

Raportti - Code Teddies

Sulevi Sihvola
Mikael Heino
Ville Saarinen
Samu Vaittinen
Miia Rämö

Valitsimme ensimmäisessä sprintissä toteutettaviksi user storyt “as a user I can add an article reference” ja “as a user I can print a reference as a BibTex”. Tässä sprintissä haasteellisimmiksi tehtäviksi nousivat Cucumber-testit, jotka lopulta toteutettiin näille user storyille vasta toisessa sprintissä. Olimme alunperin ymmärtäneet, että burndown -kaaviosta tulisi käydä ilmi, kuinka suuri osa viikottaisesta työmäärästä ollaan jo käytetty, ja paljonko sitä on käyttämättä. Saimme product ownerilta sprintin katselmoinnissa hyvät selkeät ohjeet oikeaoppiseen burndown -kaavioon ja ymmärsimme, miksi alkuperäinen ajatuksemme siitä ei olekaan toimiva käytännössä. Ylipäättään uuden projektin aloittaminen tyhjästä oli kaikkien erilaisten tekijöiden takia pelottavaa ja omien rajallisuuksien huomaaminen, kun yksinkertaisiinkin asioihin meni paljon aikaa, aiheutti päänvaivaa tuntikirjanpidossa.

Toiseen sprintiin valitsimme toteutukseen user storyt “as a user I can export references as a Bibtex-doc”, “as a user I can add a book reference” ja “as a user I can add an inproceedings reference”. Lisäksi, kuten aiemmin mainitsimme, saatoimme ensimmäisen sprintin storyt definition of done -määritelmän mukaiseksi toteuttamalla niille toimivat Cucumber-testit. Sprintin aikana suoritimme myös refaktorointia.

Kolmannessa sprintissä valitsimme toteutukseen storyt “as a user I can add all possible fields”, “as a user I can store references” ja “as a user I can list all references”. Näiden lisäksi refaktorimme jo olemassa olevaa koodia ja Cucumber-testejä sekä laajensimme Wrapper-luokkamme toiminnallisuutta sopivaksi. Skandimerkkien toteuttamisessa meillä oli pieni väärinkäsitys josta

katselmoinnissa sitten ohjeistettiin tarkemmin. Tässä sprinttissä lisäsimme myös MongoDB:n, mikä aiheutti haasteita .jar-tiedoston luomisessa.

Neljänteen sprinttiin valitsimme user storyiksi “as a user I can input references with scandics”, “as a user I can delete a reference” ja “as a user I can update a reference”. User storyjen toteutuksen lisäksi refaktoimme koodia reippaasti sekä tuotimme ajantasalla olevan luokkakaavion. Ehkä suurimmat ongelmat ilmenivät tämän sprintin aikana kun pyrimme saavuttamaan projektin loppuun ja halusimme varmistaa että koodi tosiaan oli siistiä.

Ryhmän jäsenten tiukat aikataulut vaikeuttivat työskentelyä läpi koko projektin. Seuraavalla kerralla työskentelimme suuremman osan ajasta yhdessä kommunikaation parantamiseksi. Myös user storyjen työmäärien estimoiminen yhdessä projektin alkaessa olisi helpottanut sopivan työmäärän valinnassa eri sprintteihin. Selkeät yhdessä sovitut työskentelyajat, joko paikanpäällä tai etänä, olisivat tukeneet kaikkien user storyjen toteutusta. Demotilaisuudessa jouduimme hieman tukalaan tilanteeseen tietokantayhteyden kanssa, sillä MongoDB ei toiminutkaan suoraan jar-tiedostosta ja jouduimme ratkaisemaan ongelman viime hetkellä. Tilannetta vaikeutti myös se, että tietokantayhteyden toteuttaja ei päässyt mukaan demotilaisuuteen, koska aikataulumme eivät sopineet yhteen. Tässä suhteessa opimme, että tulevaisuudessa esittelytilaisuudet edellyttävät huolellisempaa valmistelua ja parempaa kommunikointia. Lisäksi aikataulujen sopimattomuus muistutti siitä, että yhdenkään ryhmän jäsenen vastuulla ei pitäisi olla koko järjestelmän toiminnan kannalta oleellisia asioita.

Onnistuimme kuitenkin pitämään yleisesti prosessin kasassa ja pysymään perillä tehdystä työstä loppuun asti. Pidimme sprinttien jälkeen lyhyet kokoukset joissa kävimme läpi mikä edellisellä viikolla meni hyvin ja missä oli vielä parantamisen varaa. Teimme jokaisen sprintin alussa myös karkeaa tehtäväjakoja kahden pääkriteerin mukaan; millainen aikataulu kyseisellä ryhmänjäsenellä oli sprintin ajaksi varattuna (resurssit) ja millaiset tehtävät kutakin kiinnostivat. Näin jokainen pääsi tekemään sellaisia asioita, jotka itseä kiinnostivat, mikä paransi työskentelymotivaatiota myös yleisellä tasolla. Koodin integroinnin hoidimme mallikkaasti ‘pull requesteja’ käyttäen. Pull requestin tarkisti joku tiimin jäsen,

kommentoi mikäli oli tarve ja 'mergesi' sen master-haaraan, jonka jälkeen Travis hoiti integraatiotestauksen.