

UNIVERZITA KARLOVA V PRAZE
FILOZOFICKÁ FAKULTA
ÚSTAV INFORMAČNÍCH STUDIÍ A KNIHOVNICTVÍ

Studijní program: informační studia a knihovnictví

Studijní obor: informační studia a knihovnictví

Richard Pecl

**Základní metodické postupy
při tvorbě uživatelského rozhraní**

Bakalářská práce

Praha 2006

Vedoucí bakalářské práce: PhDr. Richard Papík, Ph.D.

Oponent bakalářské práce:

Datum obhajoby:

Hodnocení:

upraviti → THESIS ASSIGNMENT PAGE

Prohlášení

Prohlašuji, že jsem svou bakalářskou práci vypracoval samostatně a že jsem zde uvedl všechny informační zdroje a jiné podklady v této práci použité.

V Praze, 16. srpna 2006

podpis

Identifikační záznam

PECL, Richard. *Základní metodické postupy při tvorbě uživatelského rozhraní* [*Basic methodological techniques in user interface design*]. Praha, 2006–08–16. ix, 47 s., 5 s. příl. Bakalářská práce. Univerzita Karlova v Praze, Filozofická fakulta, Ústav informačních studií a knihovnictví. Vedoucí bakalářské práce Richard Papík.

Abstrakt

S neustálým vývojem výpočetních technologií roste i význam jejich možností ve vztahu k uživatelům. Počet lidí přicházejících do kontaktu s počítačovými systémy se rapidně zvětšuje. Ti, společně se stávajícími uživateli, vyžadují, aby pro ně bylo jednoduché naučit se ovládat moderní techniku, aby práce s ní byla rychlá, příjemná a účinná, aby počítače byly výkonnými pomocníky. Splnit tyto požadavky znamená vytvářet pro uživatele přívětivá uživatelská rozhraní. A právě o to se snaží vědní obor HCI – interakce člověk–počítač.

Tato bakalářská práce podává přehled požadavků na použitelnost uživatelských rozhraní počítačových systémů, souvisejících norem a jiných standardů a především pak přehled dnes používaných metodik pro tvorbu uživatelských rozhraní. Část textu je věnována také popisu postupů při zjišťování použitelnosti UI.

.[Autorský abstrakt].

Klíčová slova

HCI
(interakce) člověk–počítač
interaktivní systémy
použitelnost
software
(grafická) uživatelská rozhraní
vizualizace dat

Obsah

Předmluva	1
1 Úvod	3
2 Vývoj uživatelského rozhraní	5
2.1 Normativní dokumenty pro použitelnost UI	6
2.2 Terminologie	9
2.3 Fáze vývoje	11
2.4 Metodické postupy	13
2.4.1 Strukturovaný návrh uživatelského rozhraní s optimalizací interakce, pohledů a interakce, STUDIO	14
2.4.2 Modelování uživatelského rozhraní, UIM	14
2.4.3 Architektonický rámec entita, úloha, reprezentační prvek, ETP	14
2.4.4 Bridge	14
2.4.5 Návrh a vyhodnocení grafického uživatelského rozhraní, GUIDE	15
2.4.6 IDIOM	18
2.4.7 Návrh orientovaný na použití	18
2.4.8 Logický dialogový návrh orientovaný na uživatele, LUCID	20
2.4.9 Návrh objektů, pohledů a interakce, OVID	21
2.4.10 Wisdom	22
2.5 Analýza požadavků na použitelnost UI	25
2.5.1 Analýza potřeb uživatelů	26
2.5.2 Analýza úloh	26
2.6 Návrh způsobu realizace UI	26
2.6.1 Návrh s účastí uživatele (participatory design)	28
2.7 Měřítko použitelnosti UI	29
2.8 Způsoby zjišťování použitelnosti UI	29

2.8.1	Odborná revize (expert review)	29
2.8.2	Uživatelské testování použitelnosti v laboratořích (usability testing)	31
2.8.3	Průzkumy, dotazníky	35
2.8.4	Testy přijetí (acceptance tests)	36
2.8.5	Vyhodnocení během aktivního používání	36
2.8.6	Řízené psychologicky orientované experimenty	37
3	Závěr	38
	Seznam použité literatury	39
	Seznam použitých symbolů a zkratk	45
	Rejstřík	46
A	Vybrané standardy upravující tvorbu UI z hlediska použitelnosti	I
B	Přehled dotazníků používaných pro ověřování použitelnosti UI	III

Seznam obrázků

2.1	Rozdíly tradičního projektování a návrhu zaměřeného na uživatele	6
2.2	Komponenty použitelnosti	10
2.3	Vzájemné souvislosti činností ergonomického projektování	12
2.4	Schéma postupu prací během návrhu metodikou GUIDE	17
2.5	Typické pracovní prostředí pro PICTIVE	28
2.6	Typické uspořádání laboratoře pro testování použitelnosti	31
2.7	Závislost poměru výhody/cena na počtu účastníků testování	34
B.1	Dotazníková forma karet odezvy na produkt (Product Reaction Cards)	IV
B.2	Dotazník na použitelnost počítačových systémů	V

Seznam tabulek

2.1	Obsah jednotlivých částí sady norem ČSN EN ISO 9241	8
2.2	Kroky projektování UI metodikou GUIDE a jejich výstupy	16
A.1	Vybrané standardy upravující tvorbu UI z hlediska použitelnosti	II
B.1	Vybrané dotazníky používané pro ověřování použitelnosti UI	III

Předmluva

Na konci 20. století jsme byli svědky ohromného rozvoje počítačových technologií. Jejich používání už není vyhrazeno jen odborníkům, ale do kontaktu s počítačem přicházíme denně téměř všichni. Tento rozmach pokračuje s neustávající rychlostí i dnes a společně s tím vznikají potřeby umožnit i široké laické veřejnosti ovládat moderní technologie pohodlně, jednoduše, příjemně a efektivně.

V této práci se soustřeďuji na představení hlavních výsledků dosavadního vědního zkoumání v oboru interakce člověk–počítač. Podávám zde přehled nejvhodnějších postupů používaných při tvorbě *grafického uživatelského rozhraní*. Takové znalosti lze uplatnit při projektování a implementaci různých informačních systémů běžících přímo na počítači uživatele nebo i zpřístupněných přes rozhraní world wide web. Mohou to být např. systémy pro pořádání firemní dokumentace, knihovnické systémy, různé internetové vyhledávače a katalogy, webová rozhraní ke všem možným bázím dat atd.

Často dnes vidíme, že i renomované firmy nabízejí své produkty, jejichž uživatelská rozhraní vykazují při hodnocení použitelnosti základní chyby a nedostatky. Proto je více než vhodné, aby i informační pracovníci jako tvůrci informačních systémů měli určitou představu, jaká kritéria má dobře navržené uživatelské rozhraní splňovat a jakým způsobem toho lze docílit. Naštěstí se tzv. *použitelnost* uživatelského rozhraní stává stále důležitější součástí návrhu interakce člověka s počítačem a již delší dobu je zkoumána na vědecké úrovni, takže v dnešní době máme dostatek poznatků, které můžeme v praxi aplikovat a činit tak uživatelská rozhraní informačních systémů přívětivější, lákavější a efektivnější.

Hlavními zdroji informací pro tuto práci byly články v odborných periodikách (především se zaměřením na obor HCI), publikace a některé disertační práce vesměs vyhledávané v oborových bázích bibliografických dat a katalogích knihoven. V seznamu použité literatury jsou bibliografické záznamy strukturovány dle ČSN ISO 690 a ČSN ISO 690-2, vzestupně uspořádány podle prvního údaje a roku vydání a poté očíslovány. V textu jsou pak odkazovány citacemi pomocí prvního údaje a roku vydání v hranatých závorkách.

Rád bych zde poděkoval doktoru Richardu Papíkovi, vedoucímu mé bakalářské práce, za zajímavé náměty během výběru jejího zaměření, ochotnou pomoc při hledání informací k tématu a podnětné připomínky týkající se jejich zpracování. Především bych pak rád poděkoval za jeho osvícené vedení Ústavu informačních studií a knihovnictví, které studentům umožňuje do značné míry směřovat své vzdělání podle jejich zájmů a uvážení.

Zároveň děkuji také všem vyučujícím, se kterými jsem se během studia na ÚISK setkal a kteří mi pomohli naplnit cíl studia – rozšířit obzory mých vědomostí a objevit nové zajímavé poznatky.

Kapitola 1

Úvod

Ať už na počítači pracujeme, čteme denní zpravodajství, pouštíme hudbu či filmy, nebo ho využíváme příležitostně při získávání informací o dopravních spojích, při hledání dokumentů pomocí elektronického katalogu knihovny a podobně, vždy s ním musíme nějakým způsobem komunikovat. Musíme mu předat naše požadavky na to, co má vykonat, a on nám zase musí sdělit výsledky svých operací. Tomuto procesu komunikace říkáme *interakce člověk–počítač*, anglicky *human-computer interaction*, *HCI*.

Abychom mohli počítače ovládat efektivně, musíme pro interakci s nimi vytvořit vhodné prostředky – jak hardwarové (např. klávesnice, myši, monitory, tiskárny, čtečky pro slabozraké, pedály a řídicí páky v letadlech, snímače otisků prstů), tak softwarové (tj. nabídky s příkazy, tlačítka, zobrazování textů a obrázků atd.). Takové nástroje pak souhrnně nazýváme *uživatelské rozhraní*, anglicky *user interface*, *UI* nebo také *man-machine interface*, *MMI*. V oblasti tvorby softwaru jsou pak termínem *uživatelské rozhraní* obvykle míněny újeji jen softwarové prostředky a především se pak v tomto oboru používá termín *grafické uživatelské rozhraní*, anglicky *graphic user interface*, *GUI*, označující UI využívající pro zobrazení informací kromě textu také výhod počítačové grafiky (oken, tlačítek, nabídek, klávesových zkratk, ikon, grafického kurzoru apod.).

HCI jako vědní obor zkoumá interakci mezi člověkem a počítačem a navrhuje nejvhodnější postupy a řešení tvorby uživatelských rozhraní z hlediska *použitelnosti* – tj. z hlediska efektivnosti, výkonnosti a spokojenosti s užíváním. Je nasnadě, že se jedná o velmi širokou víceoborovou disciplínu využívající poznatky z počítačových věd, lingvistiky, psychologie, antropologie, fyziologie, grafického designu a mnoha dalších.

V následujícím textu se soustřeďuji na popis existujících metodik¹ pro tvorbu

¹ K použití pojmů *metodika* a *metodologie*: v anglicky psané literatuře je užíván pojem

uživatelských rozhraní vedoucích k vývoji takových UI, která by co nejvíce splňovala současná kritéria pro jejich použitelnost. Práce může být použita jako základní průvodce mezi aktuálními trendy v oblasti projektování a testování uživatelských rozhraní. Neobsahuje podrobné charakteristiky užívaných postupů, ale poskytuje náhled do jedné z oblastí široké moderní disciplíny HCI.

„methodology“, který je do češtiny překládán slovy „metodologie“ nebo „metodika“, obě česká slova jsou chápána synonymicky a vzájemně zaměňována: například Český normalizační institut určil jako překlad termínu „methodology“ pro sérii norem ISO 9000 oba zmíněné tvary a definoval je jako: „1. odvětví nebo znalosti, které se týkají metod a jejich aplikace v jednotlivých oblastech; 2. studium empirických zkoumání nebo postupů jejich použití“. V této práci volím užívání pojmu „metodika“ pro jeho snazší čitelnost a menší komplikovanost.

Kapitola 2

Vývoj uživatelského rozhraní

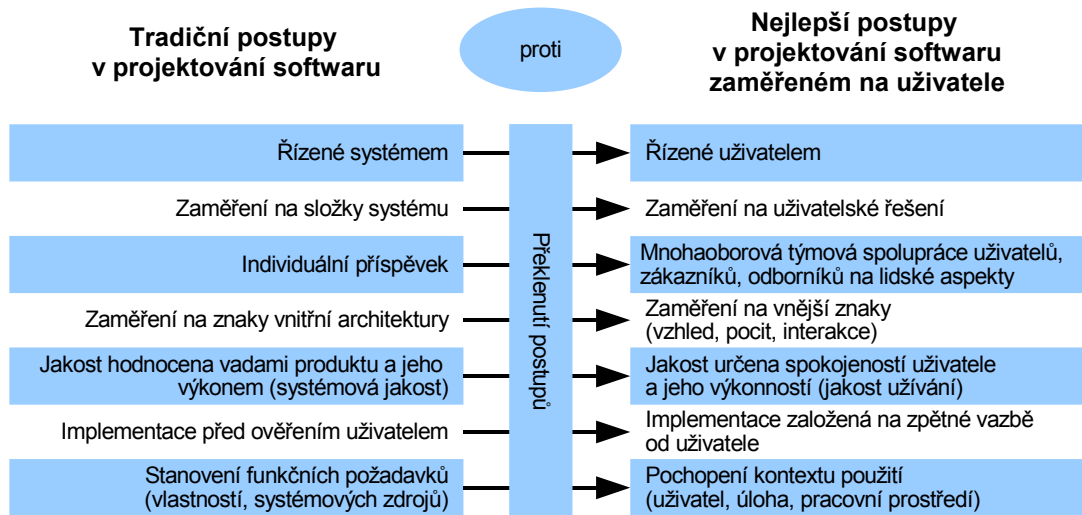
Během tvorby počítačového softwaru (včetně internetových, např. vyhledávacích, služeb) je nutné nejen stanovit funkce, jaké bude systém nabízet, ale také navrhnout formu, jakou budou tyto funkce prezentovány uživateli – tj. zhotovit uživatelské rozhraní.

Přístup k tvorbě uživatelských rozhraní se v průběhu historie měnil jak na akademické úrovni, tak u vývojářů. Vznikly tak tři pojetí projektování počítačových systémů ve vztahu k uživateli [Papík, 2001]:

1. *projektování řízené systémem (system- and technology-driven)* – člověk se měl systému přizpůsobit (probíhá od 50. let do začátku 80. let 20. stol.)
2. *projektování zaměřené na uživatele (user-centered design, UCD)* – systémy již respektují uživatele, ale jeho role je při návrhu stále pasivní
3. *projektování s pomocí uživatele a učící se systémy (learner-centered design)* – uživatel je do návrhu systému vtažen od začátku, systémy jsou schopny se samy přizpůsobovat požadavkům a zvyklostem uživatele; překrývá se s projektováním zaměřeným na uživatele

Rozdíly mezi prvním a druhým pojetím jsou znázorněny na obrázku 2.1.

Dnes je pro tvorbu UI nosný poslední přístup: *projektování s pomocí uživatele* (běžně se však stále nazývá *user-centered design, návrh zaměřený na uživatele*). Výstižně jej charakterizuje otázka, kterou bychom si na počátku projektování UI měli klást: „Kdo bude používat tento produkt?“, zdůrazňující slovo „kdo“. Po jejím zodpovězení pak navrhujeme systém, který bude odpovídat požadavkům uživatele, jeho zvyklostem, schopnostem, technickým možnostem a umožní mu co nejefektivněji a co nejpříjemněji provádět jeho pracovní úkoly.



