

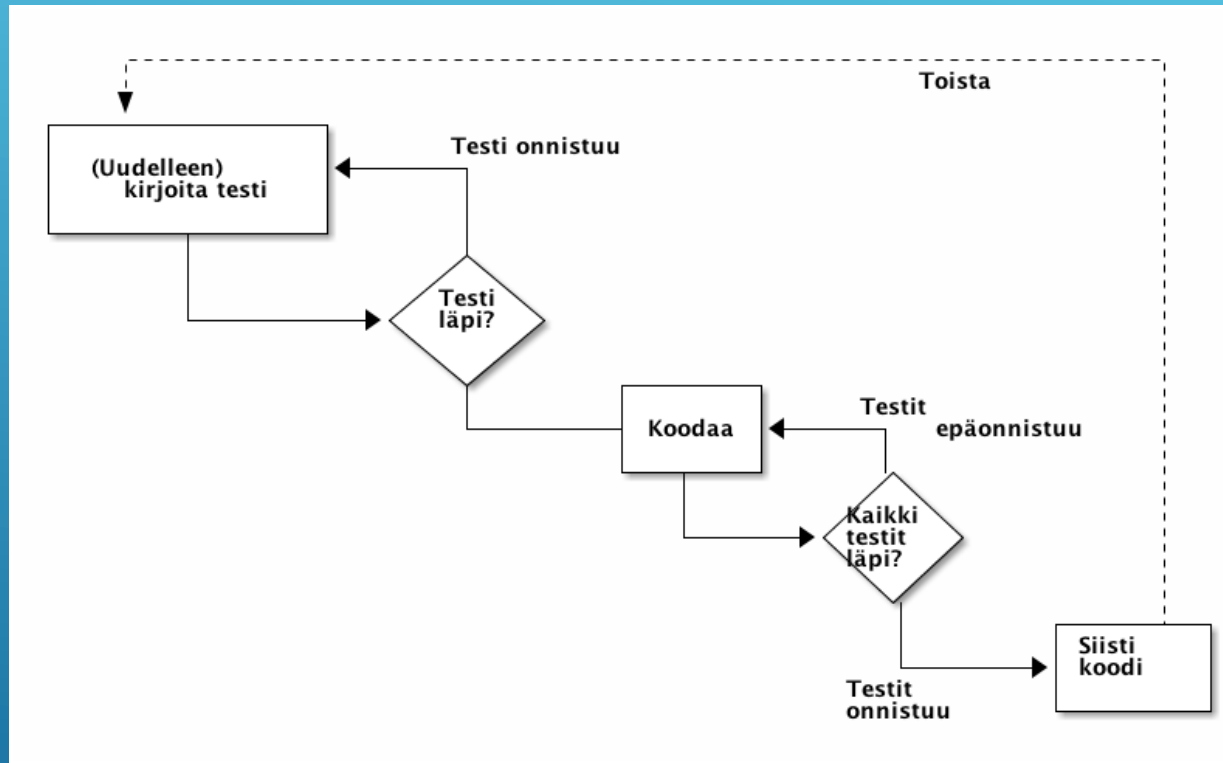
TDD

Test-driven development

© Mestarit Miko ,Riko ,Niko

PERIAATE

- Ensin luodaan uusi testitapaus.
- Vasta sen jälkeen muokataan kehitettävää ohjelmaa niin, että se läpäisee uuden testin.
- Eli yksikkötestit kirjoitetaan pienissä osissa ennen varsinaista tuotantokoodia.
- Tällä pyritään parempaan rajapintasuunnitteluun sekä myös varmistumaan ohjelmiston oikeasta toiminnasta.
- Jos yksikkötestit aiottaisiin kirjoittaa tuotantokoodin jälkeen, ne jäisivät usein tekemättä, koska jälkikäteen on vaikeampi nähdä yksikkötestien hyötyjä, joten niille ei varata aikaa.



HYÖDYT

- ▶ Kun testikoodi kirjoitetaan etukäteen saadaan jatkuvasti kehittyvä testiverkosto
- ▶ Sen varassa uusien toimintojen kehittäminen sekä virheiden korjaaminen on huomattavasti turvallisempaa.
- ▶ Jo olemassa olevia testejä suorittamalla huomataan, jos virheitä korjattaessa tulee tehneeksi uusia virheitä.
- ▶ TDD:tä voi hyödyntää myös kouluprojekteissa.

MOCKING

- ▶ "Mocking" on yksikkötestausilmiö, joka auttaa testaamaan objekteja erikseen toisistaan korvaamalla riippuvaiset objektit monimutkaisella käyttäytymisellä, testiobjekteilla ja ennalta määritetyllä/simuloidulla käytöksellä. Näitä testiobjekteja kutsutaan Mock objekteiksi.

LÄHTEET

- ▶ https://www.google.fi/search?q=TDD+malli&rlz=1C1GCEA_enFl810Fl810&source=lnms&tbm=isch&sa=X&ved=0ahUKEwif5dWMs-7cAhUyyaYKHTpXDnkQ_AUICigB&biw=1920&bih=943#imgsrc=lo6VMjDyVdRnmM periaate sivun kuvan linkki
- ▶ https://fi.wikipedia.org/wiki/Testivetoinen_kehitys periaate ja hyödyt sivun tiedot
- ▶ <https://dzone.com/articles/test-driven-development-tdd> tietoa "mockingista"