
Scenarii de testare

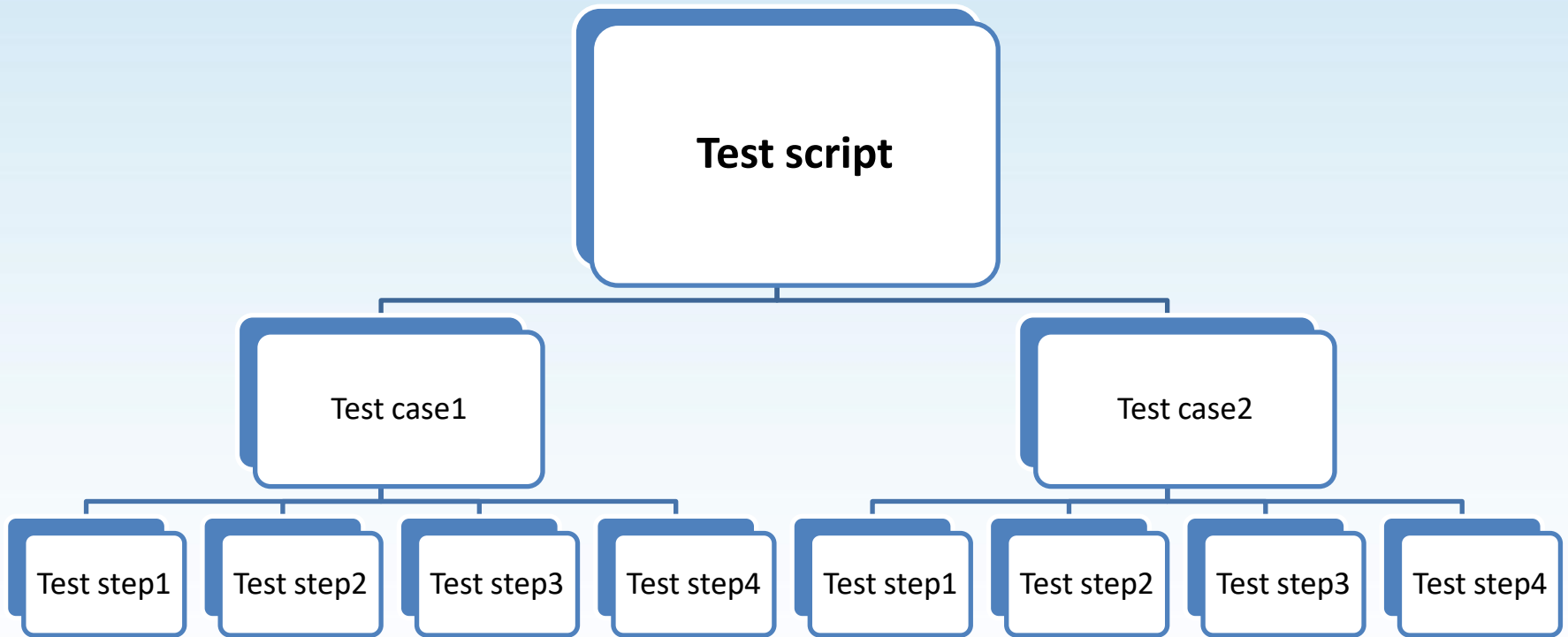


Scenariu de testare?

- Este un set de instructiuni ce urmeaza sa fie efectuate pentru a verifica functionalitatea unei parti a produsului.
- **Scenariul de testare** – este o activitate ce utilizeaza scenarii, ce ajuta testerii la implementarea testelor.

Scenariu de testare:

Structura



Caz de testare

Testing Internship 2019 Winter Session / ENDS002-3368

US001_TS01_TC002 User group displayed in "User Group" list with certain details

Edit Comment To Do Blocked Clone More Actions Execute

Details

Type: Test
Priority: High
Affects Version/s: None
Component/s: Frontend
Labels: UI User_Group
Sprint: NBC_2019 Sprint 2

Status: **PASS**
Resolution: Unresolved
Fix Version/s: Build 3

Description

Summary:
Verify that user group is displayed in "User Groups" list with the following details:

- User Group Name
- Action

Preconditions:
The Open Cart website homepage is accessed.
Logged as an administrator.
There are some user group records in the system.

