Testiautomaatio-opas

Opas testiautomaation perusteista liiketoiminnan näkökulmasta

VALA

Miksi?

Tämä opas auttaa sinua ymmärtämään testiautomaation perusteet sekä tiedostamaan, mitä kaikkea sinun tulee ottaa huomioon ennen testiautomaatioon investoimista.

Lukemisen iloa!

SISÄLTÖ

1. Mitä on testiautomaatio?
2. Testiautomaation hyödyt ja haasteet
3. Sopiiko testiautomaatio sinun tilanteeseesi?
4. Mitä ottaa huomioon ennen käyttöönottoa?
5. Mitä ottaa huomioon valittaessa työkaluja?
6. Työkalusuosituksemme erilaisiin ympäristöihin
7. Mitä tapahtuu, kun testiautomaatio-ympäristö on pystytetty?
8. Yhteenveto

1. Mitä on testiautomaatio?

Testiautomaatio (TA) tarkoittaa toimintaa, jossa sopivia työkaluja käyttämällä saadaan tietokone toteuttamaan testitapauksia, ilman ihmisen puuttumista itse testausvaiheeseen.

Tärkein asia testiautomaation ymmärtämisessä on ymmärtää, että se on ainoastaan työkalu. Se ei ole testausstrategia, eikä se ymmärrä tuon taivaallista suunnittelusta tai testien kattavuudesta ja niin edelleen.

Yleensä testiautomaatioprosessin lopputulemana tulisi olla testattavalle sovellukselle räätälöity testiautomaatio -viitekehys, joka koostuu erilaisista työkaluista.

Yleisiä vaatimuksia testiautomaatio-ratkaisulle

- Uusien testitapausten kirjoittaminen on tehty riittävän yksinkertaiseksi.
- Tulokset ovat helposti saatavissa ja helppolukuisia.
- Tulosten raportointi ulkoisiin järjestelmiin (esim. Jira, TestRail).
- Koodikannan tulisi olla helposti laajennettavissa, ylläpidettävissä ja muokattavissa.
- Kytkeminen automaattiseksi osaksi julkaisuputkea
- Tulee toimia eri käyttäjärjestelmillä (Windows, Linux. Mac, mobiili).

Erilaisia lähestymistapoja

Käyttöliittymätestaus

Rajapinnat

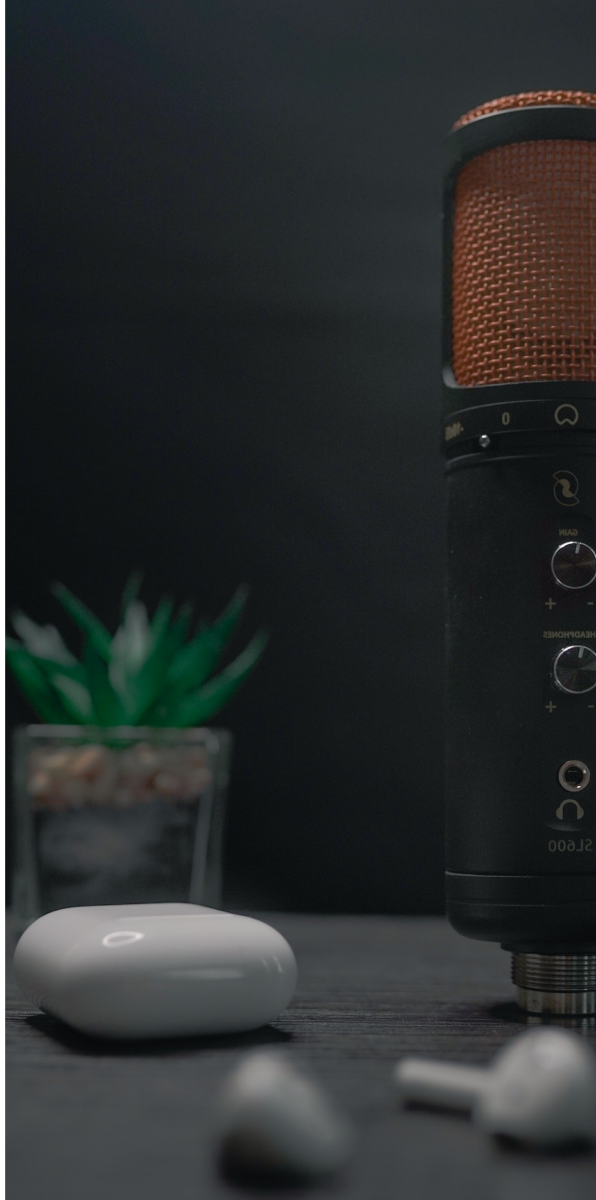
Yksikkötestaus

"Nauhoitustyökalut"

