**PLAN DE TESTARE**

**CUPRINS**

[1. **Introducere** 3](#_Toc141116993)

[1.1. Obiective 3](#_Toc141116994)

[1.2. Membrii Echipei 3](#_Toc141116995)

[2. **Scop** 3](#_Toc141116996)

[3. **Riscuri** 4](#_Toc141116997)

[4. **Abordarea testării (Test approach)** 4](#_Toc141116998)

[4.1. Testarea Automată 5](#_Toc141116999)

[5. **Tool-uri de testare** 5](#_Toc141117000)

[*6.* **Etape / Livrabile** 6](#_Toc141117001)

[6.1. Program 6](#_Toc141117002)

[6.2. Documentele livrate 6](#_Toc141117003)

# Introducere

**Planul de testare** a fost creat pentru a comunica membrilor echipei abordarea de testare. Include obiectivele, domeniul de aplicare, programul, riscurile și abordarea.

## Obiective

Compania “S.C. TEST COMPANY S.R.L.” dorește sa achizitioneze servicii pentru crearea unui site web care trebuie implementat si testat de compania noastra.

Proiectul este implementat de catre departamentul de developement. Echipa de testare este responsabila pentru testarea produsul ce urmează a fi livrat.

Echipa de testare este responsabilă de testarea produsului și se asigura că produsul satisface nevoile clientilor.

## Membrii Echipei

|  |  |
| --- | --- |
| **Resource Name** | **Role** |
| Barbusiu Bogdan | QA lead |
| Denis Tanasă | Tester |

# Scop

Faza inițială va include toate cerințele „must have”. Acestea și orice alte cerințe care sunt incluse trebuie să fie toate testate. La sfârșitul fazei 1, un tester trebuie să fie capabil să:

1. Analizeze documentele si cerintele - **Testerul trebuie să examineze cu atenție documentația și cerințele proiectului pentru a înțelege pe deplin funcționalitatea și comportamentul așteptat al aplicației.**
2. Creeze teste manuale - **Testerul trebuie să dezvolte planuri de testare și să creeze cazuri de testare manuale care să acopere toate cerințele incluse în faza inițială.**
3. Creeze teste automate – Dacă va fi cazul
4. Execute testele manuale si sa completeze statusul (Pass/Fail/Blocked) - **Testerul trebuie să execute cazurile de testare manuale și să înregistreze rezultatele testelor, notând dacă testul a fost trecut cu succes, a eșuat sau a fost blocat (în cazul în care testul nu poate fi executat din diverse motive).**
5. Raporteze defectele gasite - **Dacă sunt identificate erori sau defecte în timpul testării, testerul trebuie să le raporteze într-un sistem de bug tracking, furnizând detalii relevante pentru a facilita identificarea și rezolvarea problemei.**
6. Completeze si sa transmita un raport final - **La finalul fazei 1, testerul trebuie să compileze un raport detaliat care să conțină rezultatele testelor manuale și, dacă este cazul, rezultatele testelor automate. Raportul final ar trebui să includă starea generală a testelor și să evidențieze orice deficiențe sau probleme semnificative găsite în timpul testării.**

# Riscuri

Au fost identificate următoarele riscuri și au fost identificate măsurile adecvate pentru a atenua impactul acestora asupra proiectului. Impactul (sau severitatea) riscului se bazează pe modul în care proiectul ar fi afectat dacă riscul ar fi declanșat. Declanșatorul este momentul sau evenimentul care ar face ca riscul să devină o problemă de rezolvat.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Risk** | **Impact** | **Trigger** | **Mitigation Plan** |
| 1 | Eșecul în a identifica cazurile de testare critice sau prioritare. | High | Neidentificarea și neacoperirea cazurilor de testare critice sau prioritare. | Analiza cerințelor  Definirea criteriilor de acceptare |
| 2 | Modificările aduse funcționalității pot anula testele deja scrise și este posibil să pierdem cazuri de testare deja scrise | High | Pierderea tuturor cazurilor de testare | Exportați datele înainte de orice upgrade |
| 3 | Resursele insuficiente pentru testare (timp, oameni, infrastructură). | High | Echipa de testare va fi sub presiune și nu va putea efectua testele în mod corespunzător. | Planificare realistă  Prioritizarea și paralelizarea testelor |
| 4 | Incompletitudinea testelor și riscul de a trece cu vederea scenarii critice. | High | Anumite scenarii critice pot fi omise din planul de testare. | Definirea matricelor de testare  Reexaminarea și revizuirea testelor |

# Abordarea testării (Test approach)

Proiectul folosește o abordare **“Agile-Kanban”.**

Tot continutul este adaugat in “**To Do”** dupa care, in functie de prioritate se trece in stadiul **“In progress”, “Done”**

Testele se creaza bazate pe **“Definition of done”** atasat fiecarui task

Documentul DoD ar trebui sa contina test case-uri pentru a usura procesul de automatizare

Teste aditionale se pot adauga daca este necesar in timpul activitatii de **“test execution”.**

Testarea exploratorie va juca o mare parte a testării, deoarece echipa nu a folosit niciodată acest tip de instrument și va învăța pe măsură ce înaintează proiectul. Testele pentru funcționalitatea planificată vor fi create și adăugate la TCT pe măsură ce se executa testele.

