

Vysoká škola ekonomická v Praze

Fakulta informatiky a statistiky

Katedra informačních technologií

Studijní program : Aplikovaná informatika

Obor: Informační systémy a technologie

Analýza vlivu konvencí na použitelnost UI

DIPLOMOVÁ PRÁCE

Student : Bc. Pavel Toušek

Vedoucí : Ing. Zuzana Šedivá, Ph.D.

Oponent : doc. Ing. Alena Buchalceková, Ph.D.

2012

Prohlášení:

Prohlašuji, že jsem diplomovou práci zpracoval samostatně a že jsem uvedl všechny použité prameny a literaturu, ze kterých jsem čerpal.

V Praze dne 24. června 2012

.....
Pavel Toušek

Poděkování

Tímto bych rád poděkoval paní Ing. Zuzaně Šedivé, Ph.D., především za možnost pracovat na tomto tématu, za její pomoc a cenné rady v začátcích, za trpělivost a v neposlední řadě i za vstřícnost, se kterou do spolupráce se mnou vstupovala.

Abstrakt

Tématem práce je dopad odlišných platformních konvencí uživatelského rozhraní na hodnoty použitelnosti a uživatelské návyky. Teoretická část sbírá podklady ze souborů zásad použitelnosti a výsledků empirického výzkumu v souvisejících oblastech. Uskutečněný výzkum využívá technik nemoderovaného vzdáleného testování použitelnosti pro získání kvantitativních dat o použitelnosti tří odlišných prototypů stejné webové aplikace s různými grafickými uživatelskými rozhraními. Statistické testování získaných dat prokazuje rozdíly mezi hodnotami použitelnosti jednotlivých prototypů, zvláště významné je pak prokázání zužitkování odlišných uživatelských návyků v závislosti na typu testovaného rozhraní. Tyto výsledky prohlubují znalosti o vazbě uživatelských návyků na prostředí.

Klíčová slova

Použitelnost, uživatelské rozhraní, platformní konvence, testování použitelnosti, webové aplikace.

Abstract

This thesis focuses on the impact of user interface's compliance with different platform conventions on usability measures and user habits. Theoretical background is provided by usability guidelines and preceding research papers on related topics. Conducted research incorporates techniques of unmoderated remote usability testing for obtaining quantitative data on usability of three different prototypes of the same web application with different graphical user interfaces. Confirmatory data analysis proves differences between prototypes' usability measures, more importantly, it proves utilization of different user habits in each tested visual interface type. These results broaden the knowledge of user habits in relation to the interface environment.

Keywords

Usability, user interface, platform conventions, usability testing, web applications.

Obsah

1	Úvod.....	1
1.1	Cíle práce.....	2
1.2	Proč toto téma.....	3
2	Současný stav poznání.....	4
2.1	Vymezení pojmů.....	4
2.1.1	Uživatelské rozhraní	4
2.1.2	Použitelnost.....	5
2.1.3	Hypertextové prostředí.....	7
2.1.4	Desktopové prostředí	8
2.2	Rešerše	9
2.2.1	Wilbert O. Galitz: The Essential Guide to User Interface Design	9
2.2.2	Jenifer Tidwell: Designing Interfaces	10
2.2.3	Použitelnost domovských stránek	11
2.2.4	Could I Have the Menu Please? An Eye Tracking Study of Design Conventions	12
2.2.5	The Impact of Visual Layout Factors on Performance in Web Pages: A Cross-Language Study	14
2.3	Další teoretická východiska	16
2.3.1	Ovládací prvky	16
2.3.2	Testování použitelnosti	21
3	Příprava a průběh výzkumu	27
3.1	Princip výzkumu	27
3.1.1	Průběh testování.....	27
3.1.2	První prototyp – „hypertextový“	31
3.1.3	Druhý prototyp – „desktopový“	32
3.1.4	Třetí prototyp – „dotykový“	34
3.2	Technické zpracování.....	36
3.2.1	Architektura:.....	37
3.2.2	Výběr technologií:.....	38
3.3	Skupina účastníků testování.....	39
4	Vyhodnocení výsledků výzkumu	40
4.1	Základní údaje o souboru.....	40
4.2	Míra účinnosti.....	40
4.2.1	Nový dokument	43
4.2.2	Vypsát otázky	45
4.2.3	Vyplnit dotazník a zkontrolovat odpovědi.....	45

4.2.4	Najít a nahradit	46
4.2.5	Vrátit poslední krok.....	46
4.2.6	Uložit	46
4.2.7	Ukončit.....	47
4.3	Míra chybovosti a míra úspěšného dokončení testu	47
4.4	Alternativní postupy řešení.....	49
4.4.1	Najít a nahradit	49
4.4.2	Zpět	50
4.4.3	Ukončit.....	52
4.5	Klávesové zkratky	53
4.6	Pohyb myši.....	54
4.7	Subjektivní hodnocení a zpětná vazba.....	55
4.8	Hodnoty dle jednotlivých prototypů.....	55
4.8.1	První prototyp – „hypertextový“	55
4.8.2	Druhý prototyp – „desktopový“	56
4.8.3	Třetí prototyp – „dotykový“	56
4.9	Hodnoty dle počítačové gramotnosti	57
4.10	Výsledky	58
4.11	Zhodnocení a aplikace výsledků	58
5	Závěr	61
5.1	Přínos práce	62
	Terminologický slovník.....	63
	Seznam literatury.....	65
	Seznam tabulek, obrázků a grafů	68
	Seznam tabulek	68
	Seznam obrázků	68
	Seznam grafů	68
	Seznam příloh	69

1 Úvod

Cílem této práce je v testovacím prostředí kvantifikovat vliv konvencí uživatelských rozhraní na použitelnost webových aplikací a uživatelské návyky. Shoda s normami a platformními konvencemi patří mezi nezákladnější požadavky na kvalitní UI (Nielsen, 2005), ovšem v době překrývání platforem není vždy jednoznačné, které konvence by měly být dodrženy. Jak jsem demonstroval v dřívější práci (Toušek, 2009) na uživatelských rozhraních webových a desktopových aplikací, jednotlivé platformy se výrazně liší ve striktnosti dodržování konvencí a míře jejich detailnosti, což se výrazně projevuje v očekávání uživatelů a tím jejich schopnosti adaptovat se v rámci platformy na odlišná jednotlivá GUI. Konkrétně se ukázalo, že diverzita webového prostředí vede uživatele k vyšší adaptabilitě, ovšem existuje výrazný problém v nejednoznačnosti hranice mezi statickým obsahem a aplikací v prostředí webu.

Prostředí World Wide Webu bylo ve svých počátcích pouze hypertextové. Hypertext znamená nelineárně strukturovaný textový obsah. Stránky tedy měly čistě statický, obsahový charakter, stránka sama nedisponovala žádnou funkcionalitou. Hranice mezi stránkou a prohlížečem byla velmi jasně definovaná a jasné tím byly i možnosti uživatelské interakce.

Ve webovém prohlížeči uživatel pracoval – a dodnes většinou pracuje – s použitím sekvence: kliknutí, odeslání požadavku, čekání na překreslení stránky. (Powell, 2008) V současné době jsou běžně vyvíjeny i velmi komplexní webové aplikace, které více či méně důsledně implementují konvence desktopových uživatelských rozhraní. To mimo jiné znamená, že přinášejí do webového prostředí ovládací prvky, známé a naučené z prostředí desktopových aplikací, avšak neobvyklé v prostředí hypertextového webu. Tím, že rozhraní vzhledem připomíná

desktopovou aplikaci, vybízí uživatele k využívání návyků z těchto aplikací. Problémem je však nekonzistence. Každá aplikace implementuje tyto původně desktopové ovládací prvky v jiné míře.

Uživatel je tím vystaven nejistotě. Na webu se pohybuje v prostředí, které v jednotlivých případech plynule přechází od čistě hypertextového obsahu po aplikace věrně replikující desktopové chování. Je tím drasticky narušena spolehlivost jeho očekávání. To znamená, že nemůže spolehlivě použít své uživatelské návyky, což snižuje použitelnost rozhraní zásadní mírou. Zatím však nebylo uspokojivě prozkoumáno, do jaké míry se na webu mění postupy uživatele v závislosti na typu prostředí, kterému je vystaven.

1.1 Cíle práce

Prvním cílem této práce je v testovacím prostředí prokázat, zda existuje prokazatelný vztah mezi konkrétními platformními konvencemi pro vzhled (look and feel) grafického uživatelského rozhraní a výslednou použitelností, tedy jak účinně a přesně se toto rozhraní ovládá.

