

## **Analitikus értékelési módszerek Empirikus tesztelés (bevezető), konceptiótesztelés**

## ***Analitikus értékelési módszerek***

### **Szoftvertermékek használhatóságának vizsgálata**



Az **empirikus módszerek** (*usability testing, usability engineering*) alkalmazása során a vizsgálandó szoftver terméket - vagy annak működő prototípusát - a felhasználók kezébe adjuk és az interakciót megfelelő eszközökkel empirikusan tanulmányozzuk.

(A puding próbája ...)

Az **analitikus módszerek** (*usability inspection*) valamilyen módon szimuláljuk a felhasználó várható tevékenységét.



3

### ***Analitikus értékelési módszerek***




- Guideline review, checklists (*korábban*)
- GOMS modellen alapuló módszerek (*korábban*)
- Kognitív bejárás technikák
- Heurisztikus eljárások

4



## Kognitív bejárás (Cognitive Walkthrough)

5



## Kognitív bejárás


Olyan termék elemzésére használható a legjobban, amelyek használatát a felhasználók az *explorációs (felfedező) tanulás* útján sajátítanak el.

Az elemzés az elképzelt felhasználó lépéseinek szimulációját jelenti kérdések feltevésével.

A használat során feltárt problémákra a tervezés következő iterációs lépésében kell megoldást találni.

A vizsgálatot szakértők vagy a valós felhasználók végezhetik.

6



## Kognitív bejárás 1.

LEWIS, C. és POLSON, P. (1993) nyomán

A „*kognitív bejárás*” (*Cognitive Walkthrough*) a szoftverek egyik funkcionális tesztelési módszerének, a „*kód-bejárásnak*” (*Code Walkthrough*) az adaptálása a UI vizsgálatára


A *kód-bejárás* ⇨ programozók értékelik egymás programjait (verbálisan végigkövetik egy kiválasztott programrészlet futását)

A *kognitív bejárás* ⇨ elemzők értékelnek egy rendszert (verbálisan végigkövetik azt az utat, amin a felhasználóknak végig kell menniük egy kiválasztott feladat elvégzése érdekében)

Ahogy a *kód-bejárás* programhibákat (*bug*) talál, úgy a *kognitív bejárás* olyan pontokat azonosít, ahol a felhasználók valószínűleg nem találják meg a helyes tennivalókat

Ahogy a *kód-bejárás* nem igényel gépet, úgy a *kognitív bejárás* sem igényel valódi felhasználókat és ezért egészen korai verziók vizsgálatára is használható.

7



## Kognitív bejárás 2.

Olyan UI elemzésére használható a legjobban, amelyeket a felhasználók az *explorációs tanulás* útján sajátítanak el.

Az alkalmazás alapját a felhasználó viselkedésének a következő egyszerű modellje képezi:

- (0) A felhasználó egy nagyvonalú elképzelést alakít ki arról, hogy mi a célja, mit is akar elérni (definiálja maga számára a feladatot).
- (1) A felhasználó a felhasználói felületen keresztül felfedezi a rendszert: igyekszik olyan akciókat találni, amelyek hozzájárulhatnak célja eléréséhez.
- (2) A felhasználó kiválasztja és végrehajtja azt az akciót, amelyről leírása vagy megjelenése alapján úgy érzi, hogy leginkább illik ahhoz, amit el akar érni.
- (3) A felhasználó értelmezi a rendszer választát és megítéli, hogy közelebb jutott-e célja eléréséhez.

8

### Kognitív bejárás 3.



Az elemzés az (1), (2) és (3) lépések szimulációját jelenti a következő kérdések feltevésével:

1. Elégé nyilvánvaló-e a felhasználó számára, hogy mi a helyes akció?
2. Összekapcsolja-e a felhasználó a helyes akció leírását azzal, amit el akar érni?
3. Tudja-e értelmezni a felhasználó a rendszernek a kiválasztott akcióra adott választát, azaz el tudja-e dönteni, hogy jó vagy rossz akciót hajtott végre?

Az eredmények a fenti három kérdéshez kapcsolódó területeken problémák felfedezése: ahol a kérdésre "nem" a válasz, ott valamilyen probléma van. A problémákra a tervezés következő iterációs lépésében kell megoldást találni.

9

### Kognitív bejárás



Gyakorlat

10



### Heurisztikus elemzések (Heuristic Evaluation)



11

### Heurisztikus elemzés



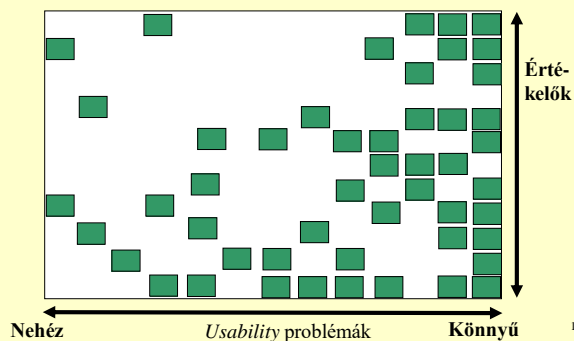
- A legtöbb módszer hátránya, hogy csak bizonyos fajta tervezési problémákra alkalmazhatók. Kevésbé kötött módszerre is szükség van.
- 4-6 szakértő végzi
- saját tapasztalataik és „design heurisztikák” alapján.
- Az elemzést a csoport tagjai egymástól függetlenül kezdik. Az elemzőket egy-egy asszisztens figyeli, s minden észrevételét rögzíti.
- Ezek után az összesített hibalistát együttes ülésen megnézik: közösen mérlegelik a megtalált problémák súlyosságát.
- A módszer nagy előnye, hogy olcsó és igen hatékony, hátránya viszont, hogy inkább a hiányosságokra koncentrál, mint a megoldásokra és csak igen kevésé megismételhető.

12

## Heurisztikus elemzés

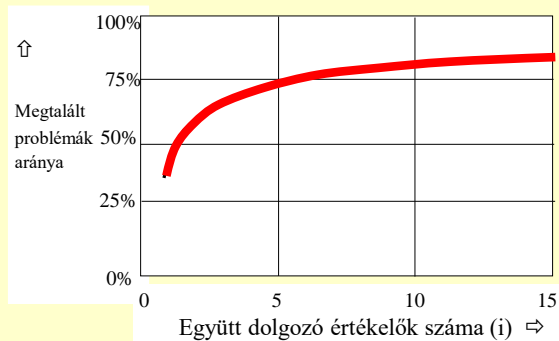


### Illusztráció:



13

## Heurisztikus elemzés.



14

## Heurisztikus elemzés



A Nielsen által javasolt „heurisztika-mankó” jellegűen alkalmazott, tervezésiirányelv-jellegű, csoportosítási célú „design heurisztikák”:

1. megfelelő visszajelzés
2. egyszerű, természetes párbeszéd
3. kontrollálható párbeszéd, „vészijárat” nem kívánt funkcióból
4. következetesség szoftveren belül
5. rendszer állapotának, munka előrehaladásának láthatósága
6. kerülje az emlékezet terhelését
7. alternatív megoldások, rövidebb utak, testreszabható
8. képernyőn nincs túl sok, nincs túl kevés információ
9. segítség a hibák észrevételében, javításában
10. megfelelő súgó és dokumentáció

15