Dr. Mileff Péter

SZOFTVERTECHNOLÓGIA

FELHASZNÁLÓI FELÜLETEK TERVEZÉSE

Miskolci Egyetem
Általános Informatikai Tanszék

Felhasználói felületek

- A felhasználói felület az elsődleges kapcsolódási pont a felhasználó és a számítógép között.
- A jól áttekinthető, gondos felhasználói felület kidolgozása fontos!
- A teljes szoftvertervezési folyamat része kell legyen
  - nagy hangsúlyt kell kapjon
  - a megfelelő színvonalú kidolgozás elengedhetetlen
- **Nem (csak) esztétikai kérdés!**
  - szubjektív
- **Tervezési elvek:**
  - Jól bevált „taktikák”
  - Segítenek a jobb eredmény érérésében

Felhasználói felületek

- **Célszerű:**
  - a kialakítandó felhasználói felület illeszteni a felhasználó szakértelméhez, tapasztalatához és elvárásaihoz.
- A felület félretervezése mindig következményeket von maga után:
  - Pl.: a felhasználó esetleg nem fér hozzá a rendszer bizonyos jellemzőihez, hibázik, és úgy érzi, hogy a rendszer segítség-nyújtás helyett inkább gátolja őt annak a célnak az elérésében.
  - Tipikus példa a verzióváltások esetén
  - Látogatók elpártolhatnak a szoftvertől
  - Anyagi vonzattal járhat!

Felhasználói felületek

- Ma már egy felhasználói felületnek számos megjelenítési formája lehet:
  - egyszerű ablakos vastagkliens alkalmazás,
    - Pl.: office, skype, thunderbird, stb
  - web-es rendszerek,
    - Pl.: webshop-ok, gmail, youtube, stb
  - mobiltelefonok, PDA-k megjelenítési megoldásai,
    - Pl.: Alkalmazások és játékok
  - Pixelgrafikus alkalmazásokról.
    - Pl.: OpenGL, DirectX programok
- **Mindegyik külön tervezést igényel!**
  - alkalmazkodni kell az eszközökhöz és környezethez
  - figyelembe kell venni a szoftvert használó emberek fizikai és mentális képességeit.

Néhány probléma

- Az emberek korlátozott rövid távú memóriával rendelkeznek.
  - Ha egyidejűleg túl sok információt kapnak, nem tudják azokat befogadni.
- Rendszerprobléma esetén a figyelmeztető üzenetektől megijednek a felhasználók
  - feszültebbé válnak, ami növeli a hibázás esélyét.
- Nem szabad saját képességeinkre tervezni a rendszert
  - feltételezni, hogy majd minden felhasználó képes lesz megbirkózni a felülettel.
  - Különböző interakciós módokat részesítünk előnyben.
    - Pl.: Linux console vs KDE menük
  - Némelyek képekkel, mások szövegekkel szeretnek dolgozni.

5

A FELÜLETEK TERVEZÉSI ELVEI...

6

Felületek tervezési elvei

- Vannak általános elvek, amelyek minden felhasználói felület tervére alkalmazhatók.
  - speciális rendszertípusok számára részletesebb tervezési irányelvek ezekből származtathatók
- Tervezésnél a kiindulópont az emberi képességek
- **1. Felhasználói jártasság:**
  - Olyan kifejezéseket és fogalmakat kell használnia, amelyek megfelelnek a rendszert legtöbbet használó emberek tapasztalatainak.
  - Nem szabad egy felületet csak azért ráerőltetni a felhasználókra, mert az kényelmesen implementálható.

