

## Automatikus tesztelési eszközök

Az egységteszt olyan program, amely egy másikat tesztl (pl. JUnit, NUnit, xUnit).

Teszt automatizáláskor a tesztelő szkripteket ír és egy másik szoftvert használ a szoftver termék tesztelésére. Ez a folyamat magában foglalja egy manuális folyamat automatizálását is. A tesztautomatizálást használják kézzel végrehajtott teszt-forgatókönyvek újra futtatására gyorsan és ismétlődően.

A regressziós tesztelésen kívül a teszt automatizálást használják az alkalmazás terheléses, teljesítmény és stressz tesztelésére is. Ez növeli a teszt lefedettséget, a pontosságot és időt és pénzt takarít meg a manuális teszteléshez képest.

Nem lehetséges mindent automatizálni egy szoftverben. Azokat a területek, amelyeken a felhasználó tranzakciókat indíthat, mint a bejelentkezési vagy regisztrációs űrlap, vagy ahol sok felhasználó használhatja a szoftvert egyidejűleg, automatizálni kell.

Ezenkívül minden GUI elemet, az adatbázis kapcsolatokat, a mező validálásokat stb. hatékonyan lehet tesztelni a manuális folyamatok automatizálásával.

Mikor kell automatizálni?

- nagy és kritikus projektek
- projektek, amelyek ugyanannak a területnek gyakori tesztelését követelik meg
- a követelmények nem változnak gyakran
- terhelés és teljesítmény sok virtuális felhasználóval
- stabil szoftver a manuális tesztelés tekintetében
- rendelkezésre álló idő Hogyan automatizáljunk?

Az automatizálást valamilyen támogató programozási nyelv (mint pl. a VB script) és egy automatizált szoftver alkalmazás segítségével végezzük. Számos eszköz áll rendelkezésre automatizálási szkriptek írására.

Az automatizálás folyamata:

- az automatizálás területeinek meghatározása a szoftverrel kapcsolatban
- a megfelelő eszköz kiválasztása a teszt automatizáláshoz
- a teszt szkriptek megírása
- a teszt készletek kifejlesztése
- a szkriptek végrehajtása
- az eredmények jelentések elkészítése
- az esetleges hibák vagy teljesítmény problémák azonosítása

## **Tesztvezérelt fejlesztés (napi build, release).**

A tesztvezérelt fejlesztés olyan szoftverfejlesztési módszertan, amelyben a teszteseteket azelőtt készítik el (és többnyire automatizálják), mielőtt a szoftver fejlesztési folyamata befejeződne és le lehetne futtatni a teszteseteket.

### **Mikor jó egy teszt környezet?**

- Közel azonos hardware  
(Ha nem is pont azonos de a generációja legyen azonos, különben nem derül ki ha a programnak túl nagy lett a processzor igénye).
- Azonos operációs rendszer  
(32 bites vagy 64 bites, service pack).
- Azonos hálózati kapcsolat  
(Ha az éles rendszerben valamiért például WIFI kapcsolatot használunk akkor a teszt környezetben is legyen ilyen lehetőség)
- Megjelenítésre használt monitor  
(itt olyan problémák lehetnek, hogy kisebb felbontásnál nem fér ki a táblázat, vagy ha nagyobb felbontásban használják nem olvashatóak a betűk)
- Csak olyan szolgáltatások fussanak ami az élő rendszeren is futni fog.