

1. A tesztelés célja a szükséges lefedettség biztosítása a teszt tárgyának. Igaz
2. A tesztelés céljai nem változnak. Hamis
3. A tesztelés a minőségellenőrzés egyik formája. Igaz
4. Egy programon csak kötelezően kitűzött teszteseteket kell futtatni. Hamis
5. Az időkorlátozás elősegíti a rohammunka kifejlődését ezzel rontsolva a fő terméket. Igaz
6. Lehetséges, hogy egy szoftver már tervezésnél is hibás. Igaz
7. A hibajelentések segítik egy rendszer környezetének kialakítását. Igaz
8. A minőséget befolyásolja a hordozhatóság. Igaz
9. Egy program minősége jó, ha funkcionálisan nem tartalmaz hibát, de néha belassul. Hamis
10. A felhasználói élmény kialakítása fontos a minőség irányítás terén. Igaz
11. A tesztelés mennyisége növelheti, vagy csökkentheti a kockázatot. Igaz
12. Ha nincsen látszólagos hiba a programban, akkor jól működik. Hamis
13. A hibafürtök csak egy bizonyos szoftver modulra vannak kihatásra. Hamis
14. A tesztelést érdemes csak a program elkészítése után elkezdni. Hamis
15. Érdemes teszt esetről, teszt esetre tesztelni a programot, ezzel is felgyorsítva a tesztelést. Hamis
16. A tesztelés függ a környezettől. PL: 1 ember v. 100 ember használ egy rendszert. Igaz
17. A tesztelés megelőzheti az anyagi károkat is. Igaz
18. A hibákat le kell naplózni. Igaz
19. Egy hiba kiváltó ok megtalálását Diagnózissal tudjuk elvégezni. Igaz

1. A szoftver működési problémájánál nem kell figyelembe venni a hardverkörnyezetet – H
2. Az emberi félreértés ugyanolyan problémát jelenthet, mint például a hardverhiba – I
3. A tesztelés az ügyfél elégedettségét növeli – I
4. A minőség fogalma nem definiálható – H
5. A használhatóság a felhasználói élményt segíti – I
6. A tesztelést a lehető legkorábban kell kezdeni – I
7. Van, amikor a hibák javítása hasztalan – I
8. Nagyvállalati programok tesztelését nehéz szimulálni – I
9. Egy weboldal tesztelésénél nem szükséges a különböző böngészők és típusaik eltéréseit figyelembe venni – H
10. A használhatóság külön tesztelési területnek számít – I
11. Az áttekinthetőség fontos a karbantarthatóság szempontjából – I
12. Környezeti változásokra nem muszáj felkészíteni a programot – H
13. A kockázat negatív jövőhöz köthető – I
14. Létezik tesztelés előtti hibátlan program – H
15. szoftvertesztelés nem igényel jó fizikai állapotot a résztvevőktől – H
16. A rendszer feltételezi, hogy az adatok változhatnak – H
17. A szoftverfejlesztés nem kerül költségekbe – H
18. Ha a teszt nem talál hibát, a kód tökéletes – H

19. A fejlesztő csapat különböző területen dolgozó emberekből áll – I
20. Érvényes konfiguráció mellett is lehet problémás a működés – I