7

Felületek tervezési elvei

- **2. Konzisztencia:**
  - a felületnek a hasonló műveleteket hasonló módon kell realizálnia.
  - **Jelentése:**
    - a rendszer parancsainak és menüinek ugyanazzal a formátummal kell rendelkezniük,
    - a paramétereket minden parancshoz ugyanúgy kell átadni, és hasonlónak kell lennie a parancsok formájának.
- **3. Minimális meglepetés:**
  - a rendszer soha ne okozzon meglepetést a felhasználóknak
  - ingerültté válnak, ha nem a várt módon viselkedik.
  - **Oka:** a felhasználók felépítik magukban a rendszer működésének gondolati modelljét.
    - Az eltérő logikai részek gondot okoznak

8

Felületek tervezési elvei

- **4. Visszaállíthatóság:**
  - a felületnek rendelkeznie kell olyan mechanizmusokkal, amelyek lehetővé teszik a hiba után történő visszaállítást.
    - **A. Az ártalmas tevékenységek megerősítése:** ha a felhasználók ártalmas tevékenységet hajtanak végre, megerősítést kell tőlük kérni.
    - **B. Visszavonási lehetőség biztosítása:** a visszavonás a rendszert visszaállítja a tevékenység bekövetkezése előtti állapotba.
    - **C. Ellenőrző pontok készítése:** célszerű a rendszer állapotát bizonyos időközönként elmenteni.

9

Felületek tervezési elvei

- **5. Felhasználói útmutatás:**
  - hiba esetén értelmes visszacsatolás a felhasználónak
  - Beépített súgó rendszer
- **6. Felhasználói sokféleség:**
  - Különböző felhasználóknak különböző interakciós lehetőségek biztosítása
    - A rendszernek lehetnek alkalmi és rendszeres felhasználói.
    - Pl.: regisztrált tagoknak többletfunkciók biztosítása

10

A FELÜLETEK TERVEZÉSI FOLYAMATA...

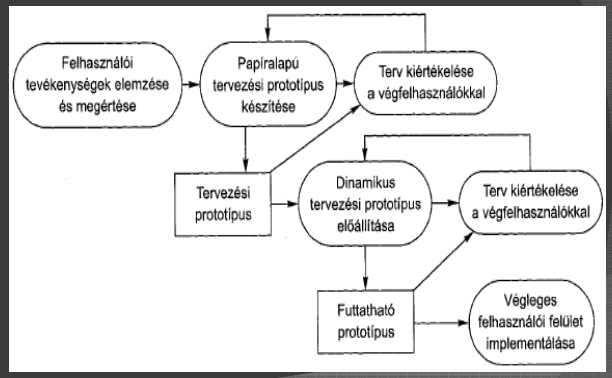
11

A folyamat jellemzői

- **Folyamat:**
  - a felhasználók együttműködnek a tervezőkkel,
  - prototípusokat készítenek a rendszer felületéről.
    - Design tervek, skicc rajzok, stb
  - Ezek alapján döntenek:
    - Milyen legyenek a felhasználói felületének jellemzői,
    - Milyen legyen a felépítése
    - Hogy nézzen ki
- Célszerű a felületek tervezését iteratív úton fejleszteni
- A prototípus külön is elkészíthető
  - párhuzamosan más szoftvertervezési tevékenységekkel
  - ez gyorsítja a rendszer kifejlesztését

12

A felület tervezési folyamata



A folyamat fő szakaszai

- A folyamat három alapvető szakasza:
  - **Felhasználók elemzése:** meg kell vizsgálni, hogy
    - a felhasználók milyen feladatokat végeznek,
    - milyen munkakörnyezetben dolgoznak,
    - milyen egyéb rendszereket használnak, munkájuk során.
  - **Prototípus-készítés:** a felhasználói felület terve alapján prototípust kell készíteni.
  - **Felületek értékelése:** a prototípus formálisabb értékelése.
    - A felhasználók interfészhasználat közben szerzett aktuális tapasztalatait gyűjtjük össze.

A FELHASZNÁLÓK ELEMZÉSE...

