

eficode

EFICODE PIKAOPAS

TESTIAUTOMAATIO ROBOT FRAMEWORK

WWW.EFICODE.FI

EFICODE PIKAOPAS

Testiautomaatio

SIVUT: 03 - 10

EFICODE PIKAOPAS

Robot Framework

SIVUT: 11 - 18



EFICODE PIKAOPAS

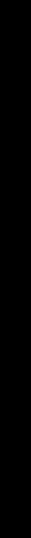
TESTIAUTOMAATIO

Lopeta toistuva
tuhlaaminen, tee vain
kerran.

Testiautomaatio ei ole pelkkä kustannusleikkuri.

**Testiautomaatio auttaa ohjaamaan kehitystä
tehokkaammin, reagoimaan asiakkaiden toiveisiin
paremmin, sekä julkaisemaan muutoksia asiakkaiden
käyttöön paljon nopeammin.**

**Testiautomaatio on
erottamaton osa modernia
ohjelmistotuotantoa.**



www.eficode.fi

TESTIAUTOMAATIO

Testiautomaatiosta on muutamassa vuodessa tullut välttämätön osa toimivaa ja asiakaskeskeistä tuotekehitystä. Jotta organisaation osaaminen saadaan käytettyä mahdollisimman tehokkaasti hyödyksi, on toistuvat rutiinitehtävät siirrettävä tietokoneille.

Testiautomaatiota on helppoa esitellä konkreettisen esimerkin avulla. Otetaan keskimääräinen ohjelmistoprojekti, jossa on noin 500 selaimella ajettavaa testitapausta. Mikäli jokaisen testitapausten ajamiseen käytetään keskimäärin viisi minuuttia aikaa, vie koko ohjelmiston testaus yhden ainoan kerran yhdeltä testaajalta työaikaa noin viikon verran. Jos lasketaan vielä ulkopuolisena konsulttina ostetun testaajan kustannuksia, maksaa yksi testiajo noin 2 000 euroa. Kun otetaan vielä huomioon testaaminen kolmella yleisemmällä selaimella, joista jokainen on oma testiajansa, kustannukset yhdestä ainoasta käsin tehdystä testauksesta nousevat helposti 6 000 euroon.

500 käsin ajettua testiä x 5 min = 42 tuntia työtä

Asiakasprojekteista keräämäämme informaatioon perustuen yhden normaalin selainpohjaisen testin automatisointiin kuluu korkeintaan kaksi tuntia, jolloin esimerkin testitapausten automatisointiin kuluisi yhteensä noin tuhatkunta tuntia. Vaikka testiautomaatioasiantuntija olisi kustannukseltaan kaksinkertainen verrattuna testaajaan, jäävät automatisoinnin kustannukset alle sadantuhannen euron. Summa vastaa alle kahtakymmentä kokonaista testiajoa käsin.

Testien automatisointi maksaa itsensä takaisin keskimäärin puolessa vuodessa.

EFICODE TUTKITUTTI

Testiautomaatio vähentää
laadunvalvonnan
kustannuksia keskimäärin yli

PROJEKTIN AIKANA

25 %

PITKÄLLÄ AIKAVÄLILLÄ

75 %

Automaattisten testien toistuva ajaminen ei käytännössä tuo mitään lisäkustannuksia. Keskimääräisessä ohjelmistoprojektissa testien automatisointi maksaa siis itsensä helposti takaisin alle puolessa vuodessa. Olemme todistaneet tämän useissa automaatioprojekteissamme.

Testiautomaatio muuttaa organisaation tapaa toimia

Ohjelmistotestaus on perinteisesti mielletty vaiheeksi, jossa toteutetun ohjelmiston toiminnallisuus varmistetaan dokumentoitujen testitapausten pohjalta kehitystyön päätteeksi.

Jotta voidaan luoda alkuperäisen tarpeen täyttämisen lisäksi kilpailukykyisiä ja miellyttäviä tuotteita, ei enää riitä että palveluiden toimivuus varmistetaan juuri ennen julkaisua. Ja jotta voidaan tehokkaasti vastata vikailmoituksiin tai käyttäjien muutospyyntöihin, ei enää riitä että kehitysehdotukset ja muutospyyntöt sysätään seuraavaan puolivuositaiseen julkaisuun.

