

## 20. ŽIVOTNI CIKLUS GREŠKE U TESTIRANJU

Kada se, za vreme testiranja softvera, uoči bilo koji defekt, on mora da prođe kroz životni ciklus defekta koji se sastoji od konačnog broja stanja u kojima se defekt može nalaziti. Nakon što tester pronađe greške one se evidentiraju kako bi se lakše mogao pratiti njihov status ili kako bi i razvojni tim imao bolji uvid u deo softvera koji je potrebno ispraviti. Postoje razni softverski alati koji omogućuju praćenje životnog ciklusa greške (npr. Redmine). Svaki defekt ima uticaj na rad sistema, pa zbog toga svakom defektu treba pristupiti prema stepenu ozbiljnosti greške.

**Kategorije ozbiljnosti greške** mogu biti:

- **kritična** ("critical") koja najčešće rezultuje prekidom rada celog sistema ili pojedinih komponenti, čime je ugrožen rad celog softvera koji postaje neupotrebljiv.
- **visoka** ("serious") koja, takođe, rezultuje prekidom rada celog sistema ili pojedinih ključnih komponenti ali postoji alternativna mogućnost korišćenja softvera kao obilazno rešenje.
- **umerena** ("moderate") koja ne izaziva prekid rada sistema, utiče na manje funkcionalnosti, ali i dalje deluje na ponašanje softvera koje nije u skladu sa zahtevima.
- **manja** ("low priority") koja ne utiče na prekid rada sistema i njeno ispravljanje nije vremenski određeno.

Osim prema ozbiljnosti, **defekti se mogu klasifikovati i prema prioritetu:**

- **hitani prioritet** imaju greške koje zahtevaju trenutno ispravljanje i bez čijeg ispravke nema smisla nastavljati testiranje, a sam softver je neupotrebljiv.
- **visoki prioritet** imaju greške koje ugrožavaju osnovnu funkcionalnost i čija ispravka mora biti u vrlo kratkom roku.
- **srednji prioritet** imaju greške koje ne ugrožavaju rad sistema, a njihov ispravak može da sačeka.
- **niski prioritet** imaju greške koje nemaju nikakav uticaj na rad sistema i čija ispravka može biti odložena dok se ne poprave greške s višim prioritetom.

Životni ciklus greške započinje u onom trenutku kada se greška detektuje, a završava kada se ispravi tj. osigura da se ona više ne pojavljuje. Životni ciklus greške može se razlikovati od organizacije do organizacije, pa čak i od projekta do projekta. Ponajviše zavisi od uslova, poput modela i specifikacije na osnovu kojih se softver razvija, od vremenskih ograničenja, politike organizacije, strukture tima.

Na sledećoj slici je prikazan uobičajeni **životni ciklus greške**:

a) kada se greška prvi puta evidentira ona može biti detektovana od strane korisnika ili od tima testera. Prvo ima status "**nepotvrđena**", dok testeri ne potvrde postojanje greške.

b) kada testeri različitim metodama potvrde da je u pitanju defekt koji treba otkloniti, dobija status "**nova**" ili "otvorena".

c) greška se upućuje određenom programeru u timu, koji treba da je otkloni, status se menja u "**dodeljena**".

d) nakon što programer napravi određene promene u kodu i potvrdi ispravke, tada dobija status "**ispravljena**" te se iz tog statusa prosleđuje timu testera na proveru.

e) ako tester prilikom retestiranja nije pronašao grešku, greška se smatra "**verifikovano**" i može se zatvoriti.

f) ako je greška i dalje prisutna tj. nije ispravljena, tada je tester vraća u obradu i postavlja status "**ponovo otvorena**", te se ponovo dodeljuje nekom programeru.

g) ako je tester utvrdio da je greška ispravljena i verifikovano je njeno otklanjanje, tada joj se postavlja status na "**zatvorena**".

Osim ovih uobičajenih statusa iz životnog ciklusa, greška iz statusa "otvorena" može biti postavljena u sledeće specifične situacije koje se u praksi retko dešavaju, ali nisu nemoguće:

h) "**duplikat**", kada je već napravljena prijava iste greške.

i) "**odbijena**" kada programer smatra da prijava nema osnovu za prijavu te uz obrazloženje objašnjava zašto se greška odbija.

j) "**odložena**" kada se ispravljanje greške postavlja na privremeno čekanje, a razlog tome može biti promena prioriteta iz visok u nizak ili ako nisu jasno definisana poslovna pravila, pa ispravka greške traži dodatnu poslovnu analizu.

k) status "**nije greška**" može se postaviti u situacijama kada je došlo do promene funkcionalnosti softvera.

