

STRATEGIA DE TESTARE PENTRU
www.Orange.ro

NUMELE PROIECTULUI: www.Orange.ro

CONTROLUL DOCUMENTELOR

Versiunea	1
Data	14/01/2023
Elaborat de	Redenstein Georgiana

SEMNAREA DOCUMENTULUI

Versiune	Stare	Data aprobată de	Aprobată de	Denumirea postului
1	Final	14/01/2023	Redenstein Georgiana	Tester Software

Cuprins

1. INTRODUCERE	5
2. SCOPUL	5
3. PREZENTARE GENERALĂ A SISTEMULUI.....	6
4. DOMENIUL DE APLICARE A TESTĂRII.....	7
4.1 ÎN DOMENIUL DE APLICARE	7
4.2 ÎN AFARA DOMENIULUI DE APLICARE.....	8
5. ABORDARE A TESTĂRII.....	9
5.1 PRINCIPII ȘI APLICARE	9
5.1.1.1 Principiul.....	9
5.2 ECHIPA - ITERATIVĂ PLANIFICATĂ	11
5.3 [FAZA DE TESTARE 1] - PRIMUL SPRINT	13
5.3.1 Obiectiv.....	13
5.3.2 Domeniul de aplicare	13
5.3.3 Pregătirea testului.....	14
5.3.3.1 Criterii de intrare.....	14
5.3.3.2 Criterii de ieșire.....	15
5.3.4 Executarea testului	15
5.4 [FAZA DE TESTARE 2] - AL DOILEA SPRINT	15
5.4.1 Obiectiv.....	15
5.4.2 Domeniul de aplicare	16
5.4.3 Pregătirea testului.....	16
5.4.3.1 Criterii de intrare.....	17
5.4.3.2 Criterii de ieșire.....	17
5.4.4. Executarea testului	17
6. CERINȚE PRIVIND MEDIILE DE TESTARE	18
7. CERINȚE PRIVIND DATELE DE TESTARE	19
8. INSTRUMENTE ȘI TEHNICI DE TESTARE	20
8.1 GESTIONAREA CERINȚELOR ȘI A CAZURILOR DE UTILIZARE	20
8.2 GESTIONAREA TESTELOR ȘI URMĂRIREA DEFECTELOR.....	22
9. ROLURI ȘI RESPONSABILITĂȚI DE TESTARE	23
10. MANAGEMENTUL TESTELOR	24
11. GESTIONAREA DEFECTELOR	25

11.1 PROCESUL DE GESTIONARE A DEFECTELOR	25
12. PROGRAM DE TESTARE	27

1. INTRODUCERE

Obiectivul proiectului www.Orange.ro este de a testa functionalitatile prin intermediul anumitor instrumente specifice testarii software precum scrierea de user story-uri in Jira, realizarea si executarea test case-urilor in Test CaseLab cat si identificarea defectelor aparute in urma executarii test case-urilor sau prin testare exploratorie, prin care ne dorim sa cunoastem modalitatea prin care acestea functioneaza atat in maniera independenta cat si in interdependenta unele cu altele.

2. SCOPUL

Această strategie de testare va oferi o viziune de nivel înalt asupra modului în care se va realiza testarea pentru proiectul www.Orange.ro. Acest document conține mai multe secțiuni, dar accentul este pus pe abordarea generală a testării pentru diferitele faze de testare care urmează să fie finalizate.

Scopul proiectului nostru este de a vedea cum functioneaza elementele din site in corelatie unele cu altele folosind instrumente specifice de lucru pentru testarea software.

De asemenea s-au aplicat si principiile testarii care atesta daca produsul asteptat de client este conform cerintei tehnice si de business.

3. PREZENTARE GENERALĂ A SISTEMULUI

Sistemul supus testării (SUT) este format din următoarele componente.

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
```

```
<!-- created with sitemap generator https://octopus.do -->
```

```
<url><loc>https://www.orange.ro/</loc><lastmod>2023-01-
```

```
13</lastmod><changefreq>always</changefreq><priority>0.5</priority></url><url><loc>https://www.orange.ro//about</loc><lastmod>2023-01-
```

```
13</lastmod><changefreq>always</changefreq><priority>0.5</priority></url><url><loc>https://www.orange.ro//about/careers</loc><lastmod>2023-01-
```

```
13</lastmod><changefreq>always</changefreq><priority>0.5</priority></url><url><loc>https://www.orange.ro//about/careers/job</loc><lastmod>2023-01-
```

```
13</lastmod><changefreq>always</changefreq><priority>0.5</priority></url><url><loc>https://www.orange.ro//about/clients</loc><lastmod>2023-01-
```

```
13</lastmod><changefreq>always</changefreq><priority>0.5</priority></url><url><loc>https://www.orange.ro//services</loc><lastmod>2023-01-
```

```
13</lastmod><changefreq>always</changefreq><priority>0.5</priority></url><url><loc>https://www.orange.ro//works</loc><lastmod>2023-01-
```

```
13</lastmod><changefreq>always</changefreq><priority>0.5</priority></url><url><loc>https://www.orange.ro//works/case</loc><lastmod>2023-01-
```

