Cofinanțat de Uniunea Europeană

Programul Educație și Ocupare 2021 - 2027
Proiect cofinanțat din Fondul Social European+
Titlu Apel: "Ține pasul" - Regiuni mai puțin dezvoltate
Cod apel: PEO/78/PEO_P9/OP4/ESO4.7/PEO_A36 / Obiectiv specific: ESO4.7
Titlu Proiect: ATAC - angajați trainuiți, angajatori competitivi, cod MySmis: 311210

ANALIST TESTARE SOFTWARE

SUPORT DE CURS







CEFP MM

ANOFM AJOFM MARAMUREȘ

ANOFM AJOFM Iași

blue CONSULTING

2025

Analist testare software



Salcianu, Monica

Salcianu, Monica

2025

Internal

CONTINUT

TESTARE DE CONFORMITATE

Ce este testarea conformității?

Tipuri de teste de conformitate

De ce avem nevoie de testare de conformitate?

Ce trebuie testat?

Când și Cum să Performați Testare de conformitate?

Testare de conformitate: opinii critice

Elemente esențiale ale testării de conformitate

Cele mai bune practici pentru testarea conformitatii

Provocări de testare a conformității

Rolul testerilor QA în conformitate

Rolul testării de conformitate

Sistemul de management al conformității

Testarea conformității vs conformitatea

Concluzie

AUTOMATIZARE IN TESTARE

Ce este automatizarea testelor?

Care sunt beneficiile testării automate?

Care sunt dezavantajele testării automatizării?

Cum decideți ce test să automatizați?

Teste care sunt de obicei automatizate

Cele 6 principii ale automatizării testelor

Testați abordări automate

Metodologii de automatizare a testelor

Instrumente automate de testare

Testarea pieței de automatizare 2025

Concluzie

Testare de Conformitate

Ce este testarea de conformitate?

Testarea conformității este ca și cum ai fi aplicatorul regulilor într-un joc de etichetă. Știi, cel care se asigură că toată lumea joacă corect și nu trișează. Dar, în loc de șmecherii pe terenul de joacă, testarea conformității asigură că software-ul funcționează conform regulilor stabilite de organismele de reglementare și standardele din industrie.

Testarea de conformitate, în forma sa cea mai de bază, este testarea software-ului pentru a se asigura că îndeplinește cerințele specifice de reglementare și industrie. Aceste reglementări abordează diverse subiecte, inclusiv securitatea, accesibilitatea și confidențialitatea datelor. Această testare funcționează ca o listă de verificare virtuală, asigurând că software-ul îndeplinește toate standardele necesare pentru a coexista cu alții în lumea computerelor. „Uau , asta sună plictisitor", ați putea crede.

Luați în considerare testele de conformitate super-eroul ascuns al lumii software, garantând că software-ul nu creează haos și ruină. Deci, într-un fel, testerii de conformitate sunt Batman sau Wonder Woman din lumea tehnologiei, ținând totul sub control și asigurând funcțiile software în limitele legale.

Tipuri de teste de conformitate

În ceea ce privește testele de conformitate, există câteva tipuri diferite de reținut. Fiecare tip are puteri și abilități unice, ca o unealtă diferită în centura de utilități a unui supererou. Așadar, să ne aprofundăm în diferitele tipuri de testare a conformității și să vedem despre ce este vorba:

- **Testare de accesibilitate** : Această formă de testare funcționează similar cu rampa care permite supereroilor să intre în clădiri și să salveze ziua. Testarea accesibilității garantează că software-ul este utilizabil de către persoanele cu deficiențe, cum ar fi nevăzătorii sau cei cu mobilitate redusă.
- **Testare de securitate** : Testarea de securitate este analogă cu scutul care protejează supereroii de daune. Această testare asigură că software-ul este sigur și că oricine nu îl poate ataca sau exploata pur și simplu.

- **Testarea confidențialității datelor** : această formă de testare este similară cu mantia de invizibilitate pe care supereroii o folosesc pentru a ascunde cine sunt. Testarea confidențialității datelor garantează că software-ul respectă regulile de protecție a datelor și că datele utilizatorului sunt sigure și securizate. Dacă aplicația dvs. implică colectarea datelor despre utilizatori prin sondaje, este esențial să utilizați **un software de sondaj conform GDPR** pentru a vă asigura că toate datele cu caracter personal sunt gestionate în conformitate cu standardele Regulamentului general privind protecția datelor.
- **Testarea performanței** : testarea performanței seamănă cu acțiunile fulgerătoare ale fulgerului supererou. Această testare garantează că software-ul funcționează eficient în diferite circumstanțe, cum ar fi utilizarea excesivă sau traficul excesiv.
- **Testare de conformitate cu reglementările** : acest tip de testare este similar cu regulile pe care supereroii trebuie să le respecte. Testele de conformitate cu reglementările garantează că software-ul respectă standardele legale specifice stabilite de organizațiile de reglementare.

Testerii de software pot combina multe metode de testare pentru a garanta că software-ul funcționează și conform tuturor legislației și standardelor aplicabile. Aceasta este similară cu o centură de utilitate de supererou.

De ce avem nevoie de teste de conformitate?

Testarea de conformitate poate să nu fie aspectul cel mai fascinant al testării software-ului, dar este esențială. Această testare este similară cu povestea de origine a unui super-erou: s-ar putea să nu aibă entuziasm, dar oferă bazele pentru tot ce urmează. Deci, să vedem de ce trebuie să testăm conformitatea:

- **Conformitate legală** : această testare garantează că software-ul respectă anumite legi și reglementări, cum ar fi legislația privind protecția datelor și standardele de accesibilitate.
- **Protejarea datelor utilizatorilor** : De asemenea, protejează datele utilizatorilor împotriva furtului sau exploatării de către părți fără scrupule. Și drepturile de confidențialitate ale utilizatorilor, inclusiv **dreptul de a li se șterge datele personale** , sunt respectate și susținute în conformitate cu reglementările relevante precum GDPR.
- **Evitarea penalităților și a amenzilor** : această testare ajută companiile să evite penalități uriașe și acțiuni legale din cauza nerespectării cerințelor de reglementare.

- **Crearea încrederii** : testarea încurajează încrederea utilizatorilor, demonstrând că o firmă apreciază confidențialitatea și securitatea datelor și se angajează să facă lucrurile corect.
- **Îmbunătățirea calității software-ului** : Această testare asigură că software-ul este de înaltă calitate și se conformează standardelor din industrie, făcându-l mai fiabil și mai puțin probabil să creeze probleme în viitor.

Testele de conformitate sunt esențiale pentru testarea software-ului, permițând programului să iasă în evidență și să funcționeze optim. Fără el, programul ar fi ca un super-erou fără pelerină.

Ce avem nevoie pentru a testa?

În ceea ce privește testele de conformitate, anumite zone ale software-ului trebuie testate pentru a asigura conformitatea cu standardele de reglementare și cele mai bune practici din industrie. Gândiți-vă la aceste zone ca la diferitele superputeri pe care trebuie să le aibă un supererou pentru a salva ziua. Să aruncăm o privire la ceea ce trebuie testat:

- **Funcționalitate** : testarea funcționalității asigură că software-ul face ceea ce trebuie și nu are erori sau erori care ar putea cauza probleme.
- **Utilizabilitate** : Această testare asigură că software-ul este ușor de utilizat și accesibil persoanelor cu nevoi și abilități diferite.
- **Securitate** : Această testare asigură că software-ul este securizat și nu poate fi piratat sau exploatat cu ușurință de către actori răi.
- **Performanță** : această testare asigură că software-ul funcționează bine în diferite condiții, cum ar fi trafic intens sau utilizare ridicată.
- **Compatibilitate** : testarea compatibilității asigură că software-ul funcționează perfect cu diferite sisteme de operare, browsere și dispozitive.
- **Confidențialitatea datelor** : această testare asigură că datele utilizatorului sunt păstrate în siguranță și securizate și respectă legile și reglementările privind protecția datelor.
- **Accesibilitate** : Această testare asigură că software-ul este accesibil persoanelor cu dizabilități și respectă standardele de accesibilitate.

Fiecare zonă este ca o superputere diferită pe care software-ul trebuie să o aibă pentru a fi conformă și a îndeplini standardele industriei. Testând temeinic fiecare

zonă, testerii de software se pot asigura că software-ul este funcțional, sigur, accesibil și în conformitate cu toate reglementările și standardele relevante.

Când și cum se efectuează testarea de conformitate?

Înainte de a începe testarea conformității, este esențial să țineți cont de toate standardele pe care trebuie să le respecte aplicația dvs. De exemplu, ar putea fi necesar să urmați standarde de accesibilitate, standarde de confidențialitate, cum ar fi UE, și ar putea exista și alte standarde care sunt stabilite pentru industria în care vă ocupați sau zona în care vă ocupați. După ce cunoașteți scopul testării dvs. de conformitate, mai jos sunt pașii pe care îi puteți urma:

1. **Începeți devreme** : este esențial să începeți testarea conformității la începutul procesului de dezvoltare. Acest lucru poate ajuta la identificarea oricăror probleme de la început și la prevenirea problemelor mai mari.

2. **Creați cazuri de testare** : creați scenarii de testare specifice pentru toate standardele care trebuie respectate de aplicația dvs. Exemple de criterii de conformitate includ compatibilitatea cititorului de ecran, navigarea cu tastatura, confidențialitatea și securitatea datelor și respectarea contrastului de culoare.

3. **Efectuați teste** : Executați testele urmând cazurile și metodele de testare stabilite, asigurând o acoperire cuprinzătoare a zonelor de conformitate.

4. **Analizați rezultatele** : analizați rezultatele testelor pentru a detecta orice defecțiuni sau încălcări ale conformității. Analizați datele pentru a obține o perspectivă asupra domeniilor care necesită îmbunătățiri sau îmbunătățiri.