Druhým cílem je prokázání adaptace odpovídajících uživatelských návyků pro rozdílná prostředí.

Za tímto účelem je využito techniky uživatelského testování na odlišných prototypch uživatelského rozhraní webové aplikace. Na základě statistického zpracování a testování získaných dat pak bude prokázána nebo vyvrácena hypotéza o vztahu mezi odlišnými konvencemi uživatelského rozhraní a použitelností:

H1: Typ uživatelského rozhraní má vliv na účinnost.

H2: Typ uživatelského rozhraní má vliv na přesnost.

Pro druhý cíl, prokázání adaptace odlišných uživatelských návyků v závislosti na prostředí, je stanovena hypotéza:

H3: Typ uživatelského rozhraní má vliv na užití odlišných uživatelských návyků.

1.2 Proč toto téma

Hlavním důvodem, proč v této práci zkoumám popsané téma, jsou nezodpovězené otázky, které vyvstaly při zpracování mé bakalářské práce a následných dalších prací. Problematikou uživatelského rozhraní webových aplikací se zabývám dlouhodobě a zde zkoumaná oblast stále zůstává otevřená a z mého pohledu nedostatečně řešená. Proto považuji za přínosné kromě analýzy stávajícího stavu poznání provést vlastní výzkum. Očekávaným přínosem je prohloubení aktuální znalosti o významu konvencí uživatelských rozhraní a jejich dopadu na použitelnost a uživatelské návyky.

2 Současný stav poznání

2.1 Vymezení pojmů

Protože názvosloví ve zkoumané oblasti je často využíváno relativně volně, je třeba si přesně vymezit používané pojmy.

Klíčovými pojmy v této práci jsou *uživatelské rozhraní*, čímž je rozuměno *grafické uživatelské rozhraní* (graphical user interface, GUI), a *použitelnost* (usability). Tyto pojmy spadají do oblasti zvané *human-computer interaction* (HCI), což je pole výzkumu zabývající se celou širokou škálou možností, jak mohou člověk a počítač vzájemně komunikovat a spolupracovat. Výzkum HCI se zaměřuje na to, co lidé chtějí a očekávají, jaké fyzické schopnosti a omezení mají, jak vnímají a zpracovávají informace a co považují za pěkné či příjemné. Cílem HCI je na základě těchto poznatků navrhnout takový způsob práce člověka s počítačem, který co nejefektivněji uspokojí potřeby jedince. (Galitz, 2002)

2.1.1 Uživatelské rozhraní

„Uživatelské rozhraní je souhrnem aspektů počítačového systému či programu, které může člověk jako uživatel vidět nebo jiným způsobem vnímat, a příkazy a mechanismy, kterými uživatel řídí činnost systému či programu a vkládá data“ (Howe, 2009) Dělí se na dvě části, vstup a výstup. Vstupem, tedy způsobem, jak předat počítači požadavky, je nejčastěji klávesnice, myš, dotyková plocha či hlasový vstup. Výstup je způsob, jakým počítač komunikuje nabídku možností, výsledky činnosti a požadavky. Dnes nejběžnějším výstupním zařízením je obrazovka, následovaná zvukovým a hlasovým výstupem. (Galitz, 2002)

2.1.2 Použitelnost

Při veliké šířce významu souhrnného označení *user experience* (UX) se pojem *použitelnost* uplatňuje v rozmanitých kontextech. Podle zaměření konkrétních oborů UX potom samotný pojem s dynamikou odpovídající této rozmanitosti mění význam.

Tabulka 1: Strategie a nástroje pro tvorbu řešení uživatelských rozhraní

Fáze		Cíl	Strategie a nástroje
Návrh	Analýza	Poznat, pochopit a analyzovat řešenou oblast a vytvořit základní požadavky na uživatele a uživatelské rozhraní	• Vytvořit a analyzovat scénáře • Různé typy analýz úkolů (task analysis) • Pozorovací techniky
	Konceptuální návrh	Vytvořit konceptuální model uživatelského rozhraní	• Uživatelské mentální modelování • Přibližné prototypy
	Detailní návrh	Přidat všechny podrobnosti do konceptuálního modelu	• <i>Guidelines</i> pro vizuální návrh • Konvence a standardy • Výběr standardních nebo nestandardních ovládacích prvků • Přibližné až velmi věrné prototypy
Vývoj		Vyřešit problémy uživatelského rozhraní plynoucí z omezení možností implementace nebo případných změn požadavků	• Analytické techniky • Vstup od programátorů • <i>Guidelines</i> pro vizuální návrh a ovládací prvky • Konvence a standardy • Přibližné až velmi věrné prototypy
Nasazení		Vyřešit problémy uživatelského rozhraní, které vyplynuly při vlastním spuštění v produkčním prostředí nebo z reálného nasazení	• Pozorování • Záznamy o incidentech • Další techniky sběru dat

Zdroj: (Parush, 2001)

Jak píše Avi Parush (Parush, 2001), při hledání řešení uživatelského rozhraní lze postupovat mnoha způsoby a každý představuje odlišný přístup nejen k vývoji, ale i k vlastnímu pojmu *použitelnost*. Parush vidí dva opačné póly: Jedním je použitelnost jako soubor standardů, zásad a široké škály dostupných výsledků vědeckého výzkumu. Na opačné straně je použitelnost konkrétní měřitelnou hodnotou, výsledkem evaluace a testování. Tyto pohledy by se potom podle něj měly v průběhu vývoje SW doplňovat, přičemž pozornost vývojářů by měla začínat u teorie a pomalu se přesouvat k empirii, jak ukazuje průběh vývoje UI během vývojového cyklu aplikace v tabulce (Tabulka 1: Strategie a nástroje pro tvorbu řešení uživatelských rozhraní).

Pro základní vysvětlení této veličiny poslouží dobře Shackelova zkrácená varianta definice, která definuje použitelnost systému jako „*způsobilost být lidmi používán snadno a účinně, kdy*

- *snadno = do stanovené míry subjektivního ohodnocení,*
- *účinně = do stanovené míry výkonnosti.“*

(Shackel, Richardson, 1991, s. 24)¹

O něco přesnější je definice dle normy ISO/IEC 25010: „*Míra, do jaké může být produkt nebo systém používán stanovenými uživateli pro dosažení stanovených cílů účinně, přesně a uspokojivě ve stanoveném kontextu použití.*“ (ISO/IEC 25010:2011, 2011) ²

¹ “The capability to be used by humans easily and effectively, where easily = to a specified level of subjective assessment, effectively = to a specified level of (human) performance.”

² “Degree to which a product or system can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.”

Tato definice nastavuje podmínky, které je důležité chápat jako její nedílnou součást: *Stanovení uživatelé* znamenají, že použitelnost má smysl se zabývat pouze z perspektivy uživatelů, pro které je produkt nebo systém určen. *Stanovené cíle* jsou ty cíle, jejichž plnění je zamýšleným účelem daného produktu nebo systému. Nelze hodnotit použitelnost při provádění úkolů, pro které produkt nebo systém nebyl navržen. *Stanovený kontext použití* potom znamená souvislosti a prostředí, se kterými se při návrhu produktu nebo systému počítá. Předpokládá se přitom, že stanovené cíle budou odpovídat cílům stanovených uživatelů a stanovený kontext bude představovat kontext, ve kterém budou tito uživatelé produkt nebo systém používat. (Barnum, 2010)

Pro účely této práce je klíčové, že použitelnost je měřitelná. V souladu se zněním definice dle ISO/IEC 25010:2011 stanovuje norma *ANSI/INCITS-354: Common Industry Format for Usability Test Reporting* výkonnostní metriky použitelnosti:

- Účinnost – jak dlouho trvá vykonání úkolu
- Přesnost – kolik chyb
- Uspokojení – subjektivní hodnocení

V druhé řadě jsou to metriky spojené s opakovaným používáním:

- Naučitelnost – jak dlouho trvá naučení
- Zapamatovatelnost – jak dlouho trvá znovunaučení

(ANSI/INCITS-354, 2001)

2.1.3 Hypertextové prostředí

Hypertextovým prostředím bude rozuměno uživatelské rozhraní odpovídající konvencím webových stránek prezentačního charakteru, bez

dynamických prvků. Taková stránka je čistě obsahem, nemá žádnou client-side funkcionalitu. Možnost práce se zobrazeným obsahem je pouze na straně funkční výbavy prohlížeče: vyhledávání na stránce, úpravy zobrazení a podobně. Interakce s obsahem pak je jedinečně synchronní: prostřednictvím hypertextového odkazu nebo odesláním formuláře je zaslána žádost na server a čeká se na odpověď. Po obdržení odpovědi se vykreslí nová stránka.