Yksi "helpoksi" koettu tapa lähestyä UI testiautomaatiota on erilaiset nauhoitustyökalut, joilla pystyy nauhoittamaan ihmisen tekemää testausta ja toistamaan tätä "automaattisesti" jälkikäteen.

Edut:
- Helppo ja nopea tapa tehdä automaatiota. Ei vaadi teknistä tietämystä testaajalta.

Haasteet:
- Pitkässä juoksussa ylläpito tulee haastavaksi ja työlääksi
- Kaikkea ei pysty automatisoimaan tällä lähestymistavalla
- Pienikin muutos käyttöliittymässä johtaa testien rikkoutumiseen
- On luultavasti kallis vaihtoehto pitkällä aikavälillä huonon ylläpidettävyytensä johdosta



2. Testiautomaation hyödyt ja haasteet

HYÖDYT	HAASTEET
Luotettava	Käyttöönoton kesto
Nopea	Vaatii työkaluja
Tarkka	Epäonnistuneen testitapauksen analysointi
Ei koskaan unohda	Automaattiset testit ovat tuotantokoodia
Parantaa testien kattavuutta	Vaatii spesifiä osaamista testaajilta
Säästää rahaa pitkässä juoksussa	Testien haavoittuvaisuus (erityisesti käyttöliittymätestauksessa)
Resurssien vapautuminen mielekkäämpiin testaustehtäviin	Hyödyt tulevat usein keskipitkällä tai pitkällä aikavälillä
	Legacy-koodi ei usein ole kirjoitettu automaatiota silmällä pitäen

3. Sopiiko testiautomatio sinun tilanteeseen?

Testiautomaatiota on mahdollista käyttää lähes kaikissa ohelmistokehitysprojekteissa. Täm ä ei kuitenkaan tarkoita, että kaikkea voi, tai pitäisi automatisoida.

Joskus ihmisen tekemä (manuuaalinen) testaus on sopivampi lähestymistapa. Seuraavalla sivulla on esimerkkejä tällaisista tilanteista.

Milloin manuaalinen testaus

- Kun tarvitaan ihmistä validoimaan tiettyä asiaa. Esimerkiksi käyttöliittymän ulkoasua tai käytettävyyttä.
- Kun ohjelmiston ominaisuudet ovat vielä kehitettävänä (niiden käyttötarkoitus tai käytettävyys voi vielä muuttua paljon).
- Kun ohjelmiston ominaisuudet ovat liian monimutkaisia, jolloin automaation pystyttäminen olisi liian työlästä.

Minkälaista testiautomaatiota?

Kuten aiemmin mainittiin, testiautomaatiota voidaan soveltaa eri "kerroksiin".

Jokaisella kerroksella on oma käyttötarkoituksensa eivätkä ne sulje toisiaan pois. Joissain tapauksissa on järkevää automatisoida kaikki kerrokset ja joskus taas vain osa.

Kun on päätetty mitä automatisoidaan, on työkalujen valinta kriittistä. Oikeiden työkalujen valinta voi määrittää eron onnistumisen ja epäonnistumisen välillä.

Käyttöliittymätestaus

Rajapinnat

Yksikkötestaus

4. Mitä ottaa huomioon ennen käyttöönottoa

Testiautomaatio ei ole mikään taikanappula, joka ratkaisee ongelmia kun sitä painetaan. Testiautomaatio on ainoastaan työkalu. Tehokas sellainen oikein käytettynä.

Kuitenkin, jos sitä käytetään väärin, voi se tehdä enemmän hallaa kuin hyvää. Ensimmäinen asia, mitä tulee pohtia, on sen hyödyt ja haitat.

