

Referaatti:

Tämä referaatti käsittelee Tero Huomo kandityötä: Ohjelmistoarkkitehtuurin sisällyttäminen ketteriin ohjelmisto- tuotantomenetelmiin.

Ketterää ohjelmistokehitystä ja arkkitehtuurilähtöistä ohjelmistotuotantoa pidetään usein vastakkainasetteluna. Ketterä ohjelmistokehitys hylkää perinteisen, sillä tiukka suunnitelmallisuus ei ole ketterien arvojen mukainen.

Käsittelyssä otetaan huomioon käytänteiden ja ketterien periaatteiden väliset eroavaisuudet sekä ristiriidat. Esiteltävät käytänteet eivät ole täysin ristiriidattomia suosituimpien ketterien ohjelmistokehitysmetodiikkojen tai ketterän ohjelmistokehityksen julistuksen kanssa.

Sprint 0 on kehitystyön alustamiseen tarkoitettu jakso. Sprint 0:ssa luodaan korkean tason suunnitelma ohjelmiston arkkitehtuurista, arkkitehtuurin ei ole tarkoitus olla lopullinen. Varhaisen arkkitehtuurin luo sama kehitysryhmä, joka vastaa ohjelmistojärjestelmän toteuttamisesta. Suosiostaan huolimatta Sprint 0, on Scrumin ja XP:n käytänteiden vastainen.

Ohjelmistoarkkitehtuuri voidaan myös eriyttää **omaksi prosessikseen**. Arkkitehtuurin suunnittelee ja mahdollisesti myös osin toteuttaa erilliset arkkitehdit. Arkkitehtiryhmän huomio keskittyy vaatimukseen, jota ohjelmistolta halutaan myöhemmissä kehitysjaksoissa, valmistelea sitä juuri riittävästi tukemaan seuraavia toiminnallisuuksia. Tyypillisesti osa arkkitehtiryhmän jäsenistä kuuluu myös kehitysryhmään. Eriytetty arkkitehtuuriprosessi voi johtaa siihen, ettei huomioida järjestelmän muuttuvia vaatimuksia. Ketterän ohjelmistokehityksen julistus painottaa, että parhaat arkkitehtuurit ja suunnitelmat kumpuavat itseorganisoituvista kehitysryhmistä.

Suunnittelupiikit ovat iteraation aikana suoritettavia työjaksoja, joissa ratkotaan järjestelmän laajentamisesta koituvia ongelmia, suunnittelun haasteita sekä uuden teknologian vaikutusta järjestelmään. Extreme Programming hyväksyy suunnittelupiikit käytäntönä. Tavoitteena on vähentää teknisten ongelmien riskiä sekä lisätä käyttäjätarinoiden aika-arvioiden luotettavuutta. Piikit eivät suoraan ole ristiriidassa ketterän kehityksen periaatteiden kanssa.

Tarvittaessa tavallisen kehitysjakson sijasta suoritetaan **arkkitehtuurijakso**. Arkkitehtuurijaksoa voidaan pitää laajennettuna variaationa suunnittelupiikistä. Etuna on koko kehitysryhmän sitoutuminen toimintaan, jolloin noudatetaan ketterien periaatteiden mukaista luottamusta ryhmään ja sen itseorganisoituvuuteen. Arkkitehtuurijaksojen ongelmat ovat yhtäläiset kuin Sprint 0:ssa. Arkkitehtuurijakson lopputuloksena on vähäisesti tai ei lainkaan asiakasta hyödyttäviä konkreettisia ominaisuuksia.

Arkkitehtuuritarinat ovat käyttäjätarinoita vastaavia vaatimuksia. Arkkitehtuuritarinat kuvaavat järjestelmän rakenteen vaatimuksia, jotka eivät näy asiakkaalle. Oleellinen osa arkkitehtuuritarinoita on työtehtävien näkyvyys työtiloissa. Arkkitehtuuritarinat eivät ole ketterän ohjelmistokehityksen julistuksen vastaisia. Ne saattavat jopa auttaa kiinnittämään huomiota jatkuvaan tekniseen erinomaisuuteen. Arkkitehtuuritarinat voivat selkeyttää myös asiakkaalle mitä lisätyötä hänen vaatimuksensa tuottavat.

Ei ole olemassa yhtä ohjetta sille, mikä käytänteistä on paras tai hyödyllisin. Tieteellistä vertailua käytäntöjen paremmuudesta tai vaikutuksista ketterään ohjelmistokehitykseen ei juuri ole.