Test Details

Freeze Test Step ID and Test Steps

Add Steps Columns

-	Test Step	Expected Result	Attachments	Actions
::	1 Click "Settings" option from "Navigation" menu.	"Settings" sub-menu is displayed.	0 attached +	🔒 0
::	2 Click "Users" option from "Settings" sub-menu.	"Users" sub-menu is displayed.	0 attached +	🔒 0
::	3 Click "User Groups" option from "Users" sub-menu.	"User Groups" list is displayed in the central part of the page with the following details: <ul style="list-style-type: none">User Group NameAction	0 attached +	🔒 0

Erori???



Caracteristicile statistice ale erorilor pot servi drept ghid pentru dezvoltatori în distribuirea eforturilor de creare a programelor. În plus, caracteristicile erorilor din procesul de proiectare a programului ajută la:

- 1) evaluarea stării reale a proiectului, planificarea intensității și duratei muncii până la finalizarea acestuia;
- 2) calcularea eficacității necesare a mijloacelor de protecție operațională împotriva erorilor primare nedetectate;
- 3) evaluarea resursele informatice necesare în termeni de memorie și performanță, luând în considerare costul eliminării erorilor;
- 4) Efectuarea cercetării și alegerii adecvate a indicatorilor de complexitate a componentelor și tehnicii de calcul, precum și a altor indicatori de calitate.

Analiza erorilor primare din programe se efectuează la două niveluri:

diferențial - luând în considerare tipurile de erori, complexitatea și gradul de automatizare a detectării lor, costul ajustărilor și etapele celei mai probabile eliminării

generalizate - în funcție de caracteristicile sumare ale detectării lor, în funcție de durata dezvoltării, funcționării și întreținerii pachetului software.

Erorile tehnologice din documentație și programele de remediere din memoria tehnicii de calcul reprezintă 5 ... 10% din numărul total de erori detectate în timpul testării. Majoritatea erorilor tehnologice sunt detectate automat prin metode standardizate.

Erorile software după număr și tipuri sunt determinate în primul rând de gradul de automatizare a programării și de profunzimea testării programelor/aplicațiilor. Numărul de erori software depinde de calificările dezvoltatorilor, de volumul total al pachetului software, de profunzimea interacțiunii logice și informaționale a modulelor și de o serie de alți factori. Când se dezvoltă programe complexe, erorile de program pot fi clasificate în funcție de tipurile de operații utilizate în următoarele grupuri: erori de tipuri de operații; erori variabile; erori de control și cicluri ș.a.

În complexul de programe și sisteme de referință, sunt utilizate diferite metode pentru a automatiza detectarea erorilor. La etapele inițiale de dezvoltare și de testare a modulelor, erorile de program reprezintă aproximativ 1/3 din toate erorile. Erorile în utilizarea operațiilor în etapele inițiale a dezvoltatorului de program ajung la 14% și apoi scad rapid pe măsură ce programatorii își îmbunătățesc abilitățile.

Erorile algoritmice sunt mult mai dificil de detectat prin metode de control automatizate standardizate decât tipurile anterioare de erori. Erorile algoritmice ar trebui să includă, în primul rând, erorile cauzate de formularea incorectă a sarcinilor funcționale, atunci când specificațiile nu stipulează pe deplin toate condițiile necesare pentru a obține rezultatul corect. Aceste condiții sunt formate și rafinate în mare măsură în procesul de testare și identificare a erorilor în rezultatele funcționării programelor. Erorile cauzate de luarea în considerare incompletă a tuturor condițiilor pentru rezolvarea problemelor sunt cele mai frecvente în acest grup și reprezintă până la 70% din toate erorile algoritmice sau aproximativ 30% din numărul total de erori la etapele inițiale de proiectare.

