

## Introducere în testarea automată

Lucrând cu o infrastructura software și hardware din ce în ce mai complexă, confrunțați cu creșterea continuă a cerințelor de calitate și cu necesitatea reducerii costurilor, firmele de software sunt nevoite să prețuiască tot mai mult soluții solide, ingineresti, de dezvoltare a produselor software.

Testarea produsului este în software o componentă majoră în procesul de dezvoltare. Firmele din industria tradițională – de ex. industria de mașini, de construcții, de produse electronice sau alimentare au de zeci de ani departamente de testare și verificare a calității. În materie de software, organizarea și automatizarea muncii de testare și verificare a produselor a început din motive istorice evidente abia în anii '80.

Testarea manuală, mult timp văzută ca singura soluție de a descoperi eventualele defecte, întârzie foarte mult lansarea pe piață a produsului și induce cheltuieli semnificative mai ales în cazul descoperirii efectelor laterale – atât în procesul dezvoltării unei aplicații cât și în cazul schimbărilor ulterioare.

Totodată procedurile de testare manuală, prin natura lor limitată, nu reușesc să descopere toate defectele și nu au nici o șansă să simuleze condiții de utilizare simultană, intensivă, a unei aplicații. Există numeroase căi de a îmbunătăți procesul de testare, una dintre cele mai eficiente fiind **testarea automată**. Testele automate execută o secvență de acțiuni fără intervenție umană și pot simula utilizarea unei aplicații simultan. În general, majoritatea produselor necesită să fie testate de mai multe ori, pe mai multe platforme software și hardware, ca și după schimbările ulterioare sau la lansarea unei noi versiuni de produs. Prin folosirea testării automate, costurile testărilor repetate se reduc aproape la zero.

Cele mai importante beneficii sunt:

- acoperirea tuturor etapelor de testare de la concepție până la lansare
- posibilitatea simulării testării cu mai mulți utilizatori
- posibilitatea repetării testelor
- creșterea siguranței în produs.

Un sistem de testare automată trebuie să aibă la bază două caracteristici principale:

- module reutilizabile
- întreținerea și urmărirea activității dintr-un singur punct de control

De aceea una din calitățile care trebuie să le aibă un astfel de sistem este posibilitatea de a fi ușor configurat și adaptat în același ritm cu software-ul ce se testează .

Sistemele de testare automată sunt o tehnologie în plină dezvoltare care au ca scop descoperirea și raportarea eventualelor defecte, mărirea longevității produsului și reducerea procesului de întreținere.

Testarea automată înseamnă mai mult decât o simplă captură de ecran și răspunsul la câteva scenarii : ea trebuie să înceapă cu planificarea și design-ul procedurii de testare, să conțină o serie de teste repetabile, o interfață de management al scenariilor de test și un mecanism de raportare și gestionare a erorilor descoperite în urma testării. Un sistem de test evoluat sprijină utilizatorii printr-un sistem de documentare pe tot parcursul acestui proces.

Într-o abordare mai detaliată testarea automată înseamnă:

1. planificare
  1. identificarea cerințelor și a funcționalităților
  2. gruparea acestora în condiții de test
  3. crearea cazurilor de test pentru aceste condiții
2. design

1. construcția scripturilor de test
2. generarea testelor de rulare
3. execuție
  1. crearea scenariului de rulare a scripturilor
  2. rularea uneltelor monitor pentru înregistrarea datelor
  3. înregistrarea rezultatelor pentru fiecare rulare
  4. raportarea și gestionarea erorilor
4. management
  1. generarea rapoartelor și graficelor
  2. controlul dintr-un singur punct de comandă
  3. documentarea permanentă a stadiului curent al proiectului

### Tipuri de testare automată :

1. *structurală (white-box testing)* - se verifică structura software-ului și necesită acces complet la codul sursă. Acest tip de testare verifică dacă structura codului este eficientă: bucle complicate, zone de date comune, mii de linii de cod încurcate sunt numai câteva din problemele care pot fi îndepărtate. Scopul acestui test este mărirea performanței aplicației și a lizibilității codului.

2. *funcțională (black-box testing)* – se definesc așteptările clientului de la aplicație și se verifică automat dacă software-ul se comportă conform acestor așteptări. Prin testele ce se execută se observă comportamentul aplicației, evidențiat prin datele de ieșire, fără a se face referire la funcțiile interne.