Jos kaikki laadunvarmistus tehdään aktiivisimman kehitysvaiheen aikana käsin, testaukseen käytettävässä ajassa ohjelmisto ehtii yleensä muuttua merkittävästi, ja pahimmillaan jo toteutetut ja aiemmin testatut ominaisuudet ovat saattaneet olla rikki useita päiviä. Nykyaikaisessa ohjelmistokehityksessä tuotteen laadusta on saatava tietoa huomattavasti nopeammin, jotta kehitystyössä voidaan paremmin ja luotettavammin keskittyä laadukkaan tuotteen tekemiseen.

Tuotekehityksessä on otettava entistä paremmin huomioon tuotteen käytettävyys ja käyttäjien tarpeet jo ennen tuotekehityksen aloittamista, ja varsinkin aktiivisen kehitystyön aikana. Testiautomaatio tarjoaa oikein käytettynä testaajille paremmat mahdollisuudet tutkivaan testaukseen, jossa tuotteesta etsitään uusia, vielä tuntemattomia ongelmia, ja varmistetaan että tuotetta on helppoa ja miellyttävää käyttää.

Automaattisten testien liittäminen toteutettavan järjestelmän vaatimuksiin auttaa seuraamaan kehityksen reaaliaikaista etenemistä luotettavammin myös liiketoiminnan näkökulmasta.

Testiautomaatio mahdollistaa ketterän organisaation ja reaaliaikaisen laadunvalvonnan

Testiautomaatio tuo laadunvarmistuksen osaksi määrittely- ja toteutustyötä. Kun tuotteelle kirjoitetut vaatimukset yhdistetään automaattisesti ajettaviin testitapauksiin, voidaan ohjelmiston toteutusta ja vaatimusten täyttymistä seurata reaaliaikaisesti. Jos automaattiset testit kirjoitetaan lisäksi luonnollisella kielellä, voi koko kehitysorganisaatio käydä avointa keskustelua vaatimusten ja siihen liittyvien automaattisten testien avulla. Tämä vähentää väärinymmärryksiä ja auttaa reagoimaan muutoksiin huomattavasti aikaisemmin.

Tuleeko testien ylläpitämisestä kustannuksia?

Automaattiset testitapaukset kannattaa mieltää osaksi tuotetta ja tuotekehitystä. Kun testitapaukset on luotu, niitä pitää ylläpitää samalla tavalla kuin tuotteen ohjelmakoodia ja sen muita riippuvuuksia. Tuotteen muuttuessa kyseiseen muutokseen liittyvät testitapaukset täytyy yleensä päivittää vastaamaan uutta toiminnallisuutta.

Se, tuleeko näiden muutosten tekemisestä ylimääräisiä kustannuksia verrattuna esimerkiksi käsin tehtävien testitapausten dokumentaation päivittämiseen, riippuu kehitettävästä tuotteesta. Jos automaattiset testitapaukset on suunniteltu hyvin ja toteutettu rakenteellisesti oikein, niin parhaimmillaan ohjelmiston muutoksiin liittyvät päivitykset täytyy tehdä vain yhteen paikkaan, jolloin ne päivittyvät samalla kaikkiin uutta ominaisuutta käyttäviin testitapauksiin. Tällaista testien kirjoittamisen tapaa kutsutaan avainsanapohjaiseksi testien kirjoittamiseksi, ja siitä voi lukea lisää toiselta puolelta tätä opasta.

Miten testiautomaatio kannattaa ottaa käyttöön?

Testiautomaatio on luonnollisesti helpointa ja kustannustehokkainta ottaa käyttöön uudessa tuotekehitysprojektissa. Se ei kuitenkaan tarkoita ettei testiautomaatio toisi kustannussäästöjä myös jo toteutettujen ja ylläpidossa olevien tuotteiden laadunvalvontaan.

Kun testiautomaatio otetaan mukaan uuteen tuotekehitysprojektiin, niin se kannattaa sitoa koko tuotteen kehityksen kaareen mukaan. Testitapaukset on

hyvä liittää toteutettaviin vaatimuksiin, ja ne kannattaa kirjoittaa samanaikaisesti kyseisen ominaisuuden toteutuksen aikana. Näin testi on valmis suunnilleen samaan aikaan tai jopa ennen uuden ominaisuuden valmistumista, ja varmistaa että se on toteutettu alun perin määritellyllä tavalla.