```
13</lastmod><changefreq>always</changefreq><priority>0.5</priority></url><url><loc>https://www.orange.ro//contacts</loc><lastmod>2023-01-
```

```
13</lastmod><changefreq>always</changefreq><priority>0.5</priority></url><url><loc>https://www.orange.ro//news</loc><lastmod>2023-01-
```

```
13</lastmod><changefreq>always</changefreq><priority>0.5</priority></url><url><loc>https://www.orange.ro//news</loc><lastmod>2023-01-
```

```
//www.orange.ro//news/contact</loc><lastmod>2023-01-  
13</lastmod><changefreq>always</changefreq><priority>0.5</priority></url><url><loc>https:  
//www.orange.ro//news/contact/email-to-mailchimp</loc><lastmod>2023-01-  
13</lastmod><changefreq>always</changefreq><priority>0.5</priority></url><url><loc>https:  
//www.orange.ro//cookies</loc><lastmod>2023-01-  
13</lastmod><changefreq>always</changefreq><priority>0.5</priority></url><url><loc>https:  
//www.orange.ro//404-error</loc><lastmod>2023-01-  
13</lastmod><changefreq>always</changefreq><priority>0.5</priority></url></urlset></priori  
ty></url></urlset>
```

4. DOMENIUL DE APLICARE A TESTĂRII

După cum arată diagrama din secțiunea 3, domeniul de aplicare al testării www.Orange.ro este de a testa elemente din site care fac parte din scopul proiectului nostru.

4.1 ÎN DOMENIUL DE APLICARE

Tipurile de testare care se află în sfera de cuprindere sunt:

- Testare statică
 - Cerințe
 - Arhitectură
 - Revizuire de cod
- Testare funcțională
- Testare non-funcțională
 - Performanță (încărcare, stres)

- Testarea securității
- Accesibilitate
- Compatibilitatea cu browserul
- Compatibilitatea cu telefoanele mobile
- Testarea acceptării de către utilizator
- Suport UAT
- Validarea defectelor UAT
- Testarea acceptării operaționale
- Implementare
- Teste de regresie (acolo unde este necesar)
 - Principiile testarii
 - Smoke testing
 - Saniy testing
 - Procesul de testare
 -

4.2 ÎN AFARA DOMENIULUI DE APLICARE

Tipurile de testare care sunt în afara domeniului de aplicare sunt:

Testarea acceptării de către utilizator (aceasta se va efectua în mediul de dezvoltare, dar va fi planificată și executată de către client).

Testarea beta fiind testarea pe structura, infrastructura si instrumentele si dispozitivele clientului.

Testarea unitara se realizeaza la nivelul cel mai mic precum clase, interfete, functii si nu este o activitate de testare, vor fi scrise de catre ecipa de dezvoltare aceste teste.

5. ABORDARE A TESTĂRII

Următoarea secțiune descrie abordarea la nivel înalt a testării.

5.1 PRINCIPII ȘI APLICARE

5.1.1.1.1 Principiul

Este important să se înțeleagă riscurile pentru afacere în ceea ce privește orice sistem care este dezvoltat. Unele părți ale sistemului vor fi critice pentru funcționarea afacerii, iar altele nu. Înțelegând profilul de risc, putem ajusta cantitatea și tipurile de teste pe care le realizăm. Acest lucru poate oferi o scară realistă de testare pentru fiecare proiect și, prin urmare, putem susține estimările pe care le prezentăm.

Cele 7 principii ale testării sunt:

- a) Testarea arată întotdeauna prezența defectelor și nu absența lor

Având în vedere acest principiu care demonstrează faptul că prin testare se arată întotdeauna că există defecte și niciodată nu se poate dovedi că nu ar exista defecte, având în vedere că toate produsele din lume au defecte.

Un exemplu în acest sens este defectul pe care l-am identificat în proiectul Orange.ro și anume - resetarea parolei Orange din My Orange, care atestă faptul că este un defect în acest context.

Un alt defect identificat ar putea fi, aranjarea în pagină în aplicația desktop este diferită în aplicația de tip mobil, datorită contextului de dimensiune și aranjare în pagină.

- b) Absența defectelor este un defect

Așa cum este prezentat la punctul a) absența defectelor este un defect, deoarece pentru ca aplicația să funcționeze în parametrii optimi înainte de a fi lansată trebuie

identificate defectele aparute, chiar daca la prima vedere se pare ca nu ar fi defecte, ele cu siguranta sunt prezente.

c) Testarea este dependenta de context

Acest tip de testare demonstreaza faptul intr-un fel se testeaza aplicatiile web, aplicatiile desktop si in altfel se testeaza aplicatiile de mobil.

Un exemplu in acest sens este faptul ca atunci cand deschidem aplicatia Orange in varinata pe desktop imaginea este mai mare, informatiile sunt in prim plan pe pagina principala, avem acces visual imediat asupra aplicatiei.