5. **Raportați probleme** : documentați și raportați orice probleme de conformitate sau încălcări în timpul testării, oferind explicații clare, dovezi și pași corectivi sugerați.

6. **Verificați corecțiile** : verificați conformitatea și eficacitatea activităților de remediere prin retestarea regiunilor afectate pentru a confirma conformitatea și eficacitatea.

7. **Automatizați problemele recurente** : automatizarea procesului de testare poate ajuta la detectarea și gestionarea problemelor recurente de conformitate mai eficient. Prima dată, testele sunt executate manual, dar

pentru a vă asigura că fiecare lansare respectă aceste standarde, poate fi automatizată.

8. **Efectuați audituri regulate** : Auditurile regulate vă pot ajuta să vă asigurați că testele de conformitate sunt actualizate și că software-ul respectă întotdeauna cele mai recente reglementări și standarde.

Urmând aceste sfaturi, testarea conformității poate fi efectuată într-un mod care să asigure că software-ul este conform și îndeplinește standardele din industrie. Testarea conformității este o parte esențială a procesului care poate ajuta la prevenirea unor probleme mai semnificative.

Testare de conformitate: opinii critice

În timp ce testarea de conformitate este o parte importantă a testării software-ului, nu este lipsită de critici. Să aruncăm o privire la câteva puncte de vedere critice ale testării de conformitate:

- **Conformitate peste inovare** : Unii critici susțin că testele de conformitate pot înăbuși inovația și creativitatea în dezvoltarea de software. Concentrându-se prea mult pe cerințele de conformitate, dezvoltatorii de software pot fi mai puțin înclinați să-și asume riscuri și să exploreze idei noi.
- **Domeniu limitat de aplicare** : Unii critici susțin că testele de conformitate pot fi limitate, concentrându-se doar pe reglementări sau standarde specifice. Acest lucru poate crea puncte oarbe în testarea software-ului, lăsând alte zone vulnerabile la probleme.
- **Resurse intensive** : testarea conformității poate necesita resurse, necesitând timp, bani și personal. Unele organizații pot avea dificultăți să dedice suficiente resurse pentru testarea conformității, ceea ce le poate expune riscului de neconformitate.
- **Fals sentiment de securitate** : testele de conformitate pot crea un fals sentiment de securitate, făcând organizațiile să creadă că sunt complet conforme și ferite de orice problemă. Cu toate acestea, această formă de testare nu este sigură și este posibil să nu detecteze orice problemă potențială sau vulnerabilitate.
- **Lipsa flexibilității** : Această testare poate fi rigidă și inflexibilă, concentrându-se doar pe cerințe specifice și nepermițând variații sau schimbări în diferite situații.

Este importantă luarea în considerare a acestor puncte de vedere critice despre testarea conformității și găsirea unor modalități de abordare a acestora în procesul de testare a software-ului. Deși testarea conformității este necesară, aceasta nu ar trebui să vină în detrimentul inovației sau al altor aspecte esențiale ale dezvoltării software.

Testarea de conformitate verifică dacă produsele sau serviciile îndeplinesc standarde specifice. Poate fi realizat de către producător, un utilizator sau o organizație independentă care ar putea fi creat standardul.

Când software-ul trece testul și primește certificarea, acesta poate fi anunțat ca fiind conform cu standardul. Această certificare asigură calitatea pentru utilizatorul final. Testarea conformității este importantă deoarece identifică potențialele probleme din timp, asigurându-se că software-ul este sigur, funcțional și accesibil pentru toți utilizatorii.

Rolul testării conformității într-un sistem de management al conformității

Testarea de conformitate verifică dacă un sistem respectă atât politicile interne, cât și reglementările externe, oferind dovada că aceste cerințe sunt îndeplinite în mod constant. Un sistem de management al conformității (CMS) vă ajută să înțelegeți cerințele de conformitate, să formeze angajații și să vă pregătească pentru audituri. Cu un CMS, orice întrerupere este rezolvată și remediată rapid, făcând conformarea o parte a proceselor de zi cu zi.

Diferite tipuri de teste de conformitate pe care trebuie să le cunoașteți

Există diferite tipuri de procese de testare a conformității implicate, începând de la testarea de penetrare până la audituri regulate de securitate. Iată câteva dintre ele:

Testarea de penetrare	Hackerii etici simulează atacuri cibernetice din viața reală asupra sistemului, software-ului, aplicațiilor sau rețelei dvs. în condiții controlate. Acest proces este crucial pentru a evalua cât de bine pot rezista măsurile dvs. de securitate unui atac.
Evaluare a riscurilor	Evaluarea riscurilor este o tehnică de evaluare strategică utilizată pentru a identifica și prioritiza riscurile potențiale pentru organizația sau proiectele dvs. Când identificați amenințările care ar putea afecta succesul proiectului dvs., evaluarea riscurilor ajută la dezvoltarea strategiilor de atenuare pentru a preveni ca aceste amenințări să devină probleme majore.

Audit de securitate	Un audit de securitate este o examinare amănunțită a apărării de securitate a informațiilor organizației dvs. Aceasta implică o revizuire aprofundată a sistemelor dvs. pentru a vă asigura că sunt protejate împotriva hackingului și a codului rău intenționat.
Managementul vulnerabilităților	Managementul vulnerabilităților este un proces continuu care permite organizației dvs. să detecteze, să evalueze, să raporteze, să gestioneze și să remedieze vulnerabilitățile de la punctele finale, sarcinile de lucru și rețelele.
Testare de securitate API	API-urile sunt conectorii care permit diferitelor părți ale sistemului dumneavoastră să comunice. Testarea de securitate API implică examinarea acestor puncte finale pentru securitate și fiabilitate, asigurându-se că respectă cele mai bune practici organizaționale.
Testarea securității aplicațiilor web	Aplicația dvs. web este poarta de acces către mediul digital al organizației dvs. Testarea securității aplicațiilor web evaluează metodic securitatea aplicațiilor dvs. web analizându-le în mod activ pentru punctele slabe, defecte tehnice și vulnerabilități.

Definiți obiectivele și domeniul de aplicare

Determinați ce sisteme, aplicații sau procese trebuie testate și ce standarde sau reglementări specifice trebuie să le respecte.

2. Creați biblioteca de cerințe

Indiferent dacă aveți un program mic de testare a conformității sau dacă începeți de la zero, următorul pas este să construiți o bibliotecă de cerințe.

Această bibliotecă va conține toate standardele și reglementările pe care trebuie să le respecte organizația dvs. Pentru a crea această bibliotecă, ar trebui să consultați un expert în produse sau în industrie care vă poate ajuta la identificarea cerințelor.

Această bibliotecă va identifica apoi controalele existente ale organizației dvs. și orice lacune care trebuie abordate pentru a atenua riscurile de conformitate.

3. Harta cerințele la funcțiile de afaceri

În continuare, fiecare cerință este legată de funcția de afaceri relevantă, iar proprietarii respectivi de afaceri colaborează pentru a defini riscurile de conformitate.

Asigurați-vă că validați aplicabilitatea fiecărei cerințe cu proprietarii de afaceri. Acest lucru ajută compania să înțeleagă fiecare risc de conformitate, de ce ar putea apărea și importanța fiecărei cerințe.

De exemplu, pentru a respecta GDPR, trebuie să vă asigurați că toate formularele de colectare a datelor clienților includ opțiuni clare de consimțământ.

1. **Efectuați o evaluare a riscurilor**

O evaluare a riscului de conformitate identifică, evaluează și ierarhizează în mod sistematic riscurile legale și de reglementare care ar putea dăuna organizației. Acest proces ajută la concentrarea resurselor pe cele mai semnificative pericole și zonele care nu au suficiente controale.

Pentru fiecare risc, evaluați riscul inerent, care este riscul încălcării unei cerințe fără niciun control, măsurând impactul și probabilitatea unei încălcări a reglementărilor.

Apoi, determinați eficacitatea controalelor care atenuează riscul. Cu aceste evaluări, puteți calcula riscul rezidual pentru fiecare cerință folosind o matrice.

2. **Creați o metodologie eficientă de testare a conformității**

În primul rând, specificați obiectivele dvs. Obiectivele sunt scopuri care trebuie îndeplinite în timp ce vă urmăriți obiectivele. Acestea pot fi la fel de simple ca demonstrarea conformității cu un standard sau construirea încrederii clienților în controalele dvs.

Iată cum efectuați testul:

- Determinați ce standarde sau linii directoare care se aplică circumstanței și contextului dvs. ar sta la baza standardului dvs. de testare.
- Elaborați un plan cuprinzător de testare care să prezinte obiectivele testului, strategiile și abordările de testare, instrumentele, programul și persoanele responsabile pentru efectuarea testului.
- Creați un set de teste concrete bazate pe aceste norme și reglementări pentru a include detalii despre ceea ce va fi testat și cum va îndeplini criteriile.
- Alegeți instrumentele și metodele potrivite prin care urmează să fie efectuate testele, care pot fi manuale sau complet automatizate.
- Efectuați testarea inițială conform strategiei dvs. și asigurați-vă că înregistrați tot ce observați.

- Rezultatele ar trebui comparate pentru a căuta modele care stau la baza amenințărilor, problemelor și vulnerabilităților. Luați în considerare aceste probleme în termeni de importanță relativă sau în ordinea mărimii efectelor lor.
- Ar trebui pregătit un raport care să cuprindă constatările, neconformitatea, riscurile și recomandările.
- Aceasta implică colaborarea cu alte părți pentru a implementa măsuri de atenuare, asigurându-se în același timp că aceștia sunt testați în mod adecvat.

