

Ohjelmistotuotanto syksy 2021

Miniprojekti

Raportti

Team Boa

Maikki Heijala

Amanda Hämynen

Helmi Karesti

Juhana Karhunen

Petro Lehtonen

Mikkel Leskelä

Ryhmän työskentely eteni hyvin jouhevasti läpi kaikkien sprinttien. Jo ensimmäisen asiakastapaamisen jälkeen saimme GitHubiin nopeasti committeja ja työn aluilleen. Eri vaiheissa projektia oli kuitenkin joitain haasteita. Ensimmäisessä sprintissä alkukonfiguroinnit veivät todella aikaa ja yksi ryhmäläisistä teki paljon ylitöitä yrittäessään saada GitHub Actionsit toimimaan. Ongelmana oli, että yksikkötestimme käyttivät ja muokkasivat suoraan tietokantaa, mitä emme saaneet mitenkään implementoitua Actionsiin. Koko ryhmä ponnisteli ongelman ratkaisemiseksi, mutta päädyimme luopumaan testien suorittamisesta palvelimella kokonaan. Testejä suoritetaan siis vain paikallisesti ja uudelleen alustetulla tietokannalla. Actionsin viraksi jäi ainoastaan pylint-tarkistaminen. Myöskään Coverage-liitännäistä ei tämän palvelimella suoritettavien testien ongelman vuoksi saatu toimimaan, vaan päädyimme antamaan ohjeet testikattavuuden generoimiseksi sekä PDF:t lopullisiin paikallisesti ajettuihin testikattavuusraportteihin README:ssä.

Toisessa sprintissä työskentely oli nopeaa, sillä siinä toteutettiin pääasiassa samoja toiminnallisuuksia eri medioille (artikkeli, video, kirja), joita oli toteutettu ensimmäisessä sprintissä blogille. Kaksi ryhmäläisistä kokeili parityöskentelyä, mikä toimi hyvin: Todettiin, että on hedelmällistä nähdä, missä järjestyksessä toinen lähtee ominaisuutta toteuttamaan. Parilta sai myös hyviä vinkkejä teknillisiin pulmiin ja parhaillaan työskentely oli nopeaa ja tarkkaa. Ominaisuus saatiin parin tunnin sessiossa rakennettua yksikkö- ja robot-testeineen valmiiksi. Yhdessä saatettiin myös huomata puutteita muussa valmiissa koodissa. Myös kolmannessa sprintissä parikoodaamista tehtiin kahdessa parissa, mutta yhdessä kokeilussa työskentelytapa hieman ontui siten, että toiminnallisuuden, jota lähdettiin kehittämään, olikin jo toinen parista ehtinyt tehdä tuotantovalmiiksi saakka ennen sessiota. Tämäkin kokeilu todettiin kuitenkin opettavaiseksi.

Kolmannessa sprintissä ehdollisiksi jätetyt viimeiset user storyt jätettiin toteuttamatta. Toisessa niistä olisi laajennettu hakutoiminnallisuutta käyttämään useampaa hakusanaa ja toisessa lisätty 'luettu'-tieto kirjavinkkiin. Jokainen laskeskeli tahoillaan, että sovelluksen saattaminen turvallisesti

tuotantomuotoon ilman, että ylittäisimme viikkotyöaikaa, tuskin antaisi tilaa vielä tietokannan muuttamiselle ja ominaisuuksien lisäämiselle. Testien ja tyyllivaatimusten ylläpitäminen vei kuitenkin aina lisätyöaikaa. Keskustelimme paljon Discordissa löydöksistämme ja jaoimme korjausvastuuta. Viimeisessä sprintissä esimerkiksi huomattiin, että tyhjillä merkkijonoilla rekisteröityminen rikkoi sovelluksen, mutta tämä ehdittiin korjata.

Videotapaamisia etäryhmällämme oli viikoittain kolme: asiakastapaamisen jälkeen pidettiin yhteinen noin tunnin sprintin suunnittelu. Suunnilleen puolivälissä sprinttiä oli yksi 15 minuutista puoleen tuntiin kestävä daily scrum ja sprintin lopussa ennen asiakastapaamista retrospektiivi, joka kesti puolesta tunnista tuntiin. Daily scrumeja olisi ehkä voinut olla vielä ainakin yksi lisää, sillä jokaisessa retrospektiivissä keskityttiin myös paljon vielä jonkin asian kirjaamiseen tai korjaamiseen sovelluksessa tai GitHubissa. Yksi daily scrum ennen retroa olisi ehkä voinut raivata palautetilaisuuteen enemmän tilaa itse sen reflektio-osuudelle. Mitään muodollista retrospektiivitekniikkaa emme oikein ehtineet käyttää.

Kaikki itse sovelluksen ominaisuuksien kehittämisen ulkopuolella tapahtuva työmäärä yllätti projektissa. Esimerkiksi viimeisessä sprintissä koko ryhmän laskennallisesti käytettävissä olevasta 36 tunnista itse user storyjen toteuttamiseen olimme resursoineet noin 13 tuntia, mutta jokaisen henkilökohtainen työaika kyllä tuli täyteen, kun otettiin huomioon myös edellisten viikkojen ylitöitä. Burndown-kaavioiden ja backlogin toteuttaminen ja ylläpitäminen veivät runsaasti aikaa, kuten myös riittävä vaadittu dokumentaatio. Oman työajan arviointi ja käytetyn työajan kirjaaminen pohdituttivat: jos jotain asiaa oli vain tutkinut, mutta se ei päätenyt toteutukseen asti, saattoi sen merkkiaaminen työskentelytunteihin arveluttaa.

Ryhmässä opittiin paljon uutta. Yksi ryhmäläisistä ei ollut tehnyt web-sovellusta aiemmin, paljon kerrattiin yksityiskohtia myös Tietokantasovellus-kurssin materiaalista ja GitHubin palvelintestien toimiminen tietokannan kanssa vaati paljon tutkimusta, vaikkei niitä lopulta toimiviksi saatu. Jatkuvat kirjaamiset ja backlogin ylläpito tuntuivat välillä turhilta. Joskus tuntui myös, että yksikkö- ja robot-testejä tehdään itse testien takia eikä ylläpidettävyyden varmistamiseksi.

Lopullisessa koodissa on paljon toisteisuutta, kun erilaisten vinkkimedioiden lisäämiset ja muut toiminnot ovat kopioita toisistaan. Jos sprinttejä olisi vielä ollut, koodia olisi ainakin näiltä osin voinut refaktoroida. Kovasti pohdimme vielä, miten tietokantaa käyttävien metodien testejä voisi

toteuttaa eristettynä varsinaisesta tietokannasta. Nyt jouduimme alustamaan tietokantaamme aina uudestaan testien ajamisen jälkeen. Herokuun alustimme tyhjän tietokannan.