In acelasi timp aplicatia Orange, in varinta de mobil la nivel de aspect visual, pagina principala este restransa si nu sunt vizibile imediat toate paginile, ci trebuie sa dam click pe o pagina pentru a ajunge la pagina dorita.

d) Testarea exhaustiva este imposibila

Se refera la faptul ca testarea cu toate conditiile, preconditionile si toate combinatiile din lume pe care un client le doreste este imposibila. Pentru testarea aplicatiei Orange am utilizat doar cateva varinate de tipuri de testare si anume testare functionala si nefunctionala, de cum arata aplicatia, cum este aranjata in pagina, cum reactioneaza aplicatia la stress daca este performanta. Cu toate acestea nu am aplicat toate testele care exista deoarece ar fi luat mult timp si ar fi fost imposibil de realizat.

e) Testarea timpurie salveaza timp si bani

Testarea ar trebui să fie efectuată pe documente de cerințe, specificații sau orice alt tip de document, astfel încât, în cazul în care cerințele sunt definite incorect, atunci acesta poate fi reparat imediat, mai degrabă decât fixarea lor în faza de dezvoltare.

Implicarea timpurie in cadrul proiectului, definirea si gasirea defectelor in prima faza vor duce la o reducere a costurilor in ciclul de viata al dezvoltarii softului, care vor putea fi identificate inca din faza de analiza a cerintelor, costul fiind mai mic, decat in fazele ulterioare ale testarii.

f) Paradoxul pesticidelor

Se refera la faptul ca aceleasi combinatii de testare si tehnici, deci utilizarea lor in mod repetat nu vor duce la rezultatul asteptat si la imbunatatirea procesului de testare de la sprint la sprint.

De exemplu daca pentru aplicatia nostra am folosi acelasi tip de testare, am efectua Stres testing, Sanity testing, Smoke testing de fiecare data, acestea nu ar duce la o imbunatatire a procesului de testare.

g) Defectele se grupeaza

Se refera la faptul ca se aplica regula lui Pareto 20% functionalitati si 80 % defecte.

Gruparea defectelor indică faptul că zona predispusă la defecte trebuie testată temeinic în timpul testării de regresie. Cu cat se testeaza o anumita parte din site la inceput pot fi mai multe defecte, dar cu trecerea timpului defectele se vor reduce deoarece se constata ca acestea s-au rezolvat intre timp.

Ca si in cazul aplicatiei Orange defectele identificate, resetarea parolei si defecte de aranjare in pagina in functie de context, desktop sau mobil, are legatura cu functionalitatea unei mici parti din site si textul “Despre comunitate” din aplicatia Orange, si dintr-un ansamblu, restul functionalitatilor vor avea un numar mai mare de defecte

5.2 ECHIPA - ITERATIVĂ PLANIFICATĂ

Testarea este o disciplină de bază în cadrul de dezvoltare al Development Company (TEAM - The Development Company Adaptive Model), cu implicare în fiecare dintre etapele cheie ale dezvoltării, după cum se arată în diagrama de mai jos.

TEAM definește șase dimensiuni, dimensiunile de bază fiind următoarele:: Proces, Echipă și Instrumente și dimensiunile de sprijin: Comunicare, Cultură și Oameni. Fiecare dintre acestea este abordată în mod specific în cadrul disciplinei de testare din cadrul metodologiei.

Există două abordări distincte de dezvoltare software descrise de TEAM:

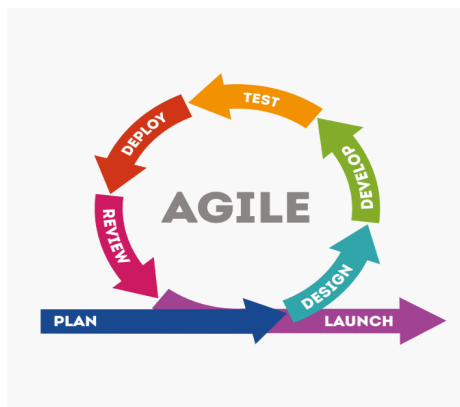
▪ Agile

In contextul de project management agil inseamna flexibil. Cu cat o metodologie de proiect are mai putine reguli cu atat proiectul este mai flexibil, mai agil.

Cea mai mare flexibilitate in proiect se obtine cu SCRUM si eXtreme Programming (XP)

Metodologia Agile are la baza 12 principii importante de schimbare si 4 zone de interes major.

Acestea sunt:



- Colaborare inainte de negocierea contractelor
- Accentual pus pe individ, pe om si nu pe process
- Livrabile mai presus de documentatie mare
- Exemple de metodologii agile: SCRUM, XP, Kanban, AUP, Crystal, DSDM, Spotify ▪

Iterativă planificată

Acest proiect va fi livrat folosind metodologia Planned Iterative.

Principiile Planned Iterative cu impact asupra testării includ:

- Testarea este finalizată de către un grup de testare independent
- Testarea pe parcursul întregului proiect

- Se concentrează pe găsirea defectelor la începutul SDLC.
- Domeniul formal de testare în cadrul fiecărei iterații (verificarea continuă a calității).
- Se concentrează pe testarea mai întâi a arhitecturii.
- Cazuri de testare derivate din cazurile de utilizare.
- Adaptați procesul și documentația la dimensiunea și complexitatea proiectului.

5.3 [FAZA DE TESTARE 1] - PRIMUL SPRINT

5.3.1 Obiectiv

