

# Testauksen työvälineet nyt ja tulevaisuudessa - akateeminen näkemys

Mika Katara, Tampereen teknillinen yliopisto

Testauksen arvostus yliopistomaailmassa on perinteisesti ollut varsin heikko. Tämä on näkynyt siten, että testauskoulutukseen ei ole panostettu läheskään yhtä paljon kuin esimerkiksi ohjelmointikursseihin. Tutkimusta testauksen alalla on jonkin verran tehty, mutta tutkimustulosten hyödyntäminen on suurelta osin vasta edessäpäin.

Testauksen osuus ohjelmistoprojektien kustannuksista on merkittävä. Täytyy siis kysyä, miksi yliopistot eivät ole panostaneet samassa suhteessa testauksen koulutukseen ja tutkimukseen. Taustalla on varmasti osaltaan testauksen heikko arvostus yleensäkin. Kenties on lähdetty myös siitä idealistisesta oletuksesta, että testaus on vain väliaikainen ilmiö; uudet ohjelmistojen kehitysmenetelmät tulevat takaamaan sen, ettei virheitä synny eikä näin testaustakaan merkittävässä määrin tarvita. Tähän mennessä tällaiset kehitysmenetelmät eivät kuitenkaan ole osoittautuneet kovinkaan käytännöllisiksi ja kustannustehokkaiksi. Niin kauan kuin ohjelmistojen tekijät ihmiset, virheitä ilmenee ja siten myös testausta

tarvitaan. Iteratiivisten prosessien myötä testauksen tarve tulee vain korostumaan.

**”Tulevaisuudessa on selvästi nähtävissä erilaisten apuvälineiden kasvava rooli”**

Järjestimme Tampereen teknillisellä yliopistolla (TTY) touku-kuussa 2004 testauskoulutusta käsittelevän työpajan. Siihen osallistui testauksen opettajia lähes kaikista testauskoulutusta tarjoavista Suomen yliopistoista. Intoa koulutuksen tason parantamiseen selvästi on. Monesti koulutus ja tutkimus käyvät käsi kädessä, joten tämä voi tarkoittaa myös

suomalaisen testaustutkimuksen nousua tulevaisuudessa.

Automaattinen testaus on erityisen mielenkiinnon kohteena yliopistomaailmasta katsottuna. Se tarjoaa mielenkiintoisia tutkimuksellisia haasteita ja toisaalta kiinnostaa opiskelijoita, jotka tulevat testauskursseille opittuaan ohjelmointitaitoja muilla kursseilla. Vaikka manuaalinen testitapausten suorittaminen on edelleen merkittävässä roolissa tehtä-

essä tietyn tyyppistä testausta, on tulevaisuudessa selvästi nähtävissä erilaisten apuvälineiden kasvava rooli. Näiden apuvälineiden avulla tullaan automatisoimaan tiettyjä työvaiheita, joita on ennen tehty käsin. Voidaan myös tehdä asioita, joita ei edes pystyttäisi tekemään manuaalisesti; koodikattavuuden mittaaminen on tästä ollut hyvä esimerkki. Kokonaan automaattiseksi testaus ei kuitenkaan muutu. Hyvä testaaja pystyy asettumaan loppukäyttäjän asemaan ja tarkastamaan hänen vaatimustensa toteutumisen paremmin kuin kone ikinä pystyy.

Tässä artikkelissa tarkastelen testauksen ja testiautomaation tulevaisuutta yliopisto-opettajan ja –tutkijan näkökulmasta. Ensin käsittelen testaustutkimusta niin kutsutun mallipohjaisen automaation kannalta ja sitten työkalujen asettamia haasteita testauskoulutukselle.

