

Proiect Final

Nume Prenume
14.10.2023

CUPRINS

I. Noțiuni teoretice

- ▶ Informații acumulate ca urmare a parcurgerii cursului de testare manuală;

II. Aspecte practice

- ▶ Punerea în aplicare a cunoștințelor dobândite;

I. Informații acumulate ca urmare a parcurgerii cursului de testare manuală

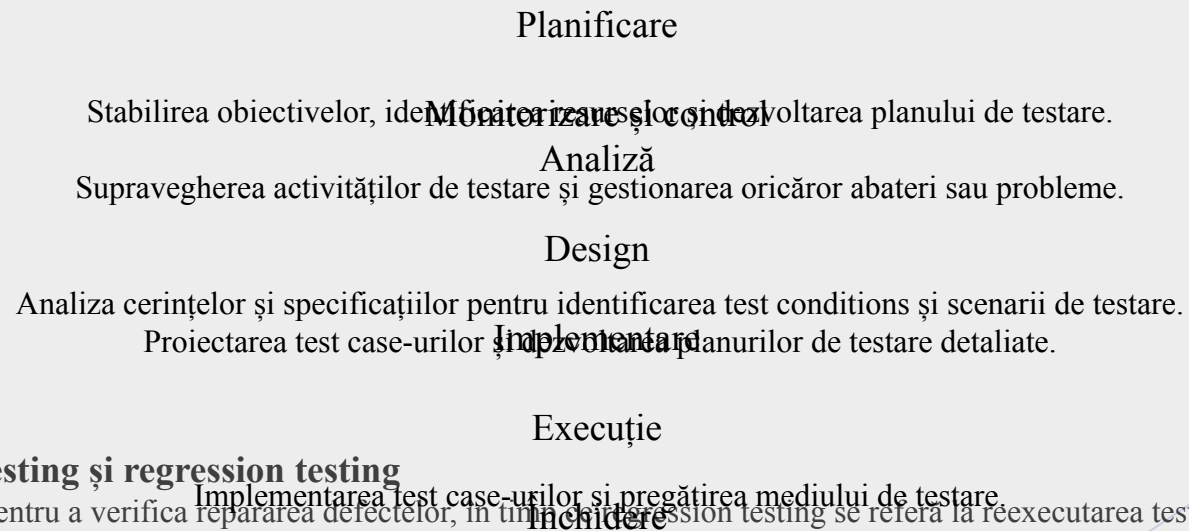
1. Explicați pe scurt ce sunt cerințele de business, la ce ne folosesc și cine le creează

- Cerințele de business sunt descrieri ale nevoilor și așteptărilor clienților și utilizatorilor pentru un produs/software. Ele sunt create de o echipă care include reprezentanți ai clienților și utilizatorilor și ghidează dezvoltarea și testarea produsului.

2. Explicați diferența între un test condition și test case

- Test condition este o descriere generală a unei situații care trebuie verificată, în timp ce test case este o specificație detaliată a pașilor și datelor de testare pentru a verifica o funcționalitate sau caracteristică.

3. Enumerați și explicați pe scurt etapele procesului de testare



4. Explicați diferența între retesting și regression testing

- Retesting se referă la reluarea testelor pentru a verifica repararea defectelor, în timp ce regression testing se referă la reexecutarea testelor existente pentru a verifica dacă modificările recente nu au introdus probleme noi.

Efectuarea testelor conform planului și înregistrarea rezultatelor.

I. Informații acumulate ca urmare a parcurgerii cursului de testare manuală

5. Explicați diferența între functional testing și non-functional testing

- ▶ Functional testing se concentrează pe verificarea funcționalității produsului, în timp ce non-functional testing se concentrează pe aspecte cum ar fi performanța, securitatea, compatibilitatea și utilizabilitatea.

6. Explicați diferența între blackbox testing și whitebox testing

- ▶ Blackbox testing se referă la testarea în care testerul nu are cunoștințe despre structura internă (cod) a produsului, în timp ce whitebox testing implică cunoștințe detaliate despre structura internă și logica produsului

7. Enumerați tehnicile de testare și grupați-le în funcție de categorie (blackbox, whitebox, experience-based)

Blackbox testing

- Partiționarea echivalentă
- Analiza valorilor limită
- Testarea tranzițiilor de stare
- Tabel decizional
- Use case

Whitebox testing

8. Explicați diferența între verification și validation

- ▶ Verification (Decision coverage) este procesul de verificare a dacă un produs respectă specificațiile și standardele, în timp ce validation se referă la evaluarea dacă produsul îndeplinește nevoile și așteptările utilizatorilor.