Obrázek 2.1: Rozdíly tradičního projektování a návrhu zaměřeného na uživatele [IBM, 2006, přeloženo z angličtiny]

2.1 Normativní dokumenty pro použitelnost UI

Ještě než se začneme zabývat konkrétními požadavky na ergonomii a použitelnost UI, uvedu zde přehled základních norem upravujících tuto problematiku. Z nich pak vybírám některé definice pojmů, které jsou použity v následujících částech této práce.

Ačkoliv nejnovější poznatky oboru najdeme samozřejmě v právě vyšlých vědeckých publikacích, obecné i velice konkrétní přístupy k tvorbě UI jsou již normované. Především se touto problematikou zabývají mezinárodní normy

- *EN ISO 6385 Ergonomic principles in the design of work systems*
- *EN ISO 13407 Human-centred design processes for interactive systems*
- *EN ISO 9241-1 až EN ISO 9241-17 Ergonomic requirements for office work with visual display terminals (VDTs).*

Všechny tři mají statut českých národních norem pod označením

- *ČSN EN ISO 6385 Ergonomické zásady navrhování pracovních systémů [ČSN EN ISO 6385, 2004]*
- *ČSN EN ISO 13407 Postupy ergonomického projektování interakčních systémů [ČSN EN ISO 13407, 2000]*

- *ČSN EN ISO 9241-1 až ČSN EN ISO 9241-17 Ergonomické požadavky na kancelářské práce se zobrazovacími terminály; první část viz ČSN EN ISO 9241-1 [1998]*

ČSN EN ISO 6385

Ergonomické zásady navrhování pracovních systémů

Tato norma je mezi normami zabývajícími se ergonomií produktů považována za základní. *Stanovuje základní ergonomické zásady jako směrnici pro navrhování pracovních systémů a definuje související základní pojmy. Popisuje integrovaný přístup k navrhování pracovních systémů při spolupráci ergonomů s dalšími projektovými pracovníky s důrazem na lidské, sociální a technické požadavky [ČSN EN ISO 6385, 2004].*

ČSN EN ISO 13407

Postupy ergonomického projektování interakčních systémů

Tato norma se zaměřuje především na způsob řízení vývoje ergonomických produktů. Je určena zejména manažerům (vedoucím projektům), kterým poskytuje zdůvodnění pro používání postupů ergonomického projektování. Píše se v ní, že „všechny pracovní systémy by měly vycházet z ergonomických principů popsaných v ISO 6385:1981“ (zmíněná norma ISO platí také v ČR pod označením ČSN EN ISO 6385 [2004]).

Zdůrazňuje zapojení uživatelů do pracovní tvorby. Tento přístup, nazývaný *návrh zaměřený na uživatele*, anglicky *user-centered design*, je dnes považován za základní v oblasti ergonomie softwaru. Znamená hlavně testování budoucího UI uživateli a získávání tak zpětné vazby, jejíž výsledky lze efektivně uplatnit ještě během vývoje.

ČSN EN ISO 9241

Ergonomické požadavky na kancelářské práce se zobrazovacími terminály

ČSN EN ISO 9241 je vlastně sada norem, které se, jak z názvu vyplývá, zabývají požadavky na zobrazovací terminály pro kancelářské práce – tzn. zpracování textů a dat. Nicméně analogicky se tyto požadavky dají použít i v mnoha jiných

oborech, kde dochází ke komunikaci mezi člověkem a počítačovými zařízeními (např. v lékařství, telekomunikacích, průmyslu atd.).

Jde o poměrně komplexní normu, která se zabývá použitelností z hlediska softwaru, hardwaru i pracovního prostředí. V tabulce 2.1 je uveden jednoduchý přehled jejích částí.

Část	Obsah
1, 2, 10, 11	obecné informace
3, 4, 7, 8, 9	požadavky na hardware
12, 13, 14, 15, 16, 17	požadavky na software
5, 6	požadavky na pracovní prostředí

Tabulka 2.1: Obsah jednotlivých částí sady norem ČSN EN ISO 9241

Některé části ještě najdeme pod označením ČSN EN 29241, což jsou původní verze těchto norem, které se později začaly nahrazovat v souladu s mezinárodními standardy převzatými normami ISO 9241.

Další standardy

Existují další normy, které se jednotlivými aspekty ergonomie produktů zabývají podrobněji. V příloze A najdeme přehled několika dalších norem (včetně již zmíněných) vztahujících se k tomuto tématu.

Kromě mezinárodních či národních norem jsou k dispozici ještě standardy soukromých společností a veřejných institucí. Většinou jde o zcela konkrétní vodítka pro návrh UI, takže se uplatní zejména ve 2. a 4. fázi jeho vývoje (viz část 2.3). Ta nejznámější uvádím zde:

- *Apple Human Interface Guidelines* [Apple Computer, 2006]: jak název napovídá, jedná se o doporučení firmy Apple Computer, Inc. ohledně tvorby uživatelského rozhraní v prostředí Mac OS X.
- *Microsoft User Interface Design and Development* [Microsoft Corporation, 2006]: doporučení firmy Microsoft Corporation.
- *GNOME Human Interface Guidelines*: doporučení způsobu používání prvků GUI „open source“ grafického uživatelského prostředí GNOME.

- *Research-Based Web Design & Usability Guidelines*: vodítka pro tvorbu použitelných a přístupných webových stránek vydaná oddělením Komunikačních technologií amerického Národního institutu pro rakovinu (National Cancer Institute) – jsou zaměřena především na vládní webová sídla.
- *Best practice – pravidla pro tvorbu přístupného webu* [Ministerstvo informatiky, 2004]: doporučení pro tvorbu webových stránek institucí veřejné správy přístupných občanům se specifickými potřebami (zrakově, sluchově, pohybově postižení apod.) – vydalo české Ministerstvo informatiky.

Podobných standardů bylo vydáno mnohem více. Některé zanikly, jiné jsou neustále aktualizovány. Jejich přehled je např. na webových stránkách firmy Experience Dynamics [Experience Dynamics, 2006].

Jak uvidíme dále, téměř všechny metodiky projektování uživatelských rozhraní doporučují použití takovýchto vodítek během návrhu vlastního vzhledu a interakčního stylu UI.

2.2 Terminologie

I když na intuitivní úrovni obsah zde používaných pojmů chápeme, je nutné je přesně definovat a vymezit. V této práci se však omezím pouze na definici pojmu *použitelnost*, neboť zaměření a rozsah práce neumožňuje podat vyčerpávající terminologické definice. Použitelnost softwarových a hardwarových produktů je však nejdůležitějším problémem, kterým se obor HCI zabývá.

Našli bychom nepřeberné množství definic termínu *použitelnost* – téměř každá významnější práce v oboru HCI vymezuje tento pojem jinak. Uvedu zde dvě definice – jednu „normovanou“ a druhou poněkud konkrétnější.

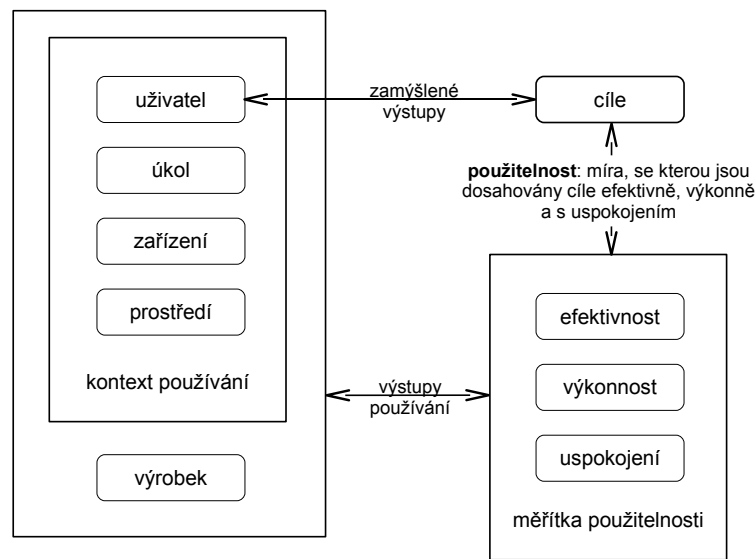
Norma ČSN EN ISO 13407 [2000] uplatňuje tuto definici použitelnosti:

Definice:

Použitelnost (usability) je rozsah v němž může určitá skupina uživatelů výrobek používat k účelnému dosažení cílů s ohledem na efektivnost, výkonnost a uspokojení ve specifickém kontextu použití [ČSN EN ISO 13407, 2000].

Pojmy efektivnost, výkonnost, uspokojení, kontext používání a uživatel definuje zmíněná norma také. Zde jejich definici neuvádím, význam je u většiny z nich

zřejmý. Význam termínu *kontext používání* bude patrný ze struktury komponent použitelnosti podle ČSN EN ISO 9241–11 [1999] znázorněné na obrázku 2.2.



Obrázek 2.2: Komponenty použitelnosti podle ČSN EN ISO 9241–11 [1999]

Význam některých pojmů z obrázku 2.2 [ČSN EN ISO 9241–11, 1999]:

cíl (goal)	očekávaný výstup
úkol (task)	činnost potřebná k dosažení cíle
výrobek (product)	část zařízení (HW, SW), pro kterou se specifikuje nebo hodnotí použitelnost

Nielsen [2003] definuje použitelnost takto (přeloženo z angličtiny):

Definice:

Použitelnost (usability) je jakostní znak, který určuje, jak snadné je používání uživatelských rozhraní. Použitelnost je určena pěti jakostními složkami:

- naučitelnost (learnability) – jak snadno se uživatelům daří splnit úkol, když se poprvé setkají s rozhraním
- výkonnost (efficiency) – jak rychle uživatelé vykonávají úkoly poté, co se seznámí s rozhraním
- zapamatovatelnost (memorability) – když se uživatelé k rozhraní vrátí po nějaké době, jak snadno opět získají zběhlost v jeho používání

- chyby (errors) – *kolik chyb uživatelé dělají, jak vážné jsou tyto chyby a jak snadno se z nich dostanou*
- spokojenost (satisfaction) – *jak je rozhraní přívětivé*

Kromě použitelnosti Nielsen zdůrazňuje *funkčnost (utility)*, což je další jakostní znak, který určuje do jaké míry dělá produkt to, co jeho uživatel potřebuje. Je stejně důležitá jako použitelnost, neboť používání nějakého softwaru může být snadné, ale je to k ničemu, pokud nedělá to, co chceme. Opačně to platí samozřejmě také – veškeré funkce software jsou na nic, pokud je neumíme použít. Pro studium funkčnosti software můžeme používat stejné metody, jako používáme pro zkoumání použitelnosti.

Další pojmy budou objasněny a upřesněny v dalších částech, kde jsou uvedeny požadavky na použitelnost.

2.3 Fáze vývoje

Obecně by měl vývoj uživatelského rozhraní probíhat ve stejných krocích jako vývoj jiných produktů a softwaru zvláště:

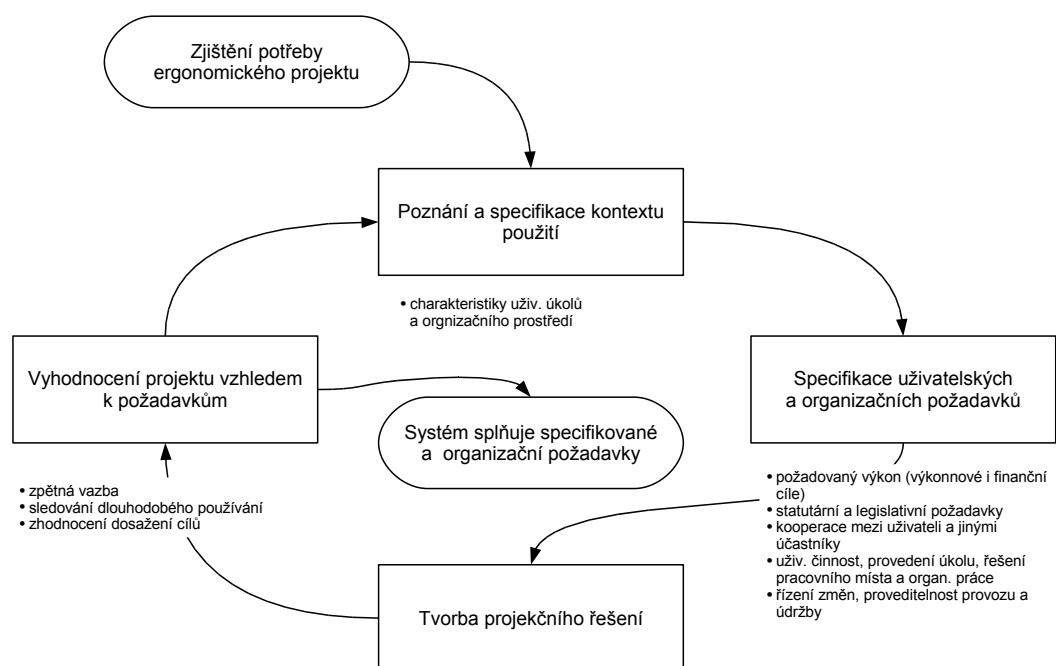
1. *analýza* - zjištění potřeb uživatele, tj. analýza úloh, které bude provádět, určení požadavků na produkt (použitelnost, standardizace, přenositelnost), plánování časových a finančních možností
2. *návrh* - popis řešení, může být vytvořeno několik alternativ
3. *testování* - zkoušení navrhovaných řešení (uživatelské testování, přezkoumání odborníky apod.)
4. *implementace* - realizace zvoleného návrhu
5. *ověření* - ověřování funkčnosti a splnění požadavků hotového produktu (příp. jeho části)

Jedna specifická vlastnost vývoje UI oproti vývoji jiných částí softwarového produktu je bod 3. Testování funkčnosti software obvykle probíhá až po jeho implementaci. UI má tu výhodu, že z velké části ho lze testovat ještě před jeho konečnou realizací. K tomu slouží např. papírové čtvrtky s nakreslenými ovládacími prvky, jednoduché prototypy tvořené v HTML, v nástrojích pro Macromedia Flash technologie

apod., anebo v nástrojích přímo specializovaných na vývoj UI. Samozřejmě, pro funkční kód software lze také vytvářet prototypy, ale ty se dělají již v cílovém programovacím jazyce a postupnou implementací se rozvíjejí do konečné podoby, takže náklady na vývoj testovacího prototypu budou vyšší než např. vytvoření sady na papíře znázorněných dialogů uživatelského rozhraní.