6. Executați procedurile de testare

Acum că aveți metodologia pregătită, urmați planul de testare pentru a efectua procedurile. Aceasta include de obicei:

- **Testare de penetrare:** simulați atacurile cibernetice pentru a verifica puterea apărării dvs.
- **Testarea securității aplicațiilor web:** identificați vulnerabilitățile din aplicațiile dvs. web.
- **Testare de securitate API:** Evaluați securitatea API-urilor dvs.
- **Scanarea vulnerabilităților:** utilizați instrumente pentru a găsi și aborda vulnerabilitățile cunoscute.
- **Alte teste:** Efectuați orice teste suplimentare relevante în funcție de nevoile dvs.

Acești pași vă ajută să identificați și să abordați problemele de securitate, asigurându-vă că sistemele dumneavoastră sunt sigure și conforme.

7. Analizați și documentați constatările

După ce ați finalizat toate acele teste, următorul lucru pe care îl faceți este să evaluați și să raportați rezultatele pe care le-ați adunat din procesele de testare. Începeți prin a face timp să citiți rezultatele testelor pentru a nota eventualele defecțiuni când vine vorba de standardele de conformitate sau securitate.

- Documentați fiecare apariție foarte meticulos. Descrierea trebuie să conțină informații despre tipul problemei, consecințele sale potențiale pentru organizația dvs. și domeniile pe care le poate viza.
- Oferiți recomandări specifice despre modul în care fiecare problemă poate fi gestionată cel mai bine. Acesta ar trebui să conțină măsuri

specifice pentru a aborda problemele identificate și pentru a îmbunătăți conformitatea generală și securitatea.

- Efectuați remedierea și efectuați verificări periodice.
- Pașii următori sunt angajarea echipelor relevante pentru a aborda problemele identificate.

8. Implementați remedierea și monitorizarea continuă

Colaborați cu echipele relevante pentru a aborda problemele identificate. Aplicați modificările recomandate și testați-le cu atenție pentru a vă asigura eficacitatea. Acest proces ajută la rezolvarea problemelor și confirmă că remediile funcționează conform intenției. Acum, cum poți simplifica asta?

Aici intervine automatizarea GRC. Sprinto va fi aliatul tău suprem pentru testarea conformității.

Software-ul de bază pentru conformitatea securității evidențiază adesea doar sarcinile, dar nu se ocupă de execuție. Derularea și finalizarea programului necesită timp, efort și coordonare.

Sprinto merge mai departe, nu doar enumerând sarcini, ci și folosind automatizarea adaptivă pentru a organiza, solicita și capta în mod continuu dovezi. Acest lucru face procesul mai ușor și mai ușor de auditat .

Cazuri de utilizare populare ale testării de conformitate

Câteva exemple de teste de conformitate sunt:

- Proceduri de backup și recuperare
- Verificarea conformității cu păstrarea datelor
- Auditul artefactelor software, inclusiv licențele
- Testarea de stres a controalelor de securitate a rețelei
- Măsuri de criptare și protecție a datelor
- Protocoale de răspuns și management la incidente
- Respectarea politicilor de păstrare a datelor
- Controale și configurații de securitate a rețelei

- Practici de gestionare a corecțiilor software
- Evaluări ale conformității furnizorilor de la terți
- Drepturile de acces ale utilizatorilor și reglementările de securitate
- Schimbarea programului și procedurile de control

Ce trebuie să verificăm pentru testele de conformitate

Testarea de conformitate înseamnă examinarea unui produs pentru a se asigura că îndeplinește standarde specifice, indiferent de domeniu.

01

Verificați că produsul este complet dezvoltat și că toate funcțiile funcționează așa cum ar trebui.

02

Pregătiți documentația și manualele de utilizare pentru o înțelegere clară și o verificare a conformității.

03

Dacă există asistență și documentație online, asigurați-vă că este cea mai recentă versiune.

04

Confirmați că testarea funcțională și de integrare este efectuată și îndeplinește criteriile de ieșire.

05

Aveți o matrice de escaladare în vigoare, împreună cu contacte pentru dezvoltare, testare și echipe de management.

06

Asigurați-vă că toate licențele sunt actualizate.

Testare de conformitate: cele mai bune practici în 2024

În acest articol, vom vorbi despre testarea conformității, de ce este esențială și pașii pe care îi puteți folosi pentru a efectua această testare.

Ce trebuie să verificăm pentru testele de conformitate

Testarea de conformitate înseamnă examinarea unui produs pentru a se asigura că îndeplinește standarde specifice, indiferent de domeniu.

01

Verificați că produsul este complet dezvoltat și că toate funcțiile funcționează așa cum ar trebui.

02

Pregătiți documentația și manualele de utilizare pentru o înțelegere clară și o verificare a conformității.

03

Dacă există asistență și documentație online, asigurați-vă că este cea mai recentă versiune.

04

Confirmați că testarea funcțională și de integrare este efectuată și îndeplinește criteriile de ieșire.

05

Aveți o matrice de escaladare în vigoare, împreună cu contacte pentru dezvoltare, testare și echipe de management.

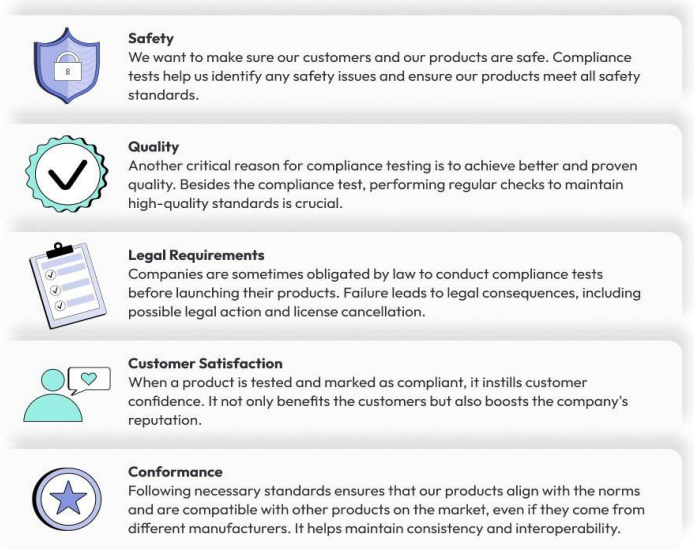
06

Asigurați-vă că toate licențele sunt actualizate.

De ce avem nevoie de teste de conformitate

Testarea de conformitate ajută la asigurarea faptului că o companie respectă standardele stabilite. Verifică dacă regulile pentru dezvoltarea software-ului sunt respectate corect, asigurând o lansare fără probleme a produsului. Această testare este crucială pentru evitarea amenzilor mari pentru neconformitate. De asemenea, ajută la menținerea standardelor de codare, facilitând gestionarea codului în viitor. În plus, testarea conformității ajută la identificarea și remedierea promptă a încălcărilor pentru a reduce riscurile de conformitate.

Există mai multe motive importante pentru efectuarea testelor de conformitate:



Elemente esențiale ale testării de conformitate

Iată câteva elemente esențiale ale testării de conformitate:

Reguli și pași de conformitate - Stabiliți reguli și pași robusti care arată promisiunea organizației de a urma și de a ghida angajații.

Programe de predare și cunoaștere - Asigurați-vă că toți angajații își înțeleg sarcinile prin implementarea unor programe detaliate de predare și conștientizare.

Vizionare și verificare - Urmăriți constant activitățile pentru a găsi orice încălcare a regulilor și pentru a vedea cât de bine funcționează regulile.

Controale interne și modul în care se desfășoară lucrurile - Configurați controale interne și metode operaționale pentru a consolida eforturile organizației de a respecta regulile, asigurând responsabilitatea la nivel general.

Sisteme de raportare - Creați modalități simple și rapide de a împărtăși rezultatele și constatările din respectarea regulilor, inclusiv acțiunile întreprinse, cu oameni influenți precum managementul de top și autoritățile de reglementare.

Cum și când se efectuează testarea de conformitate -

Se poate face în două moduri: de către echipa dumneavoastră internă sau un grup extern specializat în asta. Lucrul esențial de reținut este că testarea conformității ar trebui să fie efectuată numai de persoane sau organizații autorizate. Inițial, echipa aleasă ar trebui să înțeleagă regulile, cerințele și normele specifice care se

aplică sistemului testat, deoarece organizațiile diferite urmează seturi diferite de reguli.

Guvernul sau industriile importante stabilesc reglementări software pentru a se asigura că este sigur și securizat pentru utilizatorii săi. Unele criterii standard în industria software includ:

- **ISO/IEC/IEEE12207** este o reglementare internațională care guvernează procesele ciclului de viață al software-ului.
- **Consorțiul World Wide Web (W3C)** definește reglementări pentru dezvoltarea web pe diferite dispozitive.
- **Legea privind portabilitatea și responsabilitatea asigurărilor de sănătate (HIPAA)** stabilește standardul pentru protejarea datelor sensibile ale pacienților, făcându-l o cerință pentru sectorul asistenței medicale.
- **Regulamentul general privind protecția datelor (GDPR)** este o reglementare obligatorie în Uniunea Europeană.
- **cu industria cardurilor de plată (PCI)** este necesară pentru sistemele care se ocupă de tranzacțiile cu cardul de credit pentru a proteja clienții împotriva furtului de identitate.

Evaluatorii ar trebui să se familiarizeze cu reglementările relevante pentru țările în care software-ul va fi lansat, reglementările globale și orice reglementări specifice stabilite de organizația lor. Uneori, o organizație poate avea reglementări interne, cum ar fi linii directoare pentru ca paginile web să răspundă la anumite reguli de redirecționare. Echipa de testare a conformității ar trebui, de asemenea, să se familiarizeze bine cu aplicația testată și să colecteze și să analizeze date din controalele și rapoartele anterioare de conformitate.

În timpul testării de conformitate, domeniile de interes includ examinarea stilului de codare și criptare a software-ului. De asemenea, auditul analizează în detaliu toată documentația pentru a asigura respectarea strictă a reglementărilor stabilite.