La erorile algoritmice ar trebui să se alătore și erori în legăturile dintre module și grupurile funcționale de programe. Ele pot fi clasificate ca erori de setare incorectă a problemei. Erorile algoritmice se manifestă în contabilizarea incompletă a intervalelor de variație a variabilelor, în evaluarea incorectă a acurateții valorilor utilizate și obținute, în contabilizarea incorectă a relației dintre diferite variabile, în reprezentarea inadecvată a condițiilor formalizate pentru rezolvarea unei probleme în specificații sau schemele care urmează să fie programate etc. Aceste circumstanțe sunt motivul faptului că pentru a corecta fiecare eroare algoritmică este necesar să se schimbe mult mai mult decât în cazul erorilor software.

O parte specială a erorilor algoritmice este alcătuită din calcule greșite în utilizarea resurselor disponibile. Dezvoltarea simultană a mai multor module de către diferiți specialiști face dificilă alocarea optimă a resurselor informatice limitate pentru toate sarcinile, deoarece nu există date fiabile despre resursele necesare pentru rezolvarea fiecăruia dintre ele. Ca rezultat, există fie o sub-utilizare, fie (în majoritatea covârșitoare a cazurilor) o penurie a unor resurse computerizate pentru rezolvarea problemelor în versiunea originală. Cele mai mari erori de calcul apar de obicei la evaluarea timpului de implementare a diferitelor grupuri de programe și la distribuirea performanței sistemului

Erorile de sistem din pachetele software complexe sunt determinate în primul rând de informații incomplete despre procesele reale care apar în surse și receptori de informații. În plus, aceste procese depind adesea de algoritmi înșiși și, prin urmare, nu pot fi suficient definite și descrise în prealabil fără a studia funcționarea complexului de aplicații software în interacțiune cu mediul extern. La etapele inițiale de proiectare, nu este întotdeauna posibil să se formuleze cu acuratețe sarcina țintă a întregului sistem, precum și sarcinile țintă ale principalelor grupuri de programe, iar aceste sarcini sunt clarificate în timpul procesului de proiectare. În conformitate cu aceasta, sarcinile tehnice sau specificațiile pentru proiecte individuale sunt clarificate și concretizate și sunt identificate abateri de la sarcina specificată, care pot fi calificate drept erori de sistem.

În timpul funcționării, erorile de sistem sunt predominante (aproximativ 80% din toate erorile). De asemenea, trebuie remarcat faptul că există un număr mare corectări pentru fiecare eroare.

Scăderea erorilor în complexul de programe și intensitatea detectării lor nu este nelimitată. După o anumită perioadă de testare, intensitatea detectării erorilor în timpul celei mai active testări scade atât de mult, încât echipa de dezvoltare se încadrează într-o zonă de insensibilitate la erori și eșecuri. Cu o astfel de rată de eșec, este dificil să se prevadă timpul necesar pentru a detecta următoarea eroare. Se creează o idee că nu există deloc erori, cu privire la imposibilitatea și lipsa scopului de a le găsi, prin urmare, eforturile de testare sunt reduse și intensitatea detectării erorilor este redusă în continuare. Această rată limitativă de detectare a defecțiunilor corespunde timpului de funcționare pentru eroarea detectată, la care se oprește îmbunătățirea caracteristicilor aplicației complexe în etapele de testare.