Obiectivul [sprint 1] este de a evidenția crearea user story-urilor într-un sprint de lucru sau iterație, pe bord, pentru a fi testate ulterior prin intermediul instrumentelor software și identificarea eventualelor erori care pot apărea în acest proces.

Această etapă de testare se finalizează pentru a avea o imagine de ansamblu asupra produsului și pentru rezolvarea în timp util aplicând principiile testării, care sunt foarte importante în acest proces și în același timp identificarea defectelor din aplicație și îmbunătățirea generală a testării, înainte ca aplicația să fie lansată, în conformitate cu cerințele tehnice și de business.

5.3.2 Domeniul de aplicare

Domeniul de aplicare al testării pentru [sprint 1] este pagina de contact a aplicației, Informații privind My Orange/My Orange Fix și Accesarea paginii Cos de cumpărături.

- Testarea funcțională răspunde la întrebarea dacă aplicația, serviciul trebuie să funcționeze după cerințele de business și etichetă. În cadrul acestui tip de testare se încadrează și Smoke testing, Sanity testing, Regresion testing, Integration testing, Testare de locație, Testarea de acceptanță.

- Testarea performanței raspunde la intrebarea cat de performanta este aplicatia in comparative cu ceva, care elimina blocajele de performanta. Exemplu :Testarea de performanta va oferi informatii despre cum se comporta produsul in utiliarea zilnica, cu acces concurent din partea utilizatorilor.
- Testarea securității este de a-ti sparge proiectul pentru a vedea ce vulnerabilitati sunt in proiect.

Un exemplu in acest sens se refera la faptul ca un software personalizat nu are securitate adecvata daca o interogare SQL preia parolele reale ale utilizatorilor sai

- Testarea la stres

Se streseaza aplicatia pentru a vedea cum rezista la mai multe click-uri. Ca si in cazul aplicatiei Orange, am stresat sistemul pentru a vedea cum reactioneaza acesta la mai multe click-uri.

5.3.3 Pregătirea testului

Testarea a fost realizata cu ajutorul instrumentelor specifice de lucru si Test Case Lab si Jira.

Am creat 6 user story-uri, 3 in fiecare sprint, folosind Jira, si 10 test case-uri in Test Case Lab. User story-urile le-am creat folosind forma BDD, criterii de acceptanta, reporter si persoana desemnata, in forma completa si estimare.

Pentru test case-uri am folosit modelul specific pentru test case-uri folosind id, titlu, descriere, pasii cu pasul 1 si rezultatul asteptat, 3 pasi pentru fiecare test case.

5.3.3.3.1 Criterii de intrare

Criteriile de intrare se refera la modalitatea prin care se pot crea User Story si executa Test Case-uri, mediul de testare este productia care este pregatit si incarcat cu date pentru a avea cum sa executam testele. Avem user story-uri pentru testare si parola de test, sau sa avem si test plane-ul, aplicatia functioneaza in parametrii optimi, sau puen avea nevoie si de un query dintr-o baza de date. Un exemplu in acest sens daca atunci cand suntem pe un tab sau o pagina deschisa care ne solicita nume de utilizator si parola pot fi criterii de intrare considerate si precondii ale testarii

5.3.3.2 Criterii de ieșire

În etapa de test Design avem în vedere cu va arata template-ul test case-urilor sau infrastructura și instrumentele pe care le vom folosi.

Dupa crearea test case-urilor, a urmat faza de executie a test caseurilor prin care s-a observat daca acestea au fost functionale sau am identificat anumite erori in executia acestora. La rulara testelor aplicatia este functionala, dar la un test case s-a identificat o eroare care a fost identificata ca defect, in urma executarii acestuia.

5.3.4 Executarea testului

În faza de executie a test case-urilor s-a avut în vedere crearea test case-urilor în urma scrierii user story-urilor, conform template-ului pentru test case-uri. Pentru executia test case-urilor am folosit Test Runs din Test Case Lab, am atribuit fiecare test case cu numele meu și așa am procedat pentru fiecare.

Apoi am executat test case-urile dând click pe opțiunile pe care instrumentele de testare ni le-a pus la dispoziție, și anume ‘’esuat’’, ‘’blocat’’, ‘’trecut’’, și ‘’netstat’’. Am ales pentru fiecare test case una dintre opțiuni, cel mai des, opțiunea ‘’trecut’’, și o alta ‘’esuat’’, iar apoi ne arata în partea dreapta a ecranului rezultatul testului, și am adăugat și atasamnete, imagini din site reprezentative pentru fiecare rezultat al testului.

5.4 [FAZA DE TESTARE 2] - AL DOILEA SPRINT

5.4.1 Obiectiv