Testarea conformității cu reglementările

Testarea conformității cu reglementările este un proces care asigură aderarea unei companii la anumite reguli, reglementări și standarde stabilite de organismele de conducere sau autoritățile de reglementare dintr-o anumită industrie. Acest tip de testare este crucial pentru organizațiile care operează în sectoare cu domenii legale și de reglementare stricte, cum ar fi finanțele, sănătatea sau telecomunicațiile.

Testarea conformității cu reglementările urmărește să asigure respectarea legilor și reglementărilor relevante pentru produsele, serviciile sau procesele unei companii. Această testare ajută organizațiile să identifice și să remedieze potențialele probleme de neconformitate înainte de a se confrunta cu consecințe legale sau sancțiuni.

Aspectele critice ale testării conformității cu reglementările includ:

- **Respectarea legală** : validarea faptului că organizația respectă toate legile și reglementările aplicabile relevante pentru industria sa.
- **Securitatea datelor și confidențialitatea** : verificarea gestionării corespunzătoare a informațiilor sensibile, cum ar fi datele clienților, legile privind confidențialitatea și reglementările privind protecția datelor.
- **Standarde de calitate** : Verificarea dacă produsele sau serviciile îndeplinesc standardele de calitate specifice impuse de autoritățile de reglementare.
- **Documentație** : Asigurarea că organizația menține documentația adecvată pentru a demonstra conformitatea cu reglementările.
- **Atenuarea riscurilor** : identificarea și abordarea riscurilor potențiale care ar putea duce la neconformitate.
- **Audituri regulate** : Efectuarea de audituri regulate pentru a evalua conformitatea în curs și pentru a identifica domeniile de îmbunătățire.

Testarea conformității cu reglementările este esențială pentru construirea încrederii cu clienții, evitarea problemelor legale și menținerea unei reputații pozitive în industrie. De asemenea, oferă o abordare sistematică pentru abordarea cerințelor de reglementare și crearea unei culturi a conformității în cadrul organizației.

Cele mai bune practici pentru teste de conformitate cu succes

Continuați să vă îmbunătățiți: îmbunătățire continuă și monitorizare

- Validați că programul dvs. de conformitate se îmbunătățește prin adoptarea unei culturi de îmbunătățire continuă.
- Monitorizați și actualizați în mod regulat programul pentru a vă asigura că rămâne eficient în timp.

Faceți echipă cu autoritățile: colaborarea cu autoritățile de reglementare

- Construiți relații puternice cu autoritățile de reglementare pentru a rămâne la curent cu schimbările sau noile așteptări.
- Mențineți canale de comunicare deschise pentru a stimula colaborarea și înțelegerea reciprocă.

Cunoașterea este putere: programe de formare cuprinzătoare

- Investește în programe de formare amănunțite pentru angajați, oferindu-le abilitățile și cunoștințele necesare pentru sarcinile de conformitate.
- Echipați-vă echipa pentru a naviga cu încredere în complexitățile conformității.

Fii deschis și rapid: raportare transparentă și în timp util

- Asigurați-vă că rezultatele și constatările testelor de conformitate sunt raportate în mod transparent și prompt.
- Împărtășiți deschis acțiunile întreprinse pentru remediere pentru a promova responsabilitatea și a menține încrederea cu părțile interesate.

Protejați ceea ce contează: confidențialitatea și securitatea datelor

- Implementați măsuri puternice pentru a proteja informațiile sensibile în timpul testării de conformitate.
- Respectați reglementările privind protecția datelor pentru a asigura securitatea și confidențialitatea datelor pe parcursul proceselor de testare.

Provocări de testare a conformității în termeni simpli

- **Țineți pasul cu tehnologia și riscurile** : adaptarea la noile tehnologii și amenințări precum atacurile cibernetice și preocupările legate de confidențialitatea datelor poate însemna schimbări în abordările de testare a conformității.
- **Schimbarea regulilor** : Reglementările se schimbă în mod constant, necesitând actualizări ale metodelor de testare a conformității și asigurându-se că angajații știu ce trebuie să facă.
- **Lucrul împreună** : Asigurați-vă că testarea conformității funcționează bine cu alte funcții de asigurare, cum ar fi auditul intern și managementul riscurilor, pentru o eficiență sporită.
- **Resurse limitate** : Uneori, efectuarea tuturor testelor de conformitate dorite poate necesita mai mulți bani sau oameni.

- **Date precise** : este esențial să aveți date exacte și fiabile pentru testarea conformității.

Rolul testerilor QA în conformitate

În domeniul testării de conformitate, testerii de asigurare a calității (QA) au un rol esențial de jucat. Să analizăm importanța lor în asigurarea conformității și diferitele responsabilități pe care le au:

- **Dezvoltarea planurilor de testare** : Testerii QA sunt responsabili pentru formularea planurilor de testare cuprinzătoare, adaptate pentru a evalua conformitatea. Aceste planuri subliniază criteriile, standardele și reglementările specifice pe care software-ul trebuie să le îndeplinească. Această abordare proactivă pregătește scena pentru o testare amănunțită.
- **Efectuarea cazurilor de testare** : Efectuarea cazurilor de testare este o funcție de bază a testatorilor QA în testarea conformității. Ei evaluează riguros fiecare aspect al software-ului în raport cu standardele stabilite, identificând orice abateri sau probleme de neconformitate.
- **Identificarea și raportarea erorilor** : testerii QA, cu ajutorul instrumentelor necesare, identifică erori, erori sau cazuri de neconformitate în timpul procesului de testare. Atenția lor deosebită la detalii asigură că chiar și abaterile minore de la standarde sunt semnalate pentru rezolvare.
- **Colaborare cu experți în reglementare** : testerii QA colaborează cu experți în conformitate și reglementare din organizații cu consilii de reglementare dedicate. Acest parteneriat asigură o înțelegere holistică a standardelor și facilitează testarea eficientă aliniată cu cerințele industriei și legale.
- **Inițiative de îmbunătățire continuă** : testerii QA contribuie la inițiative de îmbunătățire continuă oferind informații valoroase în îmbunătățirea proceselor de conformitate. Ei participă activ la discuțiile privind standardele de rafinare, ridicând astfel calitatea generală a software-ului.
- **Documentare și raportare** : o documentare amănunțită a cazurilor de testare, a rezultatelor și a oricăror probleme de neconformitate identificate este o responsabilitate crucială a testatorilor QA. Raportarea clară și concisă permite dezvoltatorilor și părților interesate să înțeleagă starea de conformitate și să ia măsuri corective.
- **Automatizare pentru eficiență** : Încorporând automatizarea în testele de conformitate, testerii QA simplifică sarcinile repetitive, asigurând

evaluări eficiente și consecvente. Instrumentele de automatizare pot efectua verificări în raport cu standardele, permițând testatorilor să se concentreze asupra aspectelor de conformitate mai complexe.

- **Testare independentă de la terți** : testerii QA pot angaja servicii de testare independente de la terți în scenarii în care o perspectivă externă este crucială. Acești specialiști aduc o evaluare imparțială, oferind un nivel suplimentar de asigurare în ceea ce privește conformitatea.

Testerii QA sunt gardieni ai conformității, jucând un rol vital în respectarea standardelor și reglementărilor în cadrul ciclului de viață al dezvoltării software. Eforturile lor meticuloase identifică și remediază problemele de neconformitate și contribuie la îmbunătățirea continuă a proceselor, promovând o cultură a calității și aderarea la cele mai bune practici din industrie.

Concluzie

Testarea de conformitate se referă la verificarea dacă o companie respectă regulile și standardele de care are nevoie. Aceasta implică examinarea legilor, politicilor și liniilor directoare ale industriei pentru a găsi orice riscuri sau lacune în modul în care compania respectă aceste reguli. Scopul este de a se asigura că compania continuă să respecte aceste reguli în timp.

Principalele părți ale testului de conformitate acoperă lucruri precum regulile și procedurile companiei, programele de formare, activitățile de monitorizare, controalele interne și sistemele de raportare. Cu toate acestea, există provocări în testarea conformității. Acestea includ abordarea noilor tehnologii și riscuri, ținerea pasului cu regulile în schimbare, asigurarea faptului că testarea conformității funcționează bine cu alte verificări, a avea suficiente resurse și menținerea acurateței și fiabilității datelor.

Pentru a face teste de conformitate, este esențial să urmați cele mai bune practici. Acestea includ întotdeauna încercarea de a se îmbunătăți, colaborarea îndeaproape cu oamenii care stabilesc regulile, programe bune de formare, a fi direct și rapid atunci când raportați și a vă asigura că datele sunt private și sigure.

Calea către testarea eficientă a conformității

Deci, cum pot organizațiile să se angajeze în călătoria pentru a implementa programe eficiente de testare a conformității? Iată o foaie de parcurs care vă va ghida:

1. Evaluați peisajul dvs. de reglementare

Începeți prin a înțelege în mod cuprinzător cerințele de reglementare care se aplică organizației dvs. Identificați regulile, legile și standardele specifice care guvernează industria dvs.

2. Proiectați o metodologie de testare de control

Dezvoltați o metodologie de testare a controlului adaptată nevoilor unice ale organizației dvs. Această metodologie de testare a conformității va servi drept model pentru evaluarea controalelor și procedurilor legate de conformitate.

3. Stabiliți un program de testare a conformității

Creați un program de testare a conformității bine definit, care subliniază obiectivele, domeniul de aplicare și termenele eforturilor dvs. de testare. Asigurați-vă că se aliniază cu peisajul dvs. de reglementare.

4. Încorporați testarea automată a conformității

Luați în considerare implementarea instrumentelor și soluțiilor automate de testare a conformității. Acestea pot îmbunătăți semnificativ eficiența și acuratețea eforturilor dvs. de testare.

