Jatkuva eksperimentointi ohjelmistokehityksen tukena

Esa Kortelain

Ihmiset käyttävät nykyään mobiililaitteilla ohjelmistoja verkossa. On entistä helpompi seurata jatkuvasti miten, missä ja milloin käyttäjät ohjelmiston kanssa toimivat. Seurannan avulla voidaan todeta, haluavatko asiakkaat käyttää tuotetta.

Artikkelissa on kuvattu Leania ja ketteriä menetelmiä, mutta tiivistelmän kokorajoitteen vuoksi jätän niiden esittelyn toisiin tiivistelmiin ja keskityn artikkelissa esitettyyn eksperimentointiin.

Eksperimentoinnilla tarkoitetaan kenttäkokeisiin perustuvaa lähestymistapaa ohjelmistokehitykseen. Kokeita tehdään yleensä loppukäyttäjien kanssa, mutta myös muiden asiakkaiden, sijoittajien, kolmannen osapuolen kehittäjien ja ohjelmistoympäristökumppaneiden kanssa.

Liiketoimintamalli kuvastaa yrityksen johdon näkemystä siitä, millainen yrityksen toiminta täyttää asiakkaiden tarpeet ja tuottaa voittoa. Ohjelmistoja kehitetään liiketoimintahypoteesien perusteella. Ohjelmistot vanhenevat ja toisaalta ohjelmistoja päivitetään jatkuvasti, tuodaan uusia ominaisuuksia ja muutetaan entisiä. Riskinä on, että asiakas ei ole enää tyytyväinen tuotteeseen. A/B-testauksessa asiakas testaa uutta, kehitettävää ominaisuutta ja tehdään hypoteesi ominaisuuden vaikutuksesta. Mittareilla mitataan hypoteesin oikeellisuutta ja tulosten perusteella päätetään ohjelmistotuotteen jatkosta, päivitetäänkö laajemmin, kehitetäänkö edelleen vai lopetetaanko kehitys.

Software as a service-mallia käytetäessä on helppo kerätä jatkuvasti passiivista asiakasdataa, jonka perusteella voidaan seurata asiakastyytyväisyyttä ja sitä myöten tarvetta ohjelmiston kehittämiseen. Tässä tulee esiin vaatimus ketterien menetelmien käyttöön, koska tarvittavat ohjelmistomuutokset pitää saada nopeasti käyttöön.

Eksperimentointi selventää, että:

* kehitetään vain niitä ominaisuuksia, jotka tuottavat asiakkaalle arvoa
* päätökset eivät pohjaudu mielipiteille eivätkä päättäjän asemaan
* kehittäjät tietävät, mitä pitää kehittää
* tuote ja asiakkaiden tarpeet kohtaavat