TA vaatii aikaa ja rahaa – milloin se kannattaa?

Testiautomaation käyttöönotto vaatii aikaa ja rahaa – tämä on yksi tärkeimmistä muistettavista asioista kun pohditaan prosessin aloittamista. Vaikka kaikki muut vaadittavat asiat olisivat kunnossa (sovellus voidaan automatisoida, on tarpeeksi osaavia ihmisiä jne.) vaatii silti aina aikaa, että automaatio alkaa tuottaa tuloksia.

Testiautomaatio alkaa tuottaa hyötyjä vasta aikaisintaan keskipitkällä aikavälillä. Hyvinkin automatisoidun testitapauksen hyödyt alkavat näkyä vasta kun testitapaus on ajettu useammassa regressiosyklissä. Ajan myötä tietyn ominaisuuden automatisoitu testaus laskee kyseisen ominaisuuden testaamisen kustannusta, koska kun testi on valmis ja toimiva, ei sen ajaminen maksa juuri mitään.

TA vaatii huolellista testien suunnittelua ja hyvää regressiostrategiaa. Kuten mainittua, TA on vain työkalu. Työkalu, jonka käyttöön vaaditaan aina ihmistä.

TA vaatii aikaa ja rahaa – milloin se kannattaa?

Kaikkea ei tule automatisoida. Automatisoida kannattaa ainoastaan ne tapaukset, joista on odotettavissa riittävästi lisäarvoa.

Hyviä kandidaatteja automatisointiin ovat usein testit, joissa:

- Testaaminen vaatii ihmiseltä paljon aikaa
- Testaamiseen sisältyy paljon datan validointia
- Testaaminen tehdään jokaisessa regressiosyklissä
- Testaaminen on ihmiselle tylsää tai vaikeaa
- Testaamista ei voida tehdä muulla tapaa
    - (esim. ei-toiminnallinen testaus, suorituskykytestaus, stressitestaus)

TA on ohjelmistokehitystä – tarvitset osaavia ihmisiä

TA vaatii osaavia tekijöitä. Osaava tiimi on yksi merkittävimmistä testiautomaation onnistumista määrittävistä tekijöistä. Onnistunut testiautomaatio ei myöskään ole realismia ilman ohjelmistokehittäjien apua.

Joitakin työkaluja mainostetaan taianomaisina ratkaisuina, jotka tuottavat hienoja tuloksia ilman ohjelmointiosaamista. Tällä ei ole mitään tekemistä todelllisuuden kanssa. Jos joku myy sinulle tällaista ratkaisua, älä luota siihen. Myyjä ei joko tiedä lainkaan mistä puhuu, tai sitten hän koittaa käyttää hyväksi luottamustasi.

TA on tosiasiassa lähempänä ohjelmistokehitystä, kuin testausta. Testitapaukset ja TA frameworkit ovat tuotantokoodia. Joissakin tapauksissa tietyn testitapauksen automatisointi voi olla hyvinkin haastavaa ja vaatia melko edistynyttä ohjelmointiosaamista. Siis kertauksena, tarvitset ohjelmistokehittäjiä, osaavia sellaisia.

TA on ohjelmistokehitystä – tarvitset osaavia ihmisiä

Nykyään on tarjolla hyviä frameworkeja, jotka mahdollistavat testaajien ja kehittäjien työn paremman rajaamisen, jolloin jokainen voi keskittyä omiin vahvuuksiinsa. Siis kehittäjät kehittävät frameworkia ja testaajat käyttävät heidän luomiaan työkaluja ja frameworkeja automatisoidakseen manuaalisia testitapauksia.

Usein TA vaatii tukea myös tiimeiltä jotka kehittävät itse testattavaa sovellusta. Aina sovelluksia ei nimittäin ole kehitetty testiautomaatiota silmällä pitäen. Tällöin pienetkin muutokset itse testattavassa sovelluksessa voivat auttaa testiautomaation kannalta suuresti. Esimerkiksi käyttöliittymätestauksessa helpottaa merkittävästi, jos tietyillle elementeille on olemassa uniikit tunnisteet (ID:t).

5. Mitä ottaa huomioon työkaluja valittaessa

Tärkeimpiä tekijöitä:

BUDJETTI: Jo budjettia tutkaillessa tietyt työkalut voivat päätyä ulos listalta niiden kustannusten takia.