Obiectivul [Sprint 2] este de a evidenția crearea user story-urilor într-un sprint de lucru sau iteratie, pe bord, pentru a fi testate ulterior prin intermediul instrumentelor software și identificarea eventualelor erori care pot apărea în acest process.

Aceasta etapa de testare se finalizeaza pentru a avea o imagine de ansamblu asupra produsului si pentru rezolvarea in timp util aplicand principiile testarii, care sunt foarte importante in acest process si in acelasi timp identificarea defectelor din aplicatie si imbunatatirea generala a testarii, inainte ca aplicatia sa fie lansata, in conformitate cu cerintele tehnice si de business.

5.4.2 Domeniul de aplicare

Domeniul de aplicare al testării pentru [Sprint 2] este pagina de contact a aplicatiei, Informatii privind Afisarea item-ului Magazin de pe pagina principal, Accesarea pagini dedicate Vezi abonamentele, Caracteristicile item-ului Cauta Magazine.

- Testarea funcțională raspunde la intrebarea ca aplicatia, serviciul trebuie sa functioneze dupa cerintele de business si etnic. In cadrul acestei tip de testare se incadreaza si Smoke testing, Sanity testing, Regresion testing, Integration testing, Testare de locație, Testarea de acceptanta.
- Testarea performanței raspunde la intrebarea cat de performanta este aplicatia in comparative cu ceva, care elimina blocajele de performanta. Testarea de performanta va oferi informatii despre cum se comporta produsul in utilizarea zilnica, cu acces concurent din partea utilizatorilor.
- Testarea securității este de a-ti sparge proiectul pentru a vedea ce vulnerabilitati sunt in proiect. Un software personalizat nu are Securitate adecvata daca o interogare SQL preia parolele reale ale utilizatorilor sai
- Testarea la stres

Se streseaza aplicatia pentru a vedea cum rezista la mai multe click-uri. Ca sin cazul aplicatiei Orange, am stresat sistemul pentru a vedea cum reactioneaza acesta la mai multe click-uri.

5.4.3 Pregătirea testului

Testarea a fost realizata cu ajutorul instrumentelor specifice de lucru si Test Case Lab si Jira.

Am creat 6 user story-uri, 3 in fiecare sprint, folosind Jira, si 10 test case-uri in Test Case Lab. User story-urile le-am creat folosind forma BDD, criterii de acceptanta, reporter si persoana desemnata, in forma completa si estimare.

Pentru test case-uri am folosit modelul specific pentru test case-uri folosind id, titlu, descriere, pasii cu pasul 1 si rezultatul asteptat, 3 pasi pentru fiecare test case.

5.4.3.1 Criterii de intrare

Criteriile de intrare se refera la modalitatea prin care se pot crea User Story si executa Test Case-uri, mediul de testare este productia care este pregatit si incarcat cu date pentru a avea cum sa executam testele. Avem user story-uri pentru testare si parola de test, sau sa avem si test plane-ul. Criterii de intrare pentru sprintul 2 este introducerea orasului pe care dorim sa il cautam si butonul "localizeaza-ma". Cand ne aflam pe pagina dedicate vezi abonamentele, este un buton portocaliu prin care putem selecta si se schimba beneficiile ofertei.

5.4.3.2 Criterii de ieşire

In etapa de test Design avem in vedere cu va arata template-ul test case-urilor sau infrastructura si instrumentele pe care le vom folosi.

Dupa crearea test case-urilor, a urmat faza de executie a test caseurilor prin care s-a observat daca acestea au fost functionale sau am identificat anumite erori in executia acestora. La rularea testelor aplicatia este functionala, dar la un test case s-a identificat o eroare care a fost identificata ca defect, in urma executarii acestuia.

5.4.4. Executarea testului

Asa cum am procedat pentru sprintul 1, in acelasi fel am aplicat procedura si pentru sprintul 2.

In faza de executie a test case-urilor s-a avut in vedere crearea test case-urilor in urma scrierii user story-urilor, conform template-ului pentru test case-uri. Pentru executia test case-urilor am folosit Test Runs din Test Case Lab, am atribuit fiecare test case cu numele meu si asa am procedat pentru fiecare.

Apoi am executat test case-urile dand click pe optiunile pe care instrumentele de testare ni le-a pus la dispozitie, si anume "esuat", "blocat", "trecut", si "netstat". Am ales pentru fiecare test case una dintre optiuni, cel mai des, optiunea "trecut", si o alta "esuat", iar apoi ne arata in

partea dreapta a ecranului rezultatul testului, si am adaugat si atasamnete, imagini din site reprezentative pentru fiecare rezultat al testului.

6. CERINȚE PRIVIND MEDIILE DE TESTARE

Mediul de testare production este mediul real la care au acces clientii sau clientii clientului si este cu acces restrictionat. În mediul de producție, produsul a fost livrat și trebuie să funcționeze perfect.

Mediul de testare este production

Laptop model : Acer Z5we1 Aspire E1 Laptop, Screen Size: 15.6 Inch, Memorie RAM: 4 GB

Compomente hardware: Intel inside CORE i3

SSD: 16 GB

Tastatura, mouse-ul, microfonul și camera video digitală

Pe lângă software-ul specific clientului, este necesar următorul software.

- Windows 10 și versiunile superioare
- Office 2019 și versiunile superioare
- MS Office 2019, etc. MS Excel, MS Word, WordPerfect
- PaintBrush și Paint
- Mozilla Firefox, Internet Explorer, Google Chrome

7. CERINȚE PRIVIND DATELE DE TESTARE

Criteriile de intrare se refera la modalitatea prin care se pot creea User Story si executa Test Case-uri, mediul de testare este productia care este pregatit si incarcat cu date pentru a avea cum sa executam testele., sau Avem user story-uri pentru testare si parola de test sa avem si test plane-ul.

Criteriile de intrare pe care le putem lua in vedere sunt introducerea orasului pe care dorim sa il cautam si butonul ‘‘localizeaza-ma’’. Cand ne aflam pe pagina dedicata vezi abonamentele, este un buton portocaliu prin care putem selecta si se schimba beneficiile ofertei.

Sa ne asiguram ca aplicatia este functionala la mai multe click-uri, in acest fel se streseaza sistemul pentru a vedea cum reactioneaza aplicatia. De asemenea alte criterii importante sunt testele de performanta pentru a ne da seama cum functioneaza aplicatia, identificarea defectelor care pot sa apara si rezolvarea acestora in timp util. Datele de referinta sunt crearea de user story-uri in acest proiect.

Aplicarea testelor de regresie, smoke testing, sanity testing, teste de acceptanta ale utilizatorului care indica la nivel de utilizator cum arata aplicatia, anumite query dintr-o baza de date, sunt folosite pentru a testa anumite elemente din site si a vedea cum se comporta acestea in diferite faze de dezvoltare in ciclul de viata al acestora, si daca dau randamentul asteptat initial, de catre client, in functie de cerintele tehnice si de business.

8. INSTRUMENTE ȘI TEHNICI DE TESTARE

8.1 GESTIONAREA CERINȚELOR ȘI A CAZURILOR DE UTILIZARE

Instrumentele din cadrul disciplinei de testare se află într-un proces permanent de evaluare și personalizare, astfel încât să poată oferi cea mai bună soluție în contextul potrivit; cu toate acestea, instrumentele descrise în secțiunile următoare sunt propuse pentru a fi utilizate în acest proiect.

Ciclul de viață al unui defect reprezintă timpul și procesul efectiv de când un defect este deschis până este închis. Uneori poate fi foarte simplu, îl dăm la dezvoltatori, îl deschis, resetat, confirmat și închis, însă uneori este mai complex.

Defectul este deschis și va fi analizat de către o persoană responsabilă de proiect fiind testerii QA Dana și Alex, de obicei o persoană care are competențe tehnice care poate să analizeze orice defect din proiect.

Scopul acestei analize este de a tria defectul și în acest fel va trece prin următoarea zonă de triaj.

Trierea bug-ului este structurată în următoarele componente, adică defectele de identificare vor trece prin următoarea zonă de triaj: deschis, respins dacă nu este valid, în scop, statusul de amanat, duplicat, nou, în progress, pregătit pentru testare, testare în progress care poate fi validă sau nu.

În primul rând verificăm dacă defectul este valid sau nu, iar de vom constata că defectul pe care noi îl avem în vedere nu este defect. Un exemplu pentru aplicația Orange, un caz care nu se încadrează într-un defect ar putea fi Orange Money din pagina principală, deoarece nu putem considera că fiind defect, scrierea în limba engleză a unei asocieri de cuvinte scrise în engleză, adică defectul va avea statusul **respins**. Pe noi ne interesează un defect valid și anume resetarea parolei Orange din My Orange, dar de aceasta dată va fi respins cu statusul de **amanat**. Defectul

este valid, este in scop, asadar vom trece la urmatorul triaj. In aceasta etapa verificam daca defectul este **duplicat**.

Echipa de dezvoltare (programatorii) si anume Andreea si va creea un **nou** defect si il va muta **in progres**, in care se va lucra la acest defect resetarea parolei Orange din My Orange, apoi fixeaza defectul, moment in care va **fi pregatit pentru testare**.

