

## Testausautomaation kehittäminen ja kehittyminen

Mitro Kivinen, Qentinel Oy



Kirjoitin testauksen osaamisesta IT-Viikon Sytyke-liitteeseen joulukuussa 2004. Tarkkasilmäisimmät kyselivät, missä testausautomaatio on. Vastasin silloin kyselijöille, että testausautomaatiota varten pitää hallita kaikki samat asiat kuin testaustakin varten. Tätä asiaa pitää varmasti täsmentää ja käynkin tässä artikkelissa asiaa tarkemmin läpi.

Testausosaamisen suhteen on mahdotonta sanoa lopullista totuutta, sillä osaamistarpeet vaihtelevat sovelluksen ja sen testaamiseen sopivan automaation mukanaan tuomista vaatimuksista. Automaatio ei suinkaan ole itsensänselvä ratkaisu kaikkiin testauksen ongelmiin ja sopivan automaatiotason valinta tai automatisoimatta jättäminen vasta osaamista vaatiikin.

### Perusongelmana moniulotteisuus

Usein organisaatiossa löytyy monia erilaisia testausautomaation ilmenymiä: on nauhoittimia, työkaluja ja vielä erilaisia skriptejä eri puolilla. Kokonaisvaltainen testiautomaatiostrategia puuttuu, mikä johtuu siitä,

että organisaatioilla ei ole käytösään riittävästi testausautomaation toimintamallien laaja-alaista osaamista. Tämän vuoksi automaatio ei onnistu ja sen lupaukset jäävät lunastamatta.

“Onnistuakseen automatisointi vaatii hyvää testausosaamista”

Odotuksetkin ovat usein epärealistisia: testausautomaatio nähdään liian helposti kaikenratkaisevana asiana eikä välttämättä ymmärretä, että testausautomaatio voi ratkaista vain osan testaustarpeesta, ei kaikkea kaikilla tasoilla.

Onneksi kehittäjät itse haluavat päästä helpolla ja automatisoivat oman testaustyönsä jopa muiden sitä edes tajuamatta. Kokonaisuutena automaatiokenttä on erittäin haasteellinen hallittava.

### Mistä on kysymys

Automatisointi tulee mukaan kuvioon silloin, kun samaa asiaa halutaan toistaa useita kertoja samanlaisena. Jopa näin perustavanlaatuisen totuuden ymmärtäminen voi olla vaikeaa. Automatisointia lähdetään kehit-

tämään miettimättä sitä, mitä halutaan toistaa monta kertaa ja miksi. Lisäksi toistamisen hyötyjen tulee olla suurempia kuin automatisoinnin tuottamat kustannukset.

Toisaalta automatisointi mahdollistaa sellaisia asioita, joita olisi muin tavoin tehtynä mahdotonta toteuttaa millään järjellä työmäärällä annetussa ajassa. Esimerkkeiksi käyvät liikenteen tuottaminen puhelinverkkoon tai miljoonien rivien pohjadataan luominen testauksen tietokantaan.

## “Pitää tietää mitä halutaan toistaa paljon”

Pienessä mittakaavassa tämä on helppoa ja intuitiivista. Kun samaa kommentoijaa antaa Unixissa kolmatta kertaa, tulee asian jo samantien skriptanneeksi. Eipä tarvitse enää näpytellä komentoja ja kas - automaatiota syntyi lähes vahingossa.

Ohjelmien monimutkaistumisen ja kehittämisen haasteet vaativat entistä tehokkaampia tapoja testata laajempia kokonaisuuksia. Tätä haastetta on mahdotonta voittaa vain lisäämällä manuaalista testausta. Siksi automatisoinnin suuntaan kuikuillaan aina uudelleen.

### **Automatisoinnissa tarvittava osaaminen**

Teollisuusautomatisointi on ensisijaisesti erilaisten robottien rakentamista ja niiden ohjelmointia. Sama pätee testaukseen: testaa-

minen tulee vasta, kun automatisointi on rakennettu ja testiskriptit kirjoitettu.

Manuaalinen testaaminen on käsityötä ja ero on kuin vertaisi tehdasta ja käsityöläisverstasta. Tehdas ei kuitenkaan synny ilman valtavia investointeja. Investointi ja tehtaan rakentaminenkin pitää osata. Samoin on automaation luomisen laita: kyseessä on haastava ja monitahoinen ohjelmakehitysprojekti.

Ohjelmointi- ja skriptaustaidot ovat ehdottoman olennaisia automatisointiprojektissa. Kaikkien ei kuitenkaan tarvitse osata ohjelmointia. Lisäksi ohjelmointitapa riippuu valittavasta välineestä. Keveimmillään ohjelmointi hoituu nauhoittamalla käyttöliittymässä tehtävät hiiren klikkaukset ja syötökenttiin kirjoitettavat tekstit. Tämän jälkeen nauhoitusrobotti suorittaa annettua toimintoa vaikka loputtomiin. Eri asia on se, kuinka pitkälle tästä on hyötyä. Nauhoitustyökalujen käyttö tehostuu heti, kun mietitään etukäteen, mitä kannattaa nauhoittaa ja toistaa ja nauhoitettuja skriptejä muokataan sen mukaisesti.

Raskaimmillaan kehitetään laajalainen automaatioalusta, jonka rakentamiseksi tarvitaan useita erilaisia työkaluja. Olennaisena onnistumisen edellytyksenä on

rakenteellisen ohjelmoinnin osaaminen.

Onnistuakseen automatisointi vaatii hyvää testausosaamista. Sitä ei välttämättä tarvitse olla kaikilla automatisointiin osallistuvilla, mutta projektissa tätä osaamista on oltava.