Eșec (eng. failure) - orice deviere a comportamentului observat de la cel specificat/așteptat;

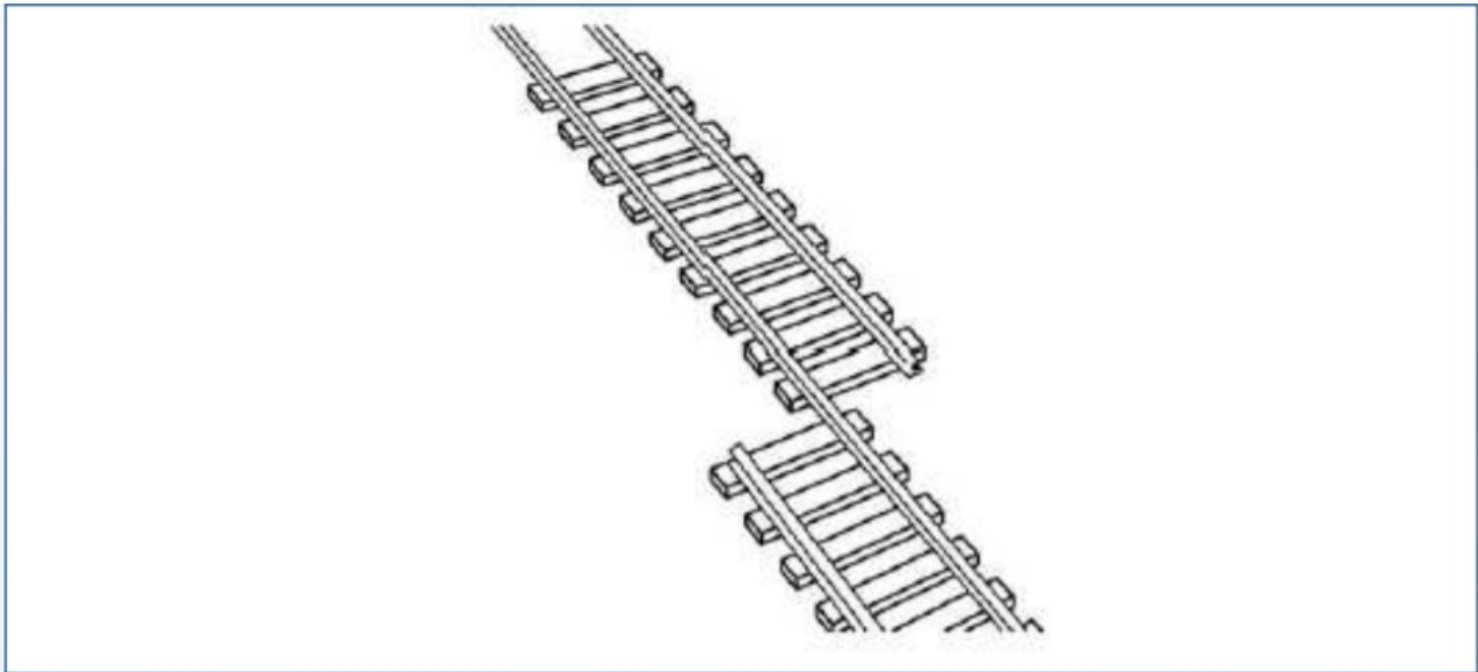
Eroare/stare de eroare (eng. erroneous state) - orice stare a sistemului în care procesările ulterioare ar conduce la eșec;

Defect (eng. fault /defect /bug) - cauza mecanică sau algoritmică a unei stări de eroare.