## Testarea Automată

Toate testele din **“Definition of done”** vor rula automat. (după caz)

*\*\* Testarea automată se referă la procesul de creare și executare automată a testelor pentru a verifica că funcționalitatea și calitatea aplicației sunt conforme cu cerințele definite în "Definition of Done" (Definiția de Finalizare). Acest proces elimină dependența de testele manuale repetitive și permite dezvoltatorilor să își concentreze eforturile pe dezvoltarea de funcționalități noi.*

# Tool-uri de testare

**GitHub** - *project management and bug tracking*

**GitHub** - *test results*

**GitHub oferă următoarele beneficii pentru project management și bug tracking:**

* Colaborare facilă între membrii echipei.
* Monitorizarea progresului proiectului prin probleme și sarcini (issues).
* Înregistrarea și gestionarea erorilor și problemelor (bugs) în mod transparent.
* Controlul versiunilor și istoricul modificărilor pentru codul sursă.
* Integrare cu diverse instrumente și servicii pentru o gestionare mai eficientă a proiectelor.

**Selenium IDE** – *automation*

**Selenium IDE** este o extensie de browser ce permite testarea automatizată a aplicațiilor web, utilizând tehnologia Selenium. Acest instrument ușurează crearea și înregistrarea de teste automate prin intermediul unui interfață grafică simplă și intuitivă. De asemenea, Selenium IDE permite editarea și reexecutarea testelor, generând înregistrări detaliate și ușor de urmărit. Cu ajutorul Selenium IDE, testele pot fi rulate pe mai multe browsere și sisteme de operare, facilitând asigurarea calității aplicației într-un mod eficient și fiabil.

**Google Sheets/Microsoft excel** - *test case management, bug creation, test status and report*

**Microsoft Excel** poate fi folosit eficient pentru gestionarea testelor, crearea rapoartelor de testare și urmărirea stării testelor în cadrul unui proiect software. Iată cum poate fi utilizat Excel pentru aceste sarcini:

* **Test Case Management (Gestionarea cazurilor de testare):**

În Excel, puteți crea tabele pentru a înregistra detaliile fiecărui caz de testare, cum ar fi descrierea, pasii de testare, datele de intrare, așteptările și rezultatele obținute.

Puteți organiza testele în mai multe foi de lucru pentru a separa testele funcționale, testele de performanță, testele de securitate etc.

Utilizarea filtrelor și sortării vă permite să găsiți rapid și să gestionați cazurile de testare.

* **Bug Creation (Crearea de erori/defecte):**

Când identificați o eroare sau un defect în timpul testării, puteți crea o nouă înregistrare într-un tablou separat pentru a urmări detaliile erorii.

Înregistrările pot conține informații precum descrierea erorii, pasii pentru reproducerea acesteia, nivelul de criticitate, starea și responsabilul pentru rezolvare.

* **Test Status (Starea testelor):**

Puteți utiliza formule pentru a calcula automat numărul de teste trecute, eșuate și nedeterminate.

Graficele și diagramele pot oferi o vedere vizuală a stării generale a testelor, precum și tendințe de testare în timp.

* **Test Reports (Rapoarte de testare):**

Excel vă permite să creați rapoarte personalizate prin utilizarea formulelor, tabelelor pivot sau a altor funcționalități avansate pentru a consolida datele din diverse tabele.

Rapoartele pot fi prezentate sub forma de tabele, grafice, diagrame sau alte tipuri de vizualizări pentru a evidenția concluziile și tendințele importante.

# Etape / Livrabile

## Program

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task Name** | **Start** | **Finish** | **Effort** | **Comments** |
| Test Plan | 12/06/2023 | 19/06/2023 |  |  |
| Review cerinte | 20/06/2023 | 23/06/2023 |  |  |
| Test design si estimari | 24/06/2023 | 28/06/2023 |  |  |
| Creare teste | 30/06/2023 | 07/07/2023 |  |  |
| Executie teste | 08/07/2023 | 13/07/2023 |  |  |
| Testare exploratorie | 14/07/2023 | 18/07/2023 |  |  |
| Monitorizare si control | 19/07/2023 | 21/07/2023 |  |  |
| Activitati de incheiere | 22/07/2023 | 24/07/2023 |  |  |

## Documentele livrate

|  |  |  |
| --- | --- | --- |
| **Livrabil** | **Pentru** | **Data** |
| Test Plan | Project Manager; QA lead; Test Team | 19/06/2023 |
| Test Cases | Project Manager; QA Lead | 07/07/2023 |
| Test Results | QA Lead | 13/07/2023 |
| Test Status and bug report | QA Lead | 21/07/2023 |