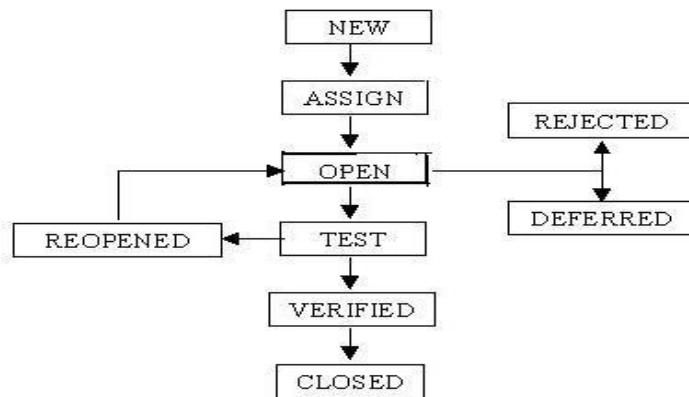


## Ciclul de viață a defectelor software

Ciclul de viață a unui defect începe atunci când se depistează defectul și se termină când un defect este închis, după ce se asigură că nu se va reproduce. Ciclul de viață a defectului este strict legat de faptul că defectul a fost găsit în timpul procesului de testare.

Defectul se poate afla în diferite stări pe parcursul ciclului de viață. Acestea pot fi prezentate printr-o diagramă:



### Ciclul de viață a unui defect poate include următoarele stări:

1. **Nou (New):** Când un defect este înregistrat și afișat pentru prima dată și se atribuie starea NOU;
2. **Asignat (Assigned):** După ce testerul a afișat defectul, liderul de echipă aprobă faptul că defectul este autentic și îl atribuie dezvoltatorului respectiv sau echipei de dezvoltatori. Defectul primește statutul ASIGNAT;
3. **Deschis (Open):** În această stare dezvoltatorul începe analiza defectului și lucrează la corectarea lui;
4. **Corectat (Fixed):** Când dezvoltatorul a efectuat modificările necesare în cod și a verificat modificările atunci defectului i se poate atribui statutul CORECTAT și este transmis echipei de testare;
5. **Retestare în așteptare (Pending retest):** După remedierea defectului, dezvoltatorul transmite codul modificat pentru retestare testerului. Aici codul este în așteptarea sfârșitului celorlalte teste. Prin urmare, statutul său este în așteptarea retestării.
6. **Retestare (Retest):** La această etapă testerul retestează codul modificat pentru a verifica dacă defectul a fost corectat sau nu.
7. **Verificat (Verified):** După retestare dacă defectul nu mai este prezent în software, se aprobă că defectul este corectat și modifică starea în "verificat".
8. **Redeschis (Reopen):** Dacă defectul rămâne prezent după ce dezvoltatorul a modificat codul testerul redeschide defectul atribuindu-i statutul "Redeschis" și se reiau pașii de la început;
9. **Închis (Closed):** Dacă defectul nu se mai reproduce după ce dezvoltatorul a modificat codul testerul îi atribuie statutul "Închis";
10. **Dublat (Duplicate):** Dacă defectul se repetă de două ori sau două defecte au aceeași descriere atunci i se atribuie statutul "Dublat";
11. **Respins (Rejected):** Dacă dezvoltatorul consideră că defectul nu este autentic, el îl respinge.
12. **Amânat (Deferred):** Defectul cu statutul "Amânat" va fi corectat în următoarele versiuni, aceasta se poate întâmpla din mai multe motive, spre exemplu prioritatea acestuia este joasă și nu afectează funcționalitățile principale ale produsului.
13. **Nu este defect (Not a bug):** Acest statut se atribuie defectelor atunci când ele nu de fapt nu modifică funcționalitățile softului, de exemplu schimbarea culorii unui text sau poziția unui buton pe ecran.

## ***Severitate vs. Prioritate***

Există două caracteristici cheie ale defectelor în testarea software:

1. Severitatea
2. Prioritatea

În ce constă diferența între acestea?