Uudet kehitysprojektit

- Valitse sellainen testiautomaatiotyökalu, josta saatavat testitulokset ovat kaikkien projektiin liittyvien henkilöiden ymmärrettävissä.
- Varmista että testiautomaatiotyökalu voidaan liittää yhteen toteutuksessa käytettävien teknologioiden kanssa.
- Varmista että testit voidaan liittää yhteen tuotteeseen toteutettavien vaatimusten kanssa. Tällöin vaatimusten toteutumista voi seurata testitulosten avulla.
- Varmista, että ne henkilöt, jotka ovat vastuussa testien kirjoittamisesta, tuntevat olonsa mukavaksi kyseisen testiautomaatiotyökalun kanssa.

Ylläpidossa olevat tuotteet

- Mieti ensimmäiseksi onko testiautomaation käyttöönotto kyseiselle tuotteelle järkevää. Kuinka suuret kustannukset nykyisestä laadunvarmistuksesta tulee, ja kuinka pitkä elinkaari tuotteella on jäljellä?
- Valitse testiautomaatiotyökalu siten, että sillä voidaan mahdollisimman vaivattomasti testata tuotetta siihen jo toteutettujen rajapintojen tai käyttöliittymien kautta.
- Pyri määrittämään avaintoiminnallisuudet, jotka tuotteesta tulisi testata ensimmäisenä. Sen lisäksi, että saat eniten hyötyä automatisoinnista, voit samalla arvioida testiautomaation käyttöönoton kustannuksia koko tuotteelle.
- Varmista, että tuotteen jatkokehitystä ja ylläpitoa tekevät henkilöt ovat motivoituneita oppimaan testiautomaatiota, ja ylläpitämään testitapauksia osana muuta ylläpito- ja kehitystyötä. Muuten testiautomaatio menee hukkaan pian sen käyttöönoton jälkeen. ■

1. Automaattiset testit säästävät aikaa ja rahaa

Automaattisten testien ajaminen on käytännössä ilmaista ja aina käsin tehtävää testausta nopeampaa. Kerran automatisoidut testit voidaan ajaa käytännössä kuinka usein tahansa ja erilaisia ympäristöjä kuten päätelaitteita tai selainversioita käyttäen.

2. Testiautomaatio luo pohjan ketterälle organisaatiolle

Vaatimuksiin liitetyt ja luonnollisella kielellä kirjoitetut automaattiset testitapaukset toimivat tukevana perustana ohjelmistokehitykselle. Kun koko organisaatio voi käydä avointa keskustelua testitapausten ympärillä, vähenee väärinymmärrysten määrä ja kehitystiimien on helpompi ottaa kokonaisvastuu omasta toteutustyöstään.

3. Automaattinen laadunvalvonta kuvaa kehitystyön tilaa reaaliaikaisesti

Automaattiset testit on helppo yhdistää tuotteen vaatimuksiin. Kun testit ajetaan jokaisen ohjelmistoon tehdyn muutoksen jälkeen, voidaan tuotekehityksen tilaa seurata luotettavasti ja reaaliaikaisesti.

4. Ihmisten osaaminen saadaan parempaan käyttöön

Kerran automatisoidut testit vapauttavat osaamista muun muassa tuotekehitykseen, parempien testitapausten suunnitteluun ja toteutettavan palvelun käytettävyyden ja käyttömukavuuden testaukseen.

5. Yhteistyö toimittajien kanssa helpottuu

Testiautomaatio kattaa suuren osan sellaisista laatuporteista, joita on helppo käyttää lähtökohtana muualta hankittavan ohjelmiston neuvotteluissa. Tämän jälkeen ei ole kenenkään mielipiteestä kiinni kuinka hyvin sovitut laatuportit ja -metriikat toteutuvat, vaan energia voidaan käyttää rakentamaan yhteistyöhön.

