

Juuso Strömberg
Essi Salmenkivi
Sami Åhl
Hilla Fred
.IO

Sprint 1

Ensimmäisen sprintin alussa kokoonnuimme ryhmässä miettimään projektille asetettuja vaatimuksia. Loimme user storyt ja niiden pohjalta vielä pyrimme tekemään yksittäisiä taskeja. Samalla mietimme, että minkälaisilla teknologioilla lähtisimme toteuttamaan. Aluksi olimme valinneet Java Spring-frameworkin ohjelmointikieleksi ja -kehikseksi. CI-työkaluksi valitsimme Jenkinsin.

Kuitenkin jo heti sprintin alussa huomasimme valintamme mahdottomuuden. Kenelläkään meistä ei ollut kovin päteviä tietoja kyseisestä frameworkista, ja Jenkinsin konfiguroinnin raskaus aiheutti jokaiselle harmaita hiuksia. Päädyimme lopulta valitsemaan toteutuskieleksi Rubyn ja sille ominaisen Rails-frameworkin. CI-työkalun vaihdoimme Travisin ja tuotantoympäristöksi Herokun. Huomasimme, että näiden palveluiden helppous sopi paljon paremmin miniprojektin kokoon.

Ensimmäinen sprintin aikana saimme lähinnä kehitysympäristön ja -pipelinen valmiiksi tulevia sprinttejä varten, sekä referenssimallin ja sille simppelit näkymät. Työ jakautui valitettavasti aika epätasaisesti ryhmämme kesken, johtuen teknologioiden tuntemattomuudesta. Backlogin kanssa työskenteleminen tuntui myös vielä uudelta asialta.

Sprint 2

Toisen sprintin alkuun suurin prioriteetti oli saada ohjelmointiympäristöt ajan tasalle kaikilla ryhmän jäsenillä. Osalla oli vielä ongelmia saada rubya yhteensopivaksi käyttöjärjestelmänsä kanssa.

Päästessämme käsiksi itse projektiin, oli korkeimmalla Sprintin backlogissa saada Definition of Done mukaisesti valmiiksi kaikki ensimmäiseen sprinttiin luvatut User Storyt ja niihin liittyvät taskit. Kun näiden taskien toiminnallisuus ja testaus alkoi näyttämään hyvältä, siirtyi osa ryhmästä jo seuraaviin taskeihin. Huomattavaa päänvaivaa ryhmän keskuudessa aiheutti kuitenkin dropdown -valikon toimivaksi saaminen referenssin luontimonisteessa. Ongelma korjaantui kuitenkin äärimmäisen toimivan suklaanvälityksen kautta ja ongelma saatiin ratkaistua.

Erona ensimmäiseen sprinttiin, työskentelimme enemmän yhdessä ja saimmekin myös tehokkaammin työstettyä taskeja pois alta. Osa työskenteli yksin, muutama kokeili pariohjelmointia. Puolet tiimistä vielä taisteli lähes tuntemattoman Ruby and Railsin kanssa. Aina oli kuitenkin vähintään irkin päässä joku joka osasi selittää ja keneltä kysyä. Myös gitin jatkuva työstäminen alkoi nopeatumaan kaikilla.

Sprint 3

Kolmas ja viimeinen sprintti sujui jo huomattavasti sutjakammin kuin kaksi edellistä (ehkä osilta sen vuoksi, ettei kenenkään tarvinnut käyttää vähäistä aikaansa ja hermojaan konffauksen parissa). Ryhmä saatiin kerralla paikalle jo kokonaisuudessaankin, jolloin työn tehokkuus oli lähes päätä huimaavaa.

Sprintissä saatiin viimeisteltyä kaikki keskeneräiset User Storyt ja valmiiksi kolmanteen iteraatioon valitut taskit. Mukaan myös otettiin Capybara-testausympäristö ja vihdoinkin myös burndown -käyrä.

Osalla ryhmästä työskentelytahti oli vielä kohtuullisen hidasta, koska uuden kielen ja kehitysympäristön opetteleminen näin lyhyessä ajassa on kuitenkin melko haastavaa. Suurin jännä kakka kuitenkin tuli vain hetki ennen demotilaisuutta, jolloin viime hetken merge hajotti jotain ohjelmassa. Tästä kuitenkin selvittiin viime hetken viillauksella eikä varsinaista ns. demo-efektiä tullutkaan.

Yhteenveto (mitä opimme, mitä pitäisi parantaa)

Konffaukselle varattava aikaa -> ei ota niin paljoa tekemistä ekaan sprinttiin koska taskien keston arvioiminen etukäteen on kohtuullisen haastavaa, jos ei ole tiiminä aiemmin työskennellyt yhdessä. User Storyt kannattaa pilkkoa todella pieniin osa-alueisiin, jolloin yksittäisten taskien valmiiksi saaminen ja ajankäytön arvioiminen helpompaa. Ryhmätyöskentely on tehokkainta konkreettisesti ryhmänä työskennellessä. Toki ymmärrettävää, että kaikilla ryhmän jäsenillä oli myös muut koulu -ja työkiireet projektin lisäksi, jotka aiheuttivat ongelmia yhteisen ajan löytämisessä. Ryhmän sisäinen kommunikaatio on tärkeää -> hyvä tietää mitä toinen tekee milläkin hetkellä tai mikä aiheuttaa hänelle vaikeuksia. Tällöin ei tule päällekkäisyyksiä ja mahdollisesti pystyy auttamaan toinen toistaan. Tässä onnistuimmekin melko hyvin: emme joutuneet käyttämään merge-konfliktien selvittämiseen paljon aikaa. Vielä tehokkaampaa työskentely olisi ollut, jos koko ryhmän tapaamisia olisi onnistuttu järjestämään aivan projektin alkuun. Tällöin projekti olisi ehkä päässyt nopeammin täyteen vauhtiin.