A felhasználók elemzése

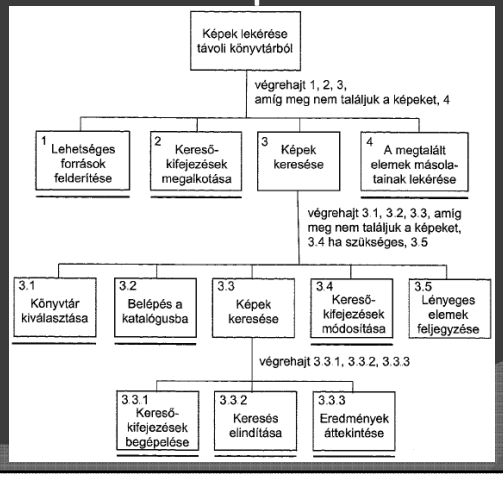
- A tervezési folyamat alapgondolata:
  - elemezni kell azokat a felhasználói tevékenységeket, amelyeket a rendszernek támogatnia kell.
  - Meg kell érteni a szoftver, és a felületek pontos célját.
- **Három alapvető technika:**
  - **feladatelemzés**,
  - **interjúztatás** vagy **kérdőívek kitöltetése**,
  - **etnográfia**.
- A feladatelemzés és az interjúztatás az egyénre és az egyén munkájára összpontosít
- Az etnográfia az emberi kapcsolatokat vizsgálja

Feladatelemzés

- Legelterjedtebb módszer: **Hierarchikus feladatelemzés** (Hierarchical Task Analysis - HTA)
- **Eredetileg:** a felhasználói kézikönyvek készítésének elősegítésére hozták létre
  - de használható annak meghatározására is, hogy mit kell a felhasználóknak tenniük egy bizonyos cél elérése érdekében.
- **A HTA elemzés alapja:**
  - a magas szintű feladatot részfeladatokra bontjuk.
  - terveket készítünk, hogy megadjuk, hogy egy adott helyzetben mi történhet.
  - A felhasználó céljából kiindulva egy hierarchiát készítünk, amely leírja, hogy mit kell tennünk a cél érdekében.
- **Jelölésrendszere:**
  - Feladat/részfeladat jele a doboz
  - Doboz alatti vonal azt jelöli, hogy az adott feladatot nem bontjuk részfeladatokra.

17

HTA példa



18

HTA jellemzői

- **Előnye:** (a természetes nyelvi forgatókönyvekkel szemben)
  - rákényszerít, hogy minden feladatot átgondoljunk és eldöntsük, hogy felbontható-e, avagy sem.
  - A forgatókönyvek használata során könnyen kimaradhatnak fontos feladatok,
  - a forgatókönyvek hosszúvá és unalmassá válhatnak, ha sok részletet szeretnénk beleírni.
- **Hátránya:**
  - leginkább csak a szekvenciális folyamatokként leírható feladatokra alkalmazható.
  - konkurens tevékenységek leírásakor a jelölésmód kényelmetlenné válik.

19

Feladatelemzés

- Nemcsak az információk begyűjtése a fontos.
- Azokat valamilyen formában dokumentálni kell.
- **Fontos a leírás módja:**
  - A felhasználói elemzéseket kell leírni hatékony módon.
  - Hogy az alapján a feladatok lényegét mind a többi tervező, mind pedig maguk a felhasználók felé kommunikálni tudják.
- A leírásra nincs követendő egyezményes eszköz.
  - Hatékony megoldást kínál az UML szekvencia diagramja.
    - Bizonyos felhasználóknak túl technikai lehet
  - Természetes nyelv mindig jó megoldás
    - Kiegészítve tetszőleges grafikai elemekkel

20

PROTOTÍPUS KÉSZÍTÉSE...

21

Prototípus készítés

- **Cél:** a felületekkel szemben támasztott követelmények kifejezése valamilyen formában
- A szöveges leírások és a diagramok nem elég jók
  - **Oka:** a felhasználói felületek mindig dinamikus folyamatot reprezentálnak.
  - Nem képesek kifejezni és leírni a folyamatot teljes egészében.
- **Hatékony megoldás:**
  - a végfelhasználó bevonásával végzett prototípus-készítés
  - egyetlen hatékony módja a grafikus felhasználói felületének tervezésének és fejlesztésének.