2.1.4 Desktopové prostředí

Pojem *desktopové prostředí* bude značit grafické uživatelské rozhraní nabízející *look and feel* aplikačního SW na běžných operačních systémech osobních počítačů. Vzhledem k penetraci a vyplývajícímu vlivu jde zejména o prostředí Microsoft Windows a Apple Mac OS / OS X. Jejich konvence se i v některých základních prvcích liší. Jedním z koncepčních rozdílů je rozdílná adaptace aplikačního menu. Apple se v systému Mac OS rozhodl pro jediné menu na společné vrchní liště, přičemž menu je platné pro právě aktivní okno. Microsoft volí jednotlivá menu pro každé okno. Další rozdíl v celkovém pojetí, kdy Apple se snaží o maximální jednoduchost, se projevuje v praktické absenci funkce pravého tlačítka myši, které v jiných prostředích vyvolává kontextovou nabídku. Jinak jsou však základní konvence stejné či hodně blízké.

Hlavním prvkem desktopových GUI, přenášeným do webových aplikací, je samotný layout, tedy rozložení grafických prvků na obrazovce a vzhled ustálených ovládacích prvků. Dalšími jsou ovládání zavedenými klávesovými zkratkami, manipulace s objekty stylem *drag and drop*. Poněkud v pozadí je ovládání pravým tlačítkem myši, možná i pro zmíněnou nekonzistenci napříč platformami.

2.2 Rešerše

Současný stav poznání zkoumaného oboru je obecně na vysoké úrovni, oblast je značně komerčně zajímavá a nové poznatky jsou velmi rychle převáděny do praxe. Zároveň však rychlý vývoj technologií otevírá nové problémy a témata k dalšímu výzkumu.

Základními zdroji v oblasti návrhu uživatelského rozhraní jsou takzvané *guidelines*³, které shrnují stávající konvence, *best-practices*⁴ i výsledky vědeckých výzkumů a formulují tak základní teoretické podklady pro návrh uživatelského rozhraní. Úroveň poznání je dále posouvána jednotlivými výzkumnými pracemi, řešícími nově se objevující problémy nebo vylepšujícími a rozšiřujícími stávající metodologii.

2.2.1 Wilbert O. Galitz: The Essential Guide to User Interface Design

The Essential Guide to User Interface Design: an Introduction to GUI Design Principles and Techniques (Galitz, 2002) patří mezi základní *GUI guidelines*, je velmi zevrubným průvodcem celým vývojem uživatelského rozhraní, od schopností a omezení člověka při práci s počítačem po evaluaci finálního nasazeného systému. Představuje konkrétní postupy pro každou oblast návrhu. Pozitivem je, že maximálně vychází z výsledků výzkumů. Negativem ovšem je omezení zpravidla na jediné řešení a až přílišné zaměření na detail bez zasazení do kontextu. Často chybí širší pohled na společné uspořádání prvků.

Jako zdroj pro potřeby této práce jsou ovšem tyto guidelines zcela nepoužitelné, protože autor zaujímá k webu velmi odmítavý postoj, a

³ Soubory doporučených postupů založené konvencích i empirických výzkumech. Zpravidla jde o rozsáhlé publikace významných odborníků v dané oblasti. Pro absenci dostatečně přesného českého překladu pojmu bude dále používán pojem původní.

⁴ Soubor řešení, která jsou osvědčená, standardizovaná či prosazovaná. Pro absenci dostatečně přesného českého překladu pojmu bude dále používán pojem původní.

pokud se mu věnuje, tak převážně v rámci technických omezení na přelomu století, což pro nás dnes nemá hodnotu vůbec žádnou. Problematice konvencí napříč platformami se nevěnuje vůbec.

2.2.2 Jenifer Tidwell: Designing Interfaces

Kniha Jenifer Tidwellové *Designing Interfaces* (Tidwell, 2005), která je podrobným průvodcem návrhovými *best-practices*, je velmi zajímavá tím, že přináší otevřený pohled na návrh uživatelského rozhraní, aniž by se zaměřovala na konkrétní platformu, přitom však dostatečně prakticky, aby se věnovala jejich rozdílům. Řeší vždy stejnou problematiku v prostředích od desktopu přes web a mobilní zařízení po spotřební elektroniku a do značné míry přistupuje ke všem stejně, jen u různých postupů upozorňuje na výhody a omezení v konkrétních prostředích.

Autorka systematicky pro každou oblast návrhu uživatelského rozhraní představuje jednotlivé *vzory* a věnuje se jejich užití v zaběhnutých *idiomech* či jiných kombinacích. *Vzory* jsou vlastně návrhové *best-practices*, ukazují běžná řešení jednotlivých problémů. *Idiomem* (interface idiom) je rozpoznatelný typ rozhraní, každý se svojí sadou objektů, akcí a zobrazení. Pro ilustraci, idiomem je například textový editor, mediální přehrávač či elektronický obchod. Uživatel tyto typy rozhraní zná a okamžitě ví, jaký je jeho smysl a jak zhruba bude fungovat. Vzorem je konkrétní řešení zobrazení či ovládání, jako *panel nástrojů*, *průvodce* (wizard) nebo *drobečková navigace*.

Zajímavá je zmínka o kombinování naučených postupů z různých platforem v jednom prostředí, například v případě společného využití tlačítek a odkazů. Klasické tlačítko, které je známé už od počátků desktopového GUI – ale vlastně jde o vizuální reprezentaci fyzického ovladače – je použito pro vyvolání akce, zatímco odkaz, typicky modrý podržený text, slouží pro zobrazení dalšího obsahu. Pro téma této práce

je zajímavé, že každý z těchto ovládacích prvků se uživatel naučil používat v jiném prostředí, a přesto se příhodně doplňují vedle sebe jak ve webových, tak v desktopových aplikacích.

Ještě dva poznatky stojí za to v kontextu našeho tématu zmínit: První, že na webu a u webových aplikací oproti desktopu ještě roste význam kvalitního uživatelského rozhraní, protože frustrovaný uživatel může vzdát snahu a přejít ke konkurenci stiskem jednoho tlačítka. Druhý, že na zařízeních, kde chybí delší historie vývoje uživatelských rozhraní, jako jsou mobilní telefony, automobilové navigační systémy či televizní a záznamové audiovizuální systémy, designéři stále hledají základní principy, co funguje a co ne. Uživatelé často tolerují nesnadné ovládání, ale to nebude trvat dlouho. Zde lze vidět analogii s raným obdobím webu a webových aplikací, kdy byly teprve hledány fungující postupy a tolerance uživatelů byla vyšší. Nároky ale rostou velmi rychle.

První vydání je z roku 2005, přesto není vidět, že by obsah výrazně zestárl. Návrhových vzorů totiž stále přibývá a ty stávající jsou dále používány. Tím, že se Tidwellová nesnaží určovat nejlepší řešení, nýbrž nabízí celou sadu možností s jejich výhodami i nevýhodami, je kniha imunní proti módním trendům a dobře vzdoruje času. V roce 2011 vyšlo druhé vydání, které zejména přidává oblast sociálních médií a mobilních zařízení, a také několik nových vzorů, které se za poslední léta prosadily. Toto vydání bych označil za aktuálně nejkvalitnějšího průvodce možnostmi návrhu grafického uživatelského rozhraní.

2.2.3 Použitelnost domovských stránek

Dílo Jakoba Nielsena a Marie Tahirové (Nielsen, Tahir, 2005) je jedním z hlavních souborů „design guidelines“ pro návrh webu. Kvalitu českého vydání bohužel sráží čtyřleté zpoždění za originálem a nepříliš dobrý

překlad. Na příkladech padesáti velkých webů Nielsen ukazuje konvenční postupy řešení jednotlivých problémů. Jeho doporučení jsou jasná: co je nejčastější postup, je konvencí, kterou uživatel zná nejlépe. Na webu je velmi málo času zaujmout návštěvníka a velmi snadné ho odradit, proto zde není prostor pro učení. Nielsen a Tahirová z těchto vybraných webů vypracovali statistiku četnosti variant řešení čtyřiceti otázek, od umístění loga po formátování odkazů. Nejfrekventovanější řešení prohlašují za konvence, které by měly být dodrženy. Jde o jeden z extrémů v polarizované otázce, zda dávat předost nejběžnějším a tedy dobře známým řešením, nebo použít řešení podpořené vědeckými experimenty za cenu prolomení uživatelského očekávání.

