Szoftvertesztelői ismeretek

**Írásbeli vizsga**

**A vizsgatevékenység, vagy részeinek leírása**

Az írásbeli vizsga ismertetése:

* 40 db feleletválasztásos tesztkérdés.
* A feleletválasztásos tesztkérdések egyetlen helyes válaszlehetőséget tartalmaznak.
* A helyes választ egyértelműen kell jelölni, egyetlen válasz aláhúzásával.
* A vizsgázó nevét a minden lap tetején kitölteni szükséges.
* A vizsgatevékenység végrehajtására rendelkezésre álló időtartam: 90 perc.
* A vizsgatevékenység aránya a teljes képesítő vizsgán belül: 50%.
* Maximálisan elérhető pontszám/százalék: 100 pont/100%.

A kérdések megoszlása:

* 3 db tesztkérdés Linux ismeretekből.
* 3 db tesztkérdés Verziókezelési ismeretekből.
* 3 db tesztkérdés a szoftverarchitektúra ismereteiből.
* 3 db tesztkérdés SQL ismeretekből.
* 28 db tesztkérdés a szoftvertesztelés elméleti ismereteiből.

A vizsgatevékenység értékelésének szempontjai:

* 3 db tesztkérdés Linux ismeretekből - (3x2,5 pont) 7,5%
* 3 db tesztkérdés Verziókezelési ismeretekből - (3x2,5 pont) 7,5%
* 3 db tesztkérdés a szoftverarchitektúra ismereteiből - (3x2,5 pont) 7,5%
* 3 db tesztkérdés SQL ismeretekből - (3x2,5 pont) 7,5%
* 28 db tesztkérdés a szoftvertesztelés elméleti ismereteiből - (28x2,5 pont) 70%
* Helyes válasz 2,5 pontot ér, a helytelen válasz 0 pontot ér.
* A rossz válasz megjelölésért pontlevonás nem jár.

A vizsga eredményessége:

* A vizsgatevékenység akkor eredményes, ha a vizsgázó a megszerezhető összes pontszám legalább 51%-át elérte.

**Linux ismeretek**

1. Melyik linux paranccsal tudjuk egy file tartalmát kiíratni a terminalra?
   1. ls
   2. cat
   3. rm
   4. list
2. Mit csinál a következő parancs: $ cp -R /etc /etc\_backup
   1. Átmásolja az etc mappát az etc\_backup mappába.
   2. Átmásolja az etc mappa tartalmát az etc\_backup mappába.
   3. Átmásolja az etc mappát és mindent, ami benne van az etc\_backup mappába.
   4. Áthelyezi az etc mappát az etc\_backup mappába.
3. Milyen jogosultsági beállításai vannak a következő file-nak: -rw-r--r-- 1 root root
   1. tulajdonos felhasználó olvasás és írás és végrehajtás, tulajdonos csoport olvasás, mindenki más olvasás.
   2. tulajdonos felhasználó olvasás és írás, tulajdonos csoport írás, mindenki más írás.
   3. tulajdonos felhasználó olvasás és írás, tulajdonos csoport olvasás, mindenki más olvasás.
   4. mindent szabad rajta, mert root a tulajdonos felhasználó és csoport is.

**Verziókezelési ismeretek**

1. A GIT verziókezelő rendszerben mi az a branch?
   1. Új vagy különálló másolat a repository-ból.
   2. Lokális másolat a repository-ból.
   3. Az összes commit gyűjteménye.
   4. Bármi, aminek emberi olvasható neve van.
2. Mik a feltétlenül szükséges GIT műveletek fejlesztés közben?
   1. checkout, add, commit, push, PR
   2. reset, add, commit, pull, push
   3. checkout, add, commit, pull, push, PR
   4. branch, add, revert, PR
3. Melyik parancs helyezi a lokálisan megváltozott file-okat az indexbe?
   1. commit
   2. branch
   3. add
   4. checkout