**Severitatea (Severity)** este măsura în care defectul poate afecta software-ul. Cu alte cuvinte, definește impactul pe care îl are un anumit defect asupra sistemului soft.

De exemplu, dacă o aplicație sau o pagină web se blochează atunci când este accesată de la distanță, impactul eșecului aplicației este grav cu toate că ar putea se întâmpla rar. Deci severitatea este de grad înalt, dar prioritatea este joasă.

**Severitatea poate fi de următoarele tipuri:**

- **Critică (Critical):** Defectul care are ca rezultat eșecul întregului sistem sau a uneia sau mai multor componente ale sistemului și provoacă o corupție extinsă a datelor. Dacă funcția afectată devine inaccesibilă și nu există o metodă alternativă acceptabilă pentru a obține rezultatele cerute, atunci severitatea va fi declarată critică.
- **Majoră (Major):** Defectul care are ca rezultat eșecul întregului sistem sau a uneia sau mai multor componente ale sistemului și provoacă o corupție extinsă a datelor. Dacă funcția afectată devine inaccesibilă dar există o metodă alternativă acceptabilă pentru a obține rezultatele cerute atunci severitatea va fi declarată ca fiind majoră.
- **Moderată (Moderate):** Defectul care nu duce la eșecul întregului sistem sau a uneia sau mai multor componente ale sistemului dar determină sistemul să producă rezultate incorecte, incomplete sau incoerente, atunci severitatea va fi declarată moderată.
- **Minoră (Minor):** Defectul care nu duce la eșec și nu afectează utilizarea sistemului, iar rezultatele așteptate pot fi ușor obținute prin ocolirea producerii acestor defecte, atunci severitatea va fi declarată minoră.
- **Cosmetică (Cosmetic):** Defectele ce țin de îmbunătățirea sistemului în care modificările sunt legate de aspectul și domeniul aplicației, atunci severitatea este declarată ca fiind cosmetică.

### **Prioritatea (Priority)**

Prioritatea definește ordinea în care ar trebui să corectăm un defect. Ar trebui să rezolvăm problema acum, sau poate să aștepte? Prioritatea este stabilită de către tester adusă la cunoștința dezvoltatorului cu menționarea intervalului de timp necesar pentru a remedia defectul. Dacă defectului i se atribuie prioritate înaltă, atunci dezvoltatorul trebuie să-l corecteze cât mai curând posibil. Prioritatea este stabilită pe baza cerințelor clientului.

**De exemplu:** dacă numele companiei este scris greșit în pagina de pornire a site-ului web, atunci prioritatea este ridicată, iar severitatea este cosmetică.

Prioritatea poate fi:

**Scăzută:** defectul este unul iritant și trebuie corectat, dar corectarea poate fi amânată până după remedierea defectelor grave.

**Medie:** defectul trebuie rezolvat pe parcursul activităților de dezvoltare. Se poate aștepta până când se creează o nouă construcție sau o versiune nouă.

**Ridicată:** Defectul trebuie rezolvat cât mai curând posibil, deoarece afectează sever aplicația sau produsul. Sistemul nu poate fi utilizat până când defectul nu este corectat.

Câteva exemple:

**Prioritate ridicată și severitate critică:** o eroare care apare la o funcționalitate de bază a aplicației și nu permite utilizatorului să utilizeze sistemul. (De exemplu, un site destinat păstrării detaliilor despre elevi,

prin salvarea înregistrării datelor personale, iar acesta nu permite salvarea înregistrării, atunci aceasta este o prioritate ridicată și un bug de severitate mare.)

**Prioritate ridicată și severitate cosmetică:** greșelile ortografice pe pagina de copertă sau în titlul unei aplicații.

**Prioritate scăzută și severitate critică:** o eroare care apare într-o funcționalitate a aplicației (pentru care nu există nici o soluție) și nu permite utilizatorului să utilizeze sistemul. Atunci se reia dezvoltarea funcționalității de la zero.

**Prioritate scăzută și severitate cosmetică:** Orice probleme cosmetice sau de ortografie care se află într-un paragraf sau în conținutul unui raport (Nu pe pagina de start sau în titlu).