Jatkuvan integroinnin ja julkaisun vaikutukset ohjelmistosuunnittelutehtäviin

31.12.2021

Ohjelmistokehityksessä on viime vuosina yleistynyt toimintatapa, jossa rakennetaan samanlaisina toistuvista rutiinitehtävistä automatisoitu toimintaketju. Tämän tarkoituksena on viedä ohjelmistosuunnittelijan toimesta valmistunut uusi ominaisuus suoraan asiakkaan kokeiltavaksi. Kyseistä toimintatapaa kutsutaan jatkuvan integroinnin- ja julkaisun/käyttöönoton menetelmäksi.

Jatkuvan integroinnin ja -julkaisun menetelmiä voidaan hyödyntää saman aikaisesti ketterien menetelmien kuten Scrumin ja Kanbanin kanssa. Tällöin ohjelmistomuutosten lisäykset tehdään suoraan kehitettävän ohjelmiston pääversioon, jonka jälkeen uusi sovellusversio testataan ja julkaistaan automaattisesti asiakkaan saataville. Tällä tavoin kehityssykleistä tulee lyhyitä ja uudet ominaisuudet saadaan nopeasti käyttöön ja niistä saadaan nopeasti palautetta.

Testivetoinen kehitys

Testitapausten suunnittelusta tulee entistä tärkeämpi osa ohjelmistokehittäjän työtä. Automatisoidun testaamisen ylläpito muuttaa työnkuvaa ja testaamisen tuloksen luotettavuus on jatkuvan julkaisun menetelmän edellytys. Mahdollinen huolimaton toiminta tulee kuitenkin automatisoidun prosessin myötä myös asiakkaan nähtäville ja voi vaikuttaa asiakastyytyväisyyteen negatiivisesti.

Jatkuvan julkaisun kohdalla ohjelmistokehittäjän harkittavaksi tulee enemmässä määrin se, milloin tehtävä on julkaisuvalmis ja saatettavissa asiakkaan saataville. Testivetoinen kehitystapa edellyttää kehittäjiltä yksikkötestauksen hoitamista ja tällöin vaaditaan omaehtoisen työskentelyn lisäksi hyvää oman työn arviointia. Ohjelmisto-osaajalta tarvitaan myös yrittäjämäistä otetta omaan tekemiseen.

Puhtaasti ketterissä menetelmissä Scrumissa ja Kanbanissa varsinainen testausvaihe alkaa tyypillisesti vasta kehityssyklin jälkeen eikä sen hoitaminen kuulu kehitystiimille. Jatkuvan julkaisun menetelmässä testaustyö integroituu osittain kehitystiimin työhön.

Vaikutukset

Jatkuvan julkaisun menetelmän yleistyminen voi vähentää testaamisen työtehtäviä, jotka ovat usein olleet ohjelmistoalan ensimmäisiä tehtäviä työuran alussa ja joista oppia on usein haettu vaativampiin työtehtäviin. Toisaalta testausautomaation järjestämisestä syntyy uusia tehtäviä, mutta ne vaativat erityisosaamista ja kehitettävän järjestelmän laajempaa tuntemusta. Toimintaketjun automatisoinnilla saavutetaan kuitenkin

vakautta ja tasaista laatua, kun prosessin inhimilliset virheet saadaan karsittua pois.

Juha Yli-Hemminki
Lehtori
SeAMK

Lähteitä:

Chen, L. 2015. (PDF) Continuous Delivery: Huge Benefits, but Challenges Too (researchgate.net)

Neely, S. & Stolt, S. 2013. Continuous Delivery? Easy! Just Change Everything (Well, Maybe It Is Not That Easy) | IEEE Conference Publication | IEEE Xplore

Linders, B. 2018. How Continuous Delivery Impacts Testing (infoq.com)