## ”2000-luvun teemana testauksen suhteen näyttäisi olevan mallipohjainen testaus”

### Testaustutkimus

Siinä missä 1990-luvulla testitapausten automaattiseen suorittamiseen panostettiin voimakkaasti, 2000-luvun teemana testauksen suhteen näyttäisi olevan mallipohjainen testaus. Mallipohjaista testausta käsitellään tässä lehdessä laajemminkin. Sen ideana on, että testitapausten sijaan testaaja laatii testimallin, josta voidaan

automaattisesti generoida ääretön määrä erilaisia testitapauksia. Testitapausten laatuun vaikuttaa testimallin lisäksi myös generointiin käytettävät heuristiikat.

## ”Perinteinen testien suorittamisen mahdollistava automaatio löytää harvoin uusia virheitä”

Perinteinen testien suorittamisen mahdollistava automaatio löytää harvoin uusia virheitä. Mallipohjaisen lähestymistavan etuna on se, että testimalleista generoidut testitapaukset sisältävät sellaisia tapahtumasekvenssejä, joita kukaan ei ole ennen edes tullut ajatelleeksikaan. Kunhan vain mallin tekemiseen panostetaan kunnolla, sen avulla voidaan löytää uusiakin virheitä. Vahvoin alue mallipohjaiselle testaukselle on rinnakkaisuuden ilmiöiden

hallinta, joka ihmisille on vaikeaa erilaisten vaihtoehtoisten suoritusten valtavan määrän takia.

TTY:ssä mallipohjaista testausta on tutkittu viime vuosikymmenen lopulta lähtien lähinnä professori Antti Valmarin tutkimusryhmässä. Mallipohjaisen testauksen taustalla olevat matemaattiset teoriat ovat vuosien saatossa kypsyneet, ja menetelmiä voidaan

alkaa soveltaa käytännössä. Nyt tarvitaan mielestäni soveltavaa tutkimusta, joka panee teorian käytäntöön. Tarvitaan käytännön esimerkkejä siitä, kuinka malli-

pohjainen testaus voidaan tuottaa ja ottaa käyttöön teollisuuden projekteissa.

Hyvä esimerkki mallipohjaisuutta hyödyntävästä lähestymistavasta on kotimaista tuotantoa edustava Conformiq Softwaren Test Generator –työkalu. Työkalua on tietääkseni tähän mennessä käytetty lähinnä ohjelmointirajapinnan läpi tehtävässä testauksessa. Mallipohjaista testausta voidaan soveltaa myös käyttöliittymän läpi tehtävään testaukseen.

### Testauskoulutus

Vastaan TTY:ssä Ohjelmistojen testaus –kurssista, jota suorittaa keväällä 2005 yli sata opiskelijaa. Testiautomaatio on keskeisessä osassa sekä luennoissa että kurssiin liittyvissä harjoitustöissä. Esimerkiksi TTCN-3-testauskielen perusteet käydään luennoilla läpi. Myös mallipohjaista testausta käsitellään, mutta pääpaino on toistaiseksi ollut perinteisissä menetelmissä ja työkaluissa.

Keväällä 2005 opiskelijoiden harjoitustyön aiheena on Mozilla-selaimen URL-jäsentäjän testaa-

minen. Harjoitustyöhön kuuluu V-mallin mukaisesti sekä testauksen suunnittelua että testitapausten ajamista. Yksikkötestaustasolla käytetään automatisointiin CppUnit-sovelluskehystä, joka on suositun JUnit-kehityksen C++-kielinen vastine. Koodista mitataan moniehtokattavuutta Testwellin CTC++-työkalujen avulla. Järjestelmätason testaus tehdään ohjelmointirajapinnan läpi käyttäen esimerkiksi Python-skriptejä. Käyttöliittymän läpi on tarkoitus testata myös manuaalisesti. Luennoilla esitellään tätä varten tutkivan testauksen (exploratory testing) tekniikoita.