5. Integrați monitorizarea conformității

Integrați monitorizarea conformității în operațiunile dvs. pentru a asigura o supraveghere continuă. Aceasta include urmărirea în timp real a activităților legate de conformitate.

6. Examinați și actualizați în mod regulat

Mențineți o abordare proactivă prin revizuirea și actualizarea regulată a programului dvs. de testare a conformității. Peisajul de reglementare evoluează, iar programul dvs. ar trebui să evolueze odată cu el.

Cadre pentru testarea conformității

Testarea de conformitate este înțeleaptă pentru diferite cadre de reglementare și standarde din industrie, în funcție de activitățile specifice ale organizației dvs., locația geografică și cerințele de reglementare. Iată câteva cadre și standarde comune pentru care este recomandată testarea conformității:

- HIPAA (Health Insurance Portability and Accountability Act): testarea conformității este esențială pentru a proteja datele sensibile ale pacienților dacă organizația dvs. se ocupă de informații de asistență medicală în Statele Unite.

- PCI DSS (Payment Card Industry Data Security Standard): Organizațiile care gestionează informații despre cardurile de plată ar trebui să efectueze teste de conformitate pentru a menține securitatea datelor deținătorilor de card și pentru a îndeplini cerințele PCI DSS.
- ISO 27001 : Testarea conformității este crucială pentru organizațiile care doresc certificarea ISO 27001. Ajută la evaluarea eficienței controalelor de securitate a informațiilor și asigură alinierea la acest standard internațional.
- GDPR (Regulament general privind protecția datelor): dacă organizația dvs. prelucrează date cu caracter personal ale persoanelor fizice din Uniunea Europeană, testarea conformității este esențială pentru a asigura conformitatea cu cerințele stricte de protecție a datelor din GDPR.
- SOX (Sarbanes-Oxley Act): Companiile cotate la bursă din Statele Unite trebuie să efectueze teste de conformitate pentru a îndeplini cerințele SOX, în special în domeniile legate de raportarea financiară și controalele interne.
- FERPA (Family Educational Rights and Privacy Act): Instituțiile de învățământ trebuie să efectueze teste de conformitate pentru a proteja confidențialitatea înregistrărilor educaționale ale elevilor, așa cum este mandatat de FERPA.
- Cadrul de securitate cibernetică NIST: organizațiile din Statele Unite, în special cele din sectoarele de infrastructură critică, beneficiază de testarea conformității pentru a se alinia cu cele mai bune practici de securitate cibernetică ale NIST.
- FISMA (Federal Information Security Management Act): agențiile federale din SUA și contractanții trebuie să efectueze teste de conformitate pentru a îndeplini cerințele FISMA pentru securitatea informațiilor și securitatea cibernetică.
- Controale CIS (Center for Internet Security Controls): Testarea de conformitate ajută organizațiile să implementeze și să evalueze eficacitatea controalelor CIS, un set de bune practici pentru îmbunătățirea securității cibernetice.

Testarea Automata

Ce inseamna automatizarea testelor?

Automatizarea testelor este o tehnică de dezvoltare software pentru automatizarea executării testelor. Implică scrierea de scripturi sau utilizarea instrumentelor software specializate pentru a efectua teste automat, fără intervenție manuală.

Scopul principal al automatizării testelor este de a crește eficiența și fiabilitatea în procesul de testare a software-ului. Prin automatizarea sarcinilor repetitive, cum ar fi testarea regresiei, dezvoltatorii și testerii pot economisi timp și resurse.

Automatizarea testelor funcționează prin simularea interacțiunilor utilizatorului cu software-ul testat. Aceasta poate include clic pe butoane, introducerea de text și navigarea prin diferite ecrane sau pagini.

Testele automate sunt scrise de obicei în limbaje de programare precum Python, Java sau JavaScript , folosind cadre precum Selenium, Appium sau Cypress. Aceste cadre oferă biblioteci și API-uri pentru a interacționa cu software-ul și a verifica comportamentul acestuia.

Automatizarea testelor poate fi aplicată la diferite niveluri ale procesului de testare, inclusiv testarea unitară, testarea integrării și testarea sistemului. Fiecare nivel se concentrează pe diferite aspecte ale software-ului, de la componente individuale la întregul sistem.

Un beneficiu cheie al automatizării testelor este repetabilitatea acestuia. Odată ce este scris un script de testare, acesta poate fi executat de mai multe ori cu rezultate consistente. Acest lucru vă ajută să vă asigurați că erorile sunt identificate și remediate în mod fiabil.

Automatizarea testelor permite, de asemenea, integrarea continuă și practicile de livrare continuă (CI/CD). Testele automate pot fi integrate în conducta de dezvoltare, permițând dezvoltatorilor să identifice și să rezolve problemele pe măsură ce apar rapid.

Cu toate acestea, automatizarea testelor are și limitări. Nu toate testele pot sau ar trebui să fie automatizate, iar menținerea scripturilor de testare poate necesita un efort semnificativ. Echilibrarea testelor automate și manuale este importantă pentru a asigura o acoperire cuprinzătoare a testelor.

Testează piața de automatizare 2025

Să examinăm evoluțiile cheie și tendințele actuale de automatizare a testelor care au un impact semnificativ asupra industriilor.

1. **Integrarea AI și ML :** se așteaptă ca utilizarea AI și ML în automatizarea testelor să crească, permițând testarea mai eficientă și mai precisă. AI poate ajuta la generarea cazurilor de testare, la detectarea defectelor și la îmbunătățirea procesului general de testare.

2. **Testare cu schimbare la stânga** : se așteaptă ca tendința de testare cu schimbare la stânga să continue, concentrându-se pe integrarea testării în ciclul de dezvoltare mai devreme și mai frecvent. Această abordare urmărește să identifice defectele din timp și să reducă costul total al testării.

3. **Testarea automatizării fără cod** : instrumentele de testare a automatizării fără cod sunt de așteptat să crească în popularitate, permițând testerilor să creeze scenarii de testare fără cunoștințe de codare. Această tendință este determinată de adoptarea tot mai mare a AI și ML în testarea software-ului.

4. **Cloud distribuit pentru testarea între browsere** : se așteaptă ca utilizarea cloudului distribuit pentru testarea între browsere să devină mai răspândită, permițând testarea pe o gamă largă de dispozitive și browsere fără a fi nevoie de dispozitive fizice.

5. **Hiperautomatizare** : se așteaptă ca hiperautomatizarea să devină mai răspândită, implicând instrumente de automatizare pentru a automatiza nu doar testarea, ci și alte aspecte ale ciclului de viață al dezvoltării software.

6. **Testarea exploratorie** : se așteaptă ca testarea exploratorie să devină mai rutină, implicând teste exploratorii pentru a identifica defectele și problemele pe care testarea automată nu le poate detecta.

7. **Testare încrucișată în cloud** : testarea încrucișată în cloud este de așteptat să devină mai populară, permițând testarea pe o gamă largă de dispozitive și browsere fără a fi nevoie de dispozitive fizice.

8. **Testarea performanței aplicațiilor bazate pe IoT** : testarea performanței pentru aplicațiile bazate pe IoT este de așteptat să crească, determinată de adoptarea tot mai mare a tehnologiei IoT în diverse industrii.

9. **Automatizarea în conformitate cu riscurile și testările de securitate** : automatizarea în conformitate cu riscurile și testarea de securitate este de așteptat să devină mai răspândită, folosind instrumente de automatizare pentru a testa riscurile de securitate și de conformitate.
10. **Testarea automatizării mobile** : este de așteptat să crească testarea automatizării mobile, determinată de adoptarea tot mai mare a dispozitivelor și aplicațiilor mobile.
11. **Utilizarea instrumentelor automate pentru aplicațiile blockchain** : Se așteaptă ca utilizarea instrumentelor automate pentru aplicațiile blockchain să crească, determinată de adoptarea tot mai mare a tehnologiei blockchain în diverse industrii.
12. **Automatizare agilă și de testare** : se așteaptă ca automatizarea agilă și de testare să continue să evolueze, permițând testarea mai eficientă și mai eficientă în medii de dezvoltare agile .

Se așteaptă că aceste tendințe vor modela viitorul automatizării testelor, permițând testari mai eficiente, mai eficiente și mai precise în diverse industrii.

Abordarea automatizarii

Abordările de automatizare a testelor variază în funcție de tipul de aplicație, cerințele de testare și resursele disponibile. Să înțelegem câteva dintre abordările cheie cu exemple:

1. Automatizare UI

Această abordare implică automatizarea interacțiunilor interfeței cu utilizatorul pentru a valida comportamentul aplicației din perspectiva utilizatorului final. Exemplele includ:

- **Selenium WebDriver** : Un instrument popular pentru automatizarea interacțiunilor cu browserul web. Suportă mai multe limbaje de programare precum Java, Python și C#, permițând testerilor să scrie scripturi pentru a simula acțiunile utilizatorului în aplicațiile web.
- **Appium** : utilizat pentru a automatiza testarea aplicațiilor mobile pe platformele iOS, Android și Windows. Le permite testerilor să scrie scripturi pentru a interacționa cu aplicațiile mobile folosind același protocol WebDriver ca Selenium.

2. Automatizare API

În automatizarea API, testele interacționează direct cu backend-ul aplicației prin intermediul API-urilor (Application Programming Interfaces) pentru a valida funcționalitatea, datele și integrările. Exemplele includ:

- **Postman** : un instrument utilizat pe scară largă pentru testarea API, care oferă o interfață ușor de utilizat pentru crearea și rularea testelor API. Testerii pot trimite cereri, valida răspunsurile și automatiza fluxurile de lucru API folosind scripturi scrise în JavaScript.
- **REST Assured** : O bibliotecă Java pentru automatizarea testării REST API. Simplifică automatizarea testelor API prin furnizarea unui limbaj specific domeniului pentru scrierea de teste concise și lizibile, făcându-l popular în rândul dezvoltatorilor Java.