- Tehnici de creștere a fiabilității sistemelor soft

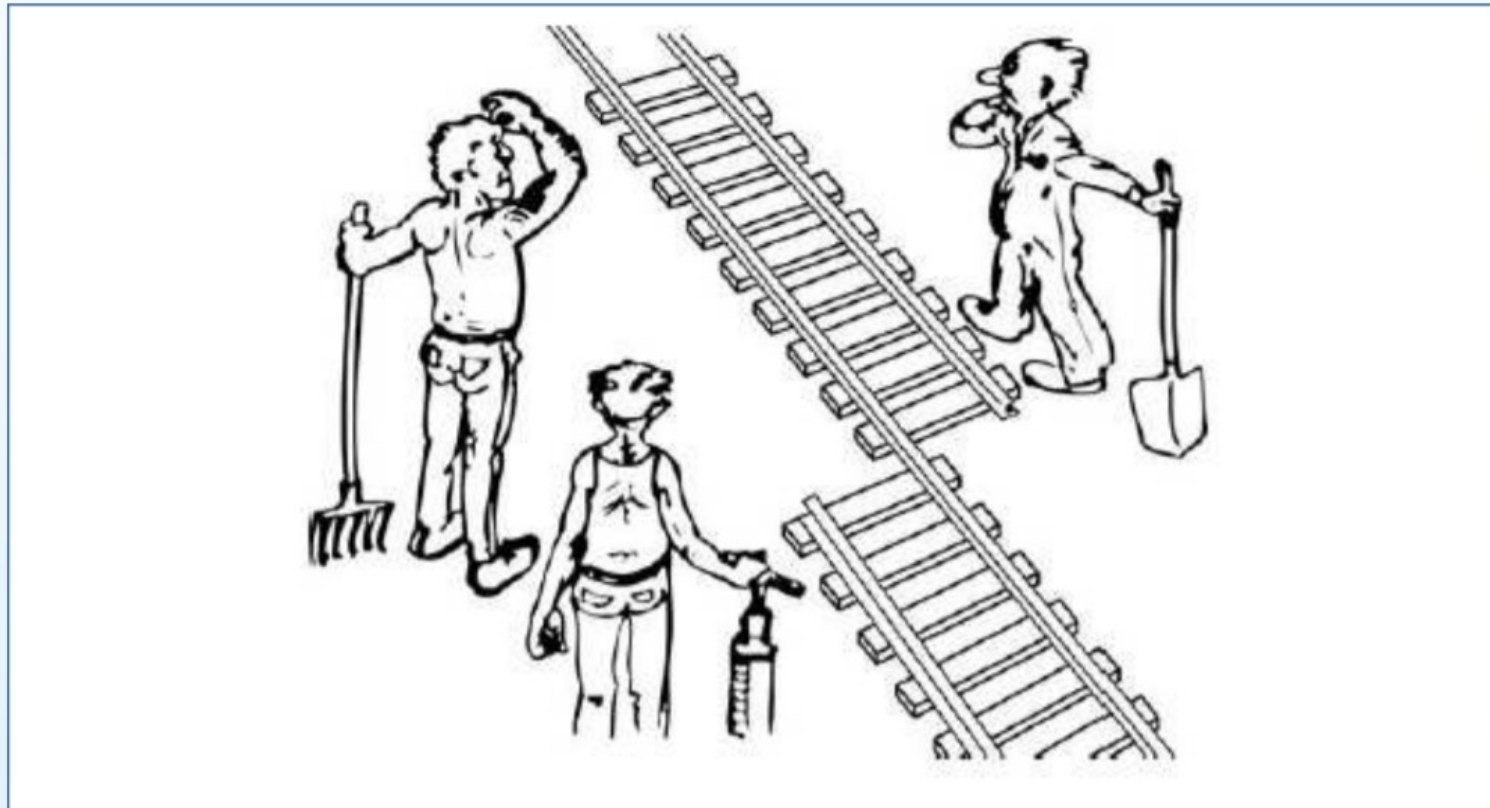
- *Tehnici privind evitarea defectelor (eng. fault avoidance techniques)*
 - identifică posibilele defecte la nivel static (fără o execuție a modelelor/codului) și încearcă să prevină introducerea acestora în sistem
 - ex.: metodologii formale - Cleanroom, Correctness by Construction
- *Tehnici privind detectarea defectelor (eng. fault detection techniques)*
 - sunt utilizate în timpul procesului de dezvoltare, pentru a identifica stările de eroare și defectele care le-au provocat, anterior livrării sistemului (ex. review, testare, depanare)
 - nu își propun recuperarea sistemului din stările de eșec induse de defectele identificate
- *Tehnici privind tolerarea defectelor (eng. fault tolerance techniques)*
 - pornesc de la premisa că sistemul poate fi livrat cu defecte și că eventualele eșecuri pot fi gestionate prin recuperare în urma lor la execuție
 - ex.: sistemele redundante utilizează mai multe calculatoare și softuri diferite pentru realizarea aceluiași sarcini

-
- Eşec, eroare sau defect?



