

Referaatti

Tero Huomo kandidaatin tutkielmassaan Ohjelmistoarkkitehtuurin sisällyttäminen ketteriin ohjelmistotuotantomenetelmiin on perehtynyt ketterien menetelmiin niiden arkkitehtuurisen rakenteen suunnittelun kannalta. Lähtökohtana hän vertaa ketteriä menetelmiä vesiputousmalliin, jossa ohjelmiston rakenne suunnitellaan, kuten koko ohjelmistokin, aluksi kokonaisuudessaan.

Tero Huomo jakaa ohjelmistokehityksen ketterissä menetelmissä kolmeen osaan: Julkaisun suunnitteluun, Iteraation suunnitteluun ja Kehitysjakson aikana tapahtuvaan suunnitteluun. Julkaisun suunnittelussa tehdään asiakkaan kanssa vaatimusmäärittely, jonka pohjalta ohjelmiston toteutus alkaa. Iteraation suunnittelussa valitaan toteutukseen tulevat vaatimukset sekä sen kuinka kehitysryhmä toteuttaa ne. Kehitysjakson aikana toteutetaan vaatimukset. Kehittäjiä ohjeistetaan turhan työn minimointiin, jonka lisäksi refaktoroimalla saavutetaan mahdollisimman yksinkertainen rakenne.

Ohjelmistoarkkitehtien keskuudesta on noussut kritiikkiä ketterien menetelmien erillisen rakenteen suunnittelun puuttumisen vuoksi. Heidän mukaan vajavainen rakennesuunnittelun vuoksi osa rajoituksista jää huomioimatta, löydetään yllättäviä riippuvuuksia, joiden vuoksi joudutaan ohjelmistoa muokkaamaan, minkä vuoksi joudutaan osa työstä tekemään uudelleen. Näiden vuoksi osa kehittäjäryhmistä ovat ottaneet käyttöön erilaisia käytänteitä, joilla huomioidaan ohjelmiston rakenne.

Eräs käytäntö on Sprint 0, joka ennen varsinaista ohjelmistotuotantoa oleva lyhyt ajanjakso, jossa voidaan tehdä myös rakennesuunnittelua. Myös arkkitehtuurinsuunnittelu voidaan eristää omaksi prosessiksi, jolloin erilliset ohjelmistoarkkitehdit suunnittelevat ohjelmistorakennetta seuraavaa suunnittelujaksoa varten. Suunnittelupiikit ovat iteraation aikana tapahtuvia lyhyitä jaksoja, jolloin voidaan tutkia erilaisia ratkaisuja ohjelmiston kehityksessä tuleviin ongelmiin. Ketterissä menetelmissä voi myös käyttää suunnittelupiikkejä ohjelmiston rakenteen suunnitteluun. Arkkitehtuurijaksossa yhden tavallisen kehitysjakson aikana keskitytään lähinnä ohjelmiston rakenteeseen. Ketterissä menetelmissä ohjelmiston vaatimukset esitetään käyttäjätarinoina, jotka kuvaavat jonkin ohjelman toiminnallisuuden, joka tuo asiakkaalle arvoa. Ohjelmiston rakenne voidaan ottaa mukaan suunnitteluun tekemällä erillisiä arkkitehtuuritarinoita.

Käytäntöjen paremmuutta on vaikea verrata ja pääosa niitä käyttäneistä ohjelmistoprojekteista olivat tyytyväisiä omaan käytäntöönsä. Koska käytännöt eivät ole ristiriidassa niitä voisi käyttää myös erilaisina yhdistelminä tai sitten niistä ja ketteristä menetelmistä voisi luoda täysin uuden projektimallin. Ketterät menetelmät painottavat valmiutta muutoksiin ja ehkä arkkitehtuurin huomioivat käytänteet ovat eräs tapa sopeutua prosessimallien kehitykseen.