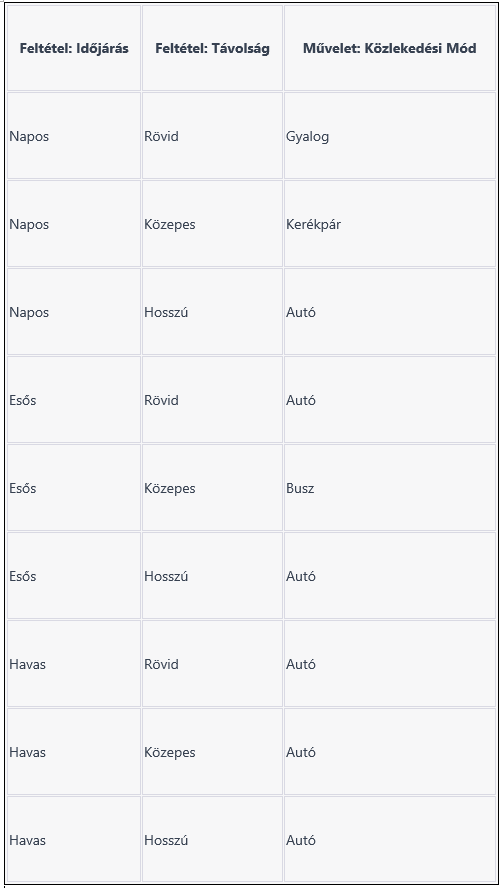
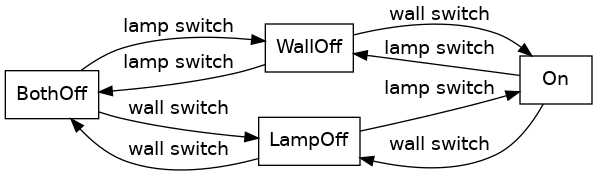
**Szoftverarchitektúra ismeret**

1. Mik az elemei az MVC architektúrának?
   1. Machine, VOIP, Control
   2. Machine, View, Client
   3. Model, VOIP, Client
   4. Model, View, Control
2. Mi jellemzi a réteges architektúrát?
   1. Minden rétegnek van egy specifikus szerepe, hívások fentről lefele mennek, jellemző elemei az adatbázis, üzleti logika és megjelenítés.
   2. Minden rétegnek van egy specifikus szerepe, hívások minden réteg között mehetnek, jellemző elemei az adatbázis, üzleti logika és megjelenítés.
   3. A rétegeknek több szerepe van, hogy az egyik kiesése esetén a többi át tudja vanni a szerepét, jellemző elemei az adatbázis, üzleti logika és megjelenítés.
   4. Minden rétegnek van egy specifikus szerepe, hívások fentről lefele mennek, jellemző elemei az kliens, szerver és adatbázis.
3. Hol bír jelentőséggel a termék architektúrája a tesztelésben?
   1. Különösebb jelentőséggel nem bír.
   2. Tervezési fázisban segít a tesztelést is megtervezni, illetve bemenettel szolgál a teszt megközelítéshez és a tesztesetek írásához.
   3. Segít a teszt management-nek párosítani a teszt feladatokat a tesztelőkhöz.
   4. Rajta keresztül tudjuk meghatározni a követelmények teszt kritériumait.

**SQL ismeret**

1. Melyik paranccsal tudsz elemet beilleszteni egy adattáblába?
   1. INSERT
   2. PUT
   3. ADD
   4. CUSTOMISE
2. Mit csinál a következő parancs: SELECT CustomerName, City FROM Customers;?
   1. CustomerName és City táblákba beilleszt egy Customers nevő recordot
   2. Customers nevű tábla CustomerName és City oszlopaiból készít egy másik táblát
   3. Customers nevű tábla recordjaihoz mutatja meg a CustomerName és City oszlopokat
   4. Customers nevű tábla recordjaihoz adja hozzá a CustomerName és City oszlopokat
3. Mi nem igaz az elsődleges kulcsra egy SQL adatbázisban?
   1. Az elsődleges kulcs egy olyan érték, amely alapján minden rekord egyértelműen és összetéveszthetetlenül beazonosítható.
   2. Az elsődleges kulcs egy teljesen egyedi érték, ez az érték nem tartalmazhat NULL-t.
   3. Egy táblának csak egyetlen egy elsődleges kulcsa lehet, amely állhat egy vagy több oszlopból is.
   4. Egy táblának több elsődleges kulcsa is lehet.