6. Erilaiset ympäristöt voi testata samalla vaivalla

Jos testit on automatisoitu oikeaoppisesti, voidaan testitapaukset ajaa samalla vaivalla erilaisia ympäristöjä, kuten eri päätelaitteita tai selainversioita, vasten. Tällöin voidaan varmistua siitä, että kaikki asiakkaiden käyttämät ympäristöt käyttäytyvät samalla tavalla, eikä ikäviä yllätyksiä esimerkiksi uusista selainversioista tule.

7. Automaatio helpottaa vaikeita ja pitkäkestoisia testauskohteita

Toisin kuin ihminen, automaatio jaksaa suorittaa sellaisiakin yksittäisiä testitapauksia, jotka kestävät tunteja tai kauemmin. Kun testaus löytää virheen, voi automaatio pureutua pinnan alle ja diagnosoida suoraan, missä ohjelmiston välikerroksessa vika piilee. Tällöin myös vian korjaaminen nopeutuu.

8. Kaikki laaturaportointi voidaan yhdistää

Ohjelmistoilla on enenevässä määrin viranomais- ja muita raportointivastuita. Sen sijaan että raportit tehdään käsin, automaatioon voidaan kehittää mekanismeja, joilla raportti saadaan tuotettua täsmälleen sitä vaativan tahon haluamassa muodossa.

9. Raportoitujen bugien hukkaaminen loppuu

Loppukäyttäjän löytämät ohjelmistovirheet voidaan jo ennen korjaamista kirjoittaa automaattisiksi testeiksi. Tällöin ne eivät vahingossakaan häviä tai jää testaamatta.

10. Ohjelmiston jatkekehitys helpottuu pitkällä aikavälillä

Pitkäikäisissä ja suurissa ohjelmistoissa suurimmaksi haitaksi muodostuu uuden toiminnallisuuden odottamattomat vaikutukset sellaisissa osissa, joista ei enää tiedetä mitkä liiketoiminnalliset päätökset johtivat toiminnallisuuteen. Sen lisäksi, että automaatio löytää odottamattomat virheet jo kehityksen aikana, ne myös dokumentoivat liiketoiminnallisia päätöksiä. On helppoa todeta, ettei liiketoiminnallinen päätös enää päde ja täten ohjelmistoa voidaan turvallisesti muuttaa. ■

HAASTE / RATKAISU

Testiautomaation aloituskustannukset ovat korkeat.

Testauksen automatisointi kannattaa aina aloittaa muutaman tärkeimmän ja selkeimmän testitapauksen automatisoinnilla, jolloin automatisoinnin kustannuksia on helpompi arvioida ja verrata nykyisiin kustannuksiin.

Tuotettamme ei voi testata automaattisesti, koska se on niin monimutkainen.

Projektihistoriassamme ei ole tullut vastaan vielä yhtään ympäristöä, jonka testausta ei olisi voinut oikeanlaisilla työkaluvalinnoilla automatisoida.

Testiautomaation tekemiseen ei ole nyt aikaa.

Testien automatisointi jää hyvin usein muiden kiireiden jalkoihin. Siksi on tärkeää aloittaa automatisointi ketterästi ja mikäli oma tuotekehitys on kiireinen, valita projektiin sopiva kumppani.

Automaattinen testiympäristö on vaikeaa tai jopa mahdotonta pystyttää.

Tuote testataan käytännössä aina jossain ennen kuin se päivitetään tuotantoympäristöön. Jos mikään muu vaihtoehto ei ole saatavilla, niin testiautomaation voi aloittaa tässä ympäristössä.

Tuotteeseen ei voi tehdä automaattisia testejä, koska tuotekehitys on ulkoistettu alihankkijalle.

Kumpi ohjaa tuotekehitystä, yritys vai sen alihankkija? Usein alihankkijat ovat valmiita kehittämään myös omaa toimintaansa, mutta ongelmatapauksissa sopimusten uusimisessa kannattaa ottaa huomioon myös testiautomaatio. ■

Testiautomaatiota on helpointa kokeilla Eficoden testiautomaatiopilotilla. Pilotin aikana asiantuntijamme kartoittavat kanssanne tärkeimmät testitapaukset, sekä ottavat niiden avulla testiautomaation käyttöön tuotteellenne. Tuotekehitykseen otetaan mukaan myös muita devopsin periaatteiden mukaisia toimintatapoja, kuten jatkuva integraatio, jonka avulla testiautomaation tuloksia ja kehitystä on tehokasta seurata reaaliaikaisesti.