## ”Teollisuudella on selvästi tarve tehostaa testausta ja pienentää siitä syntyviä kustannuksia”

Tulevaisuudessa harkintaan tulee myös muiden työkalujen käyttöönotto. Tämä asettaa kysymyksiä niiden lisensointiin liittyen. Yliopiston kannattaa hyödyntää mahdollisimman paljon aktiivisen kehittäjäyhteisön tukemia avoimen lähdekoodin ohjelmistoja, ja käyttää niitä koulutuksen apuna missä vain mahdollista. Teollisuuden käyttämät kaupalliset välineet ovat myös kiinnostavia esimerkiksi kuormitustestauksessa ja testattaessa käyttöliittymän läpi. Onneksi useimmat työkalujen maahantuojat suhtautuvat suopeasti ei-kaupallisiin yliopistolisensseihin. Nehän ovat investointi tulevaisuuteen; opiskelijat haluavat käyttää hyväksi

havaitsemiaan työkaluja myös jatkossa toimiessaan teollisuuden palveluksessa.

Työkaluvalintoja ei voi tehdä kevyesti, olipa kyseessä sitten yritys tai yliopisto. Kurssin henkilökunnan kouluttautuminen käyttämään jotakin tiettyä työkalua, voidakseen opastaa opiskelijoita, vie aikaa. Tämän takia työkaluja ei voida joka vuosi vaihtaa, vaikka mielenkiintoinen väline löytyisi eikä hintakaan olisi esteenä.

TTY:n ohjelmistotekniikan laitokselle tehdään vuosittain kymmenkunta testausaiheista diplomityötä. Takavuosina suosittu aihe on ollut

automatisoida jokin testausvaihe, joka on aikaisemmin tehty manuaalisesti, tai sitten sitä ei ole tehty ollenkaan. Nyt päällimmäisenä näyttäisivät olevan Symbian-maailman testausaasteet. Toivon näkeväni tulevina vuosina myös mallipohjaisen testauksen soveltamiseen ja käyttöönottoon liittyviä töitä. Laitoksen ensimmäinen mallipohjaiseen testaukseen liittyvä tohtorinväitös on luvassa tämän vuoden aikana.

### **Yhteenveto**

Testaus ja sen opettaminen eroaa monessa mielessä ohjelmoinnista. Yksi erottava seikka on se, että siinä missä ohjelmointipuo-

lella teknologian siirtoa tapahtuu lähinnä yliopistoista ja tutkimuskeskuksista teollisuuteen päin, testauksen tapauksessa suunta voi hyvin olla päinvastainen. Meillä yliopistojen opettajilla ja tutkijoilla on paljon opittavaa käytännön työtä tekeville.

Testiautomaation alueella pitäisi lisätä kanssakäymistä teollisuuden ja akateemisen maailman välillä. Teollisuudella on selvästi tarve tehostaa testausta ja pienentää siitä syntyviä kustannuksia. Aikajänne näihin muutoksiin on yleensä hyvin lyhyt. Akateemisella puolella taas asioita katsotaan hieman pidemmällä jännteellä, mutta käytettävissä olevat resurssit ovat usein riittämättömät. Synergiaetuja olisi löydettävissä varmasti sekä tutkimus- että koulutuspuolella.

Tekn. toht. Mika Katara työskentelee Tampereen teknillisen yliopiston ohjelmistotekniikan laitoksessa vanhempana tutkijana. Hän vastaa yksikön testauskoulutuksesta ja tutkii mm. mallipohjaista testausta. Kataran tavoittaa sähköpostitse osoitteesta [mika.katara@tut.fi](mailto:mika.katara@tut.fi) tai puhelimitse numerosta (03) 3115 5512.

TTY:n ohjelmistotekniikan laitoksesta valmistuu vuosittain noin sata diplomi-insinööriä. Laitoksessa toimii yhdeksän professoria, muuta henkilöstöä on opetus- ja tutkimustehtävissä noin 90. Merkittävissä roolissa tutkimusrahoituksen osalta ovat erilaiset yhteistyöprojektit yritysten kanssa (Tekes-projektit yms.)