Celý proces vývoje UI je iterativní. Tzn. že v každé fázi lze analýzu, návrh nebo implementaci přehodnotit a vrátit se k některému z předcházejících kroků. To se bude zřejmě dít zejména mezi kroky 1–3, nejčastěji se pak budeme vracet od bodu 3 do bodu 2. V bodě 3 např. zjistíme, že uživatelé nesdílejí naši koncepci ovládání funkcionality. Pak budeme muset tuto koncepci přehodnotit a navrhnout jiný způsob. Návrhů může být několik, během testování pak zjišťujeme, který bude nejvhodnější.

Na obrázku 2.3 jsou znázorněny činnosti ergonomického projektování a jejich souvislosti podle ČSN EN ISO 13407 [2000]. Kruhová posloupnost zdůrazňuje již zmíněnou iterativnost procesu.



Obrázek 2.3: Vzájemné souvislosti činností ergonomického projektování [ČSN EN ISO 13407, 2000]

Fáze návrhu je zde zřejmě součástí „tvorby projekčního řešení“. Fáze testování může být také součástí tohoto bodu, anebo ho lze vělenit jako vyhodnocení neúplného projekčního řešení.

Také během implementace lze ověřovat a testovat splnění požadavků na UI – v této fázi máme již reálně fungující produkt, na kterém můžeme lépe odhalit případné nedostatky návrhu.

Ověřování funkčnosti a splnění požadavků hotového produktu provádí tvůrce a hlavně zákazník, který v konečné fázi rozhoduje o úspěšnosti vývoje.

Jak již bylo zmíněno v úvodu, projektování použitelných uživatelských rozhraní (ergonomické projektování) je multidisciplinární. V řešitelském týmu by měli být [ČSN EN ISO 13407, 2000]:

- konečný uživatel (jeho aktivní zapojení do tvorby je neustále zdůrazňováno a mělo by být samozřejmostí)
- kupující (manažer uživatele)
- oborový specialista, obchodní analytik
- systémový analytik, systémový inženýr, programátor
- marketingový odborník, prodejce
- projektant uživatelských rozhraní, návrhář vzhledu
- odborník na lidské faktory a ergonomii
- technický autor, školitel a pomocný personál

V části 2.4 je přehled specifických metodik určených pro tvorbu uživatelských rozhraní (GUIDE, LUCID, OVID, IDIOM, Wisdom atd.), které jednotlivé fáze vývoje rozpracovávají s ohledem na tuto oblast vývoje.

2.4 Metodické postupy

Pro proces tvorby uživatelských rozhraní existuje mnoho metodických vodítek, která vznikla na akademické půdě, anebo je vytvořily firmy zabývající se uživatelským rozhraním.

Všechny moderní metody projektování UI používají *objektově orientovaný přístup*. Tzn. že prvky reálného světa i abstraktní pojmy, se kterými uživatel pracuje, jsou chápány jako objekty mající své vlastnosti a sady operací, které je s nimi možné provádět. Klíčovým prvkem tohoto přístupu je určení takových objektů,

jejich vlastností, operací a vzájemných vztahů mezi nimi a poté způsob jejich reprezentace pomocí prostředků uživatelského rozhraní. Popis těchto objektů pak vytváří určitý *model* zkoumaného systému. Model nám pomáhá systém pochopit, analyzovat jej a rozhodovat o problémech souvisejících s tvorbou aplikace.

2.4.1 Strukturovaný návrh uživatelského rozhraní s optimalizací interakce, STUDIO

Structured User-Interface Design for Interaction Optimization

Tuto metodiku vyvinul Dermot Browne na začátku 90. let. Je určena zejména pro vývoj klientských aplikací bohatých na uživatelské rozhraní vyvíjených v prostředí klient – server. Metodika není založena na modelování systému, ale používá mnoho prostředků na uživatele orientovaného návrhu [Nunes, 2001].

2.4.2 Modelování uživatelského rozhraní, UIM

User Interface Modeling

Je metoda vytvořená Magnusem Lifem koncem 90. let pro zjišťování uživatelských požadavků, které jsou přímo použitelné pro návrh uživatelského rozhraní informačního systému. Nestanovuje přesný postup procesu tvorby UI, neboť jej považuje za kreativní činnost, kterou nelze metodicky popsat. Místo toho ulehčuje rozhodování během návrhu poskytnutím modelů, které definují uživatelské požadavky týkající se uživatelského rozhraní [Nunes, 2001].

2.4.3 Architektonický rámec entita, úloha, reprezentační prvek, ETP

Entity, Task, Presenter Architectural Framework

Metodika ETP, publikovaná Johnem Artimem v roce 2001, definuje architekturu uživatelského rozhraní tak, že rozděluje prvky HCI, které vytvářejí popis uživatelského rozhraní do 3 hlavních skupin: entit (entities), úloh (tasks) a reprezentačních prvků (presenters) [Nunes, 2001].

2.4.4 Bridge

The Bridge

Bridge je komplexní metodika návrhu UI s účastí uživatele [Nunes, 2001].

Skládá se ze 3 kroků:

1. vyjádření uživatelských požadavků pomocí průběhu úloh – cílem je převést potřeby uživatelů týkající se úlohy na požadavky, které odrážejí průběh jejího provádění
2. přiřazení průběhu úloh objektům – cílem je přiřadit požadavky na UI samostatným jednotkám informací, kterými uživatelé manipulují během provádění úloh
3. přiřazení objektů úloh prvkům GUI – tento krok zajišťuje, že bude GUI použitelné pro provádění úloh

2.4.5 Návrh a vyhodnocení grafického uživatelského rozhraní, GUIDE

Graphical User Interface Design & Evaluation

Metodika, vytvořená Davidem Redmond–Pylem a Alanem Moorem, byla publikována v roce 1995. Jedná se o proces několika kroků, které mají své specifické výstupy. Pokud nejsou výsledky uspokojivé, kroky jsou opakovány [Griffith, 1999].

Tabulka 2.2 podává přehled jednotlivých fází projektování a jejich výstupů.

Nejprve musíme určit charakteristiky budoucích uživatelů a podle toho je *rozdělit do tříd*. Tzn. co kteří uživatelé budou provádět a jaké prostředky k tomu potřebují. Na základě těchto tříd uživatelů můžeme vytvořit *modely* jejich *úloh*. Každá úloha, kterou má uživatel provádět, musí být identifikována a popsána – zjišťujeme kdo ji bude provádět, její časovou náročnost, četnost atd. Složitější úlohy se *rozdělují na menší části*. Současně vytvoříme *scénáře* těchto úloh – tj. popis činnosti uživatele i jiných zúčastněných osob krok za krokem. Scénáře později využijeme také k vyhodnocení UI.

Když máme popsané úlohy, které bude uživatel v systému řešit, určíme *objekty*, s nimiž potřebuje uživatel během práce manipulovat (např. pro účetní to může být faktura, skladová položka, peněžní účet apod). Objekty odpovídají prvkům v uživatelově *mentálním modelu* systému. Je nutné určit typy objektů, jejich vlastnosti, které jim uživatelé mohou přisuzovat, vztahy mezi objekty a činnosti s nimi prováděné. Různí uživatelé (typicky třídy uživatelů, ale i jednotlivci) mohou mít různé mentální modely (např. správce bankovního systému se na databázi účtů dívá jinak, než klientský pracovník za přepážkou). K tomu je třeba přihlížet.

Můžeme vytvořit také *dynamický model* objektů – ty mívají různé stavy, následkem akcí přecházejí z jednoho stavu do druhého. Například peněžní banko-

Krok	Výstup
1. stanovit <i>požadavky uživatelů</i> a <i>požadavky na použitelnost</i>	popis <i>tříd (druhů) uživatelů</i> , požadavky na použitelnost
2. vytvořit <i>modely úloh</i> prováděných uživateli	modely úloh, scénáře úloh
3. vytvořit <i>modely objektů</i> používaných uživateli	modely objektů, <i>slovník pojmů</i> používaných uživateli
4. stanovit <i>vodítka návrhu</i>	vodítka návrhu aplikace
5. <i>návrh GUI</i>	návrh oken (včetně interakčního chování), návrh navigace mezi okny
6. vytvořit <i>prototyp GUI</i>	funkční prototyp GUI
7. <i>vyhodnotit GUI</i>	vyhodnocení použitelnosti GUI, problémy týkající se použitelnosti, doporučené změny GUI

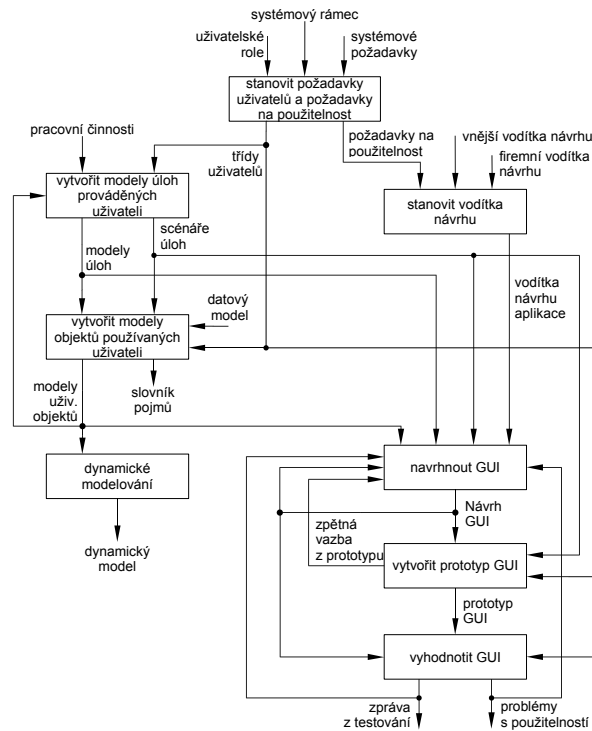
Tabulka 2.2: Kroky projektování UI metodikou GUIDE a jejich výstupy [Griffith, 1999]

mat má tyto stavy: čekání na vložení kreditní karty (stav), vložení karty (akce) přejde do stavu čekání na vložení přístupového kódu (přechod) atd. Dynamický model popisuje, jak se objekty v jistých podmínkách chovají.

Slovník pojmů určí, jak uživatelé objekty v systému a chování těchto objektů nazývají. Od toho se pak odvíjí pojmosloví uplatněné v aplikaci.

Během vývoje také vytvoříme *vodítka* pro návrh uživatelského rozhraní. Ta vymezují pravidla pro splnění stanovených požadavků na použitelnost UI a pomáhají udržet jednotný vzhled a chování prvků UI napříč celou aplikací. Jako podklad mohou posloužit vodítka firemní, externí (např. v části 2.1 zmíněná vodítka různých institucí) a normy upravující tuto problematiku. Vzniklá vodítka můžeme už v této fázi podrobit ověřování použitelnosti.

Nyní přistoupíme k návrhu UI. Určíme *pohledy* na uživatelské objekty, tj. podobu reprezentace objektů – hodnot jejich vlastností a způsobů modifikace těchto hodnot. Definujeme potřebná *okna* GUI, možné akce s těmito okny a pohledy na objekty vložíme do oken. Pokusíme se provést *scénáře úloh* s pomocí navržených



Obrázek 2.4: Schéma postupu prací během návrhu metodikou GUIDE
 [Griffith, 1999, přeloženo z angličtiny]

oken GUI. Toto testování provádíme společně s uživateli (viz část 2.8), protože ti znají své úlohy lépe a dívají se na ně jinak než projektant uživatelského rozhraní aplikace. Pokud se objeví problémy v návrhu UI, můžeme ho ihned opravovat.

Vytvoření *prototypu* nám slouží k dalšímu ověření použitelnosti rozhraní. Prototypy mohou zabírat UI v celé jeho šíři (tzv. *horizontální*) – pak nemodelují prvky detailně, nebo se mohou zaměřovat na vybrané části UI (*vertikální* prototypy). Mohou být určeny k dalšímu rozvoji nebo jen na jedno použití. Pro jejich tvorbu lze využít různých prostředků – od nejjednodušších papírových modelů až po velice sofistikované softwarové nástroje na toto specializované. Testování prototypů *šetří* nemalé *náklady* na vývoj, protože umožňuje provést změny v návrhu UI ještě předtím, než je aplikace zcela implementována (změny v hotové aplikaci jsou značně finančně a časově náročné). Stejně jako během testování návrhu, GUIDE opět doporučuje *uživatelské testování* prototypů s využitím *scénářů úloh*.

Výsledkem testování prototypu je zpráva o použitelnosti navrhovaného uživatelského rozhraní obsahující identifikované problémy a případné doporučení ke změnám. Návrhový proces můžeme v různých fázích *opakovat* tak dlouho, dokud není návrh uspokojivý.

Jak už z názvu vyplývá, metodika GUIDE rozebírá především vývoj *grafického* uživatelského rozhraní. Zaměřuje se na prezentaci uživatelských objektů a operací

grafickými interakčními prvky. Obecně však můžeme uvedené postupy aplikovat i při projektování jiných typů UI.

2.4.6 IDIOM

Mark van Harmelen začal na této metodice pracovat už na začátku 90. let jako na objektově orientované metodě s důrazem na aktivity popisující grafické uživatelské rozhraní typu WIMP (okno, ikona, nabídka, ukazovací zařízení). Vychází z předpokladu, že objektové modely mohou vyjádřit významné detaily o interakčních stránkách systému založeném na GUI [Nunes, 2001].

V roce 2001 autor zveřejnil přepracovanou verzi metodiky IDIOM, ve které už používá vytváření *scénářů* a následný návrh založený na identifikovaných úlohách (*task based*).

IDIOM obsahuje následující činnosti v procesu tvorby UI [Nunes, 2001]:

1. roztřídění uživatelů, vytvoření scénářů a modelování úloh prováděných uživateli
2. vytvoření doménového modelu (domain modeling), tzn. určení používaných objektů a jejich organizace
3. vytvoření základního modelu (core modeling) – popsání obsahu a struktury systému z pohledu uživatele, určuje doménové objekty, které mají být prezentovány nebo implementovány (ale nezávisle na konkrétním uživatelském rozhraní)
4. vytvoření tzv. pohledů (views), které stanovují vizuální prostředí provádění úloh
5. navržení podrobností týkajících se interakce s uživatelem, tj. způsobů reprezentace objektů a metod manipulace s nimi v systému GUI

V tomto procesu projektování UI není výslovně uvedeno použití průzkumů nebo návrhu s účastí uživatele, nicméně van Harmelen uvádí, že jsou s metodikou IDIOM slučitelné.

2.4.7 Návrh orientovaný na použití

Usage-Centered Design

Toto pojetí návrhu uživatelského rozhraní zdůrazňuje (oproti projektování orientovaného na uživatele) zaměření na *používání* navrhovaného systému. Tedy

nezkoumá už tolik uživatele a jeho schopnosti, ale klade důraz na analýzu *případů užití* (angl. *use-cases*) – modely užití v něm řídí celý proces vývoje [Nunes, 2001].

Autoři metody (Larry Constantine a Lucy Lockwood) doporučují provádět činnosti během projektování současně, kdykoliv je to možné.