2.2.4 Could I Have the Menu Please? An Eye Tracking Study of Design Conventions

Pro zkoumané téma je velmi zajímavá studie, kterou provedli John D. McCarthy, M. Angela Sasse a Jens Riegelsberger (McCarthy et al., 2003). Věnují se nejvhodnějšímu umístění navigačního menu. Část guidelines pro návrh webu vychází z uživatelského očekávání, jiné zase z výsledků empirického testování různých rozložení prvků. Tato studie technikou testování oční kamerou zkoumá, jestli umístění menu na nečekanou pozici má negativní dopad na použitelnost.

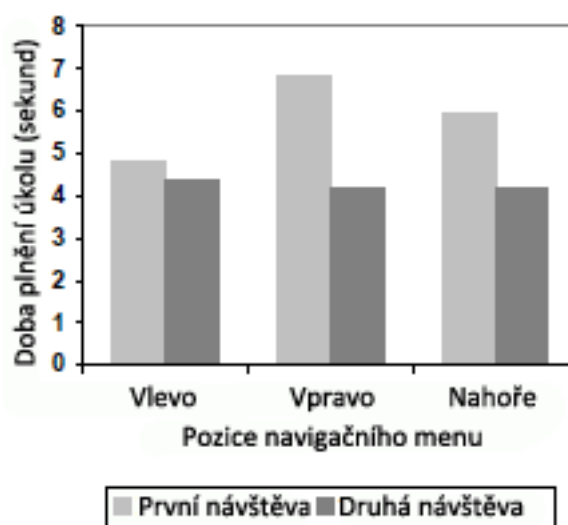
Pro umístění dle zavedených konvencí hovoří *IBM Design Guidelines* či *Jakob's Law of the Web User Experience*:

„Uživatelé tráví většinu svého času na jiných stránkách. Proto cokoliv je konvence používaná na většině jiných stránek, bude vypáleno v mozku uživatelů a můžete se od toho odklonit jen za cenu zásadních problémů s použitelností“ (Nielsen, 1999)

Na druhou stranu McCarthy s kolegy zmiňují empirické výzkumy jako ten The National Cancer Institute, který prokázal, že uživatelé

dosahovali lepších výsledků s menu zarovnaným vpravo. Stejně jako Galitz (Galitz, 2002) poukazují na blízkost scrollbaru.⁵ Podstatou této studie je srovnání výsledků při použití konvenčního umístění, a naproti tomu použití vhodnějšího umístění za cenu prolomení uživatelského očekávání.

Graf 1: Efekt očekávání při první návštěvě stránky



Zdroj: (McCarthy et al., 2003), překlad autor

Při testování na šesti různých prototypech – menu vlevo, vpravo a nahoře při vysoké a nízké komplexitě stránky – vyšlo z výsledků doby trvání i z výsledků oční kamery, že očekávání uživatele naučené na jiných stránkách hraje roli pouze při první návštěvě, při každé další už byla rychlost hledání obdobná pro všechna tři umístění menu. (Graf 1)

⁵ Konkrétní závěry těchto výzkumů dnes pravděpodobně ztratily na relevanci z důvodu masivního rozšíření kolečka myši a obdobných technologií na touchpadech notebooků. Blízkost scrollbaru byla však jen jedním z důvodů, proč bylo dosahováno lepších výsledků se zarovnáním vpravo, dále je například zmiňován přirozený tok čtení zleva doprava, čímž se uživatel přirozeně dostane k navigaci po přečtení obsahu. Studie McCarthyho, Sassové a Riegelsberga tím však na aktuálnosti neztrácí, neboť je zaměřena na prolomení uživatelských návyků.

Autoři doporučují nebát se uplatnit doporučení na základě empirických výzkumů i za cenu porušení konzistence s ostatními webovými stránkami. Zdůrazňují však nutnost zachovat konzistenci v rámci celých stránek.

2.2.5 The Impact of Visual Layout Factors on Performance in Web

Pages: A Cross-Language Study

Avi Parush, Yonit Shwarts, Avy Shtub a M. Jeya Chandra (Parush et al., 2005) se zaměřili na problematiku, která je pro zde řešené téma velmi relevantní: uplatňování klasických konvencí ve specifickém prostředí webu a rozdíly mezi webovou stránkou a běžným desktopovým uživatelským rozhraním.

Cílem jejich výzkumu bylo zjistit, jestli vizuální faktory, které mají prokázaný vliv na použitelnost desktopových uživatelských rozhraní, mají stejný dopad na webové stránky. Konkrétně podrobili testování faktory, které jsou důkladně prozkoumány v desktopovém GUI: množství informací, hustota, zarovnání, rozmístění a seskupování. Pro srovnání použili stejný přístup, jako při dřívějším testování těchto faktorů na desktopovém grafickém uživatelském rozhraní.

Dva oddělené experimenty probíhaly se stránkami v angličtině a v hebrejštině, čímž byl kontrolován dopad opačného směru čtení na vnímání rozložení prvků. Na obou potom testovali následující faktory:

- Indikace seskupení – přidání rámečku s nadpisem skupině odkazů
- Zarovnání – vzájemné vertikální zarovnání skupin odkazů
- Hustota – jednotné řádkování proti variabilnímu
- Množství odkazů – nízké množství odkazů (20 - 25) nebo vysoké množství odkazů (45 - 50)

Každý z těchto faktorů měl při testech desktopových GUI výrazný dopad na výsledky. Na webové stránce se ale ukázaly odlišné relativní váhy jednotlivých faktorů, přičemž jejich účinky byly variabilnější a více závislé na kontextu. Z výsledků i z dalších zdrojů vyplývá, že web je méně strukturovaný a o hodně dynamičtější, než tradiční grafické uživatelské rozhraní.

Tento výzkum byl jedním z prvních, který prokázal, že pravidla z klasických GUI guidelines v prostředí webu nemusí fungovat, protože jeho charakter je odlišný.

2.3 Další teoretická východiska

Tato podkapitola se zabývá teoretickou stránkou oblastí podstatných pro problematiku této práce, které si zaslouží podrobnější rozebrání.

2.3.1 Ovládací prvky

Ovládací prvky jsou ty prvky uživatelského rozhraní, které slouží k volání příkazů. Jenifer Tidwellová dělí ovládací prvky na zobrazené a skryté. Uživateli zobrazené jsou ty, které mají grafickou podobu na obrazovce aplikace: tlačítka, různá menu, nástrojové lišty, odkazy a podobně. Uživatel má o těchto prvcích vizuální přehled, navíc designér GUI má možnost jejich vzhledem, umístěním a celkovým vsazením do uživatelského rozhraní přiřadit prvku odpovídající důležitost. Skryté jsou naopak ty ovládací prvky, které nejsou vykreslené na obrazovce a uživatel musí vědět nebo hádat, že tam jsou, pokud nenapišeme na obrazovku instrukce k jejich použití. Jsou to některé akce myši, jako double-click nebo *drag and drop*, a klávesnice, tedy klávesové zkratky nebo psané příkazy. (Tidwell, 2005)

Problémem skrytých ovládacích prvků je jejich obtížnější objevování a učení se. Přesto však hrají v grafickém uživatelském rozhraní výraznou úlohu, protože často zrychlují práci se systémem, a to zejména pro pokročilejší uživatele. Mnoho jich je také natolik pevně zakotvených v uživatelských návycích, že si nelze dovolit je neimplementovat. Takovým případem jsou například zkratky Ctrl+C a Ctrl+V pro kopírování a vložení.

Protože skryté ovládací prvky nelze jednoduše objevit a rozpoznat při pohledu na uživatelské rozhraní, je jejich použití výrazně závislejší na uživatelských návycích. Pro jejich obtížnější objevování i učení by jejich přítomnost a funkce měla maximálně důsledně odpovídat existujícím konvencím.

Na webu přitom právě v případě skrytých ovládacích prvků dochází k největším konfliktům konvencí. Standardně jsou totiž zpravidla obsazeny funkcemi prohlížeče a v prostředí webu nepatří k běžnému ovládání stránky. Pokud ale webová aplikace používá idiom aplikace známé z desktopového prostředí, je často součástí tohoto idiomu i sada skrytých ovládacích prvků a aplikace by je tedy měla nabízet. Rozeberme si podrobněji situaci konkrétních prvků.