TUTKIMUS: Tutki eri vaihtoehtoja ja kerää informaatiota (internet riittää tähän hyvin). Varmista, että päätöksestä vastaavat henkilöt varmasti ymmärtävät tilanteen tarpeet ja haasteet.

OSAAMINEN: Tiimin osaaminen voi määrittää sopiiko avoimen lähdekoodin ratkaisu paremmin kuin kaupallinen tuote.

TESTAAMINEN: Kannattaa tehdä Proof of Concept tyylisiä kokeiluja sen sijaan, että vain valitset tietyn työkalun ilman käytännön evaluointia.

Avoimen lähdekoodin työkalujen hyödyt ja haitat

HYÖDYT	HAITAT
Joustava – jos sinulla on riittävästi kyvykkyyttä, voit tehdä lähes mitä vaan	Vaatii monesti vahvaa osaamista tiimiltä
Jotkut sovellukset ovat laajalti käytössä ja tukea ja yhteisöjä on saatavilla hyvin	Jos et saa yhteisöiltä tukea, olet omillasi
Ei lisenssikustannuksia	Kuluja ylläpidosta (tosin monesti tämä on vähemmän kuin lisenssit)
Pääsy lähdekoodiin	Ei paljoa valmiita ominaisuuksia "out of the box"
Useimmat työkalut tukevat useita ohjelmointikieliä	Vaatii paljon tukea kehittäjiltä (tämä voidaan nähdä myös positiivisena asiana)
Tietyt työkalut kuten esim. Selenium tai Appium ovat jo melko kypsiä	Työkalun muokkaaminen tai bugien korjaaminen voi joskus olla tarpeen

Kaupallisten työkalujen hyödyt ja haitat

HYÖDYT	HAITAT
Voi mahdollistaa nopeamman alun/käyttöönoton projektille	Lisenssikustannukset
Valmiit "built-in" ominaisuudet, joista on monesti hyötyä	Vain tuotteen tarjoaja voi lisätä ominaisuuksia
Tekninen tuki	Ei niin joustava – jotkut työkalut esim. tarjoavat tuen vain yhdelle ohjelmointikielelle
Voi olla helpompi käyttää	Ei välttämättä sovellu hyvin tulevaisuuden tarpeisiin
Ei-teknisten ihmisten voi olla mahdollista automatisoida yksinkertaisia testitapauksia (tämä ei välttämättä ole hyvä asia)	Ei pääsyä lähdekoodiin

6. Suosituksemme erilaisiin tilanteisiin

Robot Framework
Geneerinen avainsanapohjainen framework. Voidaan integroida useimpiin projekteihin.

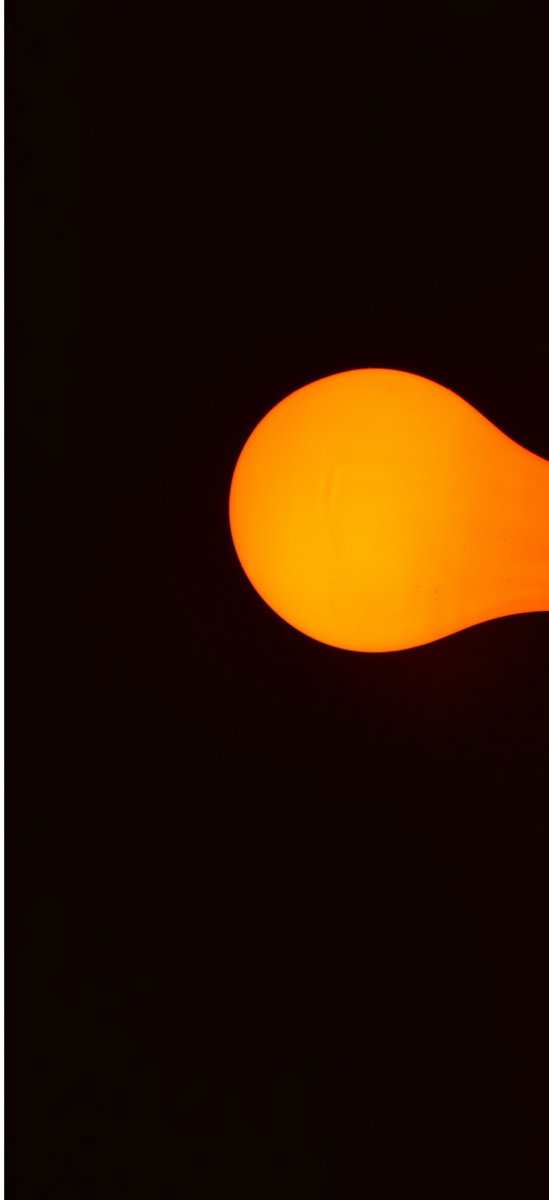
Selenium & Playwright käytetyimpiä avoimen lähdekoodin työkaluja web-testiautomaatioon.