Metodika popisuje použití následujících kroků [podle Nunes, 2001]:

- *vytvoření požadavků ve spolupráci (collaborative requirements dialogue)* mezi projektanty a uživateli (i zákazníky)
- *vytvoření modelů úloh (task modeling)* – stanovení jasného a úplného přehledu o tom, co bude systém umožňovat
- *vytvoření modelu domény (domain modeling)* – stanovení všech souvisejících konceptů a konstrukcí v aplikační doméně (oblasti)
- *vytvoření modelu pojetí rozhraní (interface concept model)* – abstraktní návrh všech interakčních prostorů, které umožní provádění úloh uživatele
- *popis modelu implementace (implementation modeling)* – konkretizace uživatelského rozhraní (tj. stanovení použití prvků a operací UI pro reprezentaci doménových objektů)
- *přezkoumání použitelnosti (usability inspection)* – zjištění možných problémů testováním s koncovými uživateli
- *uvedení do provozního kontextu (operational contextualization)* – přizpůsobení návrhu skutečným podmínkám a prostředí, ve kterých bude systém pracovat
- *stanovení standardů a stylu (standards and style definition)* – určení standardů a vodítek, které budou usměrňovat návrh UI
- *tvorba nápovědy a dokumentace (standards and style definition)* – provádí se souběžně s ostatními kroky
- *tvorba objektové struktury (object-structure design)* – uspořádání modelů základních (spodních) objektů
- *soustředná stavba (concentric construction)* – vytvoření fungujícího systému ve vrstvách
- *projekční opakování (architectural iteration)* – udržování správné interní softwarové architektury během přidávání dalších vrstev systému

2.4.8 Logický dialogový návrh orientovaný na uživatele, LUCID

Logical User Centered Interaction Design

Tuto metodiku vyvinul Charles Kreitzberg ze společnosti Cognetics Corporation [Cognetics, 2003] a uveřejnil v roce 1996.

Jde o systém šesti kroků určujících celý vývoj uživatelského rozhraní:

1. *představa (envision)* – uspořádat jednání všech účastníků tak, aby obchodní zájmy, technická omezení a vysoká uživatelská použitelnost produktu byly ve skupině vyrovnané; navrhnout jasnou sdílenou představu o produktu
2. *objevování (discovery)* – zkoumání uživatelů a určení jejich nejdůležitějších požadavků; pochopení úloh prováděných uživateli, jejich terminologie, potřebných informací a mentálních modelů
3. *základní návrh (design foundation)* – navrhnout základní vizuální vzhled, vytvořit prototyp rozhraní základní obrazovky, testovat ho a podle výsledků případně upravit
4. *podrobný návrh (design detail)* – vytvořit přesné specifikace pro všechny vizuální prvky aplikace, jejich chování a styl; provést testování specifických obrazovek
5. *sestavení (build)* – vytvořit konečný produkt; provést testování použitelnosti kritických obrazovek a provést případné řízení změn v pozdní fázi vývoje
6. *uvolnění (release)* – vytvořit plán představení produktu uživatelům, vyhodnotit jejich zkušenosti s hotovým výrobkem a uspokojení z jeho používání

Jak vidíme, metodika zřetelně kopíruje základní model vývoje zmíněný v části 2.3.

Jejím výrazným prvkem [Shneiderman, 2004] je důraz na vytvoření prototypu klíčové obrazovky, která obsahuje nejdůležitější navigační cesty v aplikaci. Tento prototyp je určen pro uživatelské testování a přezkoumání odborníky a umožňuje již v rané fázi vývoje návrh jednoduše celkově měnit a upravovat podle požadavků.

2.4.9 Návrh objektů, pohledů a interakce, OVID

Object, View and Interaction Design

Jedná se o metodiku strukturovaného návrhu objektově orientovaného uživatelského rozhraní. Strukturovaný znamená, že fáze projektování jsou rozděleny na jasně definované části. Nepokrývá analýzu úloh a určení požadavků – ty jsou jejím vstupem a je třeba je provést jinými prostředky. Výstupem je pak specifikace návrhu UI obvykle v notaci UML (Unified Modelling Language) vhodné pro následnou implementaci [IBM, 2006].

Metodika se zaměřuje na tři základní prvky návrhu UI:

- *objekty (objects)* – fyzické nebo pojmové, které uživatel vnímá jako důležité nebo užitečné pro své úkoly; snahou je, aby objekty co nejvěrněji odrážely uživatelův mentální model
- *pohledy (views)* na objekty, které jsou nabízeny – reprezentují objekty a poskytují rozhraní pro vykonávání akcí s objekty
- *interakce (interactions)*, které se vytvářejí mezi uživatelem a objekty

Důraz na objektový přístup vyplývá z moderních trendů v projektování počítačových systémů. Umožňuje návrhářům UI přímočařejší propojení mezi jejich modelem a systémem, programátorům pak zjednodušuje práci, protože dnešní nástroje pro modelování v jazyce UML, který je na objektech založen, umí automaticky generovat části zdrojového kódu programu.

OVID pracuje se třemi systémovými modely: mentálním modelem uživatele, modelem projektanta a modelem programátora [IBM, 2006].

- *uživatelův mentální model (user's conceptual model)* zastupuje to, jak uživatel chápe věci, které se dějí a proč se dějí – jaké objekty a funkce systém nabízí, jak systém reaguje na akce prováděné uživatelem, jakých cílů chce uživatel s pomocí systému dosáhnout
- *projektantův model (designer's model)* popisuje to, s čím bude uživatel pracovat, tj. vlastní interakci s počítačem – prvky uživatelského rozhraní (reprezentující objekty mentálního modelu uživatele) a jejich chování, které uživateli umožní splnit jeho cíle
- *programátorův model (programmer's model)* popisuje implementační detaily

Klíčovým prvkem metodiky je vytvoření modelu projektanta, který určuje vazbu mezi objekty v mentálním modelu uživatele a jejich implementací v modelu programátorově. Projektant musí znát uživatelův mentální model, principy návrhu použitelného UI a také možnosti a omezení prostředí, ve kterém bude programátor aplikaci implementovat. Projektantův model by měl co nejvěrněji odrážet model uživatele, protože pak je vytvářeno rozhraní, které bude pro uživatele intuitivní, tzn. že se uživatel snadno naučí aplikaci ovládat a dokáže nabyté vědomosti aplikovat i v nových situacích.

Postup při vývoji je následující:

1. *analýza úloh (task analysis)* určí počáteční sadu *objektů*; každý objekt má své vlastnosti, chování a vztahy s jinými objekty
2. definují se *pohledy*, které uživatelům umožní vidět odpovídající aspekty každého objektu
3. *úlohy se popíší* v pojmech nových objektů a pohledů
4. podrobně se popíší *interakce* mezi uživatelem a objekty

Jednotlivé kroky lze opakovat. Testováním použitelnosti prototypů vytvořených na základě vzniklého návrhu, lze zjistit, jestli je třeba další zlepšení a v čem by mělo zlepšení spočívat.

2.4.10 Wisdom

Tuto metodiku publikoval v roce 2001 Nuno Nunes [Nunes, 2001]. Měla být reakcí na některé nedostatky a omezení do té doby vytvořených metod návrhu uživatelského rozhraní. Cílem autora bylo vytvořit efektivní metodiku použitelnou v menších vývojových týmech, kterých je v oblasti tvorby softwaru naprostá většina.

Wisdom obsahuje 3 základní součásti [Nunes, 2001]:

- *vývoj dle Wisdom* – softwarový vývoj založený na evolučním, na uživateli zaměřeném modelu rychlého vytváření prototypů, který je přizpůsoben malým týmům a kde lze s výhodou uplatnit a využít možnosti zvýšené komunikace mezi členy týmu i s externími partnery, pružnosti (tzn. že vývojáři nejsou natolik svázáni tuhými administrativními postupy), jednoduché říditelnosti týmu, rychlé reakce na nové podmínky a nečekané situace, improvizace a tvořivosti zúčastněných

- *modelová architektura Wisdom* – používá sadu UML modelů, které podporují jednotlivé fáze návrhového procesu Wisdom
- *notace Wisdom* – sada notací pro modelování založená na podskupině notací UML; je vylepšena o rozšíření (kompatibilní s UML) přizpůsobená projektování interakčních systémů

Princip *rychlého evolučního vytváření prototypů* autor zavedl, aby bylo dosaženo odpovídající pružnosti a efektivity při vývoji softwaru v dnešním bouřlivém a vysoce konkurenčním prostředí. Evolučním je zde míněna snaha o postupné zdokonalování jednotlivých prototypů vedoucí až ke konečnému produktu, čímž je maximálně využita práce vývojářů (prototyp není po uživatelském otestování opuštěn, ale je využit jako základ pro další evoluční krok).

Wisdom definuje následující aktivity v procesu projektování UI, které jsou rozděleny do tří hlavních oblastí:

1. *průběh práce na požadavcích (requirements workflow)* – cílem těchto kroků je namířit vývoj směrem k produktu, který zákazníka, včetně koncového uživatele, uspokojí:
 - *zvnitřnění projektu (interiorize project)* – úvodní diskuze o představách projektu; vývojáři a koncoví uživatelé navrhnou základní pojetí systému, tj. to, co by systém měl a neměl dělat, výhody systému a případná rizika
 - *poznání souvislostí systému (understand system context)* – pochopení oblasti (domény) aplikace, a to (podle její složitosti) buď jen záležitostí zahrnutých v problému nebo i věcí týkajících se uživatelů a jejich pracovních činností
 - *určení profilu uživatelů (user profiling)* – popsání uživatelů, kterým bude systém pomáhat k provádění jejich práce; zahrnuje určení skupin uživatelů rozdělených podle významných charakteristik jako jsou dovednosti, pracovní zařazení, způsoby interakce, informační kritéria apod.
 - *zjištění požadavků (requirements discovery)* – určení funkčních i nefunkčních požadavků, tj. popsání případů užití (use-cases), ale i požadavků na interakci a vzhled (stanovení nefunkčních požadavků je již v této fázi nutné z důvodu přímého ovlivnění konečného produktu, který je evolucí odvozen z prototypu vzniklého na základě této analýzy)

2. *průběh práce na analýze (analysis workflow)* – analýza zpřesňuje a strukturuje požadavky stanovené v předchozích krocích; cílem je popis požadavků, který formuje strukturu interaktivního systému (model případů užití popisuje systém z pohledu uživatele, kdežto analytický model je interní pohled na systém z hlediska vývojářů)
 - *analýza vnitřního systému (internal system analysis)* – uspořádání požadavků, které usnadňuje jejich pochopení a řízení, tj. určení objektů, jejich vlastností, operací a vztahů mezi nimi (ale jen na vysoké úrovni abstrakce)
 - *návrh architektury rozhraní (interface architecture design)* – tzn. určení prvků reprezentujících objekty identifikované v popsáných případech užití, stanovení jejich ovládání a přechodů mezi nimi tak, aby mohly být úlohy vykonávány (obvykle se navrhne několik možností, u kterých se poté zjišťuje použitelnost uživatelským testováním)
 - *spojení interní architektury a architektury uživatelského rozhraní (relate the internal and user-interface architectures)* – tím se zaručí, že prezentační vrstva společně s implementační (interní) jsou schopny umožnit provádění úloh popsáných v případech užití
3. *průběh práce na návrhu (design workflow)* – vede systém k implementaci; zatímco analýza se soustřeďuje na funkční požadavky, návrh se zaměřuje také na nefunkční požadavky; v této fázi je bráno v úvahu vývojové prostředí (programovací jazyky, databáze, GUI, operační systémy atd.)
 - *návrh vnitřního systému (internal system design)* – návrh vnitřní implementace systému tak, aby odpovídal případům užití a průběhu úloh
 - *návrh uživatelského rozhraní (user-interface design)* – převod interakčního modelu na konkrétní uživatelské rozhraní; k tomu lze využít prostředků metodiky *Bridge* (část 2.4.4).

Prototypy tvořené v průběhu návrhu systému jsou již obvykle plně funkčními evolučními předky konečného produktu. Jejich funkčnost odráží prioritami stanovenou důležitost jednotlivých implementovaných případů použití. Opakováním návrhového cyklu se funkce prototypů postupně rozšiřuje až k výslednému produktu.

2.5 Analýza požadavků na použitelnost UI

Prvním krokem při návrhu uživatelského rozhraní je analýza. Během analýzy zjišťujeme a jejím výstupem by měly být *požadavky na použitelnost uživatelského rozhraní*.

Přitom bychom měli brát v úvahu následující uživatelské parametry: trvání pozornosti, meze krátkodobé paměti, učební zvyklosti, stupeň zkušeností s prací a manipulací s dialogovým systémem a mentální model uživatele [ČSN EN ISO 9241–10, 1997].

Zjednodušeně řečeno, vytvoření „použitelného“ systému znamená dosáhnout shody mezi požadavky uživatele a možnostmi počítačového systému [Součková, 2003].

Americký vojenský standard pro kritéria plánování návrhu s ohledem na člověka [U.S. Military Standard for Human Engineering Design Criteria, cit. dle Shneiderman, 2004] uvádí následující cíle při návrhu použitelných produktů:

- dosáhnout požadované výkonnosti u osob obsluhujících, řídicích a provádějících údržbu
- minimalizovat dovednostní a personální požadavky a čas potřebný pro školení
- dosáhnout požadované spolehlivosti pro dané kombinace personál – vybavení/software
- rozvíjet standardizaci návrhu uvnitř i mezi systémy

Nicméně Shneiderman [2004] říká, že tyto cíle mohou být dobrým začátkem, ale že efektivní rozhraní také zvyšuje kvalitu života uživatelů a zlepšuje jejich společenství. Jako cíle analýzy požadavků na použitelnost pak uvádí:

1. *zjistit potřeby uživatelů*, tzn. jejich schopnosti, jaké úlohy a podúlohy budou provádět atd.; je nutné tedy provést *analýzu potřeb uživatelů* a *analýzu úloh*
2. *zajistit náležitou spolehlivost*, tj. funkce musí pracovat tak, jak bylo popsáno, zobrazovaná data musí být správná atd., jinak uživatel ztrácí ve výrobek důvěru
3. *prosazovat vhodnou standardizaci, integraci, konzistenci a přenositelnost* – uživatelé, kteří jsou zvyklí pracovat s nějakým rozhraním, by jen těžko přecházeli na koncepčně odlišné: vedlo by to nejen k dlouhým dobám nutným pro zaškolení, ale také k možným nebezpečným chybám; nekompatibilní formáty ukládaných dat jsou zdroji neefektivnosti, zpoždění a pocitů marnosti

4. *dokončit projekty podle plánu a za daný finanční rozpočet* – v dnešním vysoce konkurenčním prostředí musí být jasné, v jakém stavu, kdy a za jakou cenu lze použitelné rozhraní vytvořit a tyto parametry je pak třeba při implementaci dodržet, jinak si uživatel vybere produkt jiný

Nezbytnou součástí všech metodik návrhu uživatelského rozhraní jsou na prvních místech *analýza potřeb uživatelů* a *analýza úloh*.