Klávesové zkratky

Dnešní majoritní internetové prohlížeče používají základní standardní klávesové zkratky pro vlastní funkce, viz (Tabulka 2)

Tabulka 2: Tabulka Standardní klávesové zkratky

Klávesová zkratka	Funkce
Ctrl+A	Označit vše
Ctrl+C	Kopírovat
Ctrl+N	Nový
Ctrl+O	Otevřít
Ctrl+P	Tisk
Ctrl+S	Uložit
Ctrl+V	Vložit
Ctrl+X	Vyjmout
Ctrl+Z	Zpět

Zdroj: (Galitz, 2002), úprava autor

U některých těchto zkratk to neznamená pro webovou aplikaci žádný problém, protože dělají i bez jakéhokoliv ošetření v rámci webové aplikace to, co uživatel očekává. To jsou zejména příkazy pro práci s

textem, tedy *označit vše*, *kopírovat*, *vyjmout*, nebo *vložit*, a dále *zpět* (*undo*). Pokud je v rámci aplikace potřeba tyto klávesové zkratky použít i jiným způsobem, než umí sám prohlížeč, jako by byla například manipulace s objekty, lze je odchytit na straně aplikace a vykonat vlastní operaci, aniž by to přitom narušilo očekávání o chování prohlížeče.

Některé standardní klávesové zkratky však mají v prohlížeči funkci vyloženě spjatou s prohlížečem, nikoliv s obsahem stránky, to jsou zejména příkazy *nový*, *otevřít* a *uložit*. Tyto zkratky má uživatel na jednu stranu velmi dobře naučené z prakticky všech desktopových GUI, na druhou stranu se často už naučil, že v prostředí prohlížeče neobslouží webovou aplikaci, nýbrž vyvolají – zpravidla nežádaně – akci prohlížeče samotného. Zde dochází k velmi problematické kolizi konvencí, protože nelze určit, zda uživatel stiskl klávesovou zkratku pro vyvolání akce prohlížeče, v souladu s chováním prohlížeče na drtivé většině webových stránek, nebo pro vyvolání akce webové aplikace, která je v souladu s konvencemi tohoto typu aplikace, bez ohledu na její přítomnost na webu. Designér rozhraní webové aplikace je postaven před volbu, jestli vyjít vstříc uživatelům očekávajícím přítomnost těchto ovládacích prvků v aplikaci, a to i za cenu narušení konzistentního chování webového prohlížeče.

Určitým kompromisem je řešení, které například volí společnost Google v některých svých webových aplikacích.⁶ Pokud je standardní klávesová zkratka Ctrl+S stisknuta během editace některého textového pole, vyvolá akci *uložení* na straně webové aplikace. Pokud je uživatel právě mimo textové pole, webová aplikace na ni nereaguje a přenechá ji prohlížeči, který vykoná svoji standardní akci, tedy vyvolá dialog uložení aktuálně načtené stránky.

⁶ Testováno v aplikacích Google Docs a Gmail

Některé webové aplikace nabízí jednoklávesové příkazy, které mají v tomto směru zásadní výhodu, že nekolidují se standardním ovládáním prohlížeče. Ovšem takové ovládání se neopírá o běžné konvence, uživatel tedy neuplatňuje svoji znalost. Částečnou výjimkou jsou klávesové příkazy, které mají původ v ovládání počítačového terminálu Lear Singer ADM-3A a byly přeneseny do textového a později i grafického uživatelského rozhraní populárního textového editoru *vi* pro operační systém Unix, respektive další *nixové systémy. Dnes některé webové stránky a aplikace například používají pro posun mezi položkami klávesy *h*, *j*, *k*, a *l*, nejčastěji *j* a *k* pro pohyb dolů a nahoru. Webová aplikace Gmail mimo to používá příkazy ze stisků dvou kláves po sobě, jako *g* a následně *i* pro příkaz *go to inbox* (přejít do složky přijaté pošty). Omezením jednoklávesových příkazů však je nemožnost jejich použití v kombinaci se zadáváním textu.

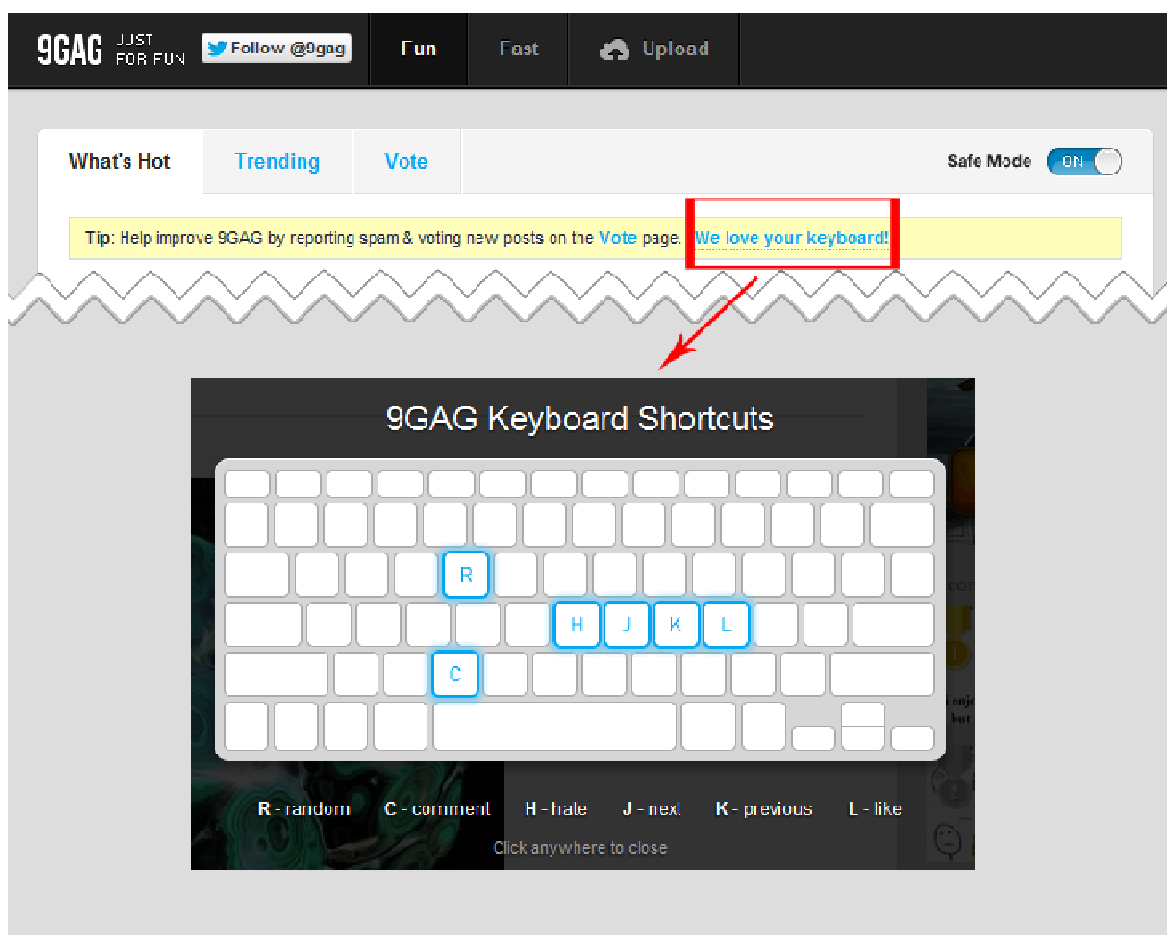
Ovládání formou psaných příkazů je v současné době na webu zcela okrajovou záležitostí, nicméně s vývojem rozpoznávání přirozeného jazyka nelze vyloučit jeho budoucí rozšíření. Z existujících projektů lze jmenovat *Ubiquity* (Mozilla Labs, v současnosti je vyvíjena pouze komunitní verze) nebo *Wolfram Alpha* od Wolfram Research.

Operace myši

Další kategorií skrytých ovládacích prvků jsou operace myši. Pokud odpovídá uživatelské rozhraní konvencím, je jasně rozpoznatelné, na které ovládací prvky lze kliknout. Méně jasná je možnost použití *drag and drop*. Skrytými prvky jsou i dvojklik a v prostředí webu za něj můžeme považovat i pravé tlačítko myši. Ani jeden z těchto ovládacích prvků není na webu standardní. Pravé tlačítko myši opět kolide s ovládáním prohlížeče, jako bylo popsáno u standardních klávesových zkratk, a v prostředí počítačů Apple nepatří k běžným ovládacím prvkům vůbec. S přibývajícím počtem uživatelů zobrazujících web na zařízení s dotykovým ovládáním potenciál jeho využití ještě dále klesá.