EFICODE PIKAOPAS

ROBOT FRAMEWORK

**Tee rutiinitestauksesta
Robotin hommaa.**

**Robot Frameworkin kehitys aloitettiin
Nokia Networksilla vuonna 2005 ja se on noussut
kymmenessä vuodessa organisaatioiden luottotyökaluksi
testausvaiheessa.**

**Robot Framework tarjoaa helpoimman
tavan yhdistää testattava ohjelmisto
luonnollisella kielellä kirjoitettuihin automaattisiin
testeihin.**



www.eficode.fi

ROBOT FRAMEWORK

Robot Frameworkin vahvuus perustuu siihen, että automaattisesti ajettavat testitapaukset kirjoitetaan uudelleenkäytettävään ja ymmärrettävään muotoon.

Testit yhdistetään testattavaan järjestelmään testikirjastojen avulla, jolloin testitapaukset voi kirjoittaa ympäristöstä riippumatta aina samalla tavalla. Tällöin koko kehitysorganisaatio voi käyttää testiautomaatioon samaa työkalua ja kirjoittaa yhdenmukaisia testitapauksia.

Robot Frameworkilla kirjoitettavat testitapaukset pohjautuvat avainsanoihin, joita voidaan uudelleenkäyttää eri testitapauksissa. Tällöin ohjelmiston toiminnallisuuden muuttuessa jokaista testitapausta ei tarvitse kirjoittaa uudestaan, vaan riittää että ohjelmiston muuttuneisiin osiin liittyvät avainsanat päivitetään vastaamaan uutta toiminnallisuutta.

Vaikka testit yleensä kirjoitetaankin englannin kielellä, Robot Framework ei rajoita testien kielivalintaa. Halutessaan testitapaukset voi siten kirjoittaa vaikka suomeksi.

Robotilla on kirjoitettu satojatuhansia testitapauksia, ja sitä käyttävät tuhannet organisaatiot ympäri maailmaa. Robot Frameworkin ympärille on lisäksi perustettu seitsemän Suomalaisen IT-yrityksen voimin säätiö, joka turvaa työkalun kehityksen riippumattomana, ilmaisena ja avoimena lähdekoodina. ■

MIKSI ROBOT FRAMEWORK?

JOUSTAVA Robot Framework taipuu lähes minkä tahansa ohjelmiston tai tuotteen testaamiseen. Parhaimmillaan se on web-, mobiili- ja sulautetun ohjelmiston sekä niihin liittyvän raudan testaamisessa

SELKOKIELINEN Robotilla kirjoitetut testit ovat kaikkien luettavissa, sillä testit voidaan kirjoittaa vaikka suomeksi. Vaatimukset saadaan linkitettyä testeihin saumattomasti.

ILMAINEN Robot Frameworkin avoimen lähdekoodin ansiosta sen käyttö testaamiseen on ilmaista. Samalla kun pidät kustannukset kurissa, voit siis heittää hyvästit myös toimittajaloukuille.

SUOSITTU Robot Framework on jo käytössä lähes kaikilla Suomen suurimmilla yrityksillä, sekä sadoilla pienemmillä. ■

ESIMERKKITESTI

Alla esitetty esimerkkitestit kuvaa tilisiirron tekemistä verkkopankissa. Testin nimi on kuvattu ensimmäisellä rivillä ja jokainen sisennetty rivi on yksi avainsana. Avainsana on yhdistetty testikirjaston avulla tiettyyn toimintoon, kuten tietojen syöttämiseen selainruudussa näkyvään tekstikenttään tai painikkeen painamiseen. Avainsanoille voi lisäksi antaa muuttujia, joiden avulla testitapauksia voi ajaa erilaisilla testidatan yhdistelmillä.

Making a wire transfer between bank accounts

```
Log in to netbank
From shortcut menu select      New payment
Insert recipients account      FI4950009420028730
Insert recipients name         Testi Anna
Insert message                  test payment
Insert amount                   100
Click continue
Insert correct pin code
Accept payment
Verify payment has been transferred correctly
```

PARHAAT KÄYTÄNNÖT

Kirjoittakaa testitapaukset yhdessä ja yhdistäkää ne vaatimuksiin

Kun testitapaukset kirjoitetaan yhdessä, kaikki kehitykseen osallistuvat henkilöt käyvät yhdessä läpi toteutettavan tuotteen vaatimuksia ja toiminnallisuuksia. Näin testitapausten kirjoittamisen kautta syntyy parempi yhteinen ymmärrys siitä mitä ollaan tekemässä.

