DEZVOLTAREA *UI/SERVICE FRAMEWORK* PENTRU AUTOMATIZAREA TESTĂRII APLICAȚIILOR *WEB*
DEVELOPMENT OF UI / SERVICE FRAMEWORK FOR WEB APPLICATION TEST AUTOMATION

Olga DARII, masterandă
CZU: 004.415.53 olga.darii@yahoo.com

This article discribes methods of specifying a web automation testing framework, its configuration and automatic tests implementation. The digital transformation demands quality assurance to achieve business objectives, such as decreased time to market, increased security, performance, and customer satisfaction. It has been demonstrated that a creation of a web automation testing framework is not a trivial task which requires new skills of quality experts.

Introducere. Schimbarea și avansarea digitalizării pot fi urmărite pretutindeni. Noile servicii digitale sunt disponibile în toate tipurile de domenii de afaceri. Pe măsură ce transformările digitale schimbă lumea, din ce în ce mai multe servicii ale unei organizații sunt expuse interacțiunii directe cu clienții. Acesta este unul dintre cei mai mari factori care impun testarea și asigurarea calității pentru atingerea obiectivelor de afaceri, cum ar fi scăderea timpului de introducere pe piață, creșterea securității, performanței și satisfacției clienților [1].

Asigurarea și testarea calității se dezvoltă rapid pentru a atinge astfel de obiective. Vom vedea tendințe ca automatizarea crescândă a practicilor de asigurare a calității și de testare, precum și schimbarea competențelor solicitate de la experții în calitate. Procesul de testare, care are loc la toate nivelurile de dezvoltare, verifică dacă produsul îndeplinește cerințele și calitatea așteptată. Acest proces este adesea efectuat manual, ceea ce este costisitor, consumator de timp și predispus la erori. Testarea automată este propusă ca soluție la această problemă, deoarece poate crește frecvența de testare, oferind feedback mai rapid dezvoltatorilor și poate rula fără interacțiunea umană [2].

Problema studierii calitative a sistemelor automatizate de testare web a intrat în vizorul mai multori savanți, și anume, Thom Garrett,

Bernie Gauf [3]. Rahul Shetty a contribuit la realizarea studiului comparativ al framework-urilor UI Selenium WebDriver, alți savanți au studiat automatizarea aplicațiilor web API: A. Doronins "Get Started with API Test Automation in Java" [4].

Avantajele testării automate sunt recunoscute pe scară largă, totuși, suportul pentru toate nivelurile – de exemplu, UI și Service în aplicații web – testarea sistemului este limitată, anume aceasta ne-a determinat să formulăm *problema cercetării*: Cum testarea software-ului poate fi îmbunătățită, extinsă și clasificată mai bine în ceea ce privește aspectele de automatizare.

Scopul cercetării: proiectarea structurii, configurarea frame work-ului de testare software și implementarea testelor automatizate tip UI/service.

Combinația testării automatizate UI și Service într-un singur Java framework, utilizând tehnologiile alese – Selenium, HttpClient, TestNG este răspândită pe larg în companii de testare software, însă proiectarea și configurarea corectă a unui astfel de framework este dificilă, totuși abordarea corectă, din punct de vedere arhitectural, aduce rezultate foarte avantajoase pe un termen îndelungat.

Înainte de a proiecta un framework, este nevoie să ne asigurăm că am făcut o alegere corectă în ceea ce ține de instrumente și tehnologii. După o analiză asiduă a instrumentelor și tehnologiilor open-source Web, ne-am oprit atenția la Maven pentru construcția și suportul proiectului, Apache HttpClient pentru testarea serviciilor, Selenium WebDriver pentru testarea interfeței de utilizator, IntelliJ IDEA pentru un mediu de dezvoltare plăcut vizibil și funcțional, TestNG pentru o configurare ușoară și eficientă a testelor, iar limbajul ales este Java, el fiind portabil, securizat și *enterprise*.

Înțelegerea părților front-end și back-end și a logicii implementate este esențială pentru o testare de succes și o acoperire maximă cu teste. Partea unui site web cu care utilizatorul interacționează direct este denumită *front-end*. Include tot ceea ce utilizatorii experimentează direct: culori și stiluri de text, imagini, grafice și tabele, butoane, culori și meniu de navigare. *Back-end* este partea serverului site-ului web. Activități, cum ar fi scrierea API-urilor, crearea de biblioteci și

lucrul cu componente de sistem fără interfețe de utilizator sau chiar sisteme de programare științifică, sunt, de asemenea, incluse în *back-end* [5]. Tehnologiile de automatizare au fost alese luând în considerare că aplicația testată utilizează *JS, nodejs, mongo, mongoose*.