Pokud jsou skryté ovládací prvky použity v prostředí, kde nejsou očekávány automaticky, tedy nejsou zcela standardním ovládacím prvkem v tomto typu prostředí, vždy platí, že by měly být výrazně viditelně popsány na obrazovce, jako v příkladě viz (Obrázek 1)

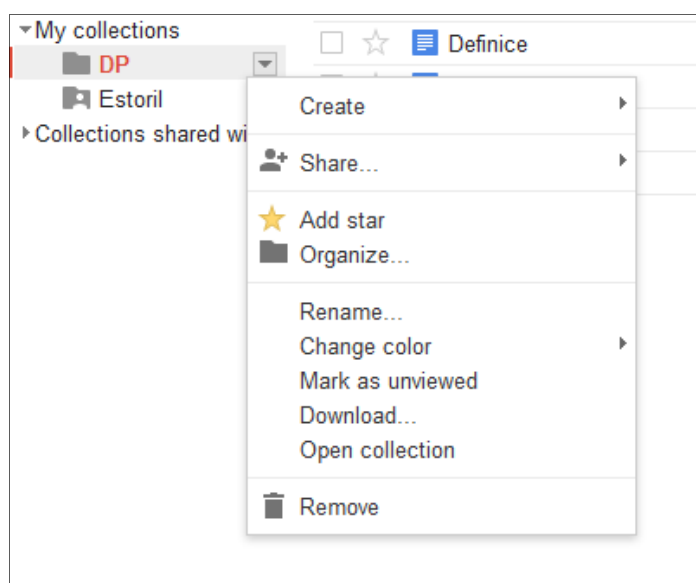
Obrázek 1: Upozornění na klávesové ovládání na webu 9gag.com



Zdroj: 9gag.com, úprava autor

Rozhodně by však skryté ovládací prvky měly mít dobře přístupnou alternativu v podobě viditelného ovládacího prvku. Příkladem je kontextové menu webové aplikace Google Docs (Obrázek 2). Kontextové menu adresáře jde vyvolat kliknutím pravým tlačítkem myši, nebo alternativně tlačítkem v pravé části pole s tímto adresářem.

Obrázek 2: Viditelné tlačítko kontextového menu v Google Docs



Zdroj: Google, úprava autor

Toto použití je příkladem, jak vhodně použít skrytý ovládací prvek. Slouží k urychlení a usnadnění práce spíše pokročilejších uživatelů, jejich neznalost však neomezuje přístup k odpovídajícím akcím.

2.3.2 Testování použitelnosti

Testování použitelnosti, také alternativně zvané uživatelské testování, je technika používaná k hodnocení produktu testováním s reprezentativními uživateli. Při testování se uživatelé snaží splnit typické úkoly a jejich počínání je zaznamenáváno hodnotiteli. Cílem je rozpoznání problémů s použitelností, sběr kvantitativních dat a zjištění míry spokojenosti účastníků s tímto produktem.

Nielsen (Nielsen, 2003) definuje tři základní komponenty testování použitelnosti:

- Reprezentativní uživatelé
- Reprezentativní úkoly
- Sledování činnosti, úspěchů a obtíží uživatelů

Testování použitelnosti se dělí podle cílů nebo podle způsobu provedení výzkumu. Nejběžnějším rozdělením podle cílů je na formativní a sumativní.

Formativní testování je kvalitativní, typicky s malým vzorkem účastníků (Graf 2)⁷ a často opakované v různých stádiích vývoje produktu. Jeho cílem je především odhalení chyb.

Graf 2: Procento odhalených chyb v závislosti na počtu účastníků



Zdroj: (Nielsen, 2000), překlad autor

Sumativní testování je kvantitativní, vyžaduje velký vzorek účastníků, protože cílem je získání statisticky signifikantních metrik. Zjišťovány jsou míry úspěšnosti při plnění úkolů, časy trvání úkolů, poměr úspěšně dokončených a předčasně opuštěných testů, množství uživatelských chyb a další hodnoty.

⁷ Ač je toto číslo často diskutované, Nielsen uvádí, že nejlepšího poměru nákladů a dosažených výsledků lze při iterativním formativním testování dosáhnout s počtem pěti účastníků (Nielsen, 2000)

Nielsen přidává ještě testování srovnávací, které slouží pro poměření s konkurenčními produkty a uplatnění poznatků o dobrých i špatných prvcích nejen z vlastního návrhu, ale i z konkurenčních.

Další dělení je možné podle způsobu provedení výzkumu. Jednoduše to lze vyjádřit podle toho, kde výzkum probíhá. Klasickým přístupem je laboratorní testování. Pro časté a podrobnější testování je vhodné zařídit laboratoř pro testování použitelnosti. Taková laboratoř, zvaná *usability lab*, může disponovat podle potřeb a finančních možností různým stupněm výbavy. Typický je obrazový a zvukový záznam chování účastníků, časté jsou i oční kamery (zařízení pro *eye tracking*) nebo zařízení pro testování mobilních zařízení.

Výzkum se také může odehrávat přímo na místě, kde má být produkt či systém nasazen, typicky na pracovišti. To pomáhá pochopit kontext práce s produktem v „opravdovém“ prostředí nebo ukázat rozdíly proti výsledkům testů v laboratoři. Provádět testování ve skutečném kontextu prostředí je zvláště důležité, pokud je práce s produktem nebo systémem výrazně závislá na vnějších faktorech, na offline datových zdrojích a podobně.

Poslední kategorií je vzdálené testování použitelnosti. Vzdálené testování probíhá bez fyzické přítomnosti uživatelů, moderátora a hodnotitelů na jednom místě. Spojení probíhá nejčastěji prostřednictvím internetu a uživatel se ho typicky účastní ze svého počítače. Vzdálené testování použitelnosti se dále dělí na moderované a nemoderované.

Moderované vzdálené testování použitelnosti znamená takový výzkum, při kterém jsou účastník s moderátorem a případně dalšími pozorovateli v synchronním kontaktu v průběhu testování. Proto je tento způsob zván také *synchronní*. Uživatel provádí testování na svém počítači a testovací

software umožňuje přímé sledování jeho činnosti. Běžné je pozorování uživatele prostřednictvím webkamery. Výhodou je nezávislost na lokaci uživatelů i členů týmu, což je podstatné zejména pro nadnárodní produkty a pomůže to také získat k testování uživatele, kteří by nebyli ochotní obětovat čas cestou do testovací laboratoře. Vyšší je i časová flexibilita. Nevýhodami jsou zprostředkovaný kontakt s účastníkem a složité technické zajištění. (Barnum, 2010)

Nemoderované, nebo také asynchronní vzdálené testování použitelnosti, je označované za „*pravděpodobně nejefektivnější způsob sběru detailních a škálovatelných souborů dat o použitelnosti*“. (Albert et al., 2010, s. 6) Asynchronní znamená, že účastník a hodnotitel nejsou v průběhu testování v kontaktu. Účastník provede testování kdykoliv se mu hodí, jeho data jsou zaznamenána a hodnotitel je může analyzovat později, typicky po získání dat všech účastníků. Stejně jako u moderovaného vzdáleného testování je výhodou nezávislost na lokaci, navíc však přibývá absolutní nezávislost na čase, kdy účastníci testování absolvují, tudíž je snazší přesvědčit vyšší počet uživatelů k účasti. Stejně tak není omezen počet účastníků, kteří absolvují testování v jeden okamžik. Všechny tyto faktory pomáhají snadno získat rozsáhlý soubor výsledků. Věrohodnosti získaných dat pomáhá, že uživatelé pracují ve svém přirozeném prostředí, a to jak fyzickém, tak softwarovém, a jejich stupeň soustředění oproti synchronním metodám testování může lépe odpovídat běžné práci na počítači.

