Ciclul de dezvoltare software

1. Primul nivel de testare ce se regaseste in piramida de testare este **unit testing.** Acest tip de testare se regaseste la baza piramidei și are ca scop izolarea codului pentru a se asigura ca acesta funcționează conform așteptărilor.

Al doilea nivel de testare ce se regaseste in piramida de testare este **integration testing**. Acesta testare are rolul de a verifica interacțiunea codului cu celălalt cod din aplicație. Acest test verifica interacțiunea codului cu componentele externe (ex. API’s, baze de date, servicii web și alte elemente asemănătoare).

Al treilea nivel de testare este **end to end testing,** fiind în varful piramidei de testare**.** Acest tip de testare verifica întregul cod al aplicației folosindu-se de environment pentru a simula functionalitatile din lumea reala. Prin acest tip de testare pot fii mai greu identificate erorile.

Aplicatie bancara

1. **Functionalitate**: bara de cautare

* O bara de cautare are ca și principala functionalitate aceea de a căuta un termen sau mai multe cuvinte intr-o baza de date. Scopul acesteia este de a ușura procesul de căutare a unui client în funcție de anumite date ca ( CNP, data de nastere, nume/prenume, etc.).
* Bara de cautare v-a fi legata de o baza de date ceea ce va presupune ca aceasta bara de cautare ar trebui sa îmi returneze rezultatul așteptat; de ex: În casuta de cautare introduc CNP-ul unui client și doresc ca după căutare sa imi afișeze informațiile clientului dorit de mine.
* Bara de cautare mai are functionalitatea de a sugera posibile rezultate pe baza primei/lor litere introdu-se.

*Scrierea testelor*

* Un prim scenariu ar fii: Sa introduc în casuta de cautare un nume de client și sa observ daca casuta de cautare îmi v-a returna rezultatul așteptat( informațiile clientului sa corespundă cu numele acestuia).
* Un alt scenariu ar putea fi introducerea în căsuța de căutare a unor elemente speciale ca( -,%,&,#,$), pentru a determina dacă casuta de cautare îmi v-a returna vreun rezultat.
* Un alt scenariu posibil ar fi introducerea în căsuța de căutare a cifrelor (15,20,etc) sau a cifrelor negative(-20,-15, etc.)

*Testarea efectiva*

* Pentru a verifica fiecare componenta a casutei de căutare se va realiza un **functional testing**. Acest lucru va însemna ca se va testa fiecare funcție pe care o va realiza casuta de cautare. De exemplu: se verifica dacă casuta de cautare este activa si se poate tasta în aceasta, se verifica dacă casuta de cautare căuta informațiile cerute de mine si-mi returnează vre-un rezultat, se verifica corectitudinea rezultatelor afișate.

*Inchiderea task-ului*

In primul rand, testele eșuate le voi retesta după efectuarea modificarilor necesare realizate de către programator.

In al doilea rand, prin procesul de retestare voi verifica dacă modificările aduse codului au rezolvat problemele sesizate sau dacă acestea nu au creat altele noi. Astfel mă voi asigura ca produsul final va fii unul de calitate.

In al treilea rand, după retestare ar trebui ca majoritatea testelor sa treaca și astfel funcția de căutare sa fie implementata corect.

Pentru a verifica dacă după modificările aduse codului specific barei de căutare; celelalte functionalitati ale aplicației nu au fost afectate (ex:afișarea soldului, crearea de carduri virtuale, afișarea codului IBAN, etc) se va realiza un **integration testing**

*Finalizarea task-ului*

Pentru ca task-ul meu “al barei de testare” să fie considerat finalizat, ar trebui sa indeplineasca următoarele criterii:

* Funcționarea corespunzătoare ale barii de căutare( să realizeze ceea ce își propune aceasta).
* Daca functia aplicației bancare “bara de cautare” îndeplinește cerințele de conformitate ale clientului, atunci task-ul poate fi considerat finalizat.
* Rezolvarea tuturor erorilor descoperite până la acest moment.
* Daca la final, “bara de cautare” este stabilă din punct de vedere software, atunci task-ul este considerat finalizat.

1. Avantajul pe care il are unit testing fata de system testing este acela ca **unit testing** testează o parte din codul viitorului proiect, astfel descoperindu-se mai repede erorile posibile. În cazul **system testingului** se verifica întreaga aplicație, astfel ingreunand la descoperirea posibilelor erori.

Testarea unitară are avantajul de a ajuta dezvoltatorii sa identifice posibilele bug-uri încă din etapa de început a dezvoltării software.

1. Avantajul pe care îl oferă end to end testing este ca acesta verifica integritatea întregii aplicații, în special al functionalitatilor și al adaptării acesteia la noi medii de producție.

Acest tip de testare este realizat la finalul fiecărei iterații în metodologia agile. Astfel sunt mai mici șansele ca aplicația să aibă multe bug-uri în etapa de producție.

Fata de unit testing, end to end testing poate ajuta la descoperirea de bug-uri ce nu au putut fii descoperite în etapa de early testing folosind testele unitare. Astfel reducandu-se timpul necesar descoperirii de bug-uri în etapa viitoare dar și al costurilor necesare reparării erorilor apărute din cauza acelor bug-uri.

1. **Calcul rata** - se va realiza la nivel de *unit testing*.

Am ales acest tip de testare deoarece se testează logica functionalitatii acestui calcul de rata.

**Postare incident in aplicatie(erori de plati, de functionalitate, de afișare a aplicației)** - se va realiza o *testare a integrării* subsistemelor.

În cazul acestei functionalitati se vor realiza teste de integrare deoarece se va verifica dacă aceste erori de plata funcționează corect și de asemenea se va verifica cerințele functionalitatii de plata, functionalitatea de afișare și dacă defectele au fost descoperite din timp. Se verifica dacă aceste parti ale aplicației funcționează corect impreuna (functionalitatea de plată și afișarea acesteia).

**Toate campurile din pagina de înregistrare sunt obligatorii** - pentru aceasta functionalitate se va realiza un *unit testing.*

Deoarece se verifica cerințele la nivelul aplicației de banking si astfel sunt descoperite bug-urile ce țin de functionalitatea acesteia.

**Plăti IBAN -** pentru aceasta functionalitate voi folosi *integration testing*

Pentru a se verifica dacă plățile sunt realizate corect si sa se verifica dacă functionalitatile de plata funcționează ( acestea pot tine de funcționarea plății, performanța sau securitatea codului IBAN).

**Investiții prin aplicația bancară** -se va realiza un integration testing.

Deoarece se testează modul în care se integrează functionalitatea de investiții prin aplicatia de banking.