

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 elemzésre 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. Elég-e 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



Közös gyakorlat

10

Heurisztikus elemzések (Heuristic Evaluation)



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 koncentrálnak, mint a megoldásokra és csak igen kevés megismételhető.

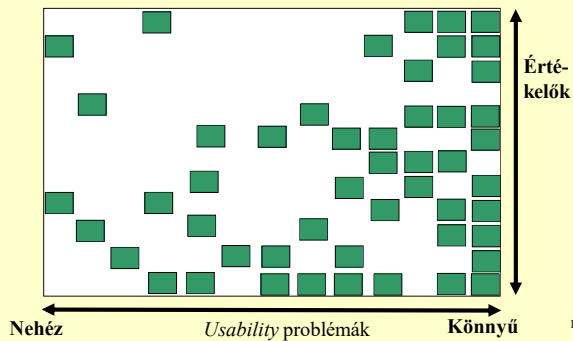
11

12

Heurisztikus elemzés

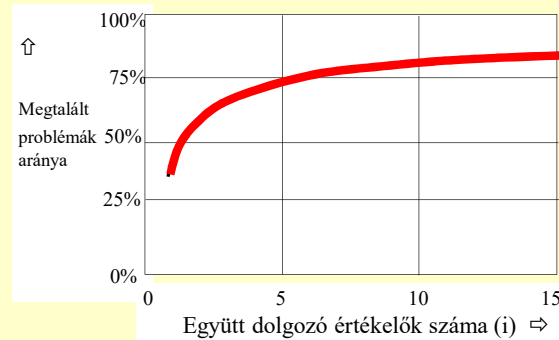


Illusztráció:



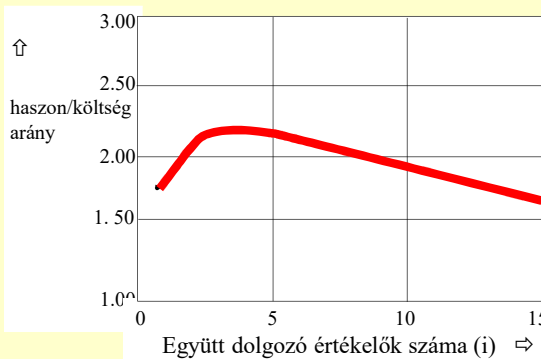
13

Heurisztikus elemzés.



14

Heurisztikus elemzés



15

Heurisztikus elemzés



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

1. megfelelő visszajelzés
2. egyszerű, természetes párbeszéd
3. kontrollálható párbeszéd, „vésszkijárat” nem kívánt funkcióból
4. következetesség szoftveren belül
5. rendszer állapotának, munka előrehaladásának láthatósága
6. kerüli 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

16

Egyszerű empirikus vizsgálatok még nem működő prototípusokkal

Egyszerű empirikus vizsgálatok még nem működő prototípusokkal



- Low-fidelity prototyping – Paper prototyping
- Óz, a nagy varázsló

18

Használhatósági tesztelés papír prototípusokkal (Paper prototyping)

Carolyn Snyder (2003): *Paper prototyping – The Fast and Easy Way to Design and Refine User Interfaces* (Morgan Kaufmann Publishers, San Francisco) című könyve alapján

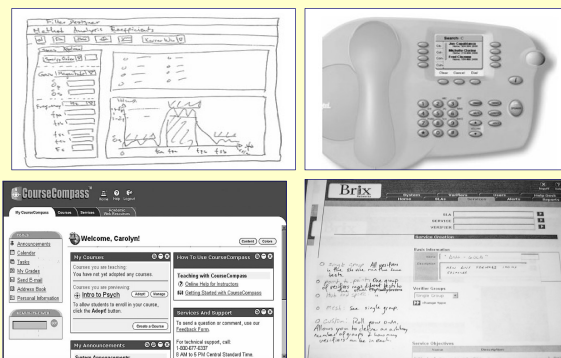
19

1.1 Meghatározás

- A papírprototípus-alapú tesztelés a **használhatósági tesztek** olyan formája,
- ahol a szoftver későbbi felhasználóinak egy **reprezentatív csoportja**
- **realisztikus feladatokat** hajt végre
- azáltal, hogy **interakciót** folytat
- a felhasználói felület **papíron** rögzített verziójával,
- amelyet egy, a számítógép szerepét játszó **valós személy** működtet.

