Tesztelés

1. **A tesztelés alapfogalmai**

Tesztelésre azért van szükség, hogy a szoftver termékben meglévő hibákat még az üzembe helyezés előtt megtaláljuk, ezzel növeljük a termék minőségét, megbízhatóságát. Abban szinte biztosak lehetünk, hogy a szoftverben van hiba, hiszen azt emberek fejlesztik, és az emberek hibáznak. Gondoljunk arra, hogy a legegyszerűbb programban, mondjuk egy szöveges menü kezelésben, mennyi hibát kellett kijavítani, mielőtt működőképes lett. Tehát abban szinte biztosak lehetünk, hogy tesztelés előtt van hiba, abban viszont nem lehetünk biztosak, hogy tesztelés után nem marad hiba. A tesztelés után azt tudjuk elmondani, hogy a letesztelt részekben nincs hiba, így nő a program megbízhatósága. Ez azt is mutatja, hogy a program azon funkcióit kell tesztelni, amiket a felhasználók legtöbbször fognak használni.

1. **A tesztelés alapelvei**

A tesztelés alapjait a következő alapelvekben foglalhatjuk össze:

1. A tesztelés hibák jelenlétét jelzi: A tesztelés képes felfedni a hibákat, de azt nem, hogy nincs hiba. Ugyanakkor a szoftver minőségét és megbízhatóságát növeli.

2. Nem lehetséges kimerítő teszt: Minden bemeneti kombinációt nem lehet letesztelni (csak egy 10 hosszú karakterláncnak 256^10 lehetséges értéke van) és nem is érdemes. Általában csak a magas kockázatú és magas prioritású részeket teszteljük.

3. Korai teszt: Érdemes a tesztelést az életciklus minél korábbi szakaszában elkezdeni, mert minél hamar találunk meg egy hibát (mondjuk a specifikációban), annál olcsóbb javítani. Ez azt is jelenti, hogy nemcsak programot, hanem dokumentumokat is lehet tesztelni.

4. Hibák csoportosulása: A tesztelésre csak véges időnk van, ezért a tesztelést azokra a modulokra kell koncentrálni, ahol a hibák a legvalószínűbbek, illetve azokra a bemenetekre kell tesztelnünk, amelyre valószínűleg hibás a szoftver (pl. szélsőértékek). 5. A féregirtó paradoxon: Ha az újratesztelés során (lásd később a regressziós tesztet) mindig ugyanazokat a teszteseteket futtatjuk, akkor egy idő után ezek már nem találnak több hibát (mintha a férgek alkalmazkodnának a teszthez). Ezért a tesztjeinket néha bővíteni kell.

6. A tesztelés függ a körülményektől: Másképp tesztelünk egy atomerőműnek szánt programot és egy beadandót. Másképp tesztelünk, ha a tesztre 10 napunk vagy csak egy éjszakánk van.

7. A hibátlan rendszer téveszméje: Hiába javítjuk ki a hibákat a szoftverben, azzal nem lesz elégedett a megrendelő, ha nem felel meg az igényeinek. Azaz használhatatlan szoftvert nem érdemes tesztelni.

1. **Tesztelési technikák**

A tesztelési technikákat csoportosíthatjuk a szerint, hogy a teszteseteket milyen információ alapján állítjuk elő. E szerint létezik:

* Feketedobozos (black-box) vagy specifikáció alapú, amikor a specifikáció alapján készülnek a tesztesetek.
* Fehérdobozos (white-box) vagy strukturális teszt, amikor a forráskód alapján készülnek a tesztesetek.

1. **A tesztelési tevékenység**

Ugyanakkor a tesztelés nem csak tesztek készítéséből és futtatásából áll. A leggyakoribb tesztelési tevékenységek: 1-5

• Tesztterv elkészítése:

1. Megtalálni a hibákat;
2. Növelni a megbízhatóságot;
3. Megelőzni a hibákat.

• Tesztesetek tervezése:

1. Körülbelül 15 felhasználói gép kipróbálása, szkennerekkel együtt
2. Szervergép tűréshatárának meghatározása, biztonsági mentések idejének beálltása
3. Váratlan leállások hibáinak kiküszöbölése
4. Pótgenerátor üzemidejének tesztelése
5. Az egyes gépek önálló működése
6. Időzítő helyes működésének ellenőrzése

• Felkészülés a végrehajtásra:

* Tesztgépek beüzemelése, szkennerekkel
* Szerver beállítása
* Szoftver installálása
* Kábelek ellenőrzése
* Emberek választása a gépek teszteléséhez

• Tesztek végrehajtása:

* A jogosultsági kódok generálása, majd működésének ellenőrzése
* A kiválasztott emberek megkezdik a tesztelést
* Hálózat túlterhelése
* Áramszünet generálása
* Pótgenerátor helyes működésének vizsgálata (megnézni, hogy áramszünetkor elindul-e a generátor)
* Szerveren a biztonsági mentések átvizsgálása
* Időzítés aktiválásának ellenőrzése, közbeni leállás és idő lejáratának generálása, adatok kielemzése

• Kilépési feltételek vizsgálata:

* A tesztesetben leírt elvárt eredmények összehasonlítása a teszt naplóban lévő valós eredményekkel

• Eredmények értékelése:

* A kiértékelés
* Újabb tesztesetek készítése, ha hibát észlelünk, és felderítjük a hasonló hibákat

• Jelentéskészítés:

* A tesztelés eredményének írásos dokumentálása