2.5.1 Analýza potřeb uživatelů

V *analýze potřeb uživatelů* je zejména nutné určit dovednosti uživatelů a jejich zkušenosti s prací s uživatelskými rozhraními, zjistit jejich věkové rozvrstvení, fyzické a kognitivní schopnosti, vzdělání, kulturní a etnické zázemí, motivace, cíle a osobnost [Shneiderman, 2004].

Všechny tyto aspekty ovlivňují budoucí vzhled a funkční možnosti UI. Často je třeba vytvořit produkt, který budou využívat rozličné skupiny uživatelů. Zvláště na Internetu budou k veřejným službám přistupovat lidé s velmi rozmanitými schopnostmi, zkušenostmi, ale také různým technickým vybavením. To vše je pak nutné v návrhu zohlednit, a buďto vytvořit co nejuniverzálnější systém (což většinou omezuje zkušenější uživatele), nebo nabídnout několik možností ovládnutí systému přizpůsobených vybraným skupinám (to zase prodražuje řešení).

2.5.2 Analýza úloh

Analýza úloh znamená identifikovat úlohy a kroky, které musí uživatel provést pro splnění svých cílů. To umožní v návrhu UI odrazit četnost úkonů, jejich návaznosti, rozvrstvení podle uživatelských rolí atd.

Poté, co je analýza hotova, je možné přistoupit k návrhu rozhraní.

2.6 Návrh způsobu realizace UI

V této fázi vývojáři na základě analýzy navrhnou možná řešení UI. Vyberou se způsoby dialogu, interakční prvky a jejich funkce a vzájemná součinnost.

Můžeme zvolit následující způsoby dialogu [Shneiderman, 2004]:

přímá manipulace, výběr z nabídky, vyplňování formulářů, příkazový jazyk, přirozený jazyk

Většinou bude UI tvořeno kombinací těchto prvků.

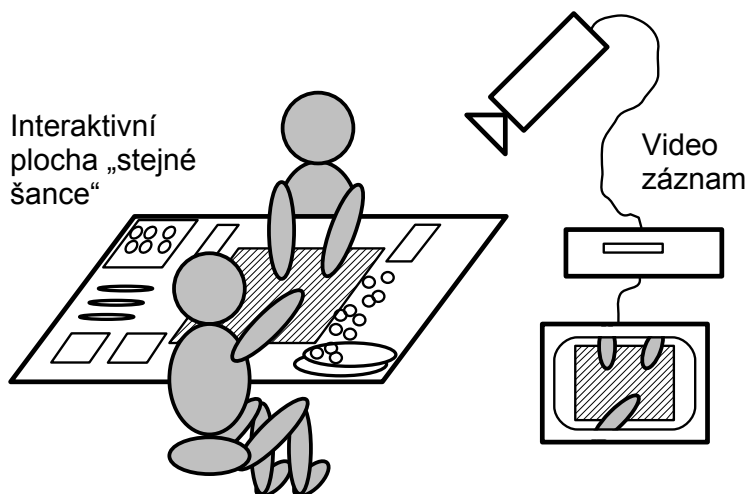
Vhodné je držet se, či alespoň se inspirovat u mnoha standardizovaných vodítek pro tvorbu UI připravených pro určité platformy, technologie nebo aplikační nasazení. Taková vodítka existují např. pro programy zpřístupňované přes WWW nebo fungující na konkrétních operačních systémech, mobilních zařízeních atd. Některé z nich jsem zmínil v kapitole 2.1.

Shneiderman dále doporučuje dodržovat osm zásad nazývaných *zlatá pravidla*. Jsou to:

1. *udržovat konzistentnost* – používat podobné sledy operací v podobných situacích, stejné termíny a grafickou podobu v celé aplikaci apod. a samozřejmě dodržovat zavedené zvyklosti a standardy
2. *vyhovět všeobecné použitelnosti* – tj. nabídnout vhodné ovládací prvky jak začátečnickům, tak pokročilým uživatelům, zohlednit uživatele se zvláštními potřebami
3. *poskytovat informačně hodnotnou zpětnou vazbu* – každá operace provedená uživatelem, by se měla navenek nějak projevit; zvláště u zřídka používaných je zpětná vazba důležitá
4. *navrhovat dialogy umožňující uzavření souvisejících operací* – sledy akcí uspořádat do skupin
5. *předcházet chybám* – navrhovat systémy, které budou uživatelům co nejvíce bránit děláním chyb
6. *umožnit jednoduché vrácení operací* – pokud uživatel ví, že může chybu napravit (vzít zpět), bude s menší úzkostí a mnohem raději prozkoumávat neznámé možnosti
7. *podporovat vnitřní pocit vlády nad aplikací* – neprovádět nečekané operace, nezobrazovat nezajímavé informace, uživatelé by měli být sami iniciátoři akcí, ne jen na akce reagovat [B. Gaines, cit. dle Shneiderman, 2004]
8. *omezit zatížení krátkodobé paměti* – zásada říká, že člověk si může najednou zapamatovat „sedm plus minus dvě porce informací“; obrazovky by proto měly být jednoduché a vždy by měla být k dispozici programová nápověda

2.6.1 Návrh s účastí uživatele (participatory design)

Do rozhodování ve fázi návrhu můžeme aktivně zapojit uživatele. Ačkoliv to může přinést nepříjemné důsledky ve formě odporu u uživatelů, jejichž návrhy nebyly přijaty, přijetí nevhodných doporučení od nekvalifikovaných uživatelů a zvýšených nákladů na vývoj, mohou uživatelé vznést mnoho inspirativních a důležitých připomínek. A – což je taky důležité – u budoucích uživatelů tento přístup zvýší pravděpodobnost kladných reakcí při převzetí hotové aplikace.



Obrázek 2.5: Typické pracovní prostředí pro PICTIVE [Muller, 1991]

Jedním z formalizovaných přístupů tohoto typu je *tvárné rozhraní pro techniky tvoření ve spolupráci pomocí zkoumání použitím obrazového záznamu* (*plastic interface for collaborative technology initiatives through video exploration, PICTIVE*) [Muller, 1991]. Jedná se o techniku využití papírových modelů – maket prvků uživatelského rozhraní (angl. mock-ups), které uživatelé a návrháři vytvářejí z obyčejného kancelářského papíru nůžkami, tužkou, příp. lepidlem. To umožňuje poměrně hravou, ale zároveň značně přizpůsobivou a nenákladnou formou testovat nápady a návrhy účastníků týkající se vzhledu a funkčnosti uživatelského rozhraní.

Návrhů uživatelského rozhraní může být několik. Během testování se pak zjistí, které řešení bude zvoleno za konečné, určené k realizaci. K rozhodnutí nám pomohou měřítka použitelnosti UI.

2.7 Měřítko použitelnosti UI

Pokud máme zjistit míru použitelnosti navrhovaného (ale také třeba již existujícího) uživatelského rozhraní co nejobektivněji, je nutné zvolit její měřítko. Možnost přesného vyhodnocení použitelnosti UI pomáhá vývojářům, hodnotitelům i zákazníkům. Normy definují měřítko jako efektivnost, výkonnost a uspokojení, což jsou dosti obecné pojmy a pro praktické využití je nutné je upřesnit.

Shneiderman [2004] uvádí tyto měřitelné faktory použitelnosti:

- *čas pro naučení* – kolik času zabere typickému zástupci skupiny uživatelů naučit se, jak udělat operace příslušné množině úloh
- *rychlost provedení* – kolik času zabere uživateli vykonat měřené úlohy
- *výskyt uživatelských chyb* – kolik a jaké druhy chyb uživatelé dělají
- *schopnost udržet znalosti* – jak dobře uživatelé udržují získané znalosti po hodině, dni nebo týdnu
- *subjektivní spokojenost* – jak moc se uživatelům líbí používat různé vlastnosti rozhraní

Je jasné, že nejlépe by bylo navrhovat systémy vyhovující všem těmto faktorům, ale v praxi jdou často některé faktory proti sobě.

2.8 Způsoby zjišťování použitelnosti UI

Použitelnost uživatelského rozhraní můžeme zjišťovat a ověřovat v různých stádiích vývoje produktu. Od toho se odvíjí plánování testů, jejich složitost a časová náročnost. Dalšími určujícími faktory při plánování testů jsou originalita produktu, počet budoucích uživatelů, cena produktu a finance určené pro testování, zkušenosti vývojového a testovacího týmu atd.

Existují různé způsoby zjišťování použitelnosti UI – odborná revize, uživatelské testování, průzkumy, testy přijetí, vyhodnocení během aktivního používání, řízené experimenty. Jsou rozvedeny v následujících částech.

2.8.1 Odborná revize (expert review)

Odbornou revizí uživatelského rozhraní je míněno jeho zhodnocení z hlediska použitelnosti odborníky na uživatelské rozhraní.

Ačkoliv u nás zatím mnoho konzultantů a firem v tomto oboru nepůsobí (rozvíjí se zde hlavně testování přístupnosti a použitelnosti webových stránek), v zahraničí je běžné u větších firem zaměstnávat specializovaný tým tvořící uživatelská rozhraní (u nás je v tomto směru průkopníkem firma Sun Microsystems Czech, s.r.o.) a u menších firem najímat odborné konzultanty pro spolupráci na návrhu nebo posouzení UI.

Revizi UI lze provádět během návrhu nebo i během implementace a po ní. Výstupem obvykle bývá zpráva hodnotící jednotlivé aspekty použitelnosti rozhraní a především obsahující návrhy a doporučení pro jeho zlepšení. Samozřejmě lze uspořádat diskuzi mezi návrháři, manažery projektu a odborníky na použitelnost.

Tento způsob ověření vhodnosti UI je mnohem efektivnější než vyhodnocování zpětné vazby od konečných uživatelů zkoušejících demoverzi produktu [J. Nielsen a R. Mack, cit. dle Shneiderman, 2004]. Už jen proto, že můžeme nechat posoudit a poté upravovat jen návrh UI bez toho, abychom investovali do implementace a poté ji nákladně měnili.

Odborné revize mohou mít následující formy [Shneiderman, 2004]:

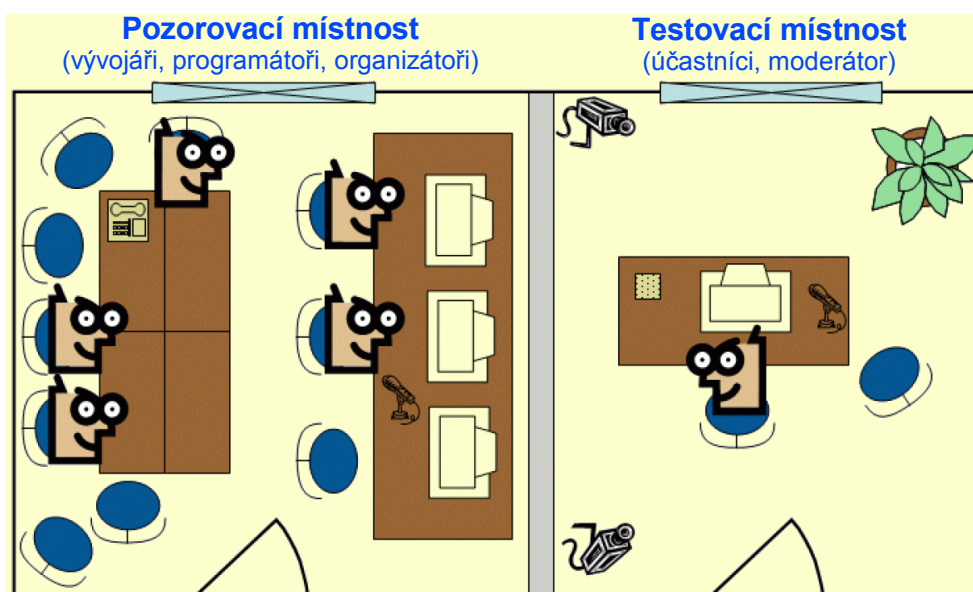
- *heuristické vyhodnocení* – odborníci hodnotí shodu prvků uživatelského rozhraní s krátkým seznamem pravidel návrhu (např. osmi zlatými pravidly); zde je důležité, aby hodnotitelé pravidla důvěrně znali a byli schopni je používat
- *revize dle vodítek* – rozhraní je hodnoceno na shodu s firemními nebo jinými vodítky pro návrh UI (některá byla zmíněna v části 2.1)
- *ověření shody* – přezkoumání konzistence mezi prvky rozhraní (terminologie, písmo, barevná schémata atd.) v sadě spolu souvisejících dialogů, nápovědě a příručkách
- *poznávací průchod rozhraním* – průchod různými částmi rozhraní tak, jak ho bude provádět konečný uživatel (měly by být zhodnoceny především často používané funkce a chování v kritických situacích)
- *formální přezkoumání použitelnosti* – společná diskuze hodnotitelů, manažerů a návrhářů řízená moderátorem nebo „soudcem“ – je užitečná zejména pro začátečníky na všech pracovních pozicích v týmu

2.8.2 Uživatelské testování použitelnosti v laboratořích (usability testing)

Velmi dobrým způsobem, jak určit problémové části uživatelského rozhraní a získat podněty pro jeho zlepšení, je testování použitelnosti UI v laboratoři skutečnými uživateli.

Typicky vypadá tak, že do testovací místnosti vybavené počítačem s instalovaným produktem nebo funkčním prototypem UI, s mikrofonom a kamerou pozveme účastníka testování (subjekt), zadáme mu několik úkolů, které má během sezení provést, požádáme ho o ústní komentování prováděných činností (tzv. *myšlení nahlas*) a jeho práci zaznamenáváme na videorekordér včetně snímání počítačové obrazovky a vstupních zařízení (klávesnice, myš).

Pro vedení uživatele a jeho psychickou pohodu je v místnosti přítomen člen testovacího týmu (moderátor), který pomůže uživateli překonat případné obtíže, ale v žádném případě mu neradí, jak úkoly provést (pokud uživatel některou situaci není schopen vyřešit, přechází se k dalšímu úkolu).



Obrázek 2.6: Typické uspořádání laboratoře pro testování použitelnosti [Franc, 2005]

Nielsen [2002] doporučuje takovou úpravu testovací místnosti, aby se v ní uživatel cítil co nejpřirozeněji a nejpohodlněji, aby jej nic nevyrušovalo a mohl se plně soustředit na plnění zadaných úkolů. Je dobré uživatele předem informovat o tom, co se bude testovat, jak bude testování probíhat, seznámit ho s prostředím a především zdůraznit, že nejde o zjišťování jeho schopností, ale pouze o ověřování použitelnosti daného uživatelského rozhraní. Samozřejmostí je zodpovězení jeho

případných dotazů.