Testerii Mirel, Dana si Alex vor lua defectul in testare cu statusul **testare in progres** in care sunt 2 variante in care statusul va fi in regula, va trece testarea, prin urmare defectul va fi inchis, defectul a fost confirmat nu se mai poate reproduce, problema este rezolvata pentru acest caz.

In cazul in care la confirmarea defectului, acesta a fie inca acolo si se considera ca nu este in regula. nu putem considera ca fiind defect, scrierea in limba engleza a unei asocieri de cuvinte scrise in engleza, asadar defectul va avea statusul **respins**. Ne intereseaza un defect valid si anume resetarea parolei Orange din My Orange, dar de aceasta dat va fi respins cu statusul de **amanat**.. In aceasta etapa verificam daca defectul este **duplicat**.

Echipa de dezvoltare (Andreea) va creea un **nou** defect si il va muta **in progres**, in care se va lucra la acest defect resetarea parolei Orange din My Orange, apoi fixeaza defectul, moment in care va **fi pregatit pentru testare**.

Testerii vor lua defectul in testare cu statusul **testare in progres** in care sunt 2 variante in care statusul va fi in regula, va trece testarea, prin urmare defectul va fi inchis, defectul a fost confirmat nu se mai poate reproduce, problema este rezolvata pentru acest caz.

Dar exista posibilitatea ca la confirmarea defectului, acesta a fie inca acolo si se considera ca nu este in regula.

8.2 GESTIONAREA TESTELOR ȘI URMĂRIREA DEFECTELOR

Avem acest ciclu de viață al defectelor și în momentul când identificăm bug-ul vom avea statusurile specifice în funcție de eroarea care va fi afișată, în cazul nostru la resetarea parolei avem statusul 501 sau diferite mesaje informaționale 103, 107.

Avem severitate mare și prioritate mică, un exemplu în acest sens este în momentul în care ne logăm și ne delogăm din aplicația Orange, ca urmare a acestui fapt aplicația nu rezistă acestei solicitări, adică se aplică principiul testelor de stress.

Severitatea este mare în acest caz deoarece, aplicația s-a dus, nu mai funcționează în parametrii optimi, adică este un impact etnic ridicat.

În schimb prioritatea de business este foarte mică dat fiind faptul că business-ul se va întreba dacă un procent mare de utilizatori vor proceda în acest fel, prin urmare prioritatea de business este foarte mică. În acest caz vom avea severitate 1 și prioritate 3.

9. ROLURI ȘI RESPONSABILITĂȚI DE TESTARE

Tabelul următor prezintă rolurile de testare pentru proiect, împreună cu persoanele implicate în efortul de testare.

Activitate	Responsabilitate/Proprietate	Nume
Crearea planului de testare	[Manager de testare]	Alex Iulian
Crearea planului fazei de testare	[Test Lead]	Cristina
Managementul testelor	[Managerul de testare al companiei de dezvoltare și liderul de testare a sistemului]	Elvira
[Analiza și proiectarea testelor]	[Inginerii de testare ai companiei de dezvoltare]	Laura
[Pregătirea, execuția și rezultatele testelor]	Inginerii de testare ai companiei de dezvoltare]	Mirel, Dana
[Depunerea defectelor de testare]	[Inginerii de testare ai companiei de dezvoltare]	Georgiana
[Rapoarte de sinteză a testelor]	[Managerul de testare al companiei de dezvoltare și liderul de testare]	Mario
[Raportarea finalizării testelor]	[Managerul de testare și conducătorul de testare al companiei de dezvoltare]	Alex Cotan
[Implementarea mediului de testare]		Alex P

10. MANAGEMENTUL TESTELOR

Responsabilitatea generală pentru proiectul de testare va reveni managerului de testare al societății de dezvoltare Orange.ro. Gestionarea zilnică a testelor va fi responsabilitatea responsabilului de testare al societății de dezvoltare Orange.ro.

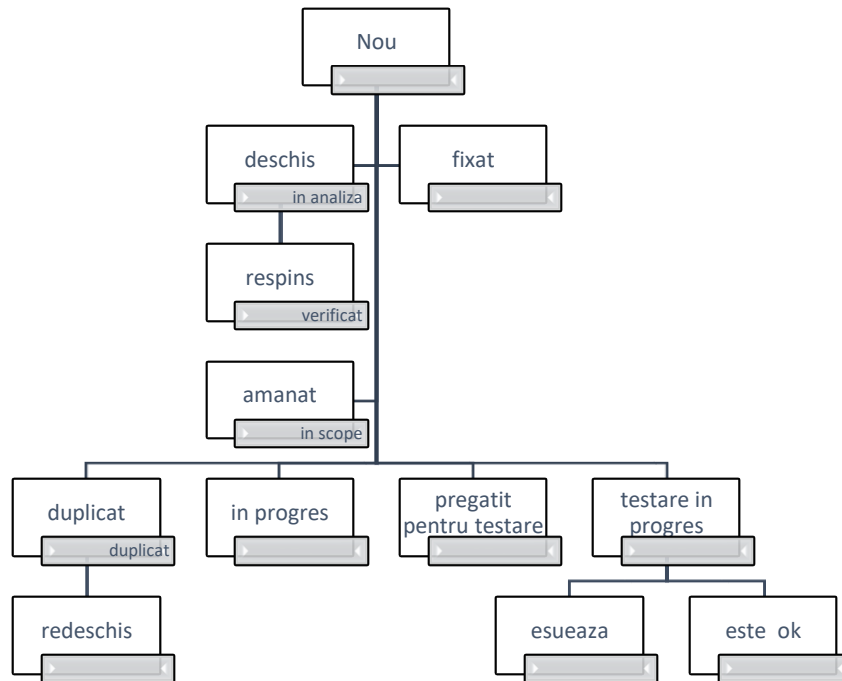
Unul dintre rolurile cheie ale echipei de gestionare a testelor este de a lucra îndeaproape cu managerul de proiect, cu liderul tehnic și cu analistul principal de afaceri pentru a se asigura că testarea este integrată și executată în mod eficient. În acest sens, echipa de gestionare a testelor va participa la reuniunea săptămânală a managerilor de proiect cu șeful echipei tehnice și cu analistul principal de afaceri, în cadrul căreia toate acțiunile, problemele și riscurile sunt analizate și progresează.

În plus, în fiecare dimineață vor avea loc întâlniri zilnice de actualizare a testelor, în cadrul cărora fiecare tester va trebui să își prezinte progresul (ce a făcut ieri, ce intenționează să facă astăzi și orice probleme care îl împiedică să progreseze).