**Szoftvertesztelés elméleti ismeret**

1. Mi a célja a szoftvertesztelésnek?
   1. Hibák keresése.
   2. Információ nyújtása a termék minőségéről.
   3. Előre megbeszélt minőségi szint elérése és információ nyújtása a minőségről.
   4. Előre megbeszélt minőségi szint elérése.
2. Milyen tevékenységeken keresztül éri el a célját a tesztelés?
   1. Hibák keresése, új hibák megelőzése, követelmények teljesülésének ellenőrzése
   2. Hibák keresése, követelmények teljesülésének ellenőrzése
   3. Dokumentáció, és hibák keresése
   4. Hibák keresése, új hibák megelőzése, követelmények teljesülésének ellenőrzése és dokumentáció
3. Mik az előnyei a korai tesztelésnek?
   1. Tesztelhetőbb követelmények létrehozása korábban.
   2. Hibák olcsóbb kijavítása.
   3. Hibák ismétlődésének megelőzése.
   4. Mindegyik.
4. Az alábbi szituációban mi az emberi eredetű hiba, defekt és a meghibásodás?  
     
   A programozót nem tájékoztatták arról, hogy hány tizedesjegy pontossággal tartsa nyilván a bank rendszere a megtakarításokat. Ezért az előzetes projekcióktól eltérő módon változott néhány betét összege.
   1. A kódban helytelenül implementált pontosság az emberi eredetű hiba, a tizedes pontosság meghatározása a defekt és a betét összeg helytelen változása a meghibásodás.
   2. A betét összeg helytelen változása az emberi eredetű hiba, a kódban helytelenül implementált pontosság a defekt és a tizedes pontosság a követelményekben a meghibásodás.
   3. A tizedes pontosság meghatározása az emberi eredetű hiba, a betét összeg helytelen változása a defekt és a kódban helytelenül implementált pontosság meghibásodás.
   4. A tizedes pontosság meghatározása az emberi eredetű hiba, a kódban helytelenül implementált pontosság a defekt és a betét összeg helytelen változása a meghibásodás.
5. Miért lényeges tényező a függetlenség a tesztelésben?
   1. Növeli a kommunikáció hatékonyságát a fejlesztő- és a tesztelő csapat között.
   2. Növeli az objektivitását a tesztelő csoportnak.
   3. Csökkenti a tesztelés költségét.
   4. Független tesztelés esetén nem kell folyamatokat meghatározni.
6. Egy meglévő termék új funkcióját fejlesztjük, melyik a legértékesebb becslés a tesztelés időigényére vonatkozóan?
   1. Teszt manager véleménye.
   2. Korábbi hasonló funkció tesztelése.
   3. Mennyi időt kapunk rá.
   4. Szakmai standard-ek.
7. Melyik teszteket érdemes automatizálni?
   1. Amiket ritkán futtatunk és kevesen ismernek.
   2. Mindent automatizálni kell, mert az javítja a hatékonyságot.
   3. Amin sokat nyerünk, ha gyorsan fut és a gép jellemzően jobban megcsinálja, mint az ember
   4. Exploratory teszteket
8. Mik az elvárt jellemzői egy követelménynek?
   1. Ellentmondásmentes, teljes, redundáns, tesztelhető, atomi, trace-able, priorizált, egyértelműen azonosított
   2. Ellentmondásmentes, teljes, konzisztens, megvalósítható, magas prioritású, trace-able, egyértelműen azonosított
   3. Teljes, konzisztens, megvalósítható, tesztelhető, atomi, trace-able, priorizált
   4. Ellentmondásmentes, teljes, konzisztens, megvalósítható, tesztelhető, atomi, trace-able, priorizált, egyértelműen azonosított
9. Melyik az az elem, ami összeköti a követelményt a teszteléssel?
   1. Követhetőség
   2. Teszt kritérium
   3. Megvalósíthatóság
   4. Teszteset
10. Mi jellemző a követelménygyűjtésre általában?
    1. A követelmények általában világosak a project kezdetekor.
    2. A követelmények mindig tesztelhetően vannak megfogalmazva.
    3. A követelmények explicit megfogalmazása mindenkinek egyértelmű igény.
    4. A követelmények általában több kör gyűjtés után stabilizálódnak.,
11. V modell hogyan kezeli a tesztelést?
    1. A tesztelés egyértelműen a fejlesztés után kezdődik el.
    2. Minden fejlesztési fázishoz szorosan kapcsolódik egy tesztelési fázis.
    3. Nincs lehetőség korrekcióra egyik fázisban sem.
    4. Ideális választás olyan projektek esetén, ahol a követelmények és tervezési részletek csak később válnak tisztává, és a rugalmasság és változáskezelés fontos szempontok.
12. Adott az alábbi döntési tábla, amely időjárási körülmények és az úti cél távolsága alapján meghatározza, mely közlekedési módot kell használni.  
      