Abordarea de întreținere a framework-ului de teste automatizate este efectuată pentru a testa dacă noile funcționalități adăugate la software funcționează bine sau nu. Ar trebui să luăm în considerare următorii factori înainte de a selecta un instrument: suportul pentru platforma dorită (web); costul licenței, dacă este cazul; preferința clientului; costul implicat în instruirea angajaților cu privire la instrument; cerințele hardware/software ale instrumentului de automatizare; politica de asistență și actualizare a furnizorului de instrumente de automatizare.

Selecția instrumentului de testare depinde, în mare măsură, de tehnologia pe care este construită aplicația testată, însă avantajul limbajului și tehnologiilor selectate și prezentate în lucrare constă în caracterul lor mai universal, așadar vom putea folosi acest framework și pentru testarea mai multor aplicații web.

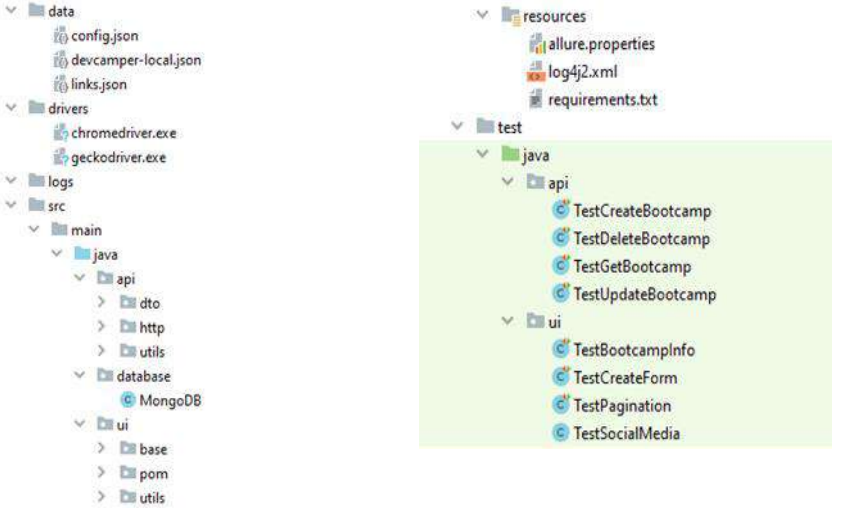


Fig. Structura framework-ului de automatizare

În proiectarea framework-ului, cu scopul de mentenanță de succes, s-a ținut cont de: configurarea dinamică a proiectului; clasificarea fișierelor în funcție de scopul acestora (testare, configurare, helpers); utilizarea tehnologiilor de înregistrare a logurilor și de raportare, și a cerințelor.

Astfel, framework-ul va fi ușor de menținut, constituind o carcasă solidă pentru crearea independentă a testelor noi și întreținerea suitelor existente (Fig).

În concluzie, proiectarea unui framework automat de testare a interfeței de utilizator și a serviciilor web este un proces anevoios, complex și consumă mult timp, deoarece necesită aplicarea gândirii abstracte. Crearea unui mediu confortabil pentru automatizarea testelor solicită cunoștințe aprofundate din domeniul IT ale experților în sfera asigurării calității.

Referinţe:

1. CHANDRA, S., AJAY, K. *Automated Software Engineering,* 2020.
2. PAGE, R., GAMBOA, R. *Essential Logic for Computer Science,* 2019.
3. GARRETT, T., GAUF, B. *Implementing Automated Software Testing: How to Save Time and Lower Costs While Raising Quality,* 2017.
4. DORONINS, A. *Getting Started with Web API Test Automation in Java,* 2018.
5. BATH, G., McKay, J. *The Software Test Engineer's Handbook Paperback.* December 1, 2018.

Recomandat
Denis NICA, dr. hab., prof. univ.

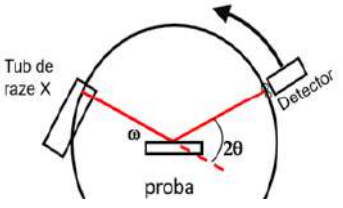
DIFRACȚIA DE RAZE X ÎN ANALIZA MATERIALELOR ȘI STRUCTURILOR SEMICONDUCTOARE

Gheorghe GHILEȚCHII, student
CZU: 539.26:621.315.592
ghiletchi.giku@gmail.com

The application of grazing incidence X-ray diffraction (XRD) along with X-ray reflectivity (XRR) has proven its applicability in multilayered structure analysis. Thin film phase composition analysis, microstrain, crystallites average dimensions and lattice parameter calculations have been carried out for nanometric epitaxial films as well.

