

Testování uživatelských rozhraní. Definice pojmu použitelnosti (usability), modely úloh, prototypování uživatelských rozhraní. Testování použitelnosti (organizace a vyhodnocování testů) (A4B39TUR)

June 10, 2012

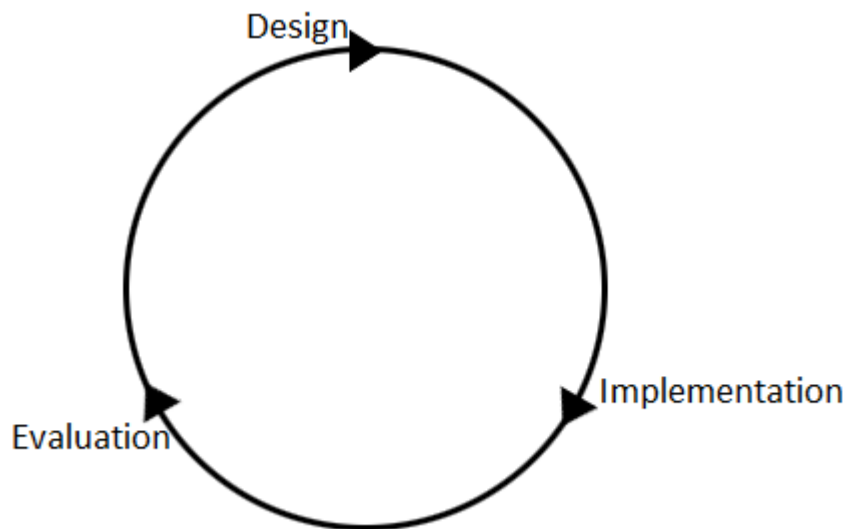
Ověření, jak systém budou používat reální uživatelé, pochopení jak ho budou používat, identifikace problémů.

1 Usability (použitelnost)

- **jednoduchost použití** – podobné systémy lze ovládat lépe (naučit se rychleji)
- **recall** – rozpomenutí se, jak se taková situace řeší
- **efektivita** – zvládnout úkol rychle a snadno
- **minimum chyb** – v případě chyby korektně informovat a říci další postup
- **spokojenost uživatele** – přesvědčit uživatele, že úkol dokončil

2 Návrh UI – 3 fáze v cyklu

- **design** – návrh dle psychologie a toho, co uživatel potřebuje
- **evaluation** – prototyp a vyhodnocení s uživateli
- **implementace** – vytvoření prototypu a návrhu



- pro vyhodnocování je nutné zavést „přesnou“ metriku hodnocení tak, aby návrháři pochopili problém a mohli ho vylepšit
- záleží na cílové skupině a jejich schopnostech

3 Testování bez uživatelů

3.1 Kognitivní průchod

- Nejprve musíme znát všechny funkce a jak systém funguje. Poté se vrátíme na začátek a postupujeme jako nezkušený uživatel
- vhodné pro ověřování „klasického“ UI, levné a rychlé testování (ale málo vypovídající)

3.1.1 Vstup

- identifikace uživatelů (cílové skupiny) – jejich zkušenosti a znalosti
- stanovení úkolů a cílů, čeho chceme dosáhnout (Např. automat -> chceme koupit jízdenku)
- stanovení postupů, jak dané akce chceme dosáhnout (Např. vybrat destinaci -> vybrat slevu -> vhodit mince -> vzít jízdenku ...)
- stanovení, co by se mohlo stát (Např. zjištění, že nemáme dostatek peněz -> storno -> vrácení peněz)

3.1.2 Výstup

- seznam nálezů a jejich hodnocení
- ve výsledném reportu popíšeme záporné výsledky, ale nenavrhujeme správné řešení (to je na vývojářích -> poté další testování)

Na začátku úkolu zodpovídáme nultou otázku

Q0: Ví uživatel, co chce udělat, čeho chce dosáhnout?

V každém kroku úkolu zodpovídáme otázky:

Q1: Ví uživatel, co má v tomto kroku udělat?

Q2: Propojí si uživatel nadpis, obrázek, vzhled s danou akcí?