Aja testit jokaisen muutoksen jälkeen

Kun testit ajetaan jokaisen muutoksen jälkeen, mahdolliset virheet ja vikatilanteet saadaan kiinni mahdollisimman nopeasti. Silloin ei myöskään rikkinäisen toteutuksen päälle vahingossakaan rakenneta uusia toiminnallisuuksia, jotka pitäisi pahimmillaan myöhemmin purkaa. Tällaista toimintatapaa kutsutaan jatkuvaksi integraatioksi ja se on yksi devops-menetelmien kulmakivistä.

Aja testit myös kehityshaaroille

Jotta tuotteen eri kehityshaaroissa toteutettavien uusien ominaisuuksien yhdistäminen yhdeksi julkaisuksi olisi mahdollisimman sujuvaa, on tärkeää että testitapaukset ajetaan myös kehityshaaroissa tehtäville muutoksille.

Aja testejä rinnakkain

Testien ajaminen rinnakkain nopeuttaa palautteen saamista ja parantaa tuotteen arkkitehtuuria. Testitapausten välille ei pääse syntymään riippuvuuksia, kun testitapaukset ajetaan samaan aikaan ja parhaimmillaan testausta varten käynnistettyjä ympäristöjä vasten.

Automaattiset testitapaukset ovat osa tuotetta

Kirjoitettuja testejä täytyy ylläpitää samaan tapaan kuin ohjelmakoodiakin. Tavoitteena on luonnollisesti mahdollisimman helposti ylläpidettävät testitapaukset, mutta mikäli testattavan järjestelmän toiminnallisuus muuttuu, niin testitapauksiin tai avainsanoihin on myös tehtävä muutoksia.

Panosta hyvään testidataan

Parhaimmillaan testiympäristö vastaa tuotantoympäristöä niin hyvin, että sitä vasten ajettujen testitapausten perusteella voidaan julkaista uusin versio asiakkaiden käyttöön. Tärkeä osa tällaista testausta on järjestelmän testauksessa käytettävä testidata, joka määrittää minkälaista tietoa tuotteen testiympäristössä näkyy. ■

TEKNISET HYÖDYT

1. Automaattiset testit takaavat ajantasaisen dokumentaation

Kun testitapaukset kirjoitetaan luonnollisella kielellä, ne kuvaavat toteutettavan järjestelmän ominaisuuksia ymmärrettävässä muodossa. Samanaikaisesti ne myös varmistavat, että tuote toimii toivotulla tavalla. Onnistuneesti ajatut testitapaukset muodostavat siten ajantasaisen kuvauksen järjestelmän toiminnasta.

2. Automaattiset testit ajetaan aina samalla tavalla

Kun testejä ajetaan käsin, on aina vaarana, että osa testeistä jää tekemättä tai testien tulokset ovat epäluotettavia testaajan näppäilyvirheen takia. Automaattiset testit sen sijaan varmistavat tuotteen toiminnan aina samalla tavalla. Jos automaattiset testit varmistavat halutun toiminnallisuuden riittävän hyvin, voidaan käsin tehtävästä testauksesta vapautuva osaaminen käyttää tuottavampaan työhön.

3. Virheet löytyvät nopeammin

Parhaimmillaan automaattiset testit ajetaan jokaisen muutoksen jälkeen, jolloin virhetilanteet saadaan kiinni lähes välittömästi. Kun virheet löydetään ajoissa, on kehitystyö miellyttävämpää ja virheiden korjaukseen kuluvat kustannukset putoavat merkittävästi.

4. Suurempi testikattavuus

Kun testitapaus on automatisoitu, se liitetään osaksi regressiotestijoukkoa, joka voidaan ajaa käytännössä kuinka usein tahansa. Tällöin jokainen automaattinen testitapaus kasvattaa toteutettavan tuotteen toiminnallisuuden varmistavaa testijoukkoa.