Toki käytettävien välineiden perustarkoitus ja työkalujen tuomat mahdollisuudet pitää osata hyvin. Jokaisella työkalutyypillä on lisäksi omat käyttökohteensa eivätkä ne sovellu kaikkeen. Kaikki palaa alkuperäiseen asetelmaan: pitää tietää, mitä halutaan toistaa paljon.

Testausympäristö vaikuttaa automatisointiin samalla lailla kuin ympäristön pystyttämiseen ja tulosten analysointiin. Automatisointi lähtee usein käyttöjärjestelmäkomentojen skriptaamisesta. Toinen tyypillinen aloitustapa on skriptata tietokannan datan syöttö ja tulosten tarkistus.

Teknisissä rooleissa testaajan on syytä kehittää yleistä järjestelmäarkkitehtuurien tuntemustaan ja ohjelmointiosaamistaan. Automaation kehittäjät eivät välttämättä ole sovelluksen kohdealueen asiantuntijoita eivätkä toiminnan asiantuntijat välttämättä ymmärrä kaikkia tekniikan hienouksia. Molempia näkökulmia tarvitaan onnistuvan testauksen ja sen automatisoinnin luomiseksi ja siksi testauksesta tulee tiimityötä.

Automaatiota voi lähestyä organisestisesti pikkuhiljaa kehittämällä tai

suurena projektina. Molemmissa tarvitaan erilaisia osaamisia.

### Automaation tilanne tänään

Laajimmin levinneitä testausautomaation ilmentymiä ovat yritys-kohtaiset skriptit. Myös erilaiset käyttöliittymän toiminnan nauhoitustyökalut ovat laajalle levinneitä. Ne ovat helpoimmin lähestyttäviä ja suhteellinen hyöty saavutetaan nopeasti. Niiden avulla toteutetaan valtaosa massatesteistä.

Laajoja automaatioalustoja on vain harvoilla. Onnistuakseen laaja automaatio vaatii organisaatiolta tietynlaista tapaa toimia. Sovelluksen arkkitehtuuri ratkaisee paljon: helpoimmin onnistutaan silloin, kun ohjelman sisäiset ja ulkoiset rajapinnat ovat harkittuja ja vakaita. Automaatio ei onnistu useissa vanhoissa järjestelmissä niissä olevien sekavien rajapintojen takia.

## ”Iso osa testaamisesta jää edelleen ihmisten tehtäväksi”

Ketterien kehitysmenetelmien rinnalla on syntynyt iso joukko näitä tukevia kehittäjien työkaluja. Esimerkiksi xUnit-työkalut mahdollistavat aiempaa helpomman yksikkötason testausautomaation. Samoin kehittyneet työkalut mahdollistavat automaattisen jatkuvan integroinnin. Molempia näkee käytössä yhä useammin.

	Testaaja	Automatisoija
Kohteen ja sen toiminnan tuntemus	Välttämätön	Hyödyllinen
Kohteen tuottaman tuloksen analysointitaito	Välttämätön	Välttämätön
Luovuus löytää uutta testattavaa	Välttämätön	Hyödyllinen
Luovuus löytää uusia tapoja testata,	Välttämätön	Välttämätön
Järjestelmällisyys käydä ahkerasti läpi kaikki erilaiset tavat testata	Välttämätön	Hyödyllinen
Priorisointitaito	Välttämätön	Välttämätön
Ohjelmointitaito	Hyödyllinen	Välttämätön
Tietotekniikka-arkkitehtuurit	Hyödyllinen	Välttämätön
Käyttöjärjestelmät	Hyödyllinen	Välttämätön
Tietokannat	Hyödyllinen	Välttämätön
Tietorakenteet ja algoritmit	Hyödyllinen	Välttämätön

### Testaajan ja automatisoijan osaamisten erot

#### Automaation rooli tulevaisuudessa

Matalan tason testausten automatisointi lisääntyy, kun testivoimien kehittämisen edut alkavat valjeta nykyistä laajemmalle yleisölle. Tämä parantaa ohjelmistojen laatua kautta linjan, mutta se ei yksin riitä.

Ohjelmistojen arkkitehtuuria rakennetaan suoraan automatisoitavaksi. Tämä mahdollistaa laajempien ja parhaimmillaan helpokäyttöisempien automaatioalustojen luomisen. Automaatioalustan rakentaminen on kuitenkin suuri investointi, eikä sitä useinkaan voida tehdä. Arkkitehtuurin selkiytyminen on kuitenkin hyvä asia ilman automatisointiakin.

Sovellusten ja järjestelmien entistä laajempi keskinäinen yhteistyöasettaa uusia vaatimuksia rajapintojen kuvaamiselle. Tämä tuo eväitä myös automaation toteuttamiselle.

Iso osa testaamisesta jää edelleen ihmisten tehtäväksi. Tutkivaa testausta ei voi luonteensa vuoksi automatisoida, eikä puutteellisesti dokumentoidulle järjestelmälle voi rakentaa laajaa automaatiota. Luovuuttakaan ei voi jättää koneiden tehtäväksi.

Valmistuvan ohjelman testaava käyttäminen paljastaa usein sellaisia asioita, joita ei testiä suunniteltaessa osattu arvata. Tätä ei voi nyt eikä ihan pian automatisoida!

Mitro Kivinen toimii testausasiantuntijana Qentinel Oy:ssä ja on mukana Testauksen osaamisyhteisön ohjausryhmässä.

Qentinel on riippumaton ohjelmistojen laadunvarmistuspalveluja tarjoava yritys.