Hlavní výhodou nemoderovaného vzdáleného testování použitelnosti je možnost provést rozsáhlá kvantitativní šetření bez nadměrných nároků na čas a finanční náročnost. Při rozsáhlém souboru účastníků lze získat statisticky signifikantní závěry i o malých rozdílech v hodnotách metrik použitelnosti. Proto se tento typ testování hodí pro srovnávací výzkumy více prototypů, srovnávací testy proti konkurenčním produktům nebo proti stanoveným kritériím. (Barnum, 2010)

Z asynchronního charakteru výzkumu plynou odpovídající nevýhody:

- Není možné vidět účastníky a jejich fyzické reakce.
- Není možné mluvit s účastníky. Ti sice zaznamenávají subjektivní hodnocení v průběhu i po ukončení testovací sekvence, ale na jejich hodnocení není možné dále reagovat a získat tak hlubší porozumění.
- Není možné zjistit důvod konkrétních uživatelských kroků nebo předčasného ukončení.
- Testovací scénáře i otázky pro subjektivní hodnocení musí být jednoduché.

(Barnum, 2010, s. 45)

Sledování pohybu myši

Sledování pohybu myši (mouse tracking) je často prezentováno jako levnější a snadněji nasaditelná alternativa k očním kamerám (eye tracking). Nicméně hrubé výsledky nepřinášejí potřebnou přesnost, aby byly srovnatelnou náhradou. Jednoduše řečeno, myš se často pohybuje značně daleko od centra uživatelského zájmu (Guo, Agichtein, 2010). Přesto je sledování pohybu myši pro různé účely běžně používáno.

Qi Guo a Eugene Agichtein v roce 2010 jako první představili metodu, jak čistě na základě pozice a pohybu myši predikovat, ve kterých okamžicích je pozice kurzoru myši úzce koordinovaná s pohledem uživatele. (Guo, Agichtein, 2010) Tato nepřiliš složitá metoda pomohla s téměř 60% přesností správně určit, kdy byla vzdálenost kurzoru myši a místa pohledu uživatele nižší, než 100 pixelů. Vzhledem k tomu, že v průběhu jejich výzkumu byla průměrná vzdálenost kurzoru myši od místa pohledu uživatele 225 pixelů, jde o výrazné zpřesnění výsledků pouhým algoritmem, bez použití dodatečných vstupů. Vývoj je v této oblasti je teprve v raném stádiu, ale ze současného stavu lze

předpokládat, že sledování pohybu myši čeká v následujících letech široké rozšíření, zejména na webu a zejména pro automatický sběr kvantitativních dat.

Prototyp

„Prototyp je primárně prostředek ke zkoumání, komunikaci a hodnocení. Jeho smyslem je získání vstupu od uživatelů a poskytnutí zpětné vazby designérům.“ (Galitz, 2002, s. 706) Prototyp je simulací nějakého systému, přičemž nejsou kladeny nároky na úplnost a přesnost, respektive jejich úroveň může být různá od prototypování na papíře po věrné interaktivní modely počítačových systémů. Slouží pro porozumění, jak má systém fungovat a vypadat.

S prototypem lze provádět testy na zjišťování široké škály údajů:

- Soulad s požadavky
- Soulad s návrhovými *guidelines* a *best-practices*
- Rozpoznání problémů v návrhu
- Snadnost učení
- Udržení si znalosti
- Rychlost plnění úkolů
- Rychlost uspokojení potřeb
- Míry chybovosti
- Subjektivní uspokojení

(Galitz, 2002, s. 709)

Na prototypu se tedy dají zjišťovat hodnoty všech metrik jako na skutečné aplikaci.

3 Příprava a průběh výzkumu

3.1 Princip výzkumu

Výzkum je založen na technice uživatelského testování tří prototypů. Všechny prototypy mají zcela shodnou funkcionalitu, liší se jen uživatelským rozhraním. Aplikace je smyšlená, postavená za účelem otestování různých vybraných situací. Připomíná jednoduchý textový editor a má deset funkcí, což je přesně nad hranicí takzvaného Millerova zákona, stanovujícího rozsah lidské krátkodobé paměti na 7 ± 2 prvků. Toto číslo sice rozhodně není přesné (Jones, 2002), nicméně pro testování je podstatné, že deset prvků by již mělo být pro uživatele velmi obtížné si zapamatovat.

Během testování je zaznamenávána veškerá interakce uživatele s aplikací. Sledovány jsou zejména:

- časy trvání úspěšného splnění úkolů
- způsob řešení zadaných úkolů
- chyby
 - o závažné, které naruší práci a uživatel musí opakovat několik kroků
 - o malé, které nemají vliv na další průběh práce.
- pohyb myši

Sledovány jsou i další hodnoty, které slouží pro kontrolu získaných výsledků, například jestli někdo neabsolvoval testování několikrát opakovaně nebo se nesnažil narušit výsledky testování.

3.1.1 Průběh testování

Testování probíhá metodou nemoderovaného vzdáleného testování použitelnosti. Pro účastníka to znamená, že probíhá online, přímo na

jeho počítači, bez nároků na jakýkoliv přidaný sledovací hardware i software. Testování také není omezeno konkrétním časem, lze ho provést v zadaném časovém intervalu zcela kdykoliv. V souladu s požadavky na nemoderované vzdálené testování použitelnosti jsou úkoly jednoduché, není jich mnoho a účastníkovi nejsou kladeny žádné složité otázky. Účastník je nejprve na vstupní stránce informován o podstatě a průběhu výzkumu. Po odsouhlasení sběru dat z testování je vybrán a spuštěn jeden ze tří prototypů. Účastníkovi se vždy zobrazuje instrukce k aktuálnímu úkolu. Úkolů je sedm a zaměřují se na různé zkoumané situace.

Otevřít nový dokument:

Akce otevření nového dokumentu je uživateli dobře známá. Tím, že je to první úkol, je zde především měřena rychlost, s jakou se uživatel zorientuje v prostředí aplikace.

Nechat vypsat otázky:

Tato akce nepatří mezi konvenční funkce žádného prostředí, uživatel ji musí najít, aniž by mu pomáhala znalost typu prostředí.

Vyplnit odpovědi a spustit jejich kontrolu:

Zde jde o dva cíle, prvním je schopnost uplatnění právě nabyté znalosti z předchozího úkolu. Funkce *kontrola odpovědí* se ve všech prototypích vyskytuje ve stejné kategorii a v bezprostřední blízkosti předchozí akce, tedy *vypsání otázek*. Jde tedy opět o akci, kterou uživatel nezná z jiných prostředí, ale už má šanci využít znalost tohoto prostředí.

Druhým cílem je získání základních údajů, které budou využity pro rozšíření výsledků o další porovnatelná kritéria. Zjišťované údaje jsou pohlaví, věk a míra počítačové gramotnosti. Vzhledem ke skutečnosti, že respondenti byli osloveni online, je zaručena alespoň základní počítačová gramotnost. Pro účely tohoto výzkumu jsou respondenti

dělení na běžné uživatele a IT profesionály či studenty informatických oborů. Pro složitost stanovování exaktní hranice je otázka položena velmi obecně: „Jste z oboru IT?“ Pro účely výzkumu je toto rozdělení dostačující.

Najít a nahradit:

Funkce *najít a nahradit* je uživatelům známá z desktopového prostředí. V aplikačním menu má dvě varianty konvenční pozice, buď v nabídce *úpravy*, nebo případně v samostatné nabídce *hledat*, pokud v aplikaci taková je. Pro tuto akci neexistuje standardní, snadno identifikovatelná ikona. Je sledováno použití standardní klávesové zkratky Ctrl+H, předpokladem je, že nepatří mezi běžně známé.

Protože úkolem je přepsat dva výskyty jednoho písmene, má uživatel možnost tuto operaci provést ručně, a to srovnatelně obtížně. Tento úkol tedy slouží především k zjištění, kolik uživatelů využije funkci aplikace. Zajímavé je především srovnání, jestli *look and feel* desktopové aplikace vede uživatele více k využití funkce naučené z tohoto prostředí právě v tomto prototypu.

Vrátit poslední krok:

Tento úkol má tři možná řešení. Prvním je opět manuálně přepsat písmena, druhým je použití funkce *najít a nahradit* a třetím je funkce *zpět* (ve smyslu *undo*). Funkce *zpět* je z desktopového prostředí velmi dobře známá, má ustálenou ikonu i dobře známou klávesovou zkratku Ctrl+Z. Ve webovém prostředí naopak není obvyklá vůbec. Opět je sledován především rozdíl v poměru využití této funkce v odlišných prototypch.

Další možnou – avšak chybnou – reakcí je stisknutí tlačítka *zpět* v prohlížeči. I tato možnost je sledována.