5. Testiautomaatio tuo varmuutta kehitystyöhön

Kun automaattiset testit varmistavat toteutettavan tuotteen toiminnon, saavutetaan kehitystyössä täysin uudenlainen turvallisuuden tunne. Kehitystyössä voidaan vapaammin keskittyä tekemään järkeviä teknisiä ratkaisuja varovaisten muutosten sijaan, kun tiedetään, että mahdollisesta virhetilanteesta saadaan automaattinen ja kohdennettu vikailmoitus. ■

ROBOT FRAMEWORK

```

*** Settings ***

Library      Selenium2Library timeout=15

Test Setup   Open browser and go to homepage
Test Teardown  Close Browser

*** Variables ***

${BROWSER}   firefox
${HOMEPAGE}  http://www.google.fi

*** Test Cases ***

Finding a blog article on Robot Framework
  Search google for      Robot Framework Eficode
  Click search result    Automatic testing with Robot
                        Framework pt. I ... - Eficode

  Wait Until Page Contains short video tutorial on the
                        power of the Robot Framework and
                        Selenium

*** Keywords ***

Open browser and go to homepage
  Open Browser  ${HOMEPAGE}  ${BROWSER}

Search google for
  [Arguments]  ${search}
  Input Text   name=q        ${search}
  Click Button name=btnG

Click search result
  [Arguments]  ${link}
  Wait Until Page Contains  ${link}
  Click Link      ${link}

```

ROBOT FRAMEWORK

Robot Framework on Eficoden yleisin vastaus, kun keskustellaan työkalun valinnasta hyväksymistestauksen automatisointiin.

Se on geneerinen ja helposti laajennettavaksi suunniteltu sovelluskehys, joka on alusta asti tehty nimenomaan hyväksymistestausta ja sen automatisointia varten. Robotissa liiketoiminnan tarpeet ilmaistaan luonnollisella kielellä testitapauksiksi ja varsinaiset testausohjelmistot toteutetaan kirjastoilla, joilla integroidaan eri testausohjelmistoja ja -teknologioita yhteen niin, että saadaan tietokone tekemään se, mikä perinteisesti on ollut testaajan käsin tehtävänä. On helppoa ottaa Robot Framework lähtökohdaksi ja pohjaksi, jonka päälle rakennetaan koko testiautomaatio, oli sitten testattavana ohjelmistona yksinkertainen mobiiliappi tai useita eri ohjelmistoja käsittävä asiakkuudenhallintajärjestelmä.

Kirjoita testitapaukset ymmärrettävästi

Testitapaukset Robot Frameworkissa voidaan määritellä luonnollisella kielellä, jolloin edustaja pystyy tarvittaessa vaikka itse kirjoittamaan, mitä testattavan ohjelmiston tulisi tehdä. Testitapaukset automatisoidaan kuitenkin ohjelmistotuotannon toimesta, jolloin testit toimivat apuvälineenä ja tarpeellisen keskustelun herättäjinä. Kommunikaatio tiivistyy ja on havainnollisesti tarkistettavissa, toimiiko toiminnallisuus halutulla tavalla; riittää että ajetaan selkokielinen testitapaus ja katsotaan mitä tapahtuu.

Avoimen lähdekoodin riemuvoitto

Robot Framework on ollut yli 10-vuotisen elinikänsä melkein koko ajan avoimen lähdekoodin tuote, vaikka sen kehitystä sponsoroikin vuosia Nokia Networks. Avoimuuden takia Robot Frameworkille on jo olemassa useita ilmaisia kirjastoja yleisimpiin testauskohteisiin, kuten web-sivujen, rajapintojen, mobiiliappien sekä tietokantojen testauksen. Avoimen lähdekoodin tuotteiden suosijat välttävät myös kalliit toimittajaloukot, sillä osajia kuin myös kilpailukykyistä koulutusta löytyy todennäköisemmin. ■