Ve druhé místnosti – pozorovací a řídicí, obvykle oddělené od testovací zrcadlem průhledným z jedné strany, jsou další členové týmu, kteří řídí testování, pořizují jeho záznam, provádějí anotaci záznamu a kteří mohou měnit úkoly podle aktuálních potřeb. Anotace záznamu je psaný protokol, kde jsou uvedeny poznámky o průběhu testování a hlavně o vzniklých problémech a u každé takové poznámky je uveden čas jejího zápisu. Anotace dlouhých (často několikahodinových) pořizovaných záznamů je výhodná pro jejich pozdější rozbor, protože nám pomůže rychleji se v nich orientovat.

Typické uspořádání laboratoře pro testování použitelnosti je znázorněno na obrázku 2.6.

Už během testovacího sezení nebo (častěji) po jeho skončení, jsou obrazové, zvukové a jiné záznamy analyzovány a jsou identifikovány situace, kdy se uživatel během plnění úkolů dostal do obtíží. Na základě tohoto rozboru jsou pak navrženy způsoby odstranění problémových prvků UI. Při prezentaci výsledků testování vývojovému týmu a vedoucím můžeme doplnit popis identifikovaných problémů o obrazové ukázky, které mohou být velmi názorné a jednoznačně přesvědčivé.

Během odborné revize UI hodnotí použitelnost lidé, kteří mají s uživatelskými rozhraními velké zkušenosti a mohou tak snadno přehlédnout některé pro ně banální, ale pro neerudované uživatele problémové prvky. Uživatelské testování nabízí možnost pozorovat při práci s dialogy přímo konečného uživatele (nebo vybraného zástupce), což poskytuje *velmi objektivní způsob* posouzení použitelnosti UI produktu. Na druhou stranu je organizačně náročnější. Výběru uživatelů, kteří budou testování provádět, je třeba věnovat velkou pozornost a pečlivě určit požadavky na jejich schopnosti, zkušenosti, kulturní dispozice apod. tak, aby věrně zastoupili budoucí konečné uživatele.

Tato metoda má ale ještě jednu podstatnou výhodu. Během testování UI několika uživateli, můžeme objevit jejich dosud neznámé *mentální modely*, které uživatelé pro plnění zadaných úkolů použijí. Takové mentální modely, se kterými jsme původně v návrhu UI nepočítali, bychom pak měli do nové verze zapracovat.

Mezi nevýhody patří nemožnost sledování výkonnosti uživatelů při práci s rozhraním po týdnu, měsíci atd. a také – z časových a finančních důvodů – velmi omezená množina testovaných prvků rozhraní. Proto se tento způsob testování často kombinuje s odbornou revizí.

Uživatelské testování může mít mnoho forem [Shneiderman, 2004]:

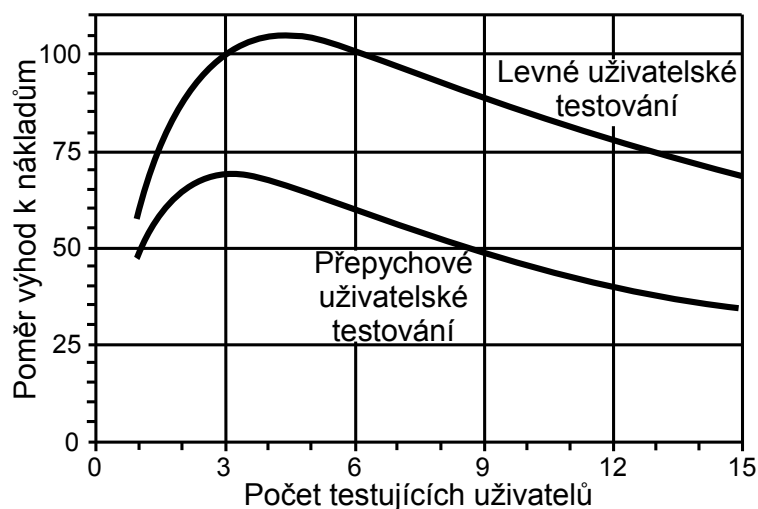
- *papírové modely (mock-ups)* – tato metoda kreslení prvků rozhraní na papír

zmíněná v části 2.6.1 je použitelná už v raných stádiích vývoje; průvodce testu sedící s uživatelem u stolu napodobuje funkci počítače a mění jednotlivé kartičky s dialogy; takové testování je poměrně levné

- *levné uživatelské testování (discount usability testing)* – využívá běžných metod, ale upravených tak, aby byly co nejméně finančně náročné; Nielsenovo levné testování [Nielsen, 1994a] je založeno na *scénářích*, zjednodušené metodě myšlení nahlas a heuristickém vyhodnocení; scénáře jsou maximálně zjednodušené prototypy rozhraní; zjednodušení metody myšlení nahlas znamená nepořizovat obrazové a zvukové záznamy, jejichž pozdější analýza je náročná, ale v průběhu testování vytvářet psané poznámky
- *srovnávací uživatelské testování (competitive usability testing)* – srovnává nové rozhraní se starší verzí nebo s jiným obdobným konkurenčním produktem
- *všestranné uživatelské testování (universal usability testing)* – testování za použití velmi odlišných softwarových a hardwarových prostředků a s množstvím různých uživatelů (např. na mezinárodní úrovni)
- *testování v terénu (field tests)* – testování nového rozhraní v reálném prostředí (např. instalace u zákazníka); je vhodné použít software k záznamu chyb, příkazů, žádostí o pomoc a měřítek produktivity
- *přenosné laboratoře (portable labs)* – umožňují přímo u zákazníka provést testování použitelnosti se zvukovým i obrazovým záznamem a často jsou využívány pro podporu testování v terénu
- *vzdálené uživatelské testování (remote usability testing)* – používané s úspěchem při ověřování webových aplikací – uživatelé nemusí cestovat do laboratoře, mohou dodávat výsledky z celého světa; nevýhodou je menší možnost kontroly prostředí a jiných podmínek
- *testy typu „umíš tohle pokazit?“ (can-you-break-this tests)* – původně používané při vývoji her, kdy jsou mladí uživatelé vyzýváni k hledání trhlin v softwaru, ale užitečné i v jiných aplikacích (např. dnes se často objevují soutěže v pronikání do zabezpečených systémů)

Nielsen [1994a], propagátor levného uživatelského testování, uvádí, že *optimální počet účastníků* (subjektů) pro uživatelské testování je 3 až 5, což při přijatelných nákladech umožňuje odhalit vysoké procento problémů v rozhraní. Na

grafu 2.7 vidíme závislost poměru výhody/cena na počtu účastníků a zvoleném způsobu testování. Přepychovým testováním je míněno využití všech prostředků používaných k uživatelskému testování.



Obrázek 2.7: Závislost poměru výhody/cena na počtu účastníků testování [Nielsen, 1994a, přeloženo z angličtiny]

Nicméně odpůrci testování s tak malým vzorkem uživatelů tvrdí, že je nedostatečné pro důkladné ověření rozsáhlejších systémů. Proto bývá testování použitelnosti rozdělováno podle počtu účastníků na [Franc, 2005]:

- *kvalitativní, tvárné (formative)* – počet účastníků bývá nejvýše 10, identifikuje kolem 90% problémů, je efektivní v průběhu vývoje, neboť umožňuje testování poměrně levně opakovat; z těchto důvodů je nejvíce využíváno komerčními firmami
- *kvantitativní, souhrnné (summative)* – počet účastníků bývá vyšší než 20, je používáno pro statistické vyhodnocení a odhad náročnosti školení, u vládních zakázek a v akademickém prostředí

I Nielsen [2000] toto dělení používá a testování s maximálně 5 uživateli doporučuje opakovaně provádět během vývoje produktu, čímž se odhalí téměř všechny problémové prvky UI. Kvantitativní testování s 20 uživateli by se pak mělo použít ke shromáždění měřítek použitelnosti UI. Kvantitativní testování a získaná měřítka jsou vhodná k porovnání dvou modelů uživatelského rozhraní.

Zkušenosti české pobočky firmy Sun Microsystems ukazují [Franc, 2005], že optimální počet účastníků pro jejich uživatelské testování je 8.

2.8.3 Průzkumy, dotazníky

Použití dotazníků umožňuje nenákladně získat zpětnou vazbu od stovek až tisíců uživatelů, která doplní informace z uživatelského testování a odborných revizí.

Cílem bývá zjistit osobní pohled uživatele na různé aspekty uživatelského rozhraní (čitelnost textů, názornost ikon, srozumitelnost systémových hlášení, účelnost nabídek a nápovědy atd.) a také osobní data uživatele (věk, pohlaví, vzdělání, zkušenosti s počítači, osobnostní vlastnosti apod.).

Příprava dotazníků je velmi náročná činnost a vyžaduje znalosti z oblasti sociologie, psychologie, statistiky atd. Je třeba je pečlivě připravit a vyzkoušet je nejprve na malém vzorku uživatelů. Naštěstí existují už vytvořené a ověřené univerzální dotazníky, které mohou vývojové týmy využít.

V příloze B je přehled několika hotových dotazníků a ukázka dvou konkrétních z nich: *dotazník na použitelnost počítačových systémů (Computer System Usability Questionnaire)* vytvořený firmou IBM [Lewis, 1995] a *karty odezvy na produkt (Product Reaction Cards)* od firmy Microsoft [Benedek aj., 2002b].

Karty odezvy na produkt (Product Reaction Cards)

Karty odezvy nebyly původně vytvořeny jako dotazník. Jedná se o systém 118 kartiček, kde na každé je jedno slovo hodnotící systém (např. „rychlý“, „přístupný“, „matoucí“ atd.). Po testovacím sezení uživatel vybere slova, která odpovídají jeho zkušenosti a pocitům z používání produktu, a z těch ještě vybere prvních pět nejvýznamnějších. Člen testovacího týmu pak s uživatelem probírá důvody zvolení těchto pěti slov a jejich význam, čímž získá konkrétnější informace.

Karty odezvy však můžeme použít i jako dotazník tak, že uživatel pouze vybere vhodná slova z daných 118 a vyplněný formulář odevzdá. Přes tento odlišný způsob dotazník podává velmi podobné výsledky jako dotazníky klasické [Tullis aj., 2004]. Výhodou je, že tato technika nezávisí na bodových stupnicích a že uživatel nemusí sám vymýšlet slova [Benedek aj., 2002a].

Obličejový dotazník (Faces Questionnaire)

Dalším, poměrně neobvyklým, způsobem získávání zpětné vazby od uživatelů je *obličejový dotazník (Faces Questionnaire)* [Benedek aj., 2002a].

Obličejový dotazník, také vytvořený firmou Microsoft, se snaží zjistit emoce, které uživatel během práce s produktem zažívá. Jde o 6 fotografií obličejů lidí vyjadřujících různé emoce s přiřazenou sedmistupňovou škálou, na které subjekty hodnotí, jak moc se jejich osobní pocity shodují s vyjádřením na fotografii.

Někteří uživatelé s menší slovní zásobou však mohou mít problémy popsat výraz na vyobrazeném obličejí a tedy vybrat odpovídající hodnocení a také hovořit o tom, jak se jejich hodnocení vztahuje k produktu.

2.8.4 Testy přijetí (acceptance tests)

Testy přijetí ověřují, zda výsledný produkt odpovídá požadavkům zadavatele. Běžně se používají pro takové vlastnosti software, jako jsou výkonnost software, čas odezvy, náročnost na hardwarové prostředky apod. Lze je ovšem zavést i pro vlastnosti uživatelského rozhraní.

Vhodné je pak v požadavcích na software dohodnutých mezi dodavatelem a zákazníkem určit *konkrétní měřitelné vlastnosti* UI. To umožní při přebírání produktu jednoznačně ověřit, jestli UI splňuje dané požadavky, nebo ne. Navíc může být posouzení ponecháno na nestranné instituci. Shneiderman [2004] zmiňuje tyto vlastnosti uživatelského rozhraní, které mohou být součástí požadavků na software: doba potřebná k naučení se určité funkce, rychlost plnění úkolů, četnost chyb uživatelů, jejich osobní spokojenost a schopnost uživatelů udržet naučené dovednosti během času.

2.8.5 Vyhodnocení během aktivního používání

Předchozí metody zjišťování použitelnosti uživatelského rozhraní byly proveditelné v různých fází vývoje. Vyhodnocení během aktivního používání lze provést až po jeho dokončení nebo vytvoření alespoň částečně funkční verze.

I když je během vývoje produktu věnována velká pečlivost návrhu UI, může se stát (a obvykle se tak stává poměrně často), že některé problémy související s použitelností nejsou odhaleny. Na ně pak narážejí koneční uživatelé při své každodenní práci. Proto je důležité získat od nich zpětnou vazbu i po nasazení software a nedostatky v dalších verzích odstraňovat. Taková zpětná vazba může mít několik podob [Shneiderman, 2004]:

- *rozhovory a skupinové diskuze* – můžeme během nich zjistit jak osobní dojmy uživatelů z práce se softwarem, tak jejich hodnotné podněty; tuto metodu jsme použili už pro výzkum požadavků na software během analýzy – toho můžeme využít a kontaktovat uživatele, kteří dobře spolupracovali

- *průběžné shromažďování hodnot výkonnostních měřítek* – sledování vlastností zmíněných v části 2.8.4 jako například četnost uživatelských chyb a žádostí o pomoc, rychlost plnění úkolů, využívání funkce „vzít zpět“ atd.; na základě takto získaných dat pak můžeme třeba změnit text systémového hlášení, upravit školící materiály apod.
- *poradci přístupní přes Internet nebo pomocí telefonu* – pro uživatele mající obtíže s produktem jsou takoví poradci velmi přínosní a tito poradci jsou pak zase důležitým zdrojem informací pro dodavatele softwaru, neboť hlásí časté nebo nepřekonatelné problémy vedoucím a vývojovému týmu
- *oznamování chyb elektronickou poštou nebo vyplněním formuláře na WWW* – tady se od uživatele očekává jistá nezbytná aktivita k tomu, aby si našel čas a chybu takto ohlásil; v případě chyb detekovatelných programově jsou efektivnější softwarové prostředky, které zcela automaticky nebo s doplněním popisu závady uživatelem odešlou hlášení dodavateli
- *elektronické diskuzní skupiny* – umožňují uživatelům vyměňovat si zkušenosti mezi sebou, žádat o pomoc dodavatele, pokud ten je také účastníkem diskuze, navrhopvat zlepšení apod.; el. diskuzní skupiny jsou oblíbené především u softwaru distribuovaného veřejně a určeného pro širokou skupinu uživatelů, u projektů tvořených na zakázku je obvyklejší telefonická a e-mailová podpora

2.8.6 Řízené psychologicky orientované experimenty

Řízené experimenty jsou finančně, časově i personálně náročné. Proto si je mohou dovolit provádět pouze největší mezinárodní firmy. Především jsou tedy součástí výzkumu ve vzdělávacích a výzkumných institucích.