3. Automatizare testare de unitate

Această abordare se concentrează pe testarea unităților sau componentelor individuale ale aplicației în mod izolat, de obicei la nivel de cod. Exemplele includ:

- **JUnit** : Un cadru de testare unitar popular pentru Java. Oferă adnotări și aserțiuni pentru a scrie și a executa teste unitare, ajutând dezvoltatorii să asigure corectitudinea codului lor.
- **NUnit** : Similar cu JUnit, dar pentru platforma .NET. Permite dezvoltatorilor să scrie teste unitare în C#, VB.NET sau F# și se integrează cu IDE-uri populare, cum ar fi Visual Studio, pentru o execuție perfectă a testelor.

4. Automatizarea testării performanței

Testarea performanței implică evaluarea modului în care o aplicație funcționează în diferite condiții de încărcare. Automatizarea ajută la simularea a mii de utilizatori virtuali pentru a măsura valorile de performanță. Exemplele includ:

- **Apache JMeter** : un instrument open-source pentru testarea încărcării și măsurarea performanței. Acesta permite testerilor să creeze planuri de testare pentru a simula diferite scenarii de utilizator, pentru a monitoriza valorile de performanță și pentru a analiza rezultatele testelor.
- **Gatling** : un instrument de testare a sarcinii de înaltă performanță construit pe Scala. Folosește un limbaj specific domeniului pentru a descrie scenariile de testare, facilitând scrierea și întreținerea testelor complexe de performanță.

Aceste abordări de automatizare abordează diferite aspecte ale testării software-ului, inclusiv validarea interfețelor utilizator și a API-urilor. Alegerea abordării potrivite depinde de factori precum natura aplicației, obiectivele de testare și expertiza disponibilă.

Metodologii de automatizare a testelor

Metodologiile de automatizare a testelor definesc abordarea și strategiile pentru testarea automată în procesele de dezvoltare software. Unele metodologii comune includ următoarele:

1. Testare bazată pe cuvinte cheie

Această metodologie folosește cuvinte cheie pentru a reprezenta acțiuni sau operațiuni efectuate pe aplicația testată. Scripturile de testare sunt compuse din cuvinte cheie reutilizabile, permițând testerilor să creeze și să mențină teste fără cunoștințe detaliate de programare.

2. Testare bazată pe date

Testarea bazată pe date implică separarea datelor de testare de scripturile de testare. Cazurile de testare sunt executate de mai multe ori cu diferite seturi de date de intrare pentru a valida diferite scenarii. Îmbunătățește acoperirea testului și permite testarea eficientă a mai multor permutări de date.

3. Testare modulară

Testarea modulară descompune aplicația în module mai mici, independente. Fiecare modul este testat separat, facilitând întreținerea mai ușoară și reutilizarea componentelor de testare. De asemenea, permite executarea testelor în paralel, reducând timpul total de testare.

4. Dezvoltare bazată pe comportament (BDD)

BDD se concentrează pe colaborarea între părțile interesate prin definirea scenariilor de testare într-un limbaj comun înțeles atât de membrii echipei tehnice, cât și de cei non-tehnici. Testele sunt scrise într-un format care poate fi citit de om folosind sintaxa Given-When-Then pentru a descrie comportamentele dorite.

5. Dezvoltare bazată pe teste (TDD)

TDD inversează procesul de dezvoltare tradițional prin scrierea de teste înaintea codului real. Dezvoltatorii creează teste unitare automate pentru a defini

funcționalitatea dorită și apoi implementează cod pentru a trece acele teste. Promovează calitatea codului, modularitatea și acoperirea testelor.

6. Testarea hibridă

Testarea hibridă combină mai multe metodologii pentru a le valorifica punctele forte. De exemplu, combinarea abordărilor bazate pe date și bazate pe cuvinte cheie permite reutilizarea scripturilor de testare cu diferite seturi de date, sporind atât eficiența, cât și acoperirea.

7. Testare continuă

Testarea continuă integrează testarea automatizată în întreaga conductă de livrare a software-ului, de la dezvoltare până la implementare. Se asigură că modificările codului sunt validate continuu în raport cu o suită de testare cuprinzătoare, permițând feedback rapid și detectarea timpurie a defectelor.

8. Testare bazată pe model

Testarea bazată pe model folosește modele pentru a reprezenta comportamentul sistemului testat. Aceste modele sunt apoi transformate automat în cazuri de testare executabile. Acest lucru ajută la generarea de scenarii de testare cuprinzătoare și la asigurarea acoperirii testelor.

9. Testare paralelă

Testarea în paralel execută mai multe cazuri de testare simultan în diferite medii sau dispozitive. Reduce semnificativ timpul de execuție a testului, permițând un feedback mai rapid și sporind eficiența testului.

10. Testare exploratorie cu suport de automatizare

Această metodologie combină testarea exploratorie, în care testerii explorează aplicația în mod dinamic, cu verificări automate pentru a valida funcționalități sau scenarii specifice. Permite flexibilitate și creativitate în testare, asigurând în același timp repetabilitate și consecvență.

Aceste metodologii oferă diverse abordări ale automatizării testelor, răspunzând diferitelor cerințe de proiect, abilități de echipă și complexități ale aplicațiilor. Alegerea metodologiei potrivite depinde de factori precum obiectivele proiectului, disponibilitatea resurselor și cultura organizațională.

Instrumente automate de testare

Instrumentele automate de testare sunt aplicații software concepute pentru a executa teste automat. Ele ajută echipele de dezvoltare software să asigure eficient calitatea produsului. Să înțelegem câteva exemple cheie de instrumente de testare automată:

1. Seleniu

Selenium este un cadru de testare open-source utilizat pe scară largă pentru aplicații web. Acesta acceptă diverse limbaje de programare, cum ar fi Java, Python și C#, permițând testerilor să scrie scripturi automate pentru a interacționa cu elementele web și pentru a valida funcționalitățile din browsere.

2. Appium

Appium este un instrument de automatizare open-source pentru aplicații mobile. Le permite testerilor să automatizeze testarea aplicațiilor web native, hibride și mobile pe platformele iOS și Android. Appium folosește protocolul WebDriver, făcându-l compatibil cu Selenium pentru testarea web.

3. JUnit

JUnit este un cadru de testare unitar popular pentru Java. Oferă adnotări și aserțiuni pentru definirea și executarea cazurilor de testare, ajutând dezvoltatorii să asigure corectitudinea unităților de cod individuale. JUnit se integrează perfect cu instrumente de construcție precum Maven și Gradle.

4. TestNG

TestNG este un alt cadru de testare pentru Java care oferă caracteristici suplimentare dincolo de JUnit. Acceptă teste parametrizate, dependențe de testare și execuție paralelă, făcându-l potrivit pentru scenarii de testare mai complexe. TestNG este utilizat pe scară largă atât în testarea unitară, cât și în cea de integrare.

5. Castravete

Castravetele este un instrument BDD care permite colaborarea între membrii echipei tehnice și non-tehnici. Folosește sintaxa Gherkin pentru a scrie scenarii care pot fi citite de om, care pot fi automatizate folosind definiții de pași. Cucumber acceptă mai multe limbaje de programare precum Java, Ruby și JavaScript.

6. JMeter

Apache JMeter este un instrument open-source pentru testarea performanței aplicațiilor web. Simulează sarcini mari pe servere, baze de date și rețele pentru a măsura parametrii de performanță, cum ar fi timpul de răspuns, debitul și concurența. JMeter acceptă diverse protocoale, inclusiv HTTP, JDBC, FTP și SOAP.

7. Poștaș

Postman este un instrument de testare API pentru API-urile RESTful. Acesta oferă o interfață ușor de utilizat pentru crearea și executarea solicitărilor API, automatizarea testelor și generarea de rapoarte cuprinzătoare. Postman acceptă scripting folosind JavaScript pentru scenarii de testare avansate.

8. Cadrul robotului

Robot Framework este un cadru generic de automatizare a testelor care acceptă abordări de testare bazate pe cuvinte cheie, comportament și date. Oferă o sintaxă ușor de citit și acceptă diverse biblioteci pentru testarea aplicațiilor web, mobile, API și desktop.

9. Sikuli

Sikuli este un instrument de automatizare vizuală care automatizează sarcinile bazate pe recunoașterea imaginilor. Acesta permite testerilor să interacționeze cu elementele GUI prin capturarea și potrivirea imaginilor de pe ecran. Sikuli este util pentru automatizarea sarcinilor în aplicații în care metodele tradiționale de automatizare sunt provocatoare.

10. SoapUI

SoapUI este un instrument de testare API open-source conceput pentru a testa serviciile web SOAP și RESTful. Oferă funcții pentru testarea funcțională, de performanță și de securitate a API-urilor, făcându-l un instrument cuprinzător pentru testarea API.

Aceste instrumente automate de testare răspund diferitelor nevoi, inclusiv testarea web, mobilă, unitară și API. Acestea permit echipelor să livreze software de înaltă calitate în mod eficient.

Beneficiile și provocările automatizării testelor

Automatizarea testelor oferă mai multe beneficii unice care sporesc eficiența și eficacitatea procesului de testare:

1. **Eficiență crescută** : automatizarea permite executarea testelor mult mai rapid decât testarea manuală, rezultând un feedback mai rapid asupra calității software-ului.
2. **Repetabilitate și consecvență** : testele automate efectuează în mod constant aceiași pași, reducând probabilitatea erorilor umane și asigurând rezultate fiabile ale testelor în mai multe execuții.
3. **Acoperire îmbunătățită a testelor** : automatizarea permite executarea multor cazuri de testare, ceea ce duce la o acoperire cuprinzătoare a testelor, care ar putea să nu fie fezabilă doar cu testarea manuală.
4. **Economii de costuri** : În timp ce configurarea și investițiile inițiale pot fi necesare, automatizarea reduce în cele din urmă costurile de testare prin economisirea de timp și resurse pe termen lung.
5. **Timp de lansare pe piață mai rapid** : automatizarea accelerează procesul de testare, ajutând la accelerarea ciclului de dezvoltare și permițând livrarea mai rapidă a produselor software pe piață.
6. **Testare de regresie** : automatizarea este deosebit de eficientă pentru testarea de regresie , unde funcționalitățile existente sunt testate pentru a se asigura că nu au fost afectate de noi modificări sau actualizări.
7. **Testare în paralel** : instrumentele de automatizare pot executa teste în paralel în mai multe medii, dispozitive sau browsere, permițând testarea simultană și feedback mai rapid.