1. A tesztelés 2 félé lehet: dinamikus vagy statikus. I
2. A tesztelés célja az is, hogy bizalmat építsen ki a tárgy minőségét illetően. I
3. A tesztelés és hibakeresés szinte ugyanaz a tevékenység. H
4. A sikeres tesztelés során mindig mindenre kiterjedő tesztelés szükséges. H
5. A tesztelésnek különböző környezetekben eltér. I
6. A tesztfolyamat a teszttervezéssel kezdődik. I
7. A tesztvégrehajtás során létrejön a tesztkörnyezet. H
8. Egy szoftver meghibásodása származhat a szoftverből és a szoftver működési környezetéből. I
9. A hardverek problémái általában nincsenek hatással a szoftver működésére. H
10. Az adatok struktúrájának sérülése nem felismerhető. H
11. Egy szoftvernek több segédprogram is része, ezek hibája a szoftver működését befolyásolják. I
12. A minőségirányítás magában foglalja a szoftver minőségével kapcsolatos alapelveket. I
13. A minőségbiztosítás egy komponense a használhatóság. H
14. A minőséget befolyásolja például a megbízhatóság, használhatóság és funkcionális hibák. I
15. Van olyan funkcionális hiba, ami mellett még elfogadható minőségű egy termék. I
16. A hatékonyság esetén azt vizsgáljuk, hogy egy adott szoftver, hogy gazdálkodik az erőforrásaival, ennek kivitelezése nagyban függ a fejlesztő tudásától. I
17. A tesztelés során figyelembe kell venni, hogy milyen célra készült a szoftver, és minden lehetséges funkciót ugyan úgy kell tesztelni. H
18. A tesztelés minél korábbi fejlesztési fázisban el kell kezdeni. I
19. A hibafürtök akár egy másik programrészben is okozhat problémát. I
20. A főregirtó paradoxon jelentése, hogy az alap tesztkészletre működő program hibamentes. H

1. **A szoftvertesztelés célja a hibák elrejtése a felhasználók előtt. – Hamis**
2. **A funkcionális hibák a szoftver működését érintik. – Igaz**
3. **Csak manuális tesztelés létezik, az automatizált tesztelés nem szükséges. – Hamis**
4. **Az elfogadási tesztelés a végső tesztelési szakasz a felhasználói elfogadás előtt. – Igaz**
5. **A tesztelési dokumentáció nem lényeges része a szoftver minőségbiztosításának. – Hamis**
6. **A szoftverhibák mindig a szoftver belső működéséből erednek. – Hamis**
7. **A hardverhiba befolyásolhatja a szoftver működését, és nehezítheti a hibák felismerését. – Igaz**
8. **A hálózati hiba csak speciális hálózatok esetén jelenthet problémát. – Hamis**
9. **Az emberi tévedés az egyik legkomolyabb szoftverhiba ok lehet. – Igaz**