      
      
    Mekkora a minimális teszteset halmaz, amivel le tudjuk fedni a táblát?
    1. 5
    2. 6
    3. 8
    4. 9
13. Adott az alábbi állapotgép:

Ha le akarjuk fedni az összes átmenetet, a BothOff állapotból akarunk indulni és az On állapotba akarunk érkezni minden tesztesettel, hány tesztesetre van szükségünk legalább?

* 1. 2
  2. 3
  3. 1
  4. 4

1. Egy program a numerikus mezőket az alábbiak szerint validálja:   
   10 alatt lévő értékeket elutasítja  
   10 és 21 közötti értékeket elfogadja  
   22 vagy annál nagyobb értékeket elutasítja.   
   Az alábbiak közül melyik tartalmazza a LEGTÖBB határértéket?
   1. 9, 10, 11, 22
   2. 9, 10, 21, 22
   3. 10, 11, 21, 22
   4. 10, 11, 20, 21

1. Egy űrlapon található mező egy szövegdobozt tartalmaz, amely numerikus értékeket fogad el 18 és 25 közötti tartományban. Az alábbi értékek közül melyik tartozik az invalid ekvivalencia osztályba?
   1. 17
   2. 19
   3. 24
   4. 21
2. Felhasználási eset főútvonalának lépései hogy épülnek fel?
   1. Actorok (userek és a rendszer) a mondatok aktív szereplői.
   2. A cselekvés röviden és objektíven van leírva.
   3. Bemenetek és kontextus tisztázva van.
   4. Mindegyik.
3. Mit jelent a hibamentesség illúziója?
   1. Nem létezik hibátlan rendszer.
   2. Teszteléssel nem tudjuk garantálni a hibamentességet.
   3. Aránytalanul sok időbe telik letesztelni minden lehetséges be-és kimenetet.
   4. A tesztelés nem emeli a termék minőségét.
4. Strukturális tesztelés esetén mit jelent egy fedés?
   1. Adott kód elemek összessége, mint 100%-hoz képest a tesztek a kódelemek mekkora részét érintették.
   2. Hány sort érintettek a kódból a tesztjeink.
   3. A követelmények teszt kritériumainak mekkora részét igazoltuk már a tesztjeinkkel.
   4. Strukturális tesztelés esetén nem használunk fedéseket, mert nincsenek követelményeink.
5. Mi igaz a black box tesztelésre?
   1. Egy tapasztalat alapú tesztelési technika.
   2. Nem szükséges hozzá specifikáció.
   3. A kód strukturájából indulunk ki és írjuk meg a tesztjeinket.
   4. A követelményekből kiindulva határozunk meg bemenet-kimenet párokat a tesztelés során.
6. Mi az az elfogadási teszt?
   1. Az ügyfél vagy felhasználók részéről történik, és azt ellenőrzi, hogy a szoftver megfelel-e az üzleti igényeknek és a végfelhasználói elvárásoknak.
   2. A szoftver legkisebb önállóan tesztelhető részeit, pl függvények vagy osztályok vizsgálja.
   3. A szoftverkomponensek közötti együttműködést vizsgálja.
   4. Azt vizsgálja, hogy a szoftver hogyan teljesít terhelés alatt vagy meghatározott terheléses feltételek mellett.
7. Mi az az integrációs teszt?
   1. Az ügyfél vagy felhasználók részéről történik, és azt ellenőrzi, hogy a szoftver megfelel-e az üzleti igényeknek és a végfelhasználói elvárásoknak.
   2. A szoftver legkisebb önállóan tesztelhető részeit, pl függvények vagy osztályok vizsgálja.