Q3: Dostane uživatel rozumnou, jasnou zpětnou vazbu?

3.2 Heuristická evaluace

- identifikace problému a návrhu UI, obecná doporučení pro správný design
- vyhodnocováno 3-5 lidmi -> každý může daný problém vidět jinak – ujistí se, že neporušuje žádné heuristiky (=teorie řešení), jinak jsou sepsány do zprávy podle nálezů v porušení heuristiky
- je méně formální

3.2.1 Nielsenova heuristika (1994)

1. **Viditelnost stavu systému** - uživatel musí být informován informován o stavu (co se právě děje)
2. **Vztah mezi systémem a skutečným světem** – systém musí „mluvit“ jako uživatel v reálném světě
3. **Uživatelská kontrola a svoboda** – musí mít na výběr, možnost proces přerušit
4. **Konzistence a standardy** – dvě stejná označení musí dělat to samé
5. **Prevence chyb** – varování jestli to opravdu chce (Format c: Yes - No)
6. **Lépe připomenout než vzpomínat** – nezatěžovat uživatele zadáváním toho, co mu mohu nabídnout/napovědět
7. **Flexibilita a efektivita využití** – podpora zkušených uživatelů pro urychlení práce – zkratky
8. **Minimalistický design** – podávat jen informace, které jsou nutné
9. **Pomoc při diagnostice a zotavení z chyb** – chybové hlášky v lidské řeči (ne kus kódu), navrhnout řešení
10. **Pomoc a dokumentace** – v případě problému poskytnout dokumentaci a přesný postup

3.2.2 Postup při heuristické evaluaci

1. **Tutorial** – seznámení testu s aplikací (není vždy nutné)
2. **Evaluate** – alespoň 2 průchody – 1. Projít systém, 2. Zaměřit se na specifické problémy
 - hlášení chyb jednotlivě a od každého testu odděleně
 - jakou heuristiku porušuje a lehké doporučení (náznak) řešení, případný komentář
3. **Stanovení priorit** – označení nálezu podle závažnosti (alespoň 5úrovní)
 - 1 test nalezne cca 35% problémů, 5 testů zhruba 75% problémů

3.3 Kognitivní průchod vs. Heuristická evaluace

3.3.1 Kognitivní průchod

- informace podle psychologie
- jeden tester/odborník
- více formální
- lepší pro hodně strukturované úkoly

3.3.2 Heuristická evaluace

- informace podle návrhových zvyků (praktice)
- více testerů
- méně formální
- vhodný pro jednodušší úkoly

3.4 KLM (keystroke level model)

- Pro měření časové náročnosti, metoda pro ohodnocení UI
- vstupem je popis úkolů
- založeno na ideálním průchodu, stanoveny časové údaje pro akce (i dle zkušeností)

3.4.1 KPHMR

K-stisk tlačítka

P-zaměření cíle myši

H-přehmat z klávesnice-myš

M-mentální příprava (myšlení)

R-reakce systému

Příklad - uložení souboru:

zaměření menu -> klik -> zaměření uložit -> klik...

total time = P+K+P+K...+mentální příprava před provedením akce

4 Testování s uživateli

- sledování je v jejich prostředí (zvyklost a „pohodlné prostředí“) – těžké na organizaci, časově náročnější
- kontrolovaný test v usability labu (riziko stresu, atp.) – soustředění jen na test, možnost opakování
- problémem je vybrat správné lidi (nejlépe pomocí screeneru)
- vhodný počet participantů (často 5-10)
- před prvním testem nejlépe vše vyzkoušet sami na sobě (vyvarování se chyb a nedostatků v popisu – **Pilot test**)

4.1 Fáze testování:

- výběr na základě „**screeneru**“ – vhodná cílová skupina, zkušenosti – oslovení k testu
- „**pre-test**“ dotazník (veřejná a neveřejná část) + instrukce k testování – nechat čas na přečtení, ...
- samotný test (procházení úkolů)
- „**post-test**“ dotazník
 - odpovědi tak, aby se daly vyhodnotit (ano, ne, ohodnocení, ...)
 - otevřené otázky – možnost vyjádřit se