	ROBOT FRAMEWORK	CUCUMBER	FITNESSE	SELENIUM
01	Avointa lähdekoodia	✓	✓	✓
02	Ydinteknologia	Python	Ruby	Java
03	Teknologiat, joilla toimii "out-of-the-box"	C#, Java	Java	C#, Python
04	Testitapaukset ovat...	...luonnollisella kielellä tiedostossa	...luonnollisella kielellä tiedostossa	...taulukko wiki-sivulla
05	"Workflow"-tyyliset testitapaukset	✓	✗	✗
06	Data-driven-tyyliset testitapaukset	✓	✗	✓
07	Gherkin-tyyliset testitapaukset	✓	✓	✗
08	Raportointi	HTML	Ulostulo terminaaliohjelmaan	HTML
09	Millä laajennettava? (Ohjelmointikielet)	Luonnollisella kielellä. Kaikki.	Ruby	Kaikki
10	Käytettävissä yleisimmistä CI-palvelimista	✓	✓	✗
11	Oma erikoistunut editori	✓	✗	✗
12	Vapaavalintainen editori	✓	✓	✗

MITEN PÄÄSEN ALKUUN?

1. Kerää ja määrittele testitapaukset

Tärkeintä on ensimmäiseksi hahmottaa, minkälaisia testitapauksia tuotteen toiminnallisuuden varmistamiseksi on jo olemassa ja minkälaisia testitapauksia niiden lisäksi tarvitaan. Ilman ymmärrystä testauksen tarpeesta, testiautomaation toteutus on hyödytöntä.

2. Automatisoi muutama testitapaus

Kun aloittaa muutamalla kriittisellä tai helposti toteutettavalla testitapauksella, huomaa nopeasti minkälaisia haasteita testiautomaation toteutuksessa on mahdollisesti tiedossa.

3. Ota testiautomaatio osaksi arkipäivää

Jotta automaattisista testeistä saisi kaiken hyödyn irti, on tärkeää että testien ajaminen ja testien kirjoittaminen otetaan osaksi tuotekehityksen arkipäivää. Tällöin testitapaukset varmasti testaavat, että tuote toimii toivotulla tavalla ja lisäksi tuotteen testikattavuus kasvaa tasaisesti.

4. Laajenna testiautomaatio kattamaan koko tuote

Jokainen automatisoitu testitapaus vähentää laadunvarmistuksessa tehtävää toistuvaa työtä jokaisen muutoksen jälkeen tai viimeistään ennen uuden version julkaisua. Mitä enemmän tuotteesta on testattu automaattisesti, sitä kustannustehokkaampaa tuotteen jatkokehitys ja ylläpito on. ■

LINKIT

Robot Frameworkin kotisivut:
robotframework.org

Konenäköä ja Robot Hardwörkkiä:
<http://eficode.fi/blogi/konenakoa-ja-robot-hardworkkia/>

Jos sen voi kirjoittaa paperille, sen voi automatisoida:
<http://eficode.fi/blogi/jos-sen-voi-kirjoittaa-paperille-sen-voi-automatisoida/>

EFICODEN ROBOT FRAMEWORK -PALVELUT

Eficode auttaa yrityksiä ottamaan Robot Frameworkin käyttöön omassa liiketoiminnassaan. Teemme Robotin käyttöönottoja, koulutuksia sekä pidempiä testiautomaatioprojekteja.

Ydinosaaminen tuotteen automaattisesta testauksesta siirretään aina asiakkaillemme, jotta tuotekehitys toimisi pitkällä aikavälillä mahdollisimman tehokkaasti eikä syntyisi ydinliiketoimintaa vaarantavia toimittajaloukkuja.



YHTEYSTIEDOT

Marko Klemetti

CTO

marko.klemetti@eficode.com

044 522 5927

HEIKKI HÄMÄLÄINEN

Head of DevOps

heikki.hamalainen@eficode.com

+358 (0) 40 766 2610

KAJ JOKINIEMI

Vice President of DevOps

kai.jokiniemi@eficode.com

+358 (0) 40 592 6257

Tatu Kairi

Domo Arigato Mr. Roboto

tatu.kairi@eficode.com

040 533 9559



Eficode Oy

Pohjoinen Rautatiekatu 25, 00100 Helsinki

Birger Jarlsgatan 18A, 2tr, 114 34 Stockholm

Åkerlundinkatu 11 A, 33100 Tampere

Uplandsgade 56, 1 sal, 2300 København S.

Marcel- Breuer- Str. 15, 80807 München

De Entree 143, 1101 Amsterdam

WWW.EFICODE.COM