Appium
Erinomainen työkalu mobiilitestiautomaatioon. Voi auttaa testien uudelleenkäyttämiseen iOS:n ja Androidin välillä. Sopii hyvin blackbox-testaukseen.

Espresso
Android testaamiseen dedikoitu työkalu. Soveltuu hyvin white-box testaukseen. Nopeampi kuin Appium. Ei tukea iOS:lle.

XCTest
Vastaava kuin Espresso, mutta iOS:lle.

OpenCV
Avoimen lähdekoodin kirjasto, joka soveltuu erityisesti testiautomaatioon, missä tarvitaan kuvantunnistusta.



7. Mitä tapahtuu käyttöönoton jälkeen

- Määrittele miten testiautomaatiota tehdään jatkossa:
  - missä tapauksissa tai mitä testejä automatisoidaan
  - Miten testiautomaatiota käytetään regressiosykleissä
- Panosta testiautomaation sisäistämiseen koko organisaation laajuudella
- Kouluta uusia ihmisiä käyttämään testiautomaatiotyökaluja ja -frameworkeja
- Rakenna automaattisia regressiosyklejä käyttämällä jo automatisoituja testejä
- Implementoi automaattinen raportointi ja KPI dashboard regressiotestien näkyvyyden ja seuraamisen parantamiseen, jotta johtokin pysyy paremmin kärryillä
- Lisää toiminnallisuuksia testiautomaatioframeworkiin
- Lisää uusia testitapauksia uusille toiminnallisuuksille
- Rakenna jatkuvan integraation putki, joissa ajetaan aina automaattitestit uuden releasen jälkeen

* Kaikki yllä oleva ei luonnollisesti välttämättä päde kaikkiin projekteihin

8. Johtopäätökset

- Testiautomaatio ei ole taikanappula vaan ennemminkin työkalu, joka oikein käytettynä voi tuoda merkittäviä hyötyjä.
- Kuten koneet yleensä, on testiautomaatio luotettava, nopea ja tarkka. Näiden ominaisuuksien johdosta se voi säästää merkittävästi rahaa ja parantaa testauksen laatua, kunhan olet kärsivällinen.
- Testiautomaation haittoja ovat kustannusten painottuminen projektin alkupäähän sekä tarve osaaville teknisille tekijöille.
- Kaikkea ei voi eikä pidä automatisoida, monet asiat on syytä testata ihmisvoimin jatkossakin.
- Hyviä kandidaatteja automatisointiin ovat esimerkiksi testit, jotka: toistetaan usein, vievät paljon aikaa tai ovat tylsiä tai vaikeita toteuttaa. Lisäksi tietynlaista testaamista ei voi tehdä muulla tapaa (esim. suorityskykytestaus).
- Oikeat työkalut ovat elintärkeitä onnistuneessa testiautomaatiossa. Ota aikaa ja valitse käytettävät työkalut huolella.
- Avoimen lähdekoodin työkalut ovat joustavampia ja mahdollistavat enemmän. kuin lisenssimaksulliset kaupalliset työkalut Toisaalta ne kuitenkin vaativat osaavampia tekijöitä.

Kiitos!

Ota yhteyttä, jos haluat kuulla lisää.

Tekstisisältö: Dragos Guberna

www.valagroup.com

Teemu Pesonen, Liiketoimintajohtaja: Laatu & Automaatio

+358 400 513 514

teemu.pesonen@valagroup.com

VALA