În ciuda beneficiilor sale, automatizarea testelor prezintă și câteva provocări:

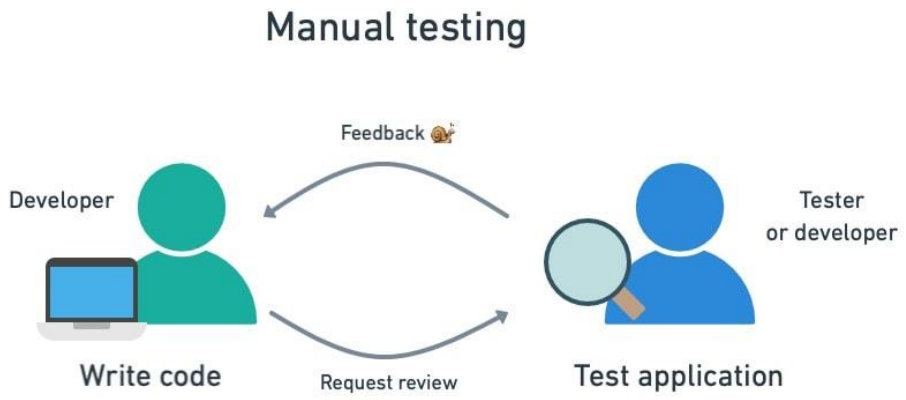
1. **Investiție inițială** : Configurarea cadrelor de automatizare și scrierea scripturilor de testare necesită timp, efort și resurse, ceea ce poate fi o investiție inițială semnificativă.
2. **Costul general de întreținere** : testele automate necesită întreținere regulată pentru a ține pasul cu schimbările software în evoluție. Actualizările aplicației pot necesita actualizări corespunzătoare pentru scripturile de testare, crescând cheltuielile de întreținere.
3. **Cerințe de competență** : automatizarea testelor necesită abilități specializate în scriptare, programare și instrumente de automatizare a testelor, care este posibil să nu fie ușor disponibile în cadrul echipei de testare.

4. **Complexitatea scenariilor de testare** : Unele scenarii de testare, în special cele care implică o logică complexă de afaceri sau interacțiuni cu utilizatorul, pot fi dificil de automatizat în mod eficient.
5. **Fals pozitive/negative** : testele automate pot produce uneori rezultate fals pozitive sau negative din cauza unor factori cum ar fi problemele legate de mediul de testare, scripturile de testare slabe sau acoperirea incompletă a testului.
6. **Raționament uman limitat** : automatizării îi lipsește intuiția și judecata umană care pot fi valoroase în testarea exploratorie sau identificarea defectelor subtile.
7. **Provocări de integrare** : Integrarea automatizării testelor în procesele existente de dezvoltare și implementare, inclusiv conducte CI/CD , poate necesita o planificare și coordonare atentă.

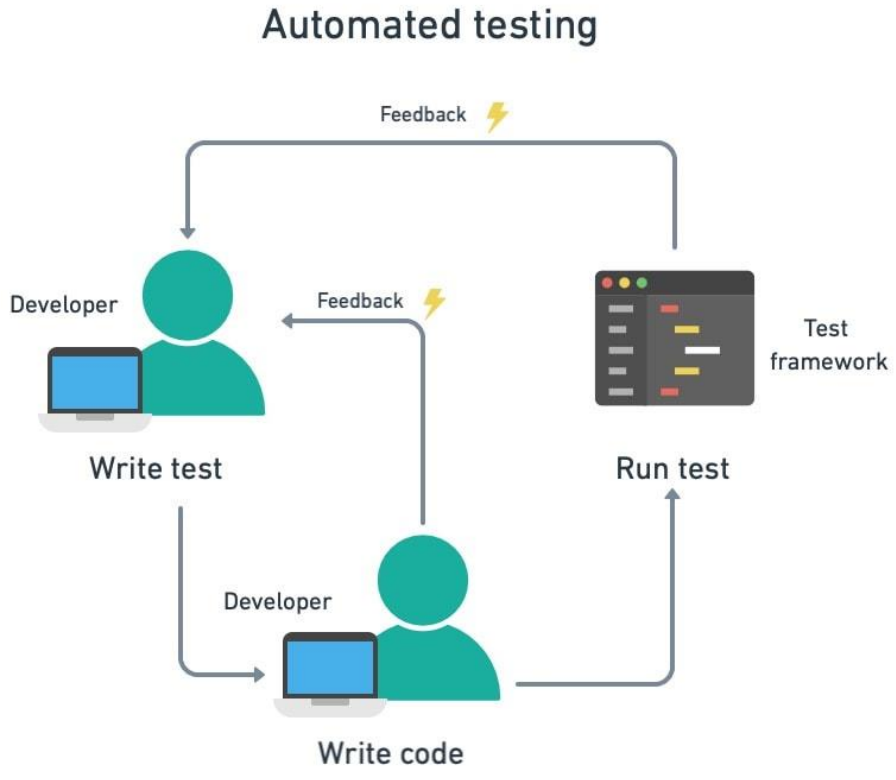
În ciuda acestor provocări, beneficiile automatizării testelor includ eficiență sporită, repetabilitate, acoperire îmbunătățită și economii de costuri. Acestea depășesc provocările, făcându-l un atu valoros în strategiile de testare a software-ului.

Principii de automatizare

Utilizăm teste în dezvoltarea de software pentru a ne asigura că codul funcționează conform așteptărilor. Testele pot fi manuale sau automate. Testarea manuală este similară cu cea a producătorilor de automobile care prăbușesc mașinile pentru a verifica dacă sunt sigure pentru drum. Funcționează, dar este mult prea scump pentru a fi făcut în mod frecvent, deci se face de obicei la sfârșitul ciclului de producție. Problema cu această metodă este că problemele găsite în această etapă pot întârzia lansarea unui produs cu luni de zile.



Testarea automată a software-ului are o structură de cost complet diferită. Există o inversare inițială plus întreținere periodică, dar odată ce automatizarea este pusă în aplicare, ne putem rula testele de câte ori avem nevoie - pentru bani.



Cu automatizarea testelor, dezvoltatorii primesc feedback continuu, permițându-le să identifice probleme foarte devreme în ciclul de producție. Iterația rapidă are ca rezultat un design îmbunătățit, o calitate mai bună și lansări mai sigure.

Principiile automatizării testelor

Cărți întregi au fost scrise exclusiv pe tema automatizării testelor. Este o abilitate pe care fiecare dezvoltator trebuie să o stăpânească la un moment dat și este mai bine să o facă mai devreme decât mai târziu.

Iată șase principii pentru a ușura curba de învățare:

1. Testele ar trebui **să îmbunătățească calitatea** .
2. Testele ar trebui **să reducă riscul** introducerii defecțiunilor.
3. Testarea ajută la **înțelegerea codului** .
4. Testele trebuie să fie **ușor de scris** .
5. O suită de teste trebuie să fie **ușor de rulat** .
6. O suită de teste ar trebui să necesite **întreținere minimă** .

Principiul 1: automatizarea testelor îmbunătățește calitatea

Calitatea este un concept evaziv. Oricât am încerca, este imposibil să-l definim numeric. Cu toate acestea, o știm când o vedem. Industria software a venit cu multe indicatori pentru a măsura calitatea: numărul de defecte, acoperirea codului, rata de eroare CI, rata de eșec al testului și așa mai departe. Fiecare surprinde un anumit aspect al ideii de calitate.

Testele automate îmbunătățesc valorile calității prin rularea continuă a sute sau mii de teste; găsirea defectelor înainte de a ajunge în producție, informarea dezvoltatorilor cu privire la problemele potențiale și verificarea dacă sistemul se abate de la așteptările utilizatorilor.

Rapoartele de testare în Semaphore arată o vedere la nivel înalt a stării unui proiect

Pe lângă valorile, știm că un design solid este o condiție prealabilă pentru calitate. Când testele stimulează dezvoltarea, dezvoltatorii pot încerca cu ușurință diferite idei și pot determina care dintre ele funcționează cel mai bine.

Principiul 2: automatizarea testelor reduce riscul

Revizuirea codului și programarea de la egal la egal, deși necesare și productive, nu se pot baza pentru a găsi erori. Experiența arată că mai mulți globi oculari nu se traduce prin mai puține erori.

Singura modalitate de a găsi erorile în mod fiabil este să construiți o suită cuprinzătoare de teste automatizate. Testele pot verifica întreaga aplicație de sus în jos. Aceștia detectează erori înainte de a putea face vreun rău, de a găsi regresii și de a rula aplicația pe diverse dispozitive și medii la o scară care, altfel, este prohibitiv de costisitoare de încercat manual.

Chiar dacă toată lumea din echipă a fost un dezvoltator excepțional de inteligent care cumva nu a greșit niciodată, dependențele de la terți pot introduce erori și pot prezenta riscuri. Testele automate pot scana fiecare linie de cod din proiect pentru erori și probleme de securitate.

Principiul 3: testele vă ajută să înțelegeți sistemul

Prea frecvent, dezvoltatorii revin la codul scris cu doar câteva zile în urmă doar pentru a realiza că au uitat complet cum funcționează. Acest lucru este și mai rău atunci când dezvoltatorii trebuie să se ocupe de codul scris de alți oameni.