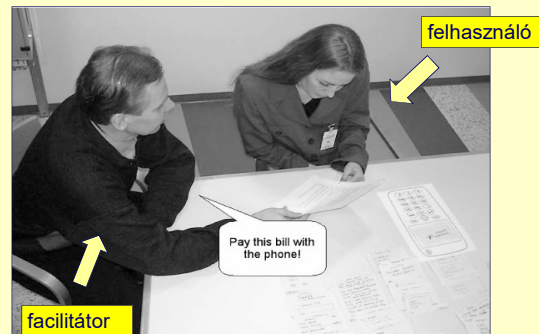
20

1.2 Hogyan néz ki egy papír prototípus?



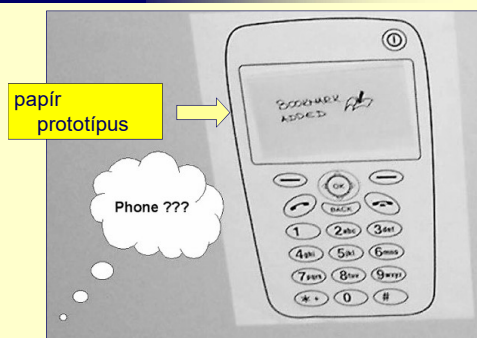
21

1.3 Hogyan néz ki egy használhatósági teszt papír prototípussal?



22

1.3 Hogyan néz ki egy használhatósági teszt papír prototípussal?



23

1.3 Hogyan néz ki egy használhatósági teszt papír prototípussal?



24

2. A módszer előnyei



2.1 Technikai: gyors és olcsó

- A prototípust nem kell leprogramozni, csak megrajzolni
- A kódolás előtt azonosítja a fő problémákat (technikai jellegűeket is)
- Pillanatok alatt módosítható, a módosítás helyben kipróbálható
- A használhatósági tesztet el lehet akkor is kezdeni, ha a fejlesztés késik
- Nem csak a hibák azonosítására, hanem a designnal kapcsolatos kérdések tisztázására is használható

25

2. A módszer előnyei



2.2.1 Pszichológiai: felhasználók

- Ha nem tűnik befejezettnek az alkalmazás, a felhasználók kevésbé „fogják vissza” magukat.
- Használható, nem szörszálhasogató kritikát kapunk.
- A műszakilag kevésbé képzeteket sem félemlíti meg.

26

2. A módszer előnyei



2.2.2 Pszichológiai: tervezők, fejlesztők

- Könnyebb megváltoztatni valamit, amibe kevés erőfeszítést fektettünk.
- Elkerülhetők a „hitviták”.



27

2. A módszer előnyei



2.2.3 Pszichológiai: döntéshozók

- Úgy érzik, van beleszólásuk a termék kialakításába.
- Kimutatja, ha a koncepcióval van alapvető probléma.
- Megkönnyíti a kommunikációt.

28

2. A módszer előnyei



Milyen jellegű problémákat találhatunk meg a segítségével?

- Rossz alapkonceptió
- Félrevezető szóhasználat
- Navigációs problémák
- Tartalmi kérdések
- Mit tartalmazzon a súgó
- Elrendezés a képernyőn

M
é
l
y
s
é
g
i

29

3. A módszer hátrányai



Milyen jellegű problémákat **nem** találhatunk meg a segítségével?

Megjelenítési hibákat:

- Ha a betűméret nem megfelelő
- Ha a képekkel gond van
- Szöveg rendezése, zárása papír prototípuson rosszabbnak látszik

Interakciós hibákat:

- Ha a nehezen észrevehető, kis változásokkal van gond, mert ezek a változások papír prototípusnál felnagyítódnak
- Gördítéssel kapcsolatos hibák nem jelennek meg
- Hosszú listák kezelése a valóságban nehezebb
- Félrekattintások, billentyűhibák nem jönnek elő
- Egérkurzor pozicionálásával kapcsolatos nehézségek nincsenek
- Válaszidő nem realisztikus

30

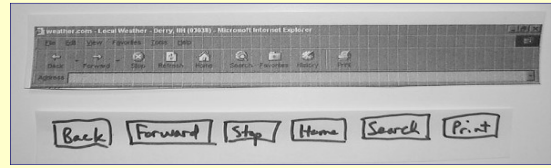
4.1 A feladatok megtervezése

Egy jó tesztfeladat:

- reális és jelentőséggel bír a felhasználók számára
- nem túl általános és nem túl specifikus (NEM: „Hogyan navigálnak a felhasználók az oldalon?”)
- véges számú, meghatározott megoldása van (NE adjunk olyan feladatot, amit lehetetlen végrehajtani)
- felismerhető végpontja van (NEM: „Nézzon körül ezen a szájon!”)
- cselekvésre, nem véleményre vonatkozik (NEM: „Mégis mit gondolt erről a weboldalról?”)
- 5-30 perc alatt megoldható
- leírása tartalmazza a feladat előfeltételeit, a megoldás lépéseit, becsült időtartamát, eredményét, a felhasználónak adandó instrukciót

31

4.2 A papír prototípus elkészítése

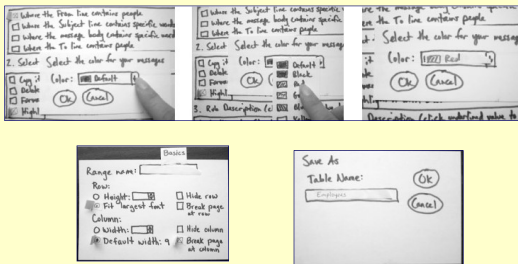


- Lehet nyomtatott képernyőtér is (olvashatóbb, de nehezebben változtatható)
- Nem kell színesnek lennie
- Általában kicsit nagyobb, mint a leprogramozott verzió lesz

32

4.2 A papír prototípus elkészítése

- Példák a felületelemek megjelenítésére és kezelésére:



33

4.3 A teszt végrehajtása

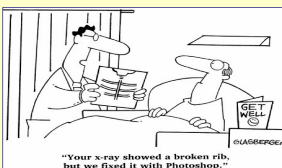
Különleges szempontok papírprototípus-alapú tesztelésnél:

- Nagyobb a helyigény, mint számítógépen történő tesztelésnél (mert több az eszköz és a résztvevő)
- Videófelvétel nem feltétlenül szükséges, mert a tevékenység lassabb, szemmel követhető, a felület pedig tesztől tesztre fejlődhet.
- Ha nincs elég megfigyelő, vagy messzebb vannak, hasznos a felvétel, de felülről kell felvételt készíteni
- Eleinte furcsa érzés a felhasználónak, mert szokatlan az interakció
- A „számítógép” szerepét játszó résztvevőt fel kell készíteni (tudja, mire mit kell reagálnia; ne beszélgesen a felhasználóval; várja meg, mit tesz a felhasználó, és arra reagáljon; ne dűlja szét a berendezést)

34

4.3 A teszt végrehajtása

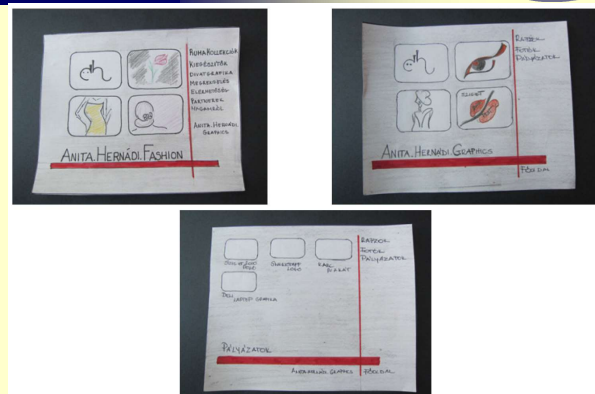
- A facilitátornak arra is figyelnie kell, hogy a „számítógép” helyesen reagál-e, és kezelnie kell azokat a helyzeteket, amire nem készültek fel (egy papír prototípus is „lefagyhat”).



- Ennél a tesztnél tesztülések között, sőt azok közben is meg lehet változtatni a felhasználói felületet (de meg kell nézni, hogy valóban megoldottuk-e a problémát)

35

Példák



Példák



„Óz, a nagy varázsló” módszer

A következő példa egy szélsőséges illusztráció olyan helyzetről, ahol nagy munka lett volna leprogramozni a mesterséges intelligenciát, de a használhatósági vizsgálat során a felhasználó számára már a **működő prototípus élményét** szimuláltuk. (A konkrét példa egyben előzetes is a későbbi virtuális valóság témából és a szemmozgáskövetés témából.)

38



39



40



3D Pointer is shown.