3. *regresivă (regression testing)* - se verifică dacă s-a modificat neașteptat comportamentul aplicației în urma implementării unor noi cerințe/schimbări.

4. *negativă (negative testing)* - se solicită aplicația, producând deliberat cazuri complicate, neobișnuite sau particulare pentru a forța apariția erorilor.

5. *de solicitare (stress testing)* - se determină capacitățile absolute ale aplicației și ale infrastructurii pe care este implementată; cu ajutorul acestui tip de test se dezvăluie caracteristicile de performanță ale unui sistem menținut în condiții de încărcare totale, adică sunt pornite și rulate toate serviciile care în mod normal ar fi fost rulate separat în timp și independent.

6. *de performanță (performance testing)* - în urma acestui tip de testare se verifică dacă performanța aplicației este adecvată pentru cerințele stabilite, în termeni de viteză de acces, resurse de sistem utilizate și procesarea cererilor de acces.

7. *de încărcare (load testing)* - se determină punctele slabe ale aplicației și dacă sunt necesare îmbunătățiri ale infrastructurii hardware sau software prin măsurarea caracteristicilor de performanță și scalabilitate a principalelor componente ale aplicației web; de regulă aceasta se realizează prin creșterea numărului de sesiuni utilizator sau a conexiunilor TCP/IP.

### Limitele testării automate

Sunt multe lucruri pe care uneltele de testare automată nu le pot face și este important să se cunoască aceste limitări pentru a alege calea cea mai potrivită. Un sistem de testare automată nu poate spune când ceva "arată bine" pe ecran sau când o poză sau fereastră nu este bine încadrată. De asemenea un test automat nu poate decide dacă logica programului are lipsuri funcționale decât în măsura în care au fost definite complet cerințele aplicației respective. Unele teste, mai ales pentru aplicații mici și pentru aplicații care se concentrează mai ales pe grafică și nu pe business logic, sunt mult mai ușor realizabile de către operatorul uman decât de către un computer. De aceea trebuie alcătuit un plan bine definit al procedurii de testare, când, unde și în ce condiții este fiabilă introducerea automatizării.

## Tipuri de unelte pentru testarea automată

Sistemele de testare automată pot include unelte de genul :

- GUI , prin folosirea metodei "înregistrare/redare"
- analizoare de cod - analizează complexitatea codului scris, respectarea unor standarde de scriere a codului
  - analizoare de memorie - detectează depășirea memoriei alocate, suprascrieri în zone nealocate și zone rămase nealocate
  - testare de solicitare/performanță - pentru testarea aplicațiilor web și client/server în diferite scenarii de solicitare
  - testare servere web - verifică validitatea și integritatea link-urilor, a codului html, programe client-side și server-side, securitatea transmiterii datelor
    - alte unelte - pentru managementul documentației, raportării erorilor, configurației, etc.

## Comparație între testarea manuală și automată

Testarea manuală și testarea automată sunt mai degrabă două procese diferite, decât două căi diferite de a executa același proces: dinamica lor este diferită precum și modul de a scoate în evidență erorile.

Testarea manuală este mai folositoare în situațiile în care este nevoie urgent de rezultatele unor tipuri de teste specifice și limitate, când se dorește ca timpul de feedback să fie foarte scurt, iar costurile să fie relativ mici. Cum îmbunătățirea calității produsului implică costuri adiționale pentru găsirea erorilor și gestionarea acestora până la repararea lor definitivă, testarea manuală s-a dovedit a fi în timp extrem de costisitoare. Testarea manuală pe scară largă presupune alocarea de resurse hardware și umane destul de mari, iar riscul să apară erori este amplificat de factorul uman.

Testarea automată necesită un efort inițial mai mare pentru planificarea, organizarea și producerea testului, criteriul principal în obținerea de rezultate bune fiind planificarea atentă, amănunțită și precisă a acestuia.

O alternativă interesantă este așa-numita "testare parțială". Această soluție este combinație de jumătate testare manuală și jumătate testare automată, aceasta din urmă fiind folosită numai acolo unde se pot obține beneficii maxime.

Testarea automată se dorește a fi soluția ideală pentru reducerea timpului de dezvoltare și a costurilor. O echipă de testeri poate să pornească unelte de testare automată, să le lase să ruleze și în final să colecteze și să analizeze rezultatele. Timpul este astfel mai bine organizat și poate fi petrecut pentru izolarea și raportarea erorilor.