22

Prototípus készítés

- **A prototípus-készítés célja:**
  - egy elkészült példafelület segítségével a tervezők tapasztalatokat szerezzenek a működéssel kapcsolatban.
- A folyamatot két lépésben hajtjuk végre:
  - 1. A folyamat elején **papíralapú prototípusok**at készítünk, és ezeket áttekintjük a végfelhasználókkal.
    - a képernyőtervek, parancssori interfészeket
  - 2. **Finomítjuk a tervet:** egyre kifinomultabb automatizált prototípusokat fejlesztünk,
    - Elkészítjük a számítógépes reprezentációját (Rajzolás, concept art)
    - tesztelésre és a tevékenységek szimulálása érdekében odaadunk a felhasználóknak.

23

Prototípus készítés

- **A papír alapú prototípusok előnyei:**
  - Olcsó és hatékony
  - Nem kell futtatható programot készíteni
  - A terveket sem kell profi módon megrajzolni
  - Csak azon képernyőket kell papírra lerajzolni, amelyeken kapcsolatba lépünk a rendszerrel
- A kezdeti tapasztalatok után a felület tervének szoftveres prototípusát kell implementálnunk.

24

Prototípus készítés

- **Három implementációs lehetőség:**

1. Szkriptvezérelt megközelítés:

- vizuális elemeket (gombokat, menüket stb.) tartalmazó képernyőket hozunk létre valamilyen szerkesztő eszköz segítségével
- ezekhez szkripteket társítunk.
- A felhasználói interakció során a szkript végrehajtásra kerül,
  - Pl.: megjelenik a következő képernyő

25

Prototípus készítés

- **2. Vizuális programozási nyelvek:**
  - Egy vizuális programozási nyelv segítségével gyorsan létrehozhatunk felületeket,
  - amely objektumaihoz komponenseket és szkripteket társíthatunk.
- **3. Internetalapú prototípus-készítés:**
  - ezek a megoldások egy web böngészőn és egy nyelven alapulnak.
  - A funkcionalitást ekkor a nyelv nyújtja (pl.: Java).
  - Ezek a kódrészletek az oldal böngészőbe töltésekor automatikusan végrehajtódnak

26

PROTOTÍPUS ÉRTÉKELÉSE...

27

Prototípus értékelése

- **Értékelés célja:**
  - megvizsgáljuk, hogy mennyire használható az adott felület,
  - ellenőrizzük, hogy megfelel-e a felhasználói követelményeknek.
- Az ellenőrzési folyamat ugyanolyan fontos, mint a tervezés, vagy a prototípus készítése!
  - Nem szabad elhanyagolni
- Szubjektív, de vannak alapvető kiértékelési jellemzők:
  - **Tanulhatóság:** mennyi idő szükséges a rendszer megtanulásához
  - **Műveleti sebesség:** a rendszer válaszideje megfelelő-e.
  - **Robusztusság:** milyen a rendszer hiba tűrőképessége.
  - **Visszaállíthatóság:** hibák esetén milyen lehetőségek vannak a visszaállításra.
  - **Adaptálhatóság:** mennyire kötött a rendszer modellje.

28

Prototípus értékelése

- **Számos eszköz a kiértékelés segítésére:**
  - **Kérdőívek:** arról gyűjtenek információt, hogy a felhasználók mit gondolnak a felületről.
    - Főleg speciális kérdések. Pl.: értékelje 1-5-ig
  - **Megfigyelés:** a rendszer felhasználóinak munka közben történő megfigyelése.
  - **Videó felvétel:** a jellegzetes rendszerhasználat videó „pillanatfelvételei”.
  - **Naplózás:** olyan kódrészlet beépítése a szoftverbe, amely a legtöbbször használt lehetőségekről és a leggyakoribb hibákról gyűjt információt.
    - Leghatékonyabb módszer

29

Felhasználói felületek tervezése (Összefoglalás)

- **A felhasználói felületek tervezése fontos!**
  - Ugyanúgy, mint minden más fejlesztési folyamat.
  - Nem csak grafikai tervezés
- A rossz UI nagymértékben rontja a szoftver minőségét
- Egy jó GUI kidolgozása szakértelmet kíván
  - Szakértő (grafikus és tervező) bevonása mindig költséges,
  - Ez később megtérül.

30

Köszönöm a figyelmet!

31