Introducere. Difracția de raze X este o tehnică analitică de caracterizare a materialelor solide, organice sau anorganice, intensitățile de difracție fiind generate în urma interacțiunilor radiațiilor X cu electronii atomilor din rețeaua cristalină. În funcție de forma probei de studiu, se pot folosi următoarele metode: pe pulbere, sub unghiuri incidente mici (GI-XRD) și reflectivitatea cu raze X (XRR).

Metoda, rezultate și interpretări. Analiza straturilor subțiri utilizând ***difracția razelor X sub unghiuri de incidență mici (GI-XRD)*** permite determinarea parametrilor rețelei cristaline și tensiunii mecanice reziduale. Aplicabilitatea practică a metodei a fost studiată utilizând structuri cu straturi subțiri Ga_2S_3/Si, CdS/ZnO:Al/Sticlă și difractometrul de raze X (cu anod din Cu $\lambda_{K\alpha1}$=1,540598 Å, $\lambda_{K\alpha2}$=1,5444260 Å) în configurația GI-XRD (Fig.1).



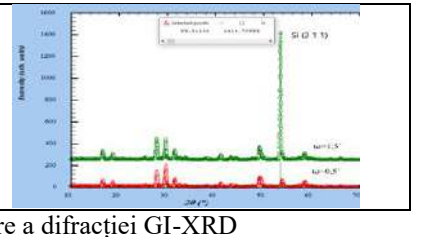


Fig. 1. Schema de cercetare a difracției GI-XRD și tabloul de difracție pentru Ga_2S_3/Si

Tubul de raze X împreună cu „optica incidentă” se fixează sub unghiul *ω* față de planul probei, iar „optica difractată” anexată la detector se mișcă cu viteza de *2θ/s* în același plan cu proba și optica

incidentă. Distribuția intensității maximelor de difracție este în conformitate cu expresia Wulff-Bragg $2d_{hkl}\sin\theta = n\lambda$ (1) [1], unde d_{hkl} este distanța dintre planurile familiei $h\ k\ l$; θ – unghiul la care s-a înregistrat maximul; λ – lungimea de undă a razelor X. În Figura 1 se observă că mărind unghiul de incidență se obțin informații de la diferite adâncimi ale probei. Astfel pentru $\omega = 1{,}5°$ se observă peak-ul provenit de la suportul de Si (3 1 1). Pentru analiza difractogramei, am utilizat setul de programe FullProf Suite [2]. Pentru determinarea parametrilor celulei elementare, am utilizat programul Dicvoll 06 care aplică metoda dihotomiei [3] și relația pătratului distanței interplanare în spațiul reciproc (2):

$$Q_I = f(a,b,c,\alpha,\beta,\gamma,h,k,l) = (d_i^*)^2 = h^2a^{*2} + k^2b^{*2} + l^2c^{*2} + hka^*b^*\cos\alpha + klb^*c^*\cos\beta + hla^*c^*\cos\gamma = \frac{1}{d_{hkl}^2} \quad (2)$$

Precizarea parametrilor rețelei cristaline și estimarea erorilor a implicat metoda *Rietveld refinement* [4], principalii parametri fiind χ^2 – goodness of fit și R-Bragg factor. Esența acestei metode constă în minimizarea divergenței intensității observate (y_i) și celei calculate (y_{ci}) (3) în fiecare punct i al difractogramei prin metoda celor mai mici pătrate.

$$y_{ci} = s\sum_{hkl} LP(2\theta_{hkl})p_{hkl}|F_{hkl}|^2F_{prof}(2\theta_i - 2\theta_{hkl})T_{hkl}A + y_{bi} (3),$$

unde s – factorul de scară; $LP(2\theta_{hkl})p_{hkl}$ – factorul de polarizare Lorentz; p_{hkl} – factorul de multiplicitate; F_{hkl} – factorul de structură; $F_{prof}(2\theta_i - 2\theta_{hkl})$ – funcția de profil a maximului i; T_{hkl} – factorul de texturare; A – factorul de absorbție; y_{bi} – intensitatea fonului în punctul i. Cele mai utilizate în practică funcții de profil sunt Lorentz (6), Gauss (7) și Pseudo-Voight (8):

$$L(x,H) = \frac{1}{\pi}\frac{H}{2}\frac{1}{(x-x_0)+(H/2)^2} (6), G(x,H) = \frac{1}{\sigma\sqrt{2\pi}}e^{\frac{-(x-x_0)^2}{2\sigma^2}} (7),$$
$$pV = I_j\eta G(x,H) + (1-\eta)L(x,H), \quad (8)$$

unde $H-$ lățimea maximului la jumate de înălțime; x_0 – poziția maximului; $\sigma = \frac{H}{2\sqrt{2\ln 2}}$.

<u>*Structura Ga_2S_3/Si.*</u> Din analiza tabloului de difracție a fost determinat că structura materialului probei este o modificație alotropă