În zilele noastre sunt multe unelte pe piață care pot ajuta la planificarea, execuția și crearea de rapoarte în activitatea de testare. Majoritatea acestor unelte necesită cunoștințe de specialitate pentru a le implementa și utiliza corespunzător. De asemenea, este bine de știut, că suitele profesionale de unelte specializate în asigurarea calității oferă întotdeauna un produs de sine stătător care preia partea de management de proiect și pe cea de raportare a erorilor. Un asemenea produs este de exemplu **TestDirector** de la firma Mercury Interactive – market leader în materie de produse pentru asigurarea calității. Acest produs poate fi corelat cu diverse sisteme de control al versiunii folosite în implementare și poate fi configurat în așa fel, încât rapoartele automate create de uneltele de testare și/sau monitorizare a aplicațiilor să fie preluate automat în sistemul, astfel încât citirea și interpretarea manuală a rapoartelor de testare nu mai este necesară. Erorile sunt parcurse și analizate de către **TestDirector**, care generează pe baza rapoartelor de testare email-uri, sms sau alte modalități de atenționare a echipei de implementare.

Este necesar de menționat că un sistem ca **TestDirector** poate fi folosit pentru partea de management a proiectului și în combinație cu testarea manuală, caz în care informațiile despre ceea ce trebuie testat se formalizează drept scenarii de testare, iar rezultatele testului manual trebuiesc introduse de către tester manual în sistem.

Soluțiile elegante pentru testarea sistemelor sofisticate sunt adesea limitate numai de imaginația tester-ilor.

Câteva dintre avantajele utilizării testării automate sunt:

- a) *avantaje privind eficiența și costurile*
  - prevenirea erorilor prin abordarea structurată a procesului de dezvoltare a proiectului
  - detecția erorilor care au ajuns până în faza de producție (prin teste de regresie automată)
  - reutilizarea informației acumulate (condiții de test, scenarii)
  - execuția automată a testelor de performanță în fazele de început ale proiectului poate evita eforturile de re-design în fazele ulterioare
  - odată ce scenariile de testare automată sunt implementate, o parte din personal poate fi redirecționat către alte necesități
- b) *avantaje privind economia de timp*
  - analiză rapidă și exactă în cazul schimbării parametrilor sistemului
  - durată scurtă a ciclurilor de testare
  - estimări mai exacte pentru procesul de planificare a testului
  - posibilitatea efectuării mai multor teste (scenariile de testare pot fi rulate și după orele de program economisind astfel timp)
    - generarea rapidă a condițiilor de testare
- c) *avantaje privind calitatea*
  - o mai bună înțelegere a scopului testării
  - o acoperire mai mare a elementelor de testat
  - rezultate mai consistente datorită repetabilității testelor
  - compararea automată a rezultatelor

### **Procesul testării automate**

Majoritatea uneltelor de testare automată sunt compatibile cu entitățile software ce intervin pe traseul de la clienți la furnizorul de aplicații.

Procesul de testare automată presupune un efort de management deosebit. Acest proces începe încă din faza de analiză a aplicației și continuă în toate etapele de dezvoltare. În diagrama următoare se pot observa etapele procesului, ordinea și frecvența acestora, precum și locul central pe care îl ocupă managementul defectelor și serviciile. Un factor important este menținerea centrală a comunicării între etape pentru managementul erorilor.



Fig.1 Procesul de testare automată

Testarea automată nu va putea înlocui în întregime testarea manuală și nici nu trebuie. Tester-ii pot să observe cum un utilizator poate interacționa cu produsul, în timp ce un sistem de testare automată nu poate întotdeauna să prevadă aceste acțiuni sau să găsească cea mai bună cale de a le testa. Dacă sunt bine folosite, programele de testare automată măresc considerabil productivitatea QA, economisesc costuri, măresc semnificativ consistența și calitatea produsului și ajută la optimizarea și accelerarea procesului de dezvoltare al unei aplicații. Deja în țările cu tradiție în dezvoltarea de software există cerința ca toate produsele software din sectorul militar, medical, guvernamental și financiar să fie testate cu unul din sistemele recunoscute de testare automată, iar rapoartele automate asupra felului cum a decurs testarea constituie baza acceptării unei aplicații de către client.