selv valgte et issue som du ville jobbe med, og skrev at du holdt på med det.

Sprint 4

Organiseringen fungerer på samme måte som i Sprint 3.

Hvordan vi er organisert

Vi valgte å dele oss inn i to grupper: Programmerings gruppe og Art gruppe. Vi brukte en forenklet git flow metode som er bedre forklart i CONTRIBUTION GUIDE på git-Prosjektet, hvor alt baserer seg på å løse issues.

Vi fastsatte 5 roller som hjalp med å gi bedre kommunikasjonsflyt og oversikt:

- Kommunikasjonansvarlig
- Git-ansvarlig
- Juridisk ansvarlig
- Kode ansvarlig
- CoachGunnar

Ting som kunne bli løst bedre

- Dårlig prioritering av issues
- Definerings av arbeidsoppgaver
- Dårlig label setting på issues