9. Explicați diferența între positive testing și negative testing și dați câte un exemplu din fiecare

- ▶ **Positive testing** se referă la testarea în care se verifică dacă sistemul funcționează așa cum se așteaptă în scenariile de utilizare normală.
 - Ghicirea erorilor
 - ▶ Exemplu: Verificarea funcționării corecte a unui formular de înregistrare atunci când sunt introduse date valide.
- ▶ **Negative testing** se referă la testarea în care se verifică modul în care sistemul gestionează scenariile de utilizare anormale sau incorecte.
 - Testarea exploratorie
 - Checklist based testing
 - ▶ Exemplu: Verificarea comportamentului sistemului atunci când se introduc date incorecte sau lipsă într-un formular de înregistrare

I. Informații acumulate ca urmare a parcurgerii cursului de testare manuală

10. Enumerați și explicați pe scurt nivelurile de testare

► Testare unitară

Acest nivel de testare se concentrează pe verificarea și validarea individuală a fiecărei componente sau module de cod. Este realizată de obicei de către dezvoltatori și are ca scop identificarea și rezolvarea erorilor la nivelul cel mai mic, asigurând funcționarea corectă a fiecărei componente în mod izolat.

► Testarea componentelor

Această etapă implică testarea grupurilor de componente sau module care interacționează între ele în cadrul unei anumite funcționalități sau servicii. Scopul este de a verifica integrarea corectă și interacțiunea între aceste componente și de a identifica și rezolva eventualele probleme de compatibilitate sau comunicare.

► Testare de integrare

Această etapă implică testarea întregului sistem sau aplicație, verificând modul în care componentele interacționează și funcționează împreună. Se testează fluxul de date, comunicarea între module și integrarea funcționalităților. Obiectivul este de a identifica și rezolva eventualele probleme de integrare și de a asigura funcționarea corectă a sistemului ca un întreg.

► Testarea sistemului

Acest nivel de testare se concentrează pe evaluarea sistemului ca un întreg, într-un mediu similar cu cel în care va fi utilizat în producție. Scopul este de a verifica dacă sistemul îndeplinește cerințele și așteptările utilizatorilor și de a identifica orice probleme sau defecte majore care pot afecta funcționalitatea sau performanța sistemului.

► Testare de acceptare a utilizatorului

Aceasta este etapa finală de testare înainte de livrarea sistemului către utilizatorul final. Scopul este de a verifica dacă sistemul îndeplinește criteriile de acceptare definite de utilizator și de a asigura că sistemul este gata să fie utilizat în producție. Utilizatorii sau clienții testează sistemul într-un mediu de producție simulat sau real, pentru a valida și confirma că acesta îndeplinește necesitățile lor și funcționează conform așteptărilor.

II. Punerea în aplicare a cunoștințelor dobândite

1. Cerințele pentru aplicația / website-ul ales - Se vor adăuga în prezentare DOUA cerințe de business (story-uri) care să reflecte atât titlul cât și descrierea story-ului. Se pot face poze direct la ce ați lucrat în Jira

II. Punerea în aplicare a cunoștințelor dobândite

2. Condițiile de testare și cazurile de testare - Condițiile de testare vor reprezenta DOAR TITLUL testelor, iar test case-urile trebuie să conțină și pașii de execuție. Condițiile de testare trebuie să fie cel puțin 10, iar testele adăugate în prezentare trebuie să fie cel puțin două

II. Punerea în aplicare a cunoștințelor dobândite

3. Matricea trasabilității - trebuie să reflecte în mod corect legătura între story-uri, teste și bug-uri. Ca să se întâmple asta trebuie să vă asigurați că testele sunt legate corect la story-uri și că bug-urile sunt legate corect la teste

II. Punerea în aplicare a cunoștințelor dobândite

4. Raportul generat din aplicația Jira din secțiunea Dashboards

II. Punerea în aplicare a cunoștințelor dobândite

5. Analiza de risc - va cuprinde riscurile de proiect și de produs identificate în scopul testării pe care ați făcut-o, NU IN GENERAL. De asemenea, riscurile vor fi prioritizate și schematizate la alegere fie pe o matrice a riscului fie pe un grafic PRISMA

II. Punerea în aplicare a cunoștințelor dobândite

6. Raportul de defect (bug ticket) - prezentarea va trebui să conțină cel puțin două rapoarte de bug care să conțină următoarele elemente: summary (titlul bug-ului), precondiții (dacă există), pași de reproducere, rezultate așteptate și rezultate actuale

II. Punerea în aplicare a cunoștințelor dobândite

7. Concluzii generale în urma testării - vor conține informații cu privire la numărul de story-uri totale, numărul de story-uri acoperite de teste, numărul total de teste scrise vs executate, numărul de bug-uri identificate împreună cu o explicație legată de severitatea acestora și cât de mult îl afectează pe utilizatorul final.

Vă mulțumesc pentru
atenție!