Pro soukromé podniky jsou ale velmi významné. Objevují nové možnosti v tvorbě uživatelských rozhraní – nové postupy vývoje i nové prvky samotných UI, hledají nejefektivnější řešení UI atd. Takové znalosti pak samozřejmě znamenají konkurenční výhodu a tak firmy na výzkumu s vědeckými institucemi často spolupracují.

Kapitola 3

Závěr

Cílem této práce bylo podat základní přehled o požadavcích na použitelnost systémů interakce člověk–počítač a především pak uvést nejnovější metodiky určené pro vývoj softwarových produktů tyto požadavky splňující. V další práci by bylo vhodné zde uvedené metodiky navzájem srovnat a na základě toho určit vhodné oblasti jejich využití. I tento text však umožňuje stanovit některé závěry o trendech v procesu tvorby uživatelských rozhraní.

Tradiční přístupy považovaly za uživatelské rozhraní produktu jen vzhled aplikace a možné způsoby jejího ovládání (klávesové zkratky, myš, programové nabídky atd.). Dnes je prvořadý význam kladen na *objekty* potřebné k tomu, aby uživatel mohl splnit své úkoly. Popis takových objektů, jejich vlastností a vazeb mezi nimi umožňuje projektantovi snáze pochopit potřeby uživatelů a lépe navrhnout systém uživatelského rozhraní. Uživatelé se pak zpětně rychleji a s menším úsilím naučí systém ovládat, protože ten věrně odráží jejich mentální modely prováděných úkolů.

V nejnovějších metodách je kladen důraz více na používání produktu (angl. usage-centered) než na uživatele (user-centered). Ačkoliv se může zdát, že je to jistý odklon od původních záměrů oboru HCI „polidštit“ uživatelská rozhraní počítačových systémů, není tomu tak. Člověk je stále centrálním bodem návrhu, neboť je stále záměrem vytvářet systémy, jejichž používání bude pro člověka vysoce intuitivní, produktivní a příjemné. Pouze se tato snaha neodvíjí přednostně od zkušeností a subjektivního uspokojení uživatelů, ale od vytváření kvalitnějších nástrojů UI, umožňujících plnění úkolů, na základě analýzy úloh prováděných uživateli.

Snad i tento text přispěje k vytváření lepších uživatelských rozhraní počítačových systémů, která si všichni přejeme.

Seznam použité literatury

1. Apple Computer, Inc. 2006. *Apple Developer Connection* [online]. Cupertino (Kalifornie) : Apple Computer, c2006 [cit. 2006-06-06]. Apple Human Interface Guidelines. Dostupný z WWW: <<http://developer.apple.com/documentation/UserExperience/Conceptual/OSXHIGuidelines/>>.
2. BENEDEK, Joey; MINER, Trish. 2002a. Measuring Desirability: New methods for evaluating desirability in a usability lab setting. In Microsoft Corporation. *Microsoft : Usability* [online]. Redmond (Washington) : Microsoft Corporation, c2006 [cit. 2006-08-05]. Dostupný z WWW: <<http://www.microsoft.com/usability/UEPostings/DesirabilityToolkit.doc>>.
3. BENEDEK, Joey; MINER, Trish. 2002b. Product Reaction Cards. In Microsoft Corporation. *Microsoft : Usability* [online]. Redmond (Washington) : Microsoft Corporation, c2006 [cit. 2006-08-05]. Dostupný z WWW: <<http://www.microsoft.com/usability/UEPostings/ProductReactionCards.doc>>.
4. BUCHALCEVOVÁ, Alena; KÁRNÍKOVÁ, Jitka; VESELÁ, Bohumila. 1998. *K úpravě dokumentů a aplikací*. Praha : Vysoká škola ekonomická, 1998. 81 s. ISBN 80-7079-034-2.
5. Cognetics Corporation. 2003. *The LUCID Framework* [online]. Princeton Junction : Cognetics Corporation, c2003 [cit. 2006-07-27]. Dostupný z WWW: <<http://www.cognetics.com/lucid/index.html>>.
6. CONSTANTINE, Larry L; LOCKWOOD Lucy A. D. 2003. Structure and Style in Use Cases for Interface Design. In Constantine & Lockwood, Ltd. *Reports, Articles, and Papers from Constantine & Lockwood, Ltd.* [online]. Rowley (Massachusetts) : Constantine & Lockwood, c2003 [cit. 2006-08-14]. Dostupný z WWW: <<http://www.foruse.com/articles/structurestyle2.pdf>>.

7. Česko. Ministerstvo informatiky. 2004. *Best practice : pravidla pro tvorbu přístupného webu* [online]. Verze 1.0. 2004 [cit. 2006-06-06]. Dostupný z WWW: <<http://www.micr.cz/dokumenty/metodicke.htm>>.
8. ČSN EN ISO 13407. 2000. *Postupy ergonomického projektování interakčních systémů*. Praha : Český normalizační institut, srpen 2000. 28 s.
9. ČSN EN ISO 6385. 2004. *Ergonomické zásady navrhování pracovních systémů*. Praha : Český normalizační institut, září 2004. 16 s.
10. ČSN EN ISO 9241-1. 1998. *Ergonomické požadavky na kancelářské práce se zobrazovacími terminály – Část 1: Obecný úvod*. Praha : Český normalizační institut, červenec 1998. 16 s.
11. ČSN EN ISO 9241-10. 1997. *Ergonomické požadavky na kancelářské práce se zobrazovacími terminály – Část 10: Základní zásady vytváření dialogu*. Praha : Český normalizační institut, listopad 1997. 16 s.
12. ČSN EN ISO 9241-11. 1999. *Ergonomické požadavky na kancelářské práce se zobrazovacími terminály – Část 11: Údaje o možnostech využití*. Praha : Český normalizační institut, květen 1999. 28 s.
13. DAVIS, Fred D. 1989. Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *MIS Quarterly*. September 1989, vol. 13, no. 3, s. 319–340. Dostupný také z WWW: <<http://www.cba.hawaii.edu/chismar/ITM704/Davis-TAM1989.pdf>>. ISSN 0276-7783.
14. Experience Dynamics. 2006. *Experience Dynamics* [online]. Portland (Oregon) : Experience Dynamics, c2006 [cit. 2006-06-06]. Science of Usability : Proven tested methodologies. Dostupný z WWW: <http://www.experiencedynamics.com/science_of_usability/>.
15. FRANC, Jakub. 2005. Testování použitelnosti : (Usability testing). In *World Usability Day – 3. listopadu 2005* [online]. Praha : Sun Microsystems Czech, 2005 [cit. 2006-08-05]. Dostupný z WWW: <http://cz.sun.com/wud/prezentace/testovani_pouzitelnosti.pdf>.
16. GRIFFITH, Richard. 1999. Notes on GUIDE. In GRIFFITH, Richard. *Richard Griffiths – University of Brighton UK* [online]. Brighton (GB) : [b.n.], 1999 [cit. 2006-08-08]. Dostupný z WWW: <<http://www.it.bton.ac.uk/staff/rng/teaching/notes/GUIDE/>>.

17. VAN HARMELEN, Mark. 2006. Interactive System Design Using Oohci Methods. In VAN HARMELEN, Mark. *Mark van Harmelen* [online]. March 2006 [cit. 2006-08-14]. Dostupný z WWW: <http://www.cs.man.ac.uk/~mark/oohci_chapter_modified.pdf>.
18. Human Factors Research Group. 1997. *MUMMS : Measuring the Usability of Multi-Media Software* [online]. Cork (Ireland) : HFRG, [1997] [cit. 2006-08-05]. Dostupný z WWW: <<http://www.ucc.ie/hfrg/questionnaires/mumms/info.html>>.
19. CHIN John P.; DIEHL Virginia A.; NORMAN Kent L. 1988. Development of an instrument measuring user satisfaction of the human-computer interface. In *Proceedings of the SIGCHI conference on Human factors in computing systems : Washington, D.C., United States, May 15-19, 1988*. New York (USA) : ACM Press, 1988, s. 213-218. Dostupný také z WWW: <<http://portal.acm.org>>. ISBN 0-201-14237-6.
20. IBM Corporation. 2006. IBM : Easy of Use. In IBM Corporation. *IBM United States* [online]. New York : IBM Corporation, 2006 [cit. 2006-08-07]. Dostupný z WWW: <<http://www.ibm.com/easy>>.
21. KIRAKOWSKI, Jurek. 1994. The Use of Questionnaire Methods for Usability Assessment. In Human Factors Research Group. *SUMI : The de facto industry standard evaluation questionnaire for assessing quality of use of software by end users* [online]. Cork (Ireland) : HFRG, [1994] [cit. 2006-08-05]. Dostupný z WWW: <<http://sumi.ucc.ie/sumipapp.html>>.
22. KIRAKOWSKI, Jurek; CLARIDGE, Nigel. 2005. *WAMMI – Home* [online]. c2005, aktualizace 7. července 2006 [cit. 2006-08-05]. Dostupný z WWW: <<http://www.wammi.com>>.
23. LEWIS, James R. 1995. IBM computer usability satisfaction questionnaires: psychometric evaluation and instructions for use. *International Journal of Human-Computer Interaction*. Jan.-March 1995, vol. 7, is. 1, s. 57-78. Dostupný také z WWW: <<http://portal.acm.org/>>. ISSN 1044-7318.
24. LIF, Magnus. 1998. *Adding Usability : Methods for Modelling, User Interface Design and Evaluation*. Uppsala : Uppsala Universitet, 1998. 359, VII s. Acta Univ. Ups., Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology. Dostupný také z WWW:

- <<http://www.it.uu.se/research/hci/publications/ml/>>.
ISBN 91-554-4186-6.
25. LIN, Han X.; CHOONG, Yee-Yin; SALVENDY Gavriel. 1997. A Proposed Index of Usability: a Method for Comparing the Relative Usability of Different Software Systems. *Behaviour & Information Technology*. July 1997, vol. 16, no. 4/5, s. 267–277. Dostupný také z WWW: <<http://journalsonline.tandf.co.uk>>. ISSN 0144-929X.
26. Microsoft Corporation. 2006. *MSDN Library* [online]. Redmond (Washington) : Microsoft Corporation, c2006 [cit. 2006-06-06]. User Interface Design and Development. Dostupný z WWW: <<http://msdn.microsoft.com/library/>>.
27. MULLER, Michael J. 1991. PICTIVE - An Exploration in Participatory Design. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology : New Orleans, Louisiana, United States, April 27 – May 02, 1991*. New York (USA) : ACM Press, 1991, s. 225–231. Dostupný také z WWW: <<http://portal.acm.org>>. ISBN 0-89791-383-3.
28. NIELSEN, Jacob. 1994a. Guerrilla HCI: Using Discount Usability Engineering to Penetrate the Intimidation Barrier. In NIELSEN, Jacob. *useit.com: Jakob Nielsen's Website* [online]. [Fremont (California)] : [b.n.], 1994 [cit. 2006-08-05]. Dostupný z WWW: <http://www.useit.com/papers/guerrilla_hci.html>.
29. NIELSEN, Jacob. 1994b. *Usability Engineering*. San Francisco : Morgan Kaufmann, 1994. 362 s. ISBN 0-12-518406-9.
30. NIELSEN, Jacob. 2000. Why You Only Need to Test With 5 Users. In NIELSEN, Jacob. *useit.com: Jakob Nielsen's Website* [online]. [Fremont (California)] : [b.n.], March 2000 [cit. 2006-08-07]. Dostupný z WWW: <<http://www.useit.com/alertbox/20000319.html>>.
31. NIELSEN, Jacob. 2002. *Web.Design*. Praha : SoftPress, c2002. 382 s. ISBN 80-86497-27-5.
32. NIELSEN, Jacob. 2003. Usability 101: Introduction to Usability. In NIELSEN, Jacob. *useit.com: Jakob Nielsen's Website* [online]. [Fremont (California)] : [b.n.], August 2003 [cit. 2006-08-07]. Dostupný z WWW: <<http://www.useit.com/alertbox/20030825.html>>.

33. NUNES, Nuno J; CUNHA, João Falcão. 2000. *Wisdom – Whitewater Interactive System Development with Object Models*. 2000. 37 s. Dostupný také z WWW: <<http://www.dsic.upv.es/~west/TIMIU'02/ficheros/Wisdom.pdf>>.
34. NUNES, Nuno J. 2001. *Object Modeling for User-Centered Development and User Interface Design: The Wisdom Approach*. Funchal (Portugal), 2001. xxvi, 259 s. Disertační práce (PhD). Universidade da Madeira. Dostupná také z WWW: <<http://dme.uma.pt/njn/publications/PhD.pdf>>.
35. PAPÍK, Richard. 2001. Vyhledávání informací II. Uživatelské rozhraní a vlivy oboru „human-computer interaction“. *Národní knihovna : knihovnická revue*. 2001, roč. 12, č. 2, s. 81–90. Dostupný také z WWW: <<http://knihovna.nkp.cz/pdf/0102/nk0102081.pdf>>. ISSN 0862-7487.
36. PERLMAN, Gary. 1994. Practical usability evaluation. In *Conference companion on Human factors in computing systems : Boston, Massachusetts, United States, April 24–28, 1994*. New York (USA) : ACM Press, 1994, s. 407–408. Dostupný také z WWW: <<http://portal.acm.org>>. ISBN 0-89791-651-4.
37. PIATTINI, Mario G.; TIGRÉAT, Hervé. 1997. Applying the „STUDIO“ method to the interface design of an environmental software system. *ACM SIGSOFT Software Engineering Notes*. July 1997, vol. 22, is. 4, s. 81–83. Dostupný také z WWW: <<http://portal.acm.org>>. ISSN 0163-5948.
38. SHNEIDERMAN, Ben; PLAISANT, Catherine. 2004. *Designing the user interface : strategies for effective human-computer interaction*. 4th ed. Boston : Pearson/Addison Wesley, 2004. xviii, 652 s. ISBN 0321197860.
39. SOUČKOVÁ, Martina. 2003. *Aspekty vztahu „člověk – počítač“ s důrazem na uživatelské rozhraní*. Praha, 2003. 89 s. Diplomová práce (Mgr.). Univerzita Karlova v Praze, Filozofická fakulta, Ústav informačních studií a knihovnictví.
40. TULLIS, Thomas S.; STETSON, Jacqueline N. 2004. A Comparison of Questionnaires for Website Usability. In *UPA 2004: Connecting Communities : June 7–11, 2004, Minneapolis, Minnesota* [online]. Bloomington (Illinois) : Usability Professionals' Association, 2004 [cit. 2006-08-06]. Dostupný z WWW: <http://www.upassoc.org/usability_resources/conference/2004/UPA-2004-TullisStetson.pdf>.

41. World Wide Web Consortium. 2006. *Web Content Accessibility Guidelines 2.0* [online]. W3C Working Draft 27 April 2006. W3C, c2006 [cit. 2006-05-09]. Dostupný z WWW:
<<http://www.w3.org/TR/2006/WD-WCAG20-20060427/>>, poslední verze:
<<http://www.w3.org/TR/WCAG20/>>.

