

Loppuraportti

DROP DATABASE; Kristian Krok, Saara Satokangas, Laura Sinikallio & Sini Suihkonen

Sprint 1

Ensimmäinen sprint sujui melko hyvin, laadimme pohjan lopulle projektin kehitykselle. Taskien kestojen arviointi oli kylmiltään melko vaikeaa, esimerkiksi konfiguraatioiden viemän ajan arvioimme alakanttiin. Kun miniprojektissa ei ollut nollasprinttiä, meille oli hieman epäselvää miten konfiguraatioihin ja projektin alustukseen liittyvät taskit merkitään. Vievät kuitenkin työtunteja, joten ne olisi hyvä saada näkyville, mutta eivät varsinaisesti ole asiakkaalle arvoa tuottavia ominaisuuksia sellaisenaan.

Sprint 2

Vaikka onnistuimme asiakastapaamisessa myös karsimaan toivottuja storyja sprintistä, päädyimme kuitenkin ottamaan niitä liikaa, varsinkin kun tiesimme etukäteen, että tietokanta-asiat tulisivat viemään paljon aikaa. Tietokanta-asioiden kertaaminen veikin ison osan ajasta, eritoten kun kaikki tekivät kaikkea, eli kertausaika söi kaikkien työtunteja. Toisaalta kaikilla oli sen jälkeen ainakin jonkinlainen käsitys tietokantaosioden toiminnasta.

Sprintin backlogin storyjen taskien seuraamisessa ilmeni haasteita. Taskien pilkkominen pienemmiksi olisi voinut edistää asiaa.

Sprint 3

Asiakkaan/ohjaajan ehdotuksesta päätimme tällä kertaa eriyttää eri sovelluksen kerrosten parissa työskentelyn tietyille tekijöille aiemman kaikki-tekee-kaikkea-mallin sijaan. Olimme sprintin alussa sopineet yhdessä kerrosten “yhdistyskohdista/-tavoista”, mutta loppujen lopuksi kerrokset toteutuivat, näin jälkikäteen todettuna, riskialttiissa järjestyksessä. Tämä kostautui ankarasti, kun viime hetken liitosyritykset eivät toimineetkaan ja korjausajan puutteessa suurin osa sprintin työstä jouduttiin heittämään roskakoriin.

Toimintojen monimutkaistuessa aiempi tekninen velka kostautui ja viimeiset työtunnit kuluivatkin aikalailla aiempien toiminnallisuuksien pelastamiseen.

Yleisesti

Alkutyyliimme, joka ainakin aluksi tuntui toimivan paremmin, johti siihen, että kaikki olivat tehneet vähän kaikkea, mutta kenelläkään ei ollut sataprosenttista otetta yhdestä kokonaisuudesta, (erityisesti Application-luokka), mikä johti ongelmiin, kun yksi tai kaksi tekijä pistettiin vastuuseen koko kerroksesta. Perehtyminen ja väärinymmärrykset veivät aikaa. Olisi ollut parempi sinnitellä aiemmalla tavalla tai eriyttää heti alussa.

Tavoitteemme olivat kuitenkin kokonaisuudessaan melko realistiset, emme tavoitelleet kuuta taivaalta. Toki kaikki suunnittelemamme ei toteutunut arvioimassamme ajassa ja arviointia sopii harjoitella vielä. Backlogimme olivat selkeitä ja ne tukivat työntekoamme. Päätökset ryhmän kesken tehtiin demokraattisesti ja kaikkien läsnä ollessa.

Pohdimme lyhyen nollasprintin tarvetta; olisiko sellainen hyvä olla? Parin päivän mahdollisuus konfata ja harjoitella tehtävien jakamista taskeiksi. Tästä voisi seurata kaikille vahvempi ymmärrys pohjasta, koska se olisi kaikkien yhteinen sprintin tavoite, ei vain yhden /kahden tekijän “sivutaski”?

Opimme olemaan konservatiivisempia arvioidemme kanssa, kommunikoinnin sekä pienissä paloissa etenemisen ja kerroksien yhdistämisen tärkeydestä. Ennen kaikkea aiemman muiden tuotoksen tutkimisen ja kommunikoinnin tärkeyden.

Huomasimme taas konkreettisemmin, että testejä pitäisi tehdä aiemmin ja enemmän.

Kaiken kaikkiaan olimme yhtä mieltä, että miniprojekti oli erittäin opettavainen kokemus:

Opimme todella paljon ohjelmistotuotannon luonteesta, toimintatavoista, viestinnästä, haasteiden ja epäonnistumisten kohtaamisesta ja niiden korjaamisesta.