   3. A szoftverkomponensek közötti együttműködést vizsgálja.
   4. Azt vizsgálja, hogy a szoftver hogyan teljesít terhelés alatt vagy meghatározott terheléses feltételek mellett.
8. Mi az az egységteszt?
   1. Az ügyfél vagy felhasználók részéről történik, és azt ellenőrzi, hogy a szoftver megfelel-e az üzleti igényeknek és a végfelhasználói elvárásoknak.
   2. A szoftver legkisebb önállóan tesztelhető részeit, pl függvények vagy osztályok vizsgálja.
   3. A szoftverkomponensek közötti együttműködést vizsgálja.
   4. Azt vizsgálja, hogy a szoftver hogyan teljesít terhelés alatt vagy meghatározott terheléses feltételek mellett.
9. Alábbiak közül melyik lehet egy test plan része?
   1. Teszt stratégia, scope és célok, tesztelési ütemterv, felelősségi körök és erőforrásmanagement.
   2. Impakt analízis, teszt itemek, automatizálási utasítások.
   3. Összefoglalás, teszteset összefoglaló jelentés, incidens összefoglaló jelentés, tervtől eltérés, incidensek és teszt futások leírása.
   4. Teszt stratégia, scope és célok, teszt itemek.
10. Mik a lényeges részei egy hibajegynek
    1. Hibajegy azonosítója, súlyosság, prioritás, hibakezelő felelős, hiba státusza, létrehozás dátuma.
    2. Hibajegy azonosítója, hiba leírása, súlyosság, prioritás, hibakezelő felelős.
    3. Csak a hiba leírása és a reprodukáláshoz szükséges információ a lényeg.
    4. Hibajegy azonosítója, hiba leírása, súlyosság, prioritás, hibakezelő felelős, hiba státusza, létrehozás dátuma.
11. Milyen részei vannak egy test summary reportnak?
    1. Összefoglalás, incidens összefoglaló jelentés, tervtől eltérés, incidensek és teszt futások leírása
    2. Összefoglalás, teszteset összefoglaló jelentés, tervtől eltérés, incidensek és teszt futások leírása
    3. Összefoglalás, teszteset összefoglaló jelentés, incidens összefoglaló jelentés, tervtől eltérés, incidensek és teszt futások leírása
    4. Teszteset összefoglaló jelentés, incidens összefoglaló jelentés
12. Mi a test item?
    1. Amit mérünk a fedésekkel.
    2. Egy-egy megfeleltetés van a teszt itemek és a tesztesetek között.
    3. Egy különálló elem, amit tesztelünk. Általában a termék több teszt item-re bontható.
    4. Az a része e követelménynek, aminek a fennállását igazoljuk a tesztekkel.
13. Mi az a teszt kritérium?
    1. Az a része e követelménynek, aminek a fennállását igazoljuk a tesztekkel.
    2. A teszt plan része, ami megmondja, hogy mikor van vége a tesztelésnek.
    3. A tesztelés indításának feltétele.
    4. Ellenőrzési lista, aminek a kitöltésével a tesztelést elvégzettnek tekinthetjük.
14. Mi az a teszteset?
    1. Bemeneti értékek, végrehajtási előfeltételek, elvárt eredmények és végrehajtási utófeltételek halmaza.
    2. Egy konkrét követelmény ellenőrzése.
    3. Egy teszt megközelítés, amelyben felbontjuk a lehetséges bemeneteket és kimeneteket egymástól elkülönülő halmazokra.
    4. Lényegében tesztadatok csoportosítása.