Uložit dokument:

Uložení je opět velmi dobře známou funkcí s dobře identifikovatelnou ikonou a v desktopovém prostředí zřejmě jednou z nejlépe naučených klávesových zkratk, Ctrl+S. Zde akce slouží hlavně právě k zjištění, kolik uživatelů použije pro uložení klávesovou zkratku.

Ukončit aplikaci:

Posledním pokynem je ukončit aplikaci. Sledován je způsob ukončení, zda uživatel zavře stránku v prohlížeči, či vybere příkaz pro ukončení na straně aplikace. V případě prototypu s desktopovým rozhraním jsou potom možnosti zavření ještě rozděleny: v menu dokument → ukončit, nebo křížkem v pravém horním rohu aplikačního okna.

Tabulka 3: Úkoly při testování

Úkol	Konvenční funkce ve webovém prostředí	Konvenční funkce v desktopovém prostředí	Standardní ikona	Existuje standardní klávesová zkratka	Co je úkolem zjišťováno
Nový dokument	ne	ano	ano	ano, ale v prohlížeči Chrome není přípustná	zorientování se v prostředí
Vypsat otázky	ne	ne	ne	ne	nalezení neznámé funkce
Vyplnit a zkontrolovat odpovědi	ne	ne	ne	ne	učení – blízká funkce právě použité
Najít a nahradit	ne	spíše ano	ne	ano	způsob řešení
Zpět	ne	ano	ano	ano	způsob řešení / nalezení dobře známé funkce
Uložit	ne	ano	ano	ano	použití klávesové zkratky
Ukončit	ano	ano	ano	ano, ale v prohlížeči nelze použít	způsob řešení funkcí aplikace X prohlížeče

Zdroj: autor

Aplikace dále obsahuje funkce, které nejsou pro plnění zadaných úkolů potřeba, ale jsou zde, protože dle konvencí do některého z prostředí patří a aby zvýšily počet ovládacích prvků nad požadovanou hranici zapamatovatelnosti. Jde o funkce *najít*, *tisk* a *o aplikaci*.

Po splnění všech zadaných úkolů je uživatel přesměrován na stránku s poděkováním za účast ve výzkumu a s formulářem pro zanechání zpětné vazby. Tímto způsobem je v souladu s běžnými postupy nemoderovaného vzdáleného testování použitelnosti získáváno subjektivní hodnocení.

Každý účastník testuje pouze jeden ze tří prototypů, protože při opakovaném provádění stejných úkolů by už znal zadání a postupy řešení, čímž by bylo narušeno měření účinnosti a přesnosti.

3.1.2 První prototyp – „hypertextový“

První prototyp vychází z konvencí běžné webové prezentace, pro uživatele je známé rozmístění prvků i přesné možnosti ovládání. Menu však místo tradiční navigace nabízí funkce pro práci s dokumentem, které nejsou na statické webové prezentaci standardní.

Rozložení stránky je konvenční dle (Nielsen, Tahir, 2005): navigační menu je vlevo a skládá se ze standardně formátovaných odkazů, v modré barvě a podtržených, dělených do logických kategorií. Odkaz *o této stránce* má běžně na webu místo mimo hlavní navigaci, proto je oddělen i zde. Společně s ním, zcela vpravo nahoře, je tlačítko *ukončit*, což odpovídá konvenčnímu umístění odkazu pro odhlášení.

Obrázek 3: První prototyp

Instrukce:

Otevřete nový dokument.
Nechte vypsat otázky.
Vyplňte odpovědi a spusťte jejich kontrolu.

[O této stránce](#) | [Ukončit](#)

Prototypové testování

Dokument:
[Nový](#)
[Uložit](#)
[Tisk](#)

Úpravy:
[Zpět](#)
[Najít](#)
[Najít a nahradit](#)

Dotazník:
[Vypsat otázky](#)
[Zkontrolovat](#)

Pohlaví:
Věk:
Jste z oboru IT?

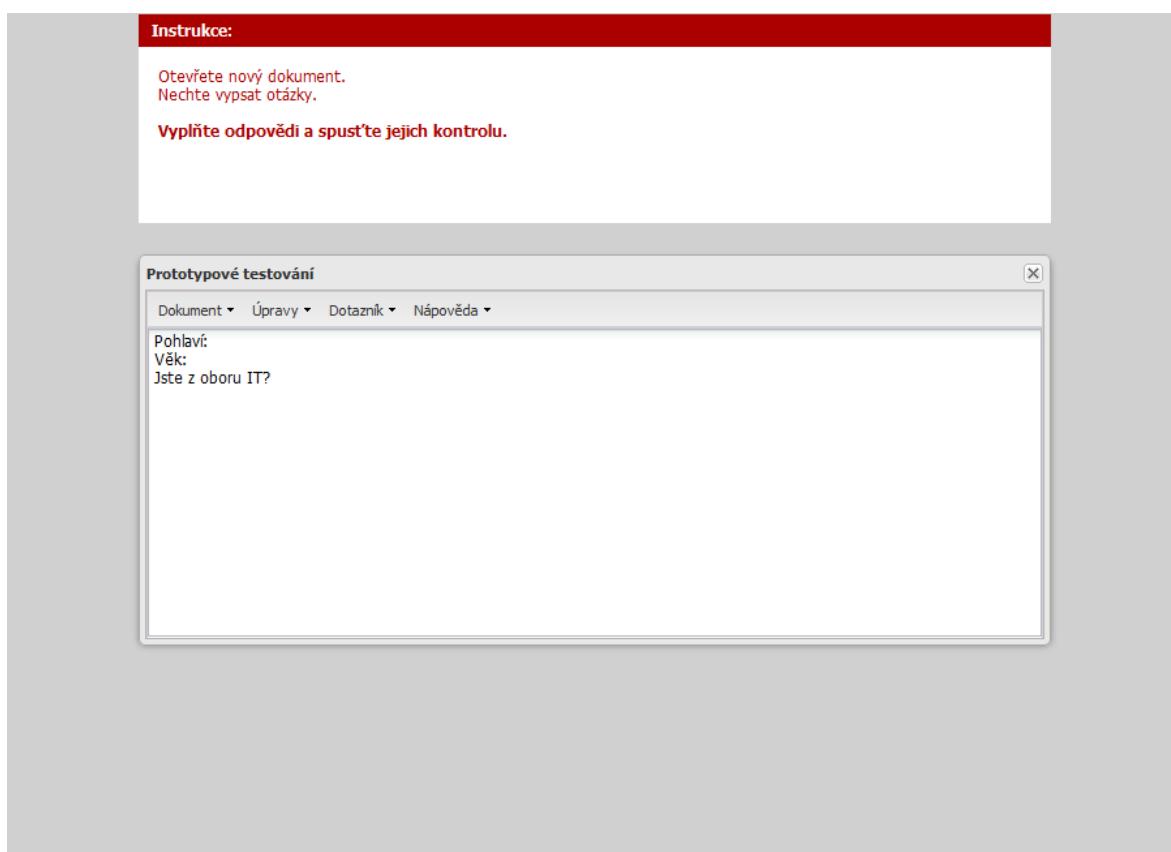
Zdroj: autor

První prototyp tedy měří výsledky uživatele v prostředí, které co nejvíce zůstává konvenčně webové.

3.1.3 Druhý prototyp – „desktopový“

Druhý prototyp je představitelem idiomu *textový editor*, s takovým typem uživatelského rozhraní se už uživatel setkal v desktopovém prostředí a proto chápe jeho smysl, rozmístění prvků a základní principy fungování, i když je přenesený na web. Podle Nielsena (Nielsen, 1999) bude uživatel při první návštěvě tápat, podle Tidwellové (Tidwell, 2005) mu naopak pochopení idiomu známého z jiného prostředí nebude dělat problém. Proto je zkoumán rozdíl mezi výkonností plnění prvního úkolu, kdy dochází k poznávání uživatelského rozhraní a učení, a následujících úkolů.

Obrázek 4: Druhý prototyp



Zdroj: autor

Rozhraní využívá aplikačního menu, kde většina funkcí a jejich rozdělení do nabídek je uživateli známá, kromě nabídky *dotazník* s funkcemi pro vypsání otázek a kontrolu odpovědí.

Tento prototyp podporuje nad rámec funkcí prohlížeče ovládání standardními klávesovými zkratkami, konkrétně jsou popsány v tabulce (Tabulka 4). V souladu s konvencemi desktopových aplikací a s potřebou na přítomnost skrytých ovládacích prvků uživatele upozornit jsou tyto klávesové zkratky v jednotlivých nabídkách aplikačního menