

Scrierea codului de programare este doar o parte a profesiei de programator. Asigurarea funcționării corecte a aplicației este la fel de importantă ca și scrierea codului în sine. Prin urmare, activitatea de programare este întotdeauna însoțită de testare detaliată. Pe lângă testare, o etapă la fel de importantă a dezvoltării produselor software este etapa de mentenanță. În rândurile următoare vor fi prezentate postulatele de bază ale testării și mentenanței software..

Ce este testarea software?

Testarea software este un termen foarte larg, care include mai multe operațiuni diferite. Mai simplu spus, testarea software implică:

- verificarea aplicației, pentru a stabili că aceasta nu are erori software,
- determinarea gradului de satisfacție a specificației software care a fost creată pe baza cerințelor clientului,
- verificarea utilizabilității produsului software din punct de vedere al utilizatorului.

Acestea sunt doar sarcinile de bază și cele mai importante ale testării software și, în funcție de tipul de produs software, testarea poate include multe alte acțiuni. Testarea confirmă corectitudinea, comportamentul funcțional și capacitatea de utilizare a aplicației înainte ca aceasta să fie livrată utilizatorilor finali.

De ce este importantă testarea?

Testarea software este un proces foarte important de dezvoltare software. Dacă se știe că mai exigentă în ceea ce privește timpul și, prin urmare, cea mai costisitoare etapă în dezvoltarea software-ului este mentenanța, se poate concluziona cu ușurință că testarea

detaliată poate economisi mult timp și bani.

Testarea software poate salva uneori vieți umane. Astfel, în 1994, o eroare software pe un avion Airbus A300 al companiei aeriene chineze a dus la pierderea a 264 de vieți.

Desigur, consecințele erorilor software la crearea aplicațiilor de afaceri nu sunt, în general, atât de grave, dar cu siguranță pot face diferența între un sistem funcțional și nefuncțional și, în cele din urmă, pot provoca pierderi de timp, bani sau clienți. Testarea oferă astfel următoarele avantaje decisive:

- feedback foarte rapid cu privire la defecțiunile aplicațiilor,
- detectarea precoce a defecțiunilor (defecțiunile sunt detectate în timpul dezvoltării),
- introducerea mai sigură a noilor funcționalități și îmbunătățiri, ceea ce duce la șanse reduse de [regresie](#),
- viteză stabilă de dezvoltare, care duce în cele din urmă la reducerea costurilor.

Tipuri de teste software

Testarea software este un termen care se referă la diferite operațiuni pentru a realiza funcționarea corectă a unui program. Prin urmare, există o serie de criterii diferite pe baza cărora se poate face distincția între tipurile de teste software:

- conform metodei de efectuare a testelor,
- conform modului în care este observat sistemul complet,
- conform nivelului sistemului testat,
- conform dimensiunii testului etc.

Clasificarea testării conform metodei de efectuare a testelor

Clasificarea de bază ia în considerare modul în care se face testarea, așadar, există:

- testare manuală,
- testare automată.

Testarea manuală presupune testarea în timpul căreia nu este utilizat niciun instrument de automatizare, adică un program special conceput pentru testare. Cu testarea manuală, persoana responsabilă cu testarea își asumă rolul utilizatorului final și astfel examinează în mod independent funcționarea sistemului. Uneori, acest tip de testare este mai mult decât acceptabil, mai ales în situațiile în care mediul de utilizator și caracteristicile sale vizuale sunt testate. Cu toate acestea, în unele situații, testarea manuală poate fi consumatoare de timp, chiar aproape imposibilă, și se recurge la utilizarea instrumentelor de testare automată.

Testarea automată presupune efectuarea testelor folosind un instrument special, adică un program scris pentru a testa alte programe. Acest tip de testare elimină necesitatea de a efectua manual acțiuni asupra aplicației testate. Testarea automată se realizează folosind logica programului, special formulată pentru testare. Un tester poate scrie singur logica pentru testarea automată sau poate folosi unul dintre instrumentele gata făcute, concepute pentru o astfel de operațiune.

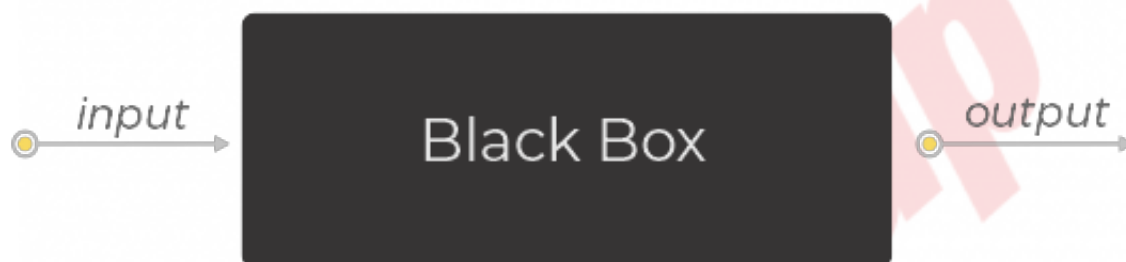
Clasificarea testării conform modului în care este observat întregul sistem

Pe lângă această clasificare de bază în funcție de modul în care se efectuează testarea, o altă clasificare de testare software se referă și la modul în care este observat întregul sistem testat. Pe baza unei astfel de clasificări există:

- testarea Black Box,
- testarea White Box.

Testarea Black Box este o tehnică care implică testarea fără nicio cunoaștere a structurii interne a sistemului. Persoana care se ocupă cu

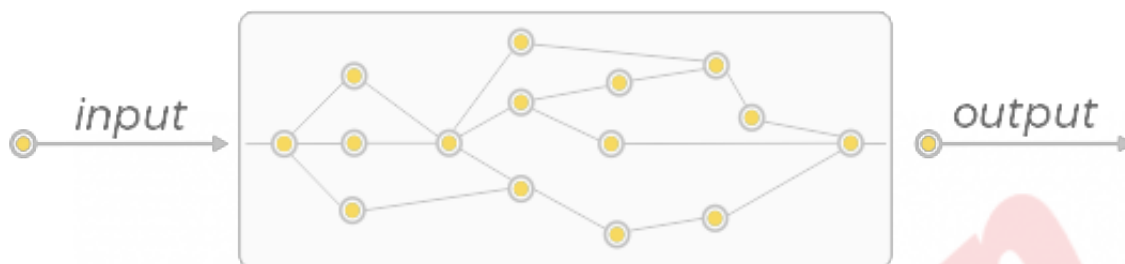
acest tip de testare nu trebuie să știe cum și cu ce tehnologii a fost scris programul. În plus, persoana care realizează testarea nici măcar nu trebuie să fie programator. Astfel, testarea Black Box se realizează prin compararea rezultatelor obținute și așteptate (imaginea 9.1.).



Imaginea 9.1. Vizualizarea Black Box asupra sistemului

Imaginea 9.1. ilustrează un program sub forma unei casete negre (*black box* în engleză). Realizarea internă a programului nu este cunoscută și singurul lucru care contează este corectitudinea ieșirii pe baza diferitor date de intrare.

Testarea White Box, spre deosebire de abordarea tocmai ilustrată, implică cunoașterea completă a logicii interne a sistemului. Persoana care se ocupă de testare într-o astfel de situație, are acces la codul sursă și este responsabilă pentru a determina dacă toate operațiunile interne ale sistemului funcționează conform specificației predeterminate (imaginea 9.2.).



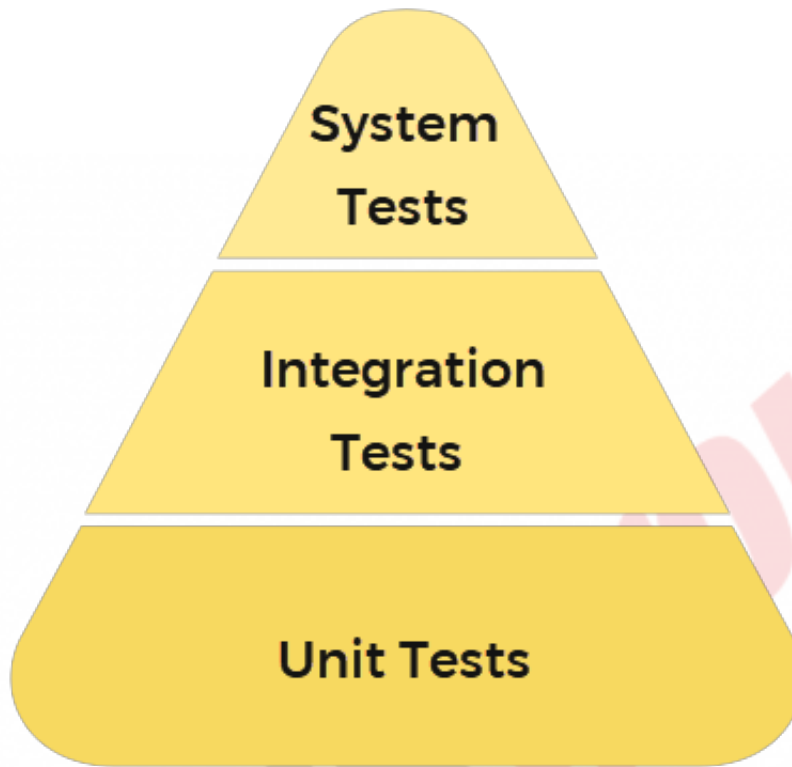
Imaginea 9.2. Vizualizarea White Box asupra sistemului

Clasificarea testării conform nivelului sistemului testat

Testarea software poate fi, de asemenea, clasificată în funcție de nivelul sistemului testat. Astfel, există următoarele teste:

- teste unitare (*unit tests* în engleză),
- teste de integrare (*integration tests* în engleză),
- teste de sistem (*system tests* în engleză).

Imaginea 9.3. ilustrează testele tocmai menționate.



Imaginea 9.3. Piramida testărilor (testări pe baza nivelului sistemului cuprins de testare)

Imaginea 9.3. ilustrează piramida de testare, care definește clasificarea în 3 grupuri de bază de teste conform nivelului sistemului cuprins de testare.

În partea de jos a piramidei se află testele unitare (***Unit Tests*** în engleză). După cum ne sugerează și numele, astfel de teste sunt concentrate pe funcționalitățile individuale care compun aplicația completă. Testele unitare testează cea mai mică unitate de cod posibilă. Cel mai adesea, acestea sunt metode de obiecte, astfel încât testele unitare determină dacă elementele individuale ale aplicației își desfășoară activitatea fără erori.

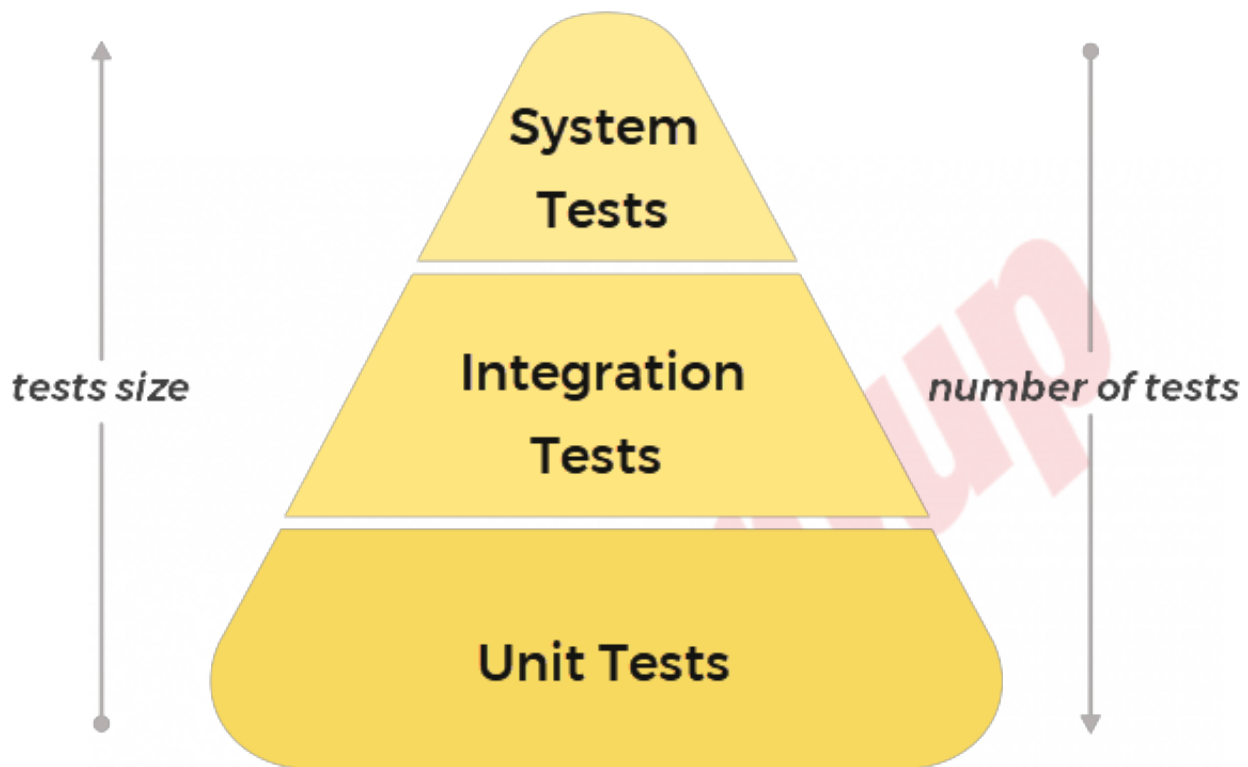
Nivelul central al piramidei de testare este ocupat de testele de

integrare (**Integration Tests** în engleză). Acestea sunt destinate testării interacțiunii dintre diferite componente mai mici ale unui sistem software. Și anume, sistemele software serioase constau în general dintr-un număr mare de componente individuale mai mici, cu cooperarea cărora este construit un program funcțional. Astfel, după testele unitare ale tuturor componentelor individuale, interacțiunea lor reciprocă este testată folosind teste de integrare.

În vârful piramidei se găsesc testele de sistem (**System Tests** în engleză). Acestea au ca scop testarea întregului sistem în ansamblu. Și anume, în condiții ideale, toate erorile interne sunt detectate prin teste unitare și de integrare, astfel încât până la urmă rămâne de verificat comportamentul întregului sistem, mai ales din punct de vedere al utilizatorului propriu-zis. Accentul acestui tip de teste se pune pe cerințe nefuncționale, cum ar fi viteza, fiabilitatea, robustețea, securitatea și altele.

Clasificarea testelor conform dimensiunii acestora

Piramida de testare din imaginea 9.3. poate servi pentru o altă clasificare de teste. De data aceasta vorbim despre clasificarea în funcție de dimensiunea testului individual (imaginea 9.4.).



Imaginea 9.4. Clasificarea testelor în funcție de număr și dimensiune

Din imaginea 9.4. puteți vedea că numărul de teste unitare este cel mai mare, pentru că și testele sunt cele mai mici. Pe de altă parte, testele de sistem sunt cele mai cuprinzătoare și, prin urmare, sunt cele mai puține. Conform unei astfel de logici, testele pot fi clasificate în următoarele grupuri:

- **teste mici** – teste unitare care verifică cele mai mici unități ale unei aplicații (metode, clase sau componente),
- **teste medii** – teste de integrare, care verifică interacțiunea dintre două sau mai multe componente ale aplicației,
- **teste mari** – teste de sistem care verifică funcționarea aplicației în ansamblu, în principal din punctul de vedere al utilizatorului.

Testele mici sunt cel mai ușor de scris și de aceea ar trebui să existe cele mai multe astfel de teste atunci când testați o aplicație. Pe măsură ce complexitatea testelor crește, numărul acestora scade, astfel încât atunci când testați o aplicație, întotdeauna trebuie să creați mult mai multe teste unitare decât este cazul testelor de integrare sau de sistem. Raportul poate varia, dar deseori se recomandă ca o împărțire a testelor să arate astfel: 70% teste mici, 20% teste medii și 10% teste mari.

Alte tipuri de testare

Pe lângă clasificările de testare software prezentate până acum, există mult mai multe clasificări și diferite tipuri de teste software. Unele dintre cele mai semnificative vor fi enumerate mai jos:

- **testul de acceptabilitate** - test care este orientat pe verificarea conformității sistemului cu cerințele clientului și dacă sistemul îndeplinește cerințele clientului, sistemul este acceptabil pentru livrare;
- **testul de instalare** - un test care verifică procesul de instalare sau actualizare a programelor pe un dispozitiv de producție și, de asemenea, astfel de teste pot include procesul de actualizare a aplicațiilor deja instalate;
- **testarea alpha** - testare cu un grup mic de utilizatori înainte ca aplicația să fie publicată sau livrată utilizatorilor finali;
- **testarea beta** - testare cu un grup restrâns de utilizatori finali în condiții reale de utilizare;
- **testarea de regresie** - testare care se efectuează, de obicei, după o anumită modificare a programului, pentru a se asigura că sistemul funcționează fără erori chiar și după modificare;
- **testarea performanței** - teste orientate pe îndeplinirea cerințelor cerute când vine vorba de performanța aplicației - timpul de răspuns, timpul maxim de așteptare pentru primirea unui răspuns sau similar.

Mentenanța software

După ce testarea software este finalizată, se decide dacă un anumit produs software sau o parte din funcționalitatea acestuia este gata de livrare către utilizatorii finali. Livrarea unui produs software nu înseamnă că munca de dezvoltare a acestuia este încheiată. După livrare, se intră într-o etapă specială de dezvoltare numită mentenanță software.

Mentenanța software este o etapă a dezvoltării software care presupune actualizarea produsului software astfel încât acesta să răspundă în mod continuu așteptărilor utilizatorilor. Este vorba de o etapă care începe abia după publicarea inițială, adică după livrarea produsului software. Corectarea erorilor, îmbunătățirea performanței, adăugarea funcționalităților noi, sunt doar câteva exemple de operațiuni care pot fi încadrate în mentenanța software.

Operațiunile efectuate în timpul mentenanței software pot fi clasificate în patru grupuri de bază:

- **mentenanța corectivă** – tip de mentenanță care presupune intervenții asupra produsului software atunci când ceva nu funcționează conform intenției, adică în situații de apariție a defecțiunilor și erorilor (bug-urilor);
- **mentenanța preventivă** - tip de mentenanță care tinde să se asigure că produsul software funcționează corect cât mai mult timp posibil, iar pentru a asigura acest lucru, este foarte important să se implementeze frecvent modificări, actualizări și adaptări preventive, care să prevină apariția problemelor în viitor;
- **mentenanța adaptivă** – tip de mentenanță pentru care apare necesitatea în cazul unor modificări ale mediului în care funcționează produsul software; acestea pot fi modificări în cadrul sistemului de operare, modificări ale unora dintre tehnologiile utilizate pentru dezvoltare sau funcționalități utilizate, dar și modificări care se referă la aspecte legale, politici și licențiere, așadar, în astfel de situații trebuie să se

facă modificări corespunzătoare pentru a adapta sistemul software la noile circumstanțe;

- **mentenanța funcțională** – tip de mentenanță care presupune îmbunătățirea produsului, adăugarea funcționalităților noi, îmbunătățirea sau înlăturarea celor existente dacă nu există nevoie reală de ele; o astfel de mentenanță tinde întotdeauna să aibă un produs software care să răspundă cel mai bine nevoilor utilizatorului.

Din rândurile anterioare se poate concluziona că mentenanța software este un termen foarte larg și că modul în care se realizează depinde de tipul de mentenanță. Mentenanța software implică, în general, implementarea tuturor etapelor de dezvoltare ale software-ului, de la analiză, prin specificare și dezvoltare, până la testare și livrare. Acest lucru este valabil mai ales atunci când implementați funcționalități noi, când este utilizată metodologia Agile.

Mentenanța software este aproape întotdeauna cea mai lungă și mai costisitoare etapă a dezvoltării unui produs software. Uneori poate reprezenta mai mult de două treimi din costul unui produs software complet. Totuși, există diverși factori care afectează costul mentenanței. Complet logic, costurile de mentenanță vor fi mai mari dacă vorbim de produse software mai vechi sau produse software implementate cu tehnologii care sunt mai rar folosite astăzi și invers.

Ca regulă generală, software-ul care este bine conceput și implementat în viitor face mentenanța mult mai ușoară. Prin urmare, mentenanța este luată în considerare în primele etape de dezvoltare. În acest fel, costul viitor de mentenanță este influențat semnificativ.