10. **A szoftver megfelelő működéséhez a konfigurációs beállításoknak nincs jelentősége.** – *Hamis*
 11. **A minőségirányítás célja a hibák elkerülése és a folyamatok optimalizálása.** – *Igaz*
 12. **A minőségbiztosítás csak a végtermék ellenőrzésére vonatkozik, nem az egész fejlesztési folyamatra.** – *Hamis*
 13. **A szoftverhasználhatóság (usability) nem befolyásolja a felhasználói élményt (UX).** – *Hamis*
 14. **A kockázatelemzés fontos része a minőségbiztosítási folyamatnak.** – *Igaz*
 15. **A minőségirányítás során az ügyfél igényeit figyelmen kívül lehet hagyni, ha a belső szabványok teljesülnek.** – *Hamis*
 16. **Bár a tesztelés kimutathatja a hibák jelenlétét, de nem képes igazolni, hogy nincsenek hibák.** – *Igaz*
 17. **A tesztelés a szoftver vagy rendszerfejlesztési életciklusban a legvégén kell elkezdni.** – *Hamis*
 18. **Ha mindig ugyanazokat a tesztekert hajtjuk végre, akkor az azonos a tesztkészlet egy idő után nem fog új hibákat találni.** – *Igaz*
 19. **A tesztelést különböző körülmények esetén különbözőképpen hajtják végre. Például egy olyan rendszert.** – *Igaz*
 20. **A kiadást megelőző tesztelés során a megtalált hibák többsége néhány modulban van, vagy legalábbis ezen modulok felelősek a működési hibák többségéért.** – *Igaz*
-
1. A tesztelés a hibák okozása és megtalálása? *Igaz*
 2. A tesztelés és a hibakeresés különálló tevékenységek? *Hamis*
 3. A tesztelés nem járul hozzá a sikerhez. *Hamis*
 4. A korai tesztelés idő és pénzt spórol. *Igaz*
 5. Hogyha ugyan azokat a tesztekert, teszteseteket ismétljük akkor hatékonyak vagyunk? *Hamis*
 6. Az emberi tévedés meghibásodáshoz vezethet. *Igaz*
 7. Probléma tud lenni a vállalat szervezési hibái? *Igaz*
 8. A teszt egyik lényege hogy az ügyfél elégedettségét növelje? *Igaz*
 9. A minőségbiztosítás önmagában egy rendszer? *Igaz*
 10. A hordozhatóság befolyásolja a minőséget. *Igaz*
 11. A nem megfelelő tesztelés a vállalatra jó hatást kelt. *Hamis*
 12. A tesztelés segít abban hogy a szoftverben ne maradjanak hibák. *Igaz*
 13. A tesztelés hozzájárul a hatékonysághoz. *Igaz*
 14. Ha egy szoftver jól van tesztelve akkor a vállalatnak az később kevesebb pénzbe kerülhet. *Igaz*
 15. Ha egy vállalat nem tesztel vagy csak keveset akkor nem vállal kockázatot. *Hamis*
 16. A karbantarthatóság hatékonyságának a növelésében a kommunikáció is szerepet játszik. *Igaz*
 17. Ha a szoftverünk minősége nem megfelelő akkor nehezen tudjuk a felhasználókat későbbi fejlesztések után rávenni hogy minőségi. *Igaz*
 18. Egy projekt során az idő és a nem megfelelő szervezés adja a legtöbb hibát! *Igaz*
 19. Ha valamit sietetünk, akkor nem lesz jó a vége. *Igaz*
 20. Nem fontos hogy legyen átfogó ismeretünk a használt technológiákban. *Hamis*

1. a tesztlezáráshoz kapcsolódó tevékenységeket a projekt mérföldkövek elérésekor hajtjuk végre. - igaz
2. a tesztelés jellemző céljai közé tartozik: hibák és a hibákat okozó tényezők megkeresése - igaz
3. a tesztelés és a hibakeresés ugyan az a tevékenység - hamis
4. az emberi hibák közé tartozik többek között a: rövid határidő, interakciók összetettsége - igaz
5. a szoftver meghibásodása csak a működési környezetből származhat - hamis
6. A szoftverek minden esetben valamilyen hardver környezetben működnek, így azok problémái közvetlen hatással vannak annak működésére - igaz
7. A rendszerek tervezésénél fontos szempont az újrafelhasználhatóság. - igaz
8. az adatbázis túlterheltségéből tipikusan hibák származnak. - igaz
9. egy elkészült rendszerhez nem szükséges meghatározni a megfelelő felhasználói környezetet. - hamis
10. a tesztelést elég a szoftver/rendszer befejezésekor elkezdni - hamis
11. a minőség önmagában nehezen definiálható, mert függ a felhasználóktól - igaz
12. ezek a szempontok meghatározzák a termék minőségét: megbízhatóság, hatékonyság, karbantarthatóság - igaz
13. funkcionális hibák esetén meghatározunk egy adott mennyiséget, típust és gyakoriságot, amin belül elfogadható minőségről beszélünk - igaz
14. egy rendszer akkor megbízható, ha a funkciói folyamatosan működnek - igaz
15. a használhatóság szempontjából nem számít a UX (user eXperience) - hamis
16. a hatékonyság esetén azt vizsgáljuk, hogy a szoftver hogyan gazdálkodik az erőforrásokkal (pl. tárhely) - igaz
17. a karbantarthatóság szempontjából, csak a felhasználói környezetet kell figyelembe venni - hamis
18. a ticket rendszert a hibák számontartására használják - igaz
19. a hordozhatóság szempontjából úgy kell felépíteni a rendszert, hogy a környezet változására megfelelően reagáljon - igaz
20. a kockázat az a tényező, ami a jövőben negatív következményeket okozhat – igaz