Seznam použitých symbolů a zkratk

GUI *graphic user interface*, česky *grafické uživatelské rozhraní*

HCI *human–computer interaction*, česky *interakce člověk–počítač*

UCD *user–centered design*, česky *návrh zaměřený na uživatele*

UI *user interface*, česky *uživatelské rozhraní*

UML *unified modeling language*, česky *jazyk unifikovaného modelování*

WIMP *window, icon, menu, pointing device*, česky *okno, ikona, nabídka, ukazovací zařízení*

Rejstřík

- acceptance tests, *viz* testy přijetí
- analýza, 11, 25
 - úloh, 22, 26
 - cíle analýzy, 25
 - potřeb uživatelů, 26
- Bridge, 14
- cíl, *viz* použitelnost
- ČSN
 - 6385, 6, 7
 - 9241, 7
 - 13407, 6, 7
- dotazníky, *viz* průzkumy
- efektivnost, *viz* měřítka použitelnosti
- ergonomické projektování, 13
- ETP, 14
- expert review, *viz* odborná revize
- fáze vývoje, 11
- goal, *viz* použitelnost
- GUIDE, 15
- HCI, *viz* interakce člověk–počítač
- heuristika, 30, 33
- human-computer interaction, *viz* interakce člověk–počítač
- IDIOM, 18
- implementace, 11, 26
- interakce člověk–počítač, 3
- iterativnost, 12
- laboratoř, *viz* uživatelské testování
- LUCID, 20
- man-machine interface, *viz* uživatelské rozhraní
- metodické postupy, 13
- metodologie, *viz* metodické postupy
- měřítka použitelnosti, 29
- míra použitelnosti, *viz* měřítka použitelnosti
- MMI, *viz* uživatelské rozhraní
- mock-ups, *viz* papírové modely, *viz* papírové modely
- model, 14
 - programátorův, 21
 - projektantův, 21
 - uživatelův, 21
- návrh, 11
- normy, 6
 - standardy, 8
 - vodítka, 8
- objektově orientovaný, 13
- odborná revize, 29
- osm zásad, *viz* zlatá pravidla
- ověření, 11
- OVID, 21
- papírové modely, 28, 32
- participatory design, 28

PICTIVE, 28
použitelnost, 9, 10
product, *viz* použitelnost
prototypy, 17
 horizontální, 17
 na jedno použití, 17
 vertikální, 17
průzkumy, 35

realizace, *viz* implementace
revize, *viz* odborná revize

scénáře, 33
STUDIO, 14

task, *viz* použitelnost
task analysis, *viz* analýza úloh, *viz* ana-
 lýza úloh
testování, 11
testy přijetí, 36

UI, *viz* uživatelské rozhraní
UIM, 14
UML, 23
usability, 9, 10
usability testing, *viz* uživatelské testování
user interface, *viz* uživatelské rozhraní
user needs, *viz* analýza
user-centered design, 7
uspokojení, *viz* měřítka použitelnosti
uživatelské rozhraní, 3
 grafické, 3
uživatelské testování, 31

úkol, *viz* použitelnost

výkonnost, *viz* měřítka použitelnosti
výrobek, *viz* použitelnost

Wisdom, 22
zlatá pravidla, 27

Příloha A

Vybrané standardy upravující tvorbu UI z hlediska použitelnosti

V tabulce A.1 najdeme přehled několika norem vztahujících se k tématu tvorby uživatelského rozhraní. Některé z nich se nezabývají softwarem, na který je tato práce zaměřena, avšak pro názornost je zde uvádím. Každá z nich je mezinárodní standard, ale pokud k němu existuje český ekvivalent, uvádím označení české normy.

Jde jen o výběr a rozhodně zde není uveden úplný výčet, který by byl pro tento dokument příliš rozsáhlý.

Označení	Název	Poznámka
ČSN EN ISO 6385	Ergonomické zásady navrhování pracovních systémů	
ČSN EN ISO 13407	Postupy ergonomického projektování interakčních systémů	
ČSN EN ISO 9241–(1–17)	Ergonomické požadavky na kancelářské práce se zobrazovacími terminály	sada norem, části 1–17
ISO 10075:1991	Ergonomic principles related to mental work-load – General terms and definitions	

Tabulka A.1 – pokračování na další straně

Tabulka A.1 – pokračování z předchozí strany

Označení	Název	Poznámka
ČSN EN ISO 10075–(2–3)	Ergonomické zásady ve vztahu k mentální pracovní zátěži	sada norem, části 2–3, navazují na ISO 10075:1991
ČSN EN ISO 13406–(1–2)	Ergonomické požadavky na práce se zobrazovacími displeji založenými na plochých panelech	sada norem, části 1–2
ISO/IEC 9126–(1–4):2001	Software engineering – Product quality	sada norem, části 1–4
ČSN ISO IEC 9126–1	Softwarové inženýrství – Jakost produktu – Část 1: Model jakosti	český překlad ISO/IEC 9126–1
ČSN ISO/IEC 10741–1+Amd.1	Informační technologie – Rozhraní uživatelských systémů – Dialogová interakce – Část 1: Řízení kurzoru pro editování textu	
ČSN ISO/IEC 11581–(1, 2, 3, 6)	Informační technologie – Rozhraní uživatelských systémů a symboly – Symboly a funkce ikon	sada norem, části 1, 2, 3 a 6
ISO/IEC 11581–5:2004	Information technology – User system interfaces and symbols – Icon symbols and functions – Part 5: Tool icons	sada norem, části 1, 2, 3, 5 a 6, oproti českému překladu navíc pátá část
ČSN ISO/IEC 9995–1	Informační technologie – Uspořádání klávesnic pro textové a kancelářské systémy	sada norem, části 1–7

Tabulka A.1: Vybrané standardy upravující tvorbu UI z hlediska použitelnosti

Příloha B

Přehled dotazníků používaných pro ověřování použitelnosti UI

V tabulce B.1 je uvedeno několik vybraných dotazníků používaných pro ověřování použitelnosti UI, které mohou testovací týmy případně upravit a aplikovat v jejich práci. Některé z nich jsou volně k dispozici, u jiných je vyžadován licenční poplatek. Pro testovací týmy však představují velkou výhodu – ušetření času a nemalých finančních prostředků při přípravě vlastních dotazníků.

Zkratka	Název	Autor
QUIS	Questionnaire for User Interface Satisfaction	Chin aj. [1988]
PUEU	Perceived Usefulness and Ease of Use	Davis [1989]
NAU	Nielsen's Attributes of Usability	Nielsen [1994b]
NHE	Nielsen's Heuristic Evaluation	Nielsen [1994b]
CSUQ	Computer System Usability Questionnaire	Lewis [1995]
ASQ	After Scenario Questionnaire	Lewis [1995]
PHUE	Practical Heuristics for Usability Evaluation	Perlman [1994]
PUTQ	Purdue Usability Testing Questionnaire	Lin aj. [1997]
SUMI	Software Usability Measurement Inventory	Kirakowski [1994]
MUMMS	Measurement of Usability of Multi Media Software	HFRG [1997]
WAMMI	Website Analysis and Measurement Inventory	Kirakowski aj. [2005]

Tabulka B.1: Vybrané dotazníky používané pro ověřování použitelnosti UI

Na obrázku B.1 je seznam 118 slov hodnotících produkt, používaných na kartách odezvy při hodnocení dojmů z používání produktu. Tuto metodu zjišťování použitelnosti produktů vyvinula firma Microsoft. Zde uvádím její dotazníkovou formu.

<input type="checkbox"/> Accessible	<input type="checkbox"/> Creative	<input type="checkbox"/> Fast	<input type="checkbox"/> Meaningful	<input type="checkbox"/> Slow
<input type="checkbox"/> Advanced	<input type="checkbox"/> Customizable	<input type="checkbox"/> Flexible	<input type="checkbox"/> Motivating	<input type="checkbox"/> Sophisticated
<input type="checkbox"/> Annoying	<input type="checkbox"/> Cutting edge	<input type="checkbox"/> Fragile	<input type="checkbox"/> Not Secure	<input type="checkbox"/> Stable
<input type="checkbox"/> Appealing	<input type="checkbox"/> Dated	<input type="checkbox"/> Fresh	<input type="checkbox"/> Not Valuable	<input type="checkbox"/> Sterile
<input type="checkbox"/> Approachable	<input type="checkbox"/> Desirable	<input type="checkbox"/> Friendly	<input type="checkbox"/> Novel	<input type="checkbox"/> Stimulating
<input type="checkbox"/> Attractive	<input type="checkbox"/> Difficult	<input type="checkbox"/> Frustrating	<input type="checkbox"/> Old	<input type="checkbox"/> Straight Forward
<input type="checkbox"/> Boring	<input type="checkbox"/> Disconnected	<input type="checkbox"/> Fun	<input type="checkbox"/> Optimistic	<input type="checkbox"/> Stressful
<input type="checkbox"/> Business-like	<input type="checkbox"/> Disruptive	<input type="checkbox"/> Gets in the way	<input type="checkbox"/> Ordinary	<input type="checkbox"/> Time-consuming
<input type="checkbox"/> Busy	<input type="checkbox"/> Distracting	<input type="checkbox"/> Hard to use	<input type="checkbox"/> Organized	<input type="checkbox"/> Time-Saving
<input type="checkbox"/> Calm	<input type="checkbox"/> Dull	<input type="checkbox"/> Helpful	<input type="checkbox"/> Overbearing	<input type="checkbox"/> Too Technical
<input type="checkbox"/> Clean	<input type="checkbox"/> Easy to use	<input type="checkbox"/> High quality	<input type="checkbox"/> Overwhelming	<input type="checkbox"/> Trustworthy
<input type="checkbox"/> Clear	<input type="checkbox"/> Effective	<input type="checkbox"/> Impersonal	<input type="checkbox"/> Patronizing	<input type="checkbox"/> Unapproachable
<input type="checkbox"/> Collaborative	<input type="checkbox"/> Efficient	<input type="checkbox"/> Impressive	<input type="checkbox"/> Personal	<input type="checkbox"/> Unattractive
<input type="checkbox"/> Comfortable	<input type="checkbox"/> Effortless	<input type="checkbox"/> Incomprehensible	<input type="checkbox"/> Poor quality	<input type="checkbox"/> Uncontrollable
<input type="checkbox"/> Compatible	<input type="checkbox"/> Empowering	<input type="checkbox"/> Inconsistent	<input type="checkbox"/> Powerful	<input type="checkbox"/> Unconventional
<input type="checkbox"/> Compelling	<input type="checkbox"/> Energetic	<input type="checkbox"/> Ineffective	<input type="checkbox"/> Predictable	<input type="checkbox"/> Understandable
<input type="checkbox"/> Complex	<input type="checkbox"/> Engaging	<input type="checkbox"/> Innovative	<input type="checkbox"/> Professional	<input type="checkbox"/> Undesirable
<input type="checkbox"/> Comprehensive	<input type="checkbox"/> Entertaining	<input type="checkbox"/> Inspiring	<input type="checkbox"/> Relevant	<input type="checkbox"/> Unpredictable
<input type="checkbox"/> Confident	<input type="checkbox"/> Enthusiastic	<input type="checkbox"/> Integrated	<input type="checkbox"/> Reliable	<input type="checkbox"/> Unrefined
<input type="checkbox"/> Confusing	<input type="checkbox"/> Essential	<input type="checkbox"/> Intimidating	<input type="checkbox"/> Responsive	<input type="checkbox"/> Usable
<input type="checkbox"/> Connected	<input type="checkbox"/> Exceptional	<input type="checkbox"/> Intuitive	<input type="checkbox"/> Rigid	<input type="checkbox"/> Useful
<input type="checkbox"/> Consistent	<input type="checkbox"/> Exciting	<input type="checkbox"/> Inviting	<input type="checkbox"/> Satisfying	<input type="checkbox"/> Valuable
<input type="checkbox"/> Controllable	<input type="checkbox"/> Expected	<input type="checkbox"/> Irrelevant	<input type="checkbox"/> Secure	
<input type="checkbox"/> Convenient	<input type="checkbox"/> Familiar	<input type="checkbox"/> Low Maintenance	<input type="checkbox"/> Simplistic	

Obrázek B.1: Dotazníková forma karet odezvy na produkt (Product Reaction Cards) [Benedek aj., 2002b; Tullis aj., 2004]

Na obrázku B.2 je sada 19 otázek dotazníku na použitelnost počítačových systémů (Computer System Usability Questionnaire) se sedmibodovou hodnotící škálou vytvořený firmou IBM [Lewis, 1995].

			1	2	3	4	5	6	7		NA
1.	Overall, I am satisfied with how easy it is to use this system	strongly disagree	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	strongly agree	<input type="checkbox"/>
2.	It was simple to use this system	strongly disagree	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	strongly agree	<input type="checkbox"/>
3.	I can effectively complete my work using this system	strongly disagree	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	strongly agree	<input type="checkbox"/>
4.	I am able to complete my work quickly using this system	strongly disagree	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	strongly agree	<input type="checkbox"/>
5.	I am able to efficiently complete my work using this system	strongly disagree	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	strongly agree	<input type="checkbox"/>
6.	I feel comfortable using this system	strongly disagree	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	strongly agree	<input type="checkbox"/>
7.	It was easy to learn to use this system	strongly disagree	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	strongly agree	<input type="checkbox"/>
8.	I believe I became productive quickly using this system	strongly disagree	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	strongly agree	<input type="checkbox"/>
9.	The system gives error messages that clearly tell me how to fix problems	strongly disagree	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	strongly agree	<input type="checkbox"/>
10.	Whenever I make a mistake using the system, I recover easily and quickly	strongly disagree	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	strongly agree	<input type="checkbox"/>
11.	The information (such as online help, on-screen messages, and other documentation) provided with this system is clear	strongly disagree	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	strongly agree	<input type="checkbox"/>
12.	It is easy to find the information I needed	strongly disagree	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	strongly agree	<input type="checkbox"/>
13.	The information provided for the system is easy to understand	strongly disagree	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	strongly agree	<input type="checkbox"/>
14.	The information is effective in helping me complete the tasks and scenarios	strongly disagree	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	strongly agree	<input type="checkbox"/>
15.	The organization of information on the system screens is clear	strongly disagree	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	strongly agree	<input type="checkbox"/>
16.	The interface of this system is pleasant	strongly disagree	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	strongly agree	<input type="checkbox"/>
17.	I like using the interface of this system	strongly disagree	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	strongly agree	<input type="checkbox"/>
18.	This system has all the functions and capabilities I expect it to have	strongly disagree	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	strongly agree	<input type="checkbox"/>
19.	Overall, I am satisfied with this system	strongly disagree	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	strongly agree	<input type="checkbox"/>

Obrázek B.2: Dotazník na použitelnost počítačových systémů [Lewis, 1995]