Adesea, testele de citire sunt cel mai bun loc pentru a înțelege un sistem, deoarece arată cum funcționează lucrurile prin exemplu. Deci, atunci când au dubii, dezvoltatorii se pot referi la suita de teste.

Testele, de exemplu, pot arăta unui alt dezvoltator cum ar trebui să răspundă un API , permițându-i să nu mai consulte documentația.

Nu sunteți sigur dacă este necesară o linie de cod? Comentează-l pentru a vedea care test eșuează. Aveți o idee pentru a îmbunătăți o funcție? Trebuie să refactorizezi o bucată de cod? Încercați-l și rulați testele automate. Veți fi surprins cât de multe puteți afla despre un sistem din testele sale.

Principiul 4: testele automate ar trebui să fie ușor de scris

Unele teste își încep viața ca teste manuale și se automatizează pe drum. Dar, de cele mai multe ori , acest lucru duce la teste prea complicate, lente și incomode. Cele mai bune rezultate vin atunci când testele și codul au o anumită sinergie. Actul de a scrie un test îi determină pe dezvoltatori să producă un cod mai modular, ceea ce, la rândul său, face testele mai simple și mai granulare.

Simplitatea testelor este importantă deoarece nu este practic să scrieți teste pentru teste. Codul ar trebui să fie, de asemenea, simplu de citit și de scris. În caz contrar, riscăm să introducem eșecuri cu testul propriu-zis, ducând la fals pozitive și scăderi.

Multe cadre de testare folosesc limbaje specifice domeniului (DSL) pentru a defini testele în limba engleză simplă. Poate cel mai notabil exemplu este Gherkin, limbajul folosit de cadrul de testare Cucumber :

Pentru a rezuma, este o idee bună să respectați câteva elemente fundamentale atunci când scrieți teste:

- Scrieți o singură afirmație pe test.
- Păstrați codul separat de teste, adică codul de producție nu ar trebui să includă teste.
- Păstrați testele independente unele de altele, deoarece dependențele pot transforma rapid bulgărele de zăpadă într-o mizerie care provoacă dureri de cap.
- Păstrați suprapunerea testelor la minimum, adică nu testați același cod de două ori.
- Nu rupeți încapsularea codului testat. În schimb , testați doar interfețele externe.

Principiul 5: testele ar trebui să fie ușor de executat

Dacă un dezvoltator trebuie să deschidă o listă de verificare pentru a începe o rulare de testare, testele dvs. nu vor fi rulate atât de des pe cât ar trebui.

Pentru ca testele să fie ușor de executat, trebuie îndeplinite câteva condiții:

- **Idempotenta** : testele nu trebuie sa aiba efecte secundare. Efectele secundare includ scrierea în fișiere, salvarea într-o bază de date sau, în general, modificarea datelor. Dezvoltatorii ar trebui să poată rula în siguranță aceleași teste de câte ori.
- **Determinist** : testele ar trebui să dea întotdeauna același rezultat, având aceleași intrări. Când testele au nevoie de date externe care sunt în afara controlului dezvoltatorului, cum ar fi data/ora sau un răspuns de la un API, acestea ar trebui să fie falsificate cu falsuri sau stub-uri.
- **Independente** : testele ar trebui să fie independente unele de altele, iar dezvoltatorii trebuie să le poată rula în orice ordine.
- **Ușoare** : testele trebuie să fie suficient de ușoare pentru a rula pe computerul dezvoltatorului într-un timp rezonabil .
- **Granular** : dezvoltatorii trebuie să poată rula suita de testare fragmentar.

Rularea testelor pe mașina dezvoltatorului este doar o parte a ecuației. Testarea trebuie să aibă loc, de asemenea, în cadrul conductei de integrare continuă . Conducta dumneavoastră CI/CD acționează ca o poartă de calitate; rulează suita de testare la fiecare comitere, oferind feedback instantaneu și permițând dezvoltatorilor să detecteze când a fost introdus un eșec.



Principiul 6: o suită de testare automată ar trebui să necesite o întreținere redusă

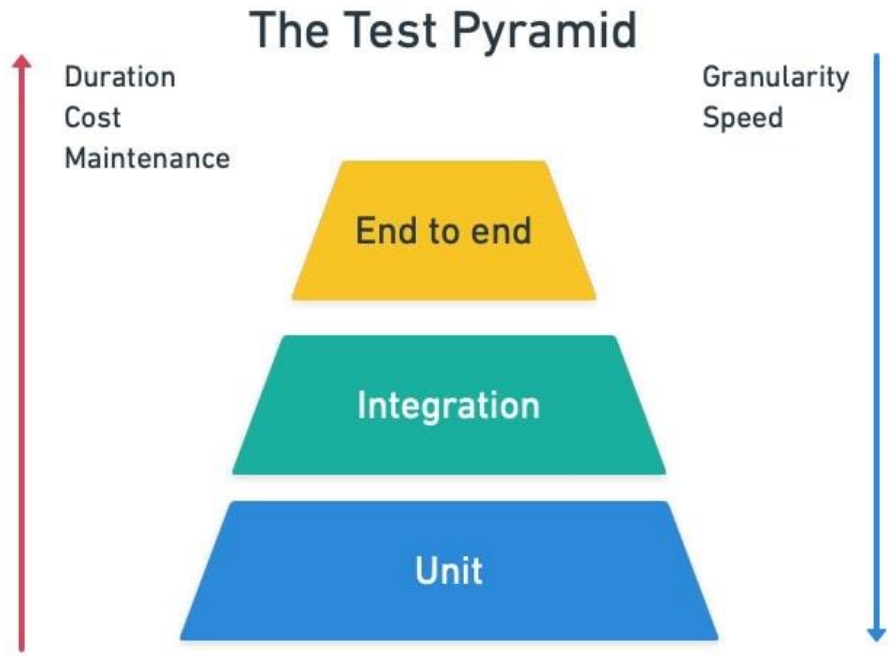
Ultimul principiu este un corolar al celor cinci anterioare. Adică îl primești gratuit dacă le îndeplinești bine pe celelalte. Totuși, este important, așa că este bine să facem un punct.

Dezvoltatorii doresc să facă o muncă creativă și plină de satisfacții. Automatizarea permite mașinilor să se ocupe de greoaiele testării. O buclă de feedback pozitiv este creată atunci când testele sunt ușor de scris și sunt executate frecvent. Dezvoltatorii tind să aprecieze modul în care automatizarea le face viața mai ușoară și, prin urmare, sunt motivați să scrie și să mențină teste.

Desigur, va fi necesară o întreținere periodică pentru a vă menține testele în stare bună. Iată patru recomandări pentru scrierea și întreținerea suitei de teste:

- Scrieți suficiente teste pentru a fi eficiente (dar nu mai multe). Dacă erorile trec, aveți nevoie de mai multe teste. În schimb, dacă descoperiți că testele se întrerup cu mici modificări, trebuie să eliminați unele teste.
- Alegeți cel mai bun tip de test pentru situație. Testele unitare sunt rapide și concentrate pe laser, în timp ce testele end-to-end acoperă interfața de utilizare și sunt grele și mai cuprinzătoare. O suită de teste care urmează

  piramida de testare are o varietate sănătoasă de teste.



Piramida de testare

- Păstrați testele de încredere. Un test care eșuează atunci când codul este corect se numește fals pozitiv. Testele care uneori eșuează fără un motiv aparent se numesc teste scazute . Ambele cauzează probleme într-o suită de teste, deoarece sunt uriașe pierderi de timp și surse de frustrare.
- Păstrați testele rapid. O suită de teste lente va pune frâna dezvoltării.

Concluzie

Cei care cred că testarea este costisitoare nu sunt pe deplin conștienți de costul calității proaste. Individual, impactul erorilor și defectelor asupra valorii produsului poate fi greu de măsurat, dar poate scăpa rapid de sub control dacă nu sunt abordate. Din fericire, puteți preveni acest lucru prin construirea și rafinarea suită de testare automată, pentru a servi drept fundație pentru o experiență excelentă de dezvoltator și un software remarcabil, de calitate.

Cum se compară cu alte metode de testare

Consumatorii se așteaptă ca aplicațiile lor web și mobile să funcționeze atunci când au nevoie de ele. Și asta înseamnă testarea robustă a software-ului și QA este o cerință.

Există trei tipuri de testare software pe care echipele trebuie să le înțeleagă și să le implementeze. Iată o detaliere rapidă a elementelor de bază ale testării automate vs. testării manuale vs. testării continue:

Testare automată

Utilizează instrumente speciale pentru a rula mai multe teste software simultan. Testarea automată a fost concepută pentru a accelera procesul de testare și pentru a crește acoperirea globală a testării. Utilizarea acestuia duce la o mai bună detectare a erorilor, deoarece instrumentele de automatizare a testelor pot acoperi mai multe sisteme de operare, dispozitive și scenarii decât testarea manuală.

Testare manuală

Cel mai tradițional tip de testare. În testarea manuală a software-ului, analiștii și testerii creează testele și le execută manual. De obicei, testarea manuală necesită mult timp, dar este benefică, deoarece testarea practică permite o înțelegere mai contextuală și nuanțată a testului care este executat.

Testare continuă

Procesul de executare a testelor automate într-o manieră repetabilă și scalată. Testarea continuă are loc atunci când testarea software-ului este complet automatizată și optimizată. Avantajele testării continue includ feedback rapid, lansare rapidă și aplicații web și mobile de calitate.

Provocări comune în testarea automată

Există 5 provocări de bază în testarea automată astăzi:

1. Lipsa abilităților necesare.
2. Echipele nu stabilizează testele înainte de automatizare.
3. Managementul laboratorului este împovărător.
4. Automatizarea testelor devine dinamică.
5. Sortarea tuturor datelor.