

Taustaa ryhmästäämme

Diplodocus ryhmämme oli alunperin nelijäseninen, mutta jo ensimmäisessä tapaamisessa ryhmämme koko kasvoi viiteen henkeen. Alkuperäisestä suunnitelmasta tehdä sovellus Ruby On Rails teknologialla luovuttiin myöskin tällöin, ja päädyimme käyttämään Javaa ja Spring-sovelluskehystä projektin suorittamisessa. Kenelläkään ryhmässämme ei ollut isompaa taustaa ohjelmistotuotantoon tai agileihin menetelmiin liittyvissä asioissa. Osa ryhmästäämme oli toki tehnyt yhdessä isompia koko periodin kestäviä ryhmä projekteja, mutta varsinaisesti niissä ei oltu scrumin periaatteita noudatettu. Kolme meistä oli muun muassa Web-palvelinohjelmoinnin harjoitustyötä tehdessä käyttänyt Travisia, jossa tutustuimme continuous deliveryn käsitteeseen. Git-versionhallinta työkalu oli suurimmalle osalle ryhmästäämme hyvinkin tuttu ja lopuillekin ryhmämme jäsenille perusteet työkalun käyttöön olivat hyvin hallussa.

Ensimmäisen sprintin ongelmat ja onnistumiset

Ensimmäisessä asiakastapaamisessa saimme user storyjä pienen haparoinnin jälkeen hyvin aikaan. Päädyimme käyttämään internetissä toimivaa taulua taskien ylläpitoon, oman huoneen puuttuessa. Backlogimme saatiinkin ohjaajan pienoisella avustuksella tehtävänannon vaatimaan DEEP muotoon ja kaikki storyt hiottiin INVEST mallia mukaillen. Ryhmämme ensimmäisessä varsinaisessa tapaamisessa suunnittelimme ohjelman rakennetta ja yritimme saada Spring-sovelluskehysten konfiguroitua henkiin. Tässä kohtaa erityisesti Springin tuomat tekniset vaikeudet alkoivat nousta pintaan. Emme saaneet koko ensimmäisen sprintin aikana springiä toimimaan ja näin ollen myös kaikki user storyt jäivät suorittamatta ryhmämme Definition of Donen vaatimalla tavalla. Myös projektin rakenteen suunnittelu jäi puolitiehen ja monia asioita piti keksiä pitkin projektia täysin lennosta.

Ryhmämme keskinäinen työjärjestys alkoi myös muotoutua jo ensimmäisen sprintin aikana. Noin puolet ryhmästäämme keskittyi Springin kanssa taisteluun ja toinen vähemmän Springiä käyttänyt osa keskittyi kirjoittamaan testejä ja kirjoittamaan puhtaalla Javalla ohjelman vaatimia toiminnallisuuksia. Tässä järjestelyssä onnistuimme mielestämme koko projektin ajan hyvin, jokaisella oli lähes aina jotain tehtävänä ja tarvittaessa toisia ryhmäläisiä autettiin pahimpien vaikeuksien yli.

Toisen ja kolmannen sprintin ongelmat ja onnistumiset

Ensimmäinen henkilökohtainen asiakastapaaminen oli ryhmällemme pettymys. Sillä emme olleet saaneet yhtään user storyä valmiiksi, vaikka olimmekin edenneet muutaman kohdalla hyvinkin pitkälle. Sovimme asiakkaan kanssa, että jatkamme samoille user storyillä tämänkin viikon. Toisen viikon aikana saimme springin toimimaan lokaalisti ja pääsimme viimein kunnolla tekemään kehitystä ohjelmassa. Mavenin asetukset aiheutti vielä pitkään ongelmia projektin Continuous Deliveryn kanssa, sillä emme saaneet tuotostamme Heroku palveluun ennen kuin viimeisellä viikolla. Uusina ongelmina kohtasimme taskien rinnakkaisen toteuttamisen olevan paikoin vaikeaa. Varsinkin etänä tehtäessä toisinaan täytyi vain odottaa, milloin joku saa seuraavan ominaisuuden valmiiksi. Tätä tapahtui kyllä myös ollessa fyysisesti läsnä, mutta tällöin pystyi aina auttamaan tehokkaasti vierestä ja etsimään toiselta kantilta ratkaisuja ongelmiin.

Git-työskentely oli ryhmässämme enimmäkseen hyvää, mutta se aiheutti myös isoja murheita. Kaikki ryhmämme jäsenet käyttivät aktiivisesti haaroja ja puskipivat säännöllisesti committeja ulos ja tuotanto-haara pysyi ehjänä 99% prosenttia ajasta. Suurimmaksi Git-ongelmaksi muodostui enimmäkseen toiminnallisuuksien saamisen tuotanto-haaraan. Toisinaan muutokset vain jäivät omaan haaraan ja niistä ei tehty pull requesteja. Toisinaan taas pull requestit jäivät viikonlopuksi odottamaan itseään ja maanantaihin mennessä oli ohjelmaa kehitetty niin paljon eteenpäin, että yhdistäminen ei enää ollut mahdollista. Nämäkin ongelmat olisi tehokkaasti voitu välttää, mikäli ryhmämme olisi pystynyt työskentelemään paremmin samassa tilassa samaan aikaan. Tämä ei projektin aikana toteutunut kovinkaan montaa kertaa. Suurin syy oli, ettemme ymmärtäneet asian tärkeyttä ja viidellä hengellä töiden, opintojen ja kaiken muun yhteen sovittaminen oli vaikeaa.

Tekniset ongelmat piinasivat jälleen projektin loppupuolella kun, Selenium testaus ei tahtonut onnistua Herokussa tai TravisCI-palvelussa. Emme saaneet asetettua testaus serveriä päälle niin, että olisimme voineet ajaa testit Herokussa tai TravisCI:ssä.

Kokonaisuuteen jonka saimme kasaan kolmessa viikossa olimme kuitenkin hyvin tyytyväisiä, ohjelmamme toimi. Sprintin backlogiin ei ollut jäänyt mitään, koodi oli enimmäkseen hyvänlaatua, ohjelmassa oli hyvin testejä ja demotilaisuus meni hyvin.

Vaikkakin projektimme oli enemmänkin scrumbuttia, kuin itse scrumia, saimme kuitenkin hyvän kuvan metodin tarkoituksesta ja siitä miten sen tulisi toimia.