Igaz - Hamis

- 1.** A tesztelés egyik fő célja, hogy a meghibásodások okát, és a hibákat megtaláljuk

2. A tesztelésnek nem célja, hogy csökkentjük a szoftverminőség kockázati szintjét.
3. Azért tesztelünk, hogy a hibák jelenlétét mutassuk ki, nem a hiányát.
4. A hibakeresési folyamat első lépése az ok kijavítása.
5. A QA minőség biztosítást/tesztelést jelenti.
6. A kimerítő teszt lehetséges.
7. A korai tesztelés időt és pénzt is tud spórolni.
8. A tesztelés minden környezetben ugyan úgy történik.
9. A teszt tevékenységeket párhuzamosan is lehet végezni, nem kötelező a sorrend.
10. Meghibásodás csak és kizárólag a szoftver működéséből eredhet.
11. A hardver környezet közvetlen hatással van a program működésére.
12. A rendszerek tervezésénél nem számít az újrafelhasználhatóság.
13. A szoftver meghibásodás eredhet elkészítés során elkövetett hibákból.
14. Az emberi tévedés nem vezethet hibára.
15. Azért tesztelünk, hogy a hibák gyakoriságát csökkentjük, ne megszüntessük.
16. A funkcionális hibák nem befolyásolják a minőséget.
17. A minőség „Az a szint, amikor egy komponens, rendszer vagy folyamat megfelel a meghatározott követelményeknek és/vagy a felhasználó/ügyfél elvárásainak”
18. A hatékonyság nem befolyásolja a minőséget.
19. Fontos a minőségnek a karbantarthatóság.
20. A tesztesetek rendszeres felül vizsgálása felesleges.

1. A helytelenül működő szoftver sok veszteséghez vezethet. Igaz
2. A szoftvertesztelés segít a szoftver működés közbeni meghibásodási kockázatának csökkentésében. Igaz
3. A tesztelés csak statikus. Hamis
4. A rendszerek hibája az ember tévedése miatt is származhat. Igaz
5. Nem fontos, hogy a csapat számára egyértelmű legyen a cél. Hamis
6. Hibák gyakorisága attól függ, hogy a hibás funkció milyen gyakran van használatban. Igaz
7. A használhatóság angol megfelelője a usability. Igaz
8. Hatékonyság esetén azt vizsgáljuk, hogy egy adott szoftver hogyan gazdálkodik az erőforrásokkal. Igaz
9. Karbantarthatóság szempontjából, olyan megoldásokat kell választani, ami azt biztosítja, hogy a rendszer üzemeltetése során, könnyen áttekinthető legyen. Igaz
10. A kockázat nem okozhat pozitív következményeket. Hamis

11. A tesztelés költséghatékony módszert kínál a hibák észlelésére. Igaz
12. Emberi hibákat emberi lények követnek el. Igaz
13. A minőségi szint egy mértékrendszer. Igaz
14. A tesztelés feladata hogy növelje az ügyfél elégedettségét. Igaz
15. A tesztelés kimutathatja a hibák jelenlétét, de azt nem képes igazolni, hogy nincsenek hibák. Igaz
16. A tesztelést elég elkezdni a projekt befejezésénél. Hamis
17. A rendszereknél tervezésénél fontos szempont az újra felhasználhatóság. Igaz
18. A tesztelés és a hibakeresés ugyan az a tevékenység. Hamis
19. A hálózati probléma is okozhatja a meghibásodást. Igaz
20. A szoftverek számos komponensből épülnek fel. Igaz

1. A szoftvertesztelés a szoftverminőségét értékeli, illetve segít a szoftver működés közbeni meghibásodási kockázatának csökkentésében. – I
2. A tesztelés fő célja többek között, bizalom kiépítése a teszt tárgyának minőségével kapcsolatban – I
3. A teszt tárgy tesztelése során kizárólag a szoftver tesztelésére kell odafigyelni a teszt tárgy hibáira nem. – H
4. Egy hiba kijavítása nem okozhat további hibákat. – H
5. A tesztelők biztosítják, hogy a felhasználók igényeit a fejlesztési életciklus során figyelembe vegyék. – I
6. A tesztelés és a minőségbiztosítás szinonima szavak és ugyan azt is jelentik. – H
7. A minőségbiztosítás mind a fejlesztési, mind a tesztelési folyamatra értelmezhető és a projektekből mindenki felelős. – I
8. A tesztelés a hibák jelenlétét mutatja ki nem azok hiányát. – I
9. A tesztet elég csak a projekt befejeztével elvégezni hiszen így több idő és pénz spórolható meg. – H
10. A hibafürtök megjelenése kizárólag egy szektorban okoznak hibát. – H
11. A tesztelés során a teszt adatokat módosítani kell ahhoz, hogy a tesztek hatékonyak legyenek. – I
12. A szoftver meghibásodások oka származhat a szoftver működési környezetéből és magából a szoftverből. – I
13. Hardver hiba lehet például egy olyan érzékelő meghibásodása is, amely adatokkal lát el összetettebb szoftvereket. Például, ha egy hőfok vagy nyomás érzékelő hamis adatokat ad, akkor a rendszer szoftver a hamis adatok alapján vezérli az eszközt. - I
14. Ha az adatok áramlása az egyes számítógépek között nem működik megfelelően akkor az csak minimálisan lassítja annak működését. – H
15. A szoftverek meghibásodása származhat a külső körülményeken túl olyan okokból, amely az elkészítéskor elkövetett hibákra vezethetők vissza. – I
16. A szoftverek számos komponensből épülnek fel, amelyek külön-külön való működése valósítja meg szoftver célját. - H

17. A szoftverfejlesztés és tesztelés igen komoly szellemi tevékenység, ami jó szellemi és fizikai állapotot kíván meg a projekt résztvevőitől. – I
18. A minőség irányítás figyelmen kívül hagyja a minőséggel kapcsolatos alapelveket. – H
19. A használhatóság célja, hogy a szoftver minél gyorsabban futhasson így a felhasználói élményt mellőzve. – H
20. A hibák megtalálása és javítása hasztalan, ha a kifejlesztett rendszer használhatatlan, és nem felel meg a felhasználók igényeinek, elvárásainak. – I

1. A tesztelés az összes hibát feltárja. – H
2. A tesztelés különböző környezetekben eltérő módon történik. – I
3. A hardver problémái nincsenek hatással a szoftver működésére. – H
4. A használhatóság célja, hogy a felhasználói élmény minél jobb legyen. – I
5. A hordozhatóság szempontjából fontos, hogy a rendszer ne reagáljon a környezet változásaira. – H
6. A tesztelés célja az ügyfél elégedettségét növelni. – I
7. A kockázat az a tényező, amely a jövőben pozitív következményeket okozhat. - H
8. A rendszer megbízhatatlan, ha folyamatosan leáll, belassul és így a funkciók nem használhatóak. - I
9. Az emberi hibát nem kell számításba venni teszteléskor - H
10. Ha a rendszer nincs felkészítve adatsérülésre akkor az nem megfelelő működéshez vezet. - I
11. Az emberi tévedés nem gyakori hiba ok. - H
12. A tesztlezáráshoz kapcsolódó tevékenységeket projektmérföldkövek elérésekor hajtjuk végre. - I
13. A tesztervezés részét nem képezik a tesztcélok meghatározása. - H
14. A tesztmegvalósítás során létrehozuk vagy beszerezünk a tesztvégrehajtáshoz szükséges tesztvert. - I
15. A tesztelést elszigetelten végzik. - H
16. A tesztelés költséghatékony módszert kínál a hibák észlelésére. - I
17. A minőségbiztosítás önmagában is egy rendszer. – I
18. A hatás számítása nem becsülhető tapasztalti úton. - H
19. A tesztelés és a hibakeresés különálló tevékenységek. - I
20. A tesztelés képes igazolni, hogy nincsenek hibák. - H

1. A tesztelés és a hibakeresés nem különálló tevékenység. H
2. A tesztelés egy költséghatékony módszer a hibák észlelésére. I
3. A tesztelők biztosítják, hogy a felhasználók igényei a fejlesztés során figyelembe legyenek véve. I
5. A tesztelés a minőségellenőrzés egyik formája. I
6. Ugyanazon tesztek lefuttatása számos alkalommal nem ront a hatékonyságán új hibák megtalálásában. H
8. A szoftver hardverkörnyezete nincs hatással annak működésére. H
9. Az egyik legkomolyabb hiba ok az emberi tévedés. I

10. A segédprogramok hibája a szoftver működését is befolyásolja. I
11. A rendszereknek számos különböző környezetben kell működni. I
12. A szoftver tervezésekor nincs meghatározva, hogy milyen input
13. és output adatokkal működik megfelelően. H
14. A tesztelés szerepe közé tartozik a szoftver megbízhatóságának
15. növelése. I
16. A szoftver használhatósága nem befolyásolja a minőséget. H
17. A használhatóság célja, hogy a felhasználói élmény
18. minél jobb legyen. I
19. Karbantartás szempontjából fontos az áttekinthetőség. I
20. A tesztelést különböző körülmények között különbözőképpen
21. hajtják végre. I
22. Felesleges a lehető legkorábban elkezdeni a tesztelést. H
23. Ha a tesztelés során nem találunk hibát akkor a
24. rendszer tökéletes. H
25. Fontos a csapat számára egyértelmű célok és feladatok. I
26. A tesztelés lehet statikus vagy dinamikus. I

1. A statikus tesztelésnél a szoftvert futtatni kell? **Hamis**
2. A tesztelés garantálja, hogy a szoftver soha nem fog meghibásodni? **Hamis**
3. A tesztelés nem csak a szoftvertesztelők feladata? **Igaz**
4. A szoftvertesztelés során csak a szoftver működését kell ellenőrizni. **Hamis**
5. A szoftverhibák nem mindig a szoftver hibájából adódnak. **Igaz**
6. Az adatsérülés mindig a rendszer teljes összeomlásához vezet. **Hamis**
7. Egy érzékelő hibája okozhat nagyobb problémát egy összetett rendszerben. **Igaz**
8. A hardverhibák csak az új eszközöknél fordulhatnak elő. **Hamis**
9. A hálózati hibák csak a szoftverek működését befolyásolják. **Hamis**
10. A fejlesztőknek jó fizikai és mentális állapotban kell lenniük a hatékony munkához.
Igaz
11. A felhasználó szempontjából a minőség szubjektív. **Igaz**
12. A szoftver minőségét csak a funkcionális hibák befolyásolják. **Hamis**
13. A tesztelést különböző körülmények esetén is ugyanúgy hajtják végre. **Hamis**
14. A tesztelést elég akkor elkezdeni, amikor kész a program. **Hamis**
15. Nem létezik tökéletes program. **Igaz**
16. A felhasználói élmény köré egy külön terület nőtt ki magát. **Igaz**

17. Az ismerethiány lehet technológia, fejlesztéstechnikai, módszertani vagy üzleti ismerethiány. **Igaz**
18. Időnként a tesztek elavulnak és helyettük újakat kell írni. **Igaz**
19. Egy szoftver hiba akár milliós kárt okozhat. **Igaz**
20. A projekt tervezése során oda kell figyelni, hogy megfelelő körülmények biztosítottak legyenek a csapat számára. **Igaz**

1. Ha ugyanazokat a tesztek ismétljük akkor azok nem lesznek hatékonyak új hibák megtalálásában. I
2. A tesztelés minden környezetben ugyanolyan módon történik. H
3. A teszttervezés része a tesztcélok meghatározása és egy erre alkalmas megközelítés kiválasztása. I
4. Lineáris tesztelésnek nevezzük amikor közvetlenül a teszt tárgyában találhat hibákat. H
5. Dinamikus tesztelésnél a szoftver hibáit tárjuk fel. I
6. A szoftver meghibásodások oka származhat a szoftver működési környezetéből és magából a szoftverből. I
7. A hardver hiba soha nem vezethet teljes működésképtelenséghez. H
8. Hálózati hibákat nagyon egyszerű feltárni egy tesztelőnek. H
9. A rendszerek tervezésénél fontos szempont az újrafelhasználhatóság. I
10. A szoftverek meghibásodása származhat a külső körülményeken túl olyan okokból, amely az elkészítéskor elkövetett hibákra vezethetők vissza. I
11. Van olyan hogy az alkalmazás nem működik megfelelően más típusú böngészővel. I
12. A szoftverfejlesztés nem igényel jó szellemi állapotot. H
13. Tesztelés feladata a hibák csökkentése, megbízhatóság növelése és az előírásoknak való megfelelés biztosítása. I
14. Funkcionális hibák esetén meghatározzuk azt a mennyiséget, típust és gyakoriságot, ami mellett meg elfogadható a minősége. I
15. A használhatóság célja, hogy a fejlesztőnek egyszerűbb legyen tesztelnie. H
16. A tesztelés mennyiségénél figyelembe kell venni a rendszer célját. I
17. A legtöbb idővesztést egy projekt alatt az ismeretek hiánya, a nem megfelelő szervezés és az időnyomás generálja. I
18. A tesztelés képes igazolni hogy nincs hiba a szoftverbe. H
19. A tesztelést a lehető legkorábban el kell kezdeni, és előre meghatározott célokra kell összpontosítani. I
20. A féregirtó paradoxon ellen a teszteseteket rendszeresen felül kell vizsgálni. I

1. A tesztelés kimutatja a hibák jelenlétét. Igaz
2. Létezik hibamentes rendszer. Hamis
3. A különböző rendszereket különbözőképpen kell tesztelni. Igaz

4. A tesztelést a szoftver legkorábbi fázisában kell elkezdni. Igaz
5. Gyakran kell cserélni a teszteseteket. Igaz
6. Nem számít tesztcélnak a hibák megtalálása. Hamis
7. Nem lehet hiba nélküli szoftvert írni. Igaz
8. A tesztelés és a hibakeresés különálló tevékenységek. Igaz
9. A statikus tesztelés közvetlenül találja meg a hibákat. Igaz
10. A tesztelés egy drága módszer a hibák észlelésére. Hamis
11. Sokszor hardver hiba okozza a szoftver nem megfelelően működését. Igaz
12. A rendszerek tervezésénél nem fontos szempont az újrahasználatosság. Hamis
13. Felhasználó nem okozhat hibát. Hamis
14. Adatsérülés nem okozhat hibát. Hamis
15. Ha nem olyan környezetben van a szoftver, ahova tervezték, akkor nem garantálható a működése. Igaz
16. A tesztelések csökkentik a szoftver megbízhatóságát. Hamis
17. A minőség: az a szint, amikor egy komponens, rendszer vagy folyamat megfelel a meghatározott követelményeknek és a felhasználó elvárásainak. Igaz
18. A usability célja, hogy a felhasználói élmény minél jobb legyen. Igaz
19. A szoftverekben nem fontos a karbantarthatóság és a hatékonyság. Hamis
20. A tesztelés mennyiségének megállapítása során nem fontos figyelembe venni, hogy a szoftver milyen célra készült. Hamis