|  |  |  |
| --- | --- | --- |
|  | **TESTUL Nr.2** |  |
| 1 | Enumerati avantajele si dezavantajele modelului ciclului de viata Cascada (Waterfall) a produsului software.   |  |  | | --- | --- | | ***Avantaje*** | ***Dezavantaje*** | | 1. Simplu și ușor de înțeles și de utilizat | 1. Niciun software de lucru nu este produs până in etapele de sfarsit ciclului de viață. | | 1. Ușor de gestionat datorită rigidității modelului. Fiecare fază are rezultate specifice și un proces de revizuire. | 1. Niveluri ridicate de risc și incertitudine. | | 1. Etape clar definite. | 1. Este dificil să se măsoare progresele înregistrate în cadrul etapelor. | | 1. Funcționează bine pentru proiecte mai mici, unde cerințele sunt simple foarte usor înțelese. | 1. Nu este un model bun pentru proiecte complexe și orientate pe obiecte. | | 1. Repere/ etape bine înțelese. | 1. Este complicat adapta cerințelor în schimbare. | |  |  | |  |  | | 4p. |
| 2. | Care este diferenta dintre Severitatea (Severity) si Prioritatea (Priority) defectului? Da-ti exemple.   |  |  | | --- | --- | | ***Priority*** | ***Severity*** | | Este definit ordinea în care dezvoltatorul ar trebui să rezolve un defect | Este definită ca gradul de impact pe care un defect îl are asupra funcționării produsului | | Este clasificată în trei tipuri:   * Low * Medium * High | Este clasificată în cinci tipuri:   * Critical * Major * Moderate * Minor * Cosmetic | | Indică cât de curând trebuie remediată eroarea | Indică gravitatea defectului asupra funcționalității produsului | | Prioritatea defectelor este decisă în consultare cu managerul / clientul | Persoana ce se ocupa de QA determină gradul de severitate al defectului | | Statutul de prioritate se bazează pe cerințele clienților | Starea de severitate se bazează pe aspectul tehnic al produsului | | 4p. |
| 3 | Tehnici de testare. White Box, Grey Box, Black Box – comparați-le   |  |  |  | | --- | --- | --- | | ***White Box*** | ***Grey Box*** | ***Black Box*** | | Cunoașterea structurii interne de lucru (codificarea software-ului) este necesară în mod necesar pentru acest tip de testare. | Este necesară cunoașterea parțială a structurii interne de lucru. | Cunoașterea structurii interne de lucru (cod) nu este necesară pentru acest tip de testare. Numai GUI (Graphical User Interface) este necesar pentru cazurile de testare. | | Se efectuează prin verificarea limitelor sistemului și a domeniilor de date inerente software-ului, deoarece nu lipsesc cunoștințele interne de codificare. | Dacă testerul are cunoștințe de codificare, atunci se procedează prin validarea domeniilor de date și a limitelor sistemului intern al software-ului. | Abordarea față de testare include *trial techniques* și *error* *guessing method* deoarece testerul nu are nevoie de cunoștințe de cod interne a software-ului. | | Spațiul de testare a tabelelor pentru intrări (intrări care urmează să fie utilizate pentru crearea cazurilor de testare) este mai mic în comparație cu *Black Box Testing*. | Spațiul de testare a tabelelor pentru intrări (intrări care trebuie utilizate pentru crearea cazurilor de testare) este mai mic decât la *White Box & Black Box Testing*. | Spațiul de testare a tabelelor pentru intrări (intrări care trebuie utilizate pentru crearea cazurilor de testare) este destul de imens și cel mai mare dintre toate tipurile de testare (White/ Grey Box Box Testing). | | Durează mult timp pentru a proiecta cazuri de testare din cauza codului lung. | Proiectarea cazurilor de testare se poate face într-o perioadă scurtă de timp. | Consumul de timp în *Black Box Testing* depinde de disponibilitatea specificațiilor funcționale. | | Tester-ul și Developer-ul fac parte in procesul de testare. | Tester-ul, Developer-ul și utilizatorul final poate face parte in procesul de testare. | Tester-ul, Developer-ul și utilizatorul final poate face parte in procesul de testare. | | Rezistența și securitatea împotriva atacurilor de virusi (*viral attacks*) nu sunt acoperite in cadrul acestui tip de test. | Rezistența și securitatea împotriva atacurilor de virusi (*viral attacks*) nu sunt acoperite in cadrul acestui tip de test. | Rezistența și securitatea împotriva atacurilor de virusi (*viral attacks*) sunt acoperite in cadrul acestui tip de test. | | Consuma/ Necesita cel mai mult timp pentru a fi efectuat | Consuma/ Necesita mai putin timp ca *White Box Testing* | Consuma/ Necesita cel mai putin timp pentru a fi efectuat | | 7p. |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 4. | Utilizatorul trebuie sa aibă posibilitatea de a căuta zboruri pe pagina Homepage>Flightschedule>Flight search. Rezultatul căutării va fi afișat după completarea datelor pentru câmpuri: Destination, Departure, Arrival.    In timpul executării testului, ați identificat un defect: rezultatele căutării arată datele zborurilor din trecut. Elaborați raportul despre acest defect (*defect report*), specificând atributele de baza pentru descrierea defectului.   |  |  | | --- | --- | | ***ID*** | A404 | | ***PROJECT*** | Flight America – Flight scheduling | | ***PRODUCT*** | International Flight Scheduler | | ***RELEASE VERSION*** | 1.4.0a | | ***MODULE*** | Homepage>Flightschedule>Flight | | ***DETECTED BUILD VERSIION*** | 1.4.0.3a | | ***SUMMARY*** | As a result it was displaying results dated in past, not in the time range indicated in the fields | | ***DESCRIPTION*** | While running several defect tests on finding flight schedules completing the *Destination, Departure, Arrival,* with different data, multiple results were displayed dated in past, not in the time range indicated (Departure - Arrival). | | ***STEPS TO REPLICATE*** | 1.1 ”Choose destination” → Moscow  1.2 ”Departure” →13:30 30.11.2021  1.3 ”Arrival” → 21:30 30.11.2021  2.1 ”Choose destination” → London  2.2 ”Departure” →11:30 30.11.2021  2.3 ”Arrival ” → 00:30 1.12.2021  3.1 ”Choose destination” → Zurich  3.2 ”Departure” →15:00 30.11.2021  3.3 ”Choose destination” → 20:30 30.11.2021 | | ***ACTUAL RESULT*** | *To: Moscow 17:30 28.11.2021 22:30 28.11.2021*  *To: London 11:30 28.11.2021 00:30 29.11.2021*  *To: Zurich 15:30 28.11.2021 21:00 28.11.2021* | | ***EXPECTED RESULT*** | *To: Moscow 13:30 30.11.2021 21:30 23.11.2021*  *To: London 11:30 30.11.2021 00:30 01.12.2021*  *To: Zurich 15:00 30.11.2021 20:30 30.11.2021* | | ***ATTACHMENT*** | *[SCREENSHOT]\* [SCREENSHOT]* | | ***REMARKS*** | The problem seems to be in the JS code which is responsible of filtering the flight schedules | | ***DEFECT PROBABILITY*** | *HIGH* | | ***DEFECT SEVERITY*** | *MAJOR* | | ***DEFECT PRIORITY*** | *HIGH* | | ***REPORTED BY*** | Fernanda BARRERO | | ***ASSIGNED TO*** | Catalin PAVLOVSCHI | | ***STATUS*** | ASSIGNED | | ***FIXED BUILD VERSION*** | 1.4.0.5a | |  |  |   \**[TAG\_NAME]* – este un tag pt substituire a imaginilor, semnaturilor, logorilor in caz ca acestea nu pot fi recreate | 10p. |