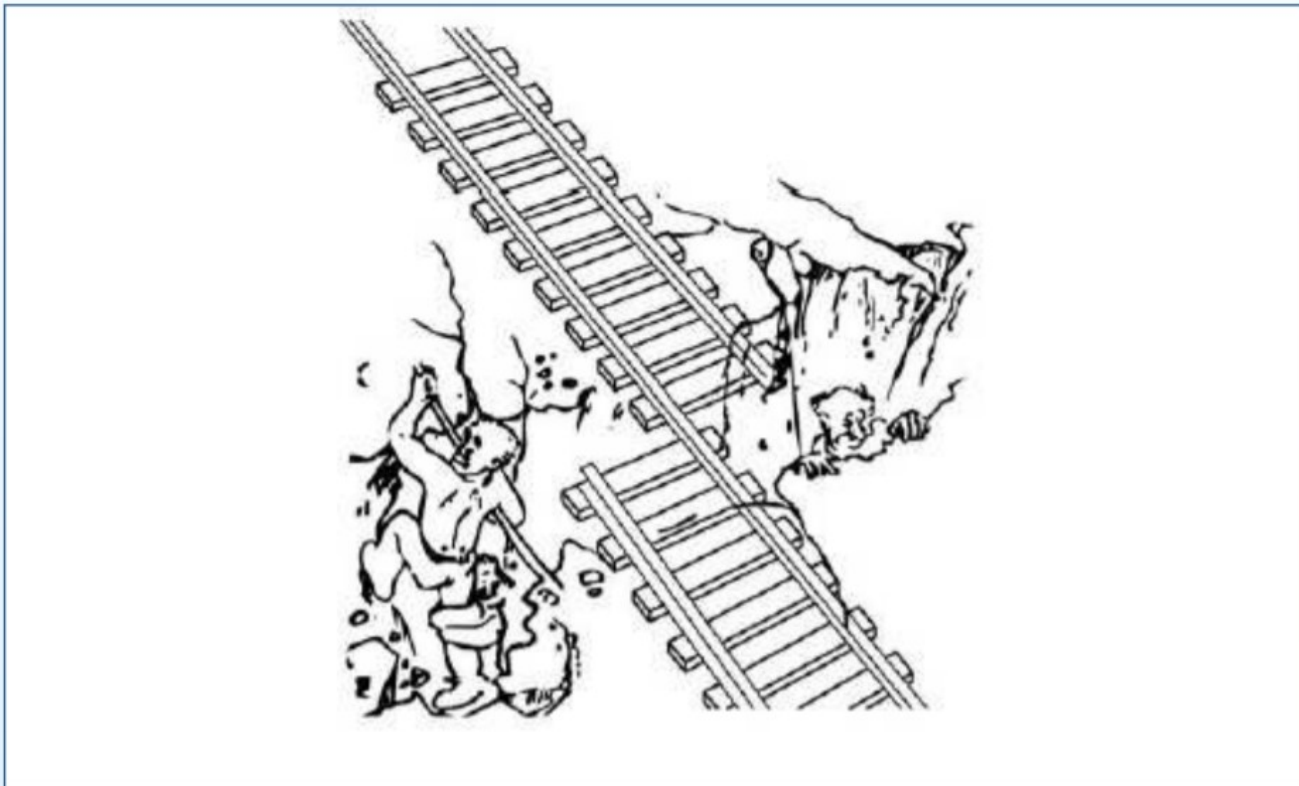
- Defect algoritmic

- omiterea inițializării unei variabile
- setarea unei variabile folosite drept index în afara valorilor permise



- Defect mecanic

- defect la nivelul mașinii virtuale
- pană de curent



- Eroare