La sfârșitul fiecărei faze de testare, se va organiza o reuniune a Comitetului de evaluare a testării. Consiliul de evaluare a testelor (TRB) este un consiliu de evaluare care lucrează în cadrul unui proiect, alcătuit din principalele părți interesate și factori de decizie pentru proiect. Acesta este conceput ca un mecanism de revizuire a gradului de pregătire pentru testare, a stării de testare și de luare a deciziilor strategice formale ale proiectului privind testarea și gestionarea portofoliului.

Echipa de proiect (managementul testelor, managementul proiectului, analiza de afaceri și dezvoltarea) va participa, de asemenea, la o reuniune zilnică a consiliului de revizuire a defectelor. Aceasta va avea loc la sfârșitul fiecărei zile, în timp ce execuția testelor este în desfășurare. Consiliul de revizuire a defectelor (Defect Review Board - DRB) va evalua erorile și problemele care apar în timpul testării. Scopul principal al DRB este de a preveni ca testarea să fie distrasă de probleme cu prioritate scăzută care le scot din calea critică. DRB va evalua cu atenție implicațiile funcționale, tehnice și de date, precum și impactul asupra afacerii.

11. GESTIONAREA DEFECTELOR



11.1 PROCESUL DE GESTIONARE A DEFECTELOR

- Subiectul trebuie să fie concis, având o cantitate maximă de informații cu un număr minim de cuvinte.
- Comentariile privind defectele trebuie să conțină repro, rezultate așteptate și rezultate reale.
- O bună practică pentru un Tester este să încerce întotdeauna să reducă pașii de repro la un număr minim de pași; acest lucru este extrem de util pentru programatorul care trebuie să găsească defectul.
- Amintiți-vă că singura persoană care poate închide un defect este cea care l-a deschis. Oricine îl poate rezolva, dar numai persoana care a detectat defectul poate fi cu adevărat sigură că ceea ce a găsit este rezolvat.

- Nu există reproș înseamnă că nimeni nu ar putea reproduce vreodată defectul. Programatorii folosesc adesea acest lucru atunci când din raportul de defect lipsesc pașii de repro.
- Fiecare build al software-ului care este dat la testare trebuie să aibă un număr de identificare a build-ului cu număr de revizuire. Un raport de defecte fixate pentru construcția respectivă ar trebui să fie pus la dispoziția testeristilor printr-un jurnal de modificări furnizat de PM, în care sunt prezentate mediul, numărul construcției și lista de modificări. În acest caz, testerul este informat despre neconformitățile care sunt rezolvate în această versiune și nu trebuie să testeze din nou defectul pe o versiune a software-ului în care nici măcar nu trebuia să fie rezolvat.
- Dacă vi se pare că reproducerea unui bug este foarte greu de explicat, având mai mulți pași, interacțiunea cu mai multe sisteme oferiți mai multe dovezi vizuale prin încărcarea de imagini și videoclipuri.
- Nu este o practică bună să țineți evidența neconformităților atât în Bugzilla cât și în foi de calcul sau documente word. Totuși, dacă trebuie, atunci asigurați-vă că includeți ID-urile neconformităților din Bugzilla și folosiți întotdeauna Bugzilla ca sursă de date atunci când creați un document word sau o foaie de calcul.
- Orice solicitare de modificare va avea mențiunea "Enhancement".

12. PROGRAM DE TESTARE

Nmarul de referinta	Etapă	Etapă de referință a proiectului	Data
TM101	Etapă de pregătire a testului	Documentul planului de testare a fazei finalizat	16/01/2023
TM102	Etapă de pregătire a testului	Analiza de testare finalizată pe baza cerințelor detaliate și a documentației tehnice	17/01/2023
TM103	Etapă de pregătire a testului	Condițiile/cazurile/scrierile de testare completate și semnate.	7/01/2023
TM104	Etapă de execuție a testului	Execuția scripturilor de testare a fost finalizată.	9/01/2023
TM201	Etapă de pregătire a testului	Documentul planului de testare a fazei finalizat	16/01/2023
TM202	Etapă de pregătire a testului	Analiza de testare finalizată pe baza cerințelor detaliate și a documentației tehnice	17/01/2023
TM203	Etapă de pregătire a testului	Condițiile/cazurile/scrierile de testare completate și semnate.	7/01/2023
TM204	Etapă de execuție a testului	Execuția scripturilor de testare a fost finalizată.	9/01/2023

13. DOCUMENTE DE REFERINȚĂ

Tabelul următor identifică documentația utilizată pentru elaborarea prezentului plan de testare:

#	Document	Autorul	Descriere
1	Planul de testare	[Manager de testare] Dana	Acest document oferă informații cu privire la testele specifice care vor fi finalizate în cadrul proiectului.
4	Documentații de proiect	[Manager de proiect] Ana Maria	Acest document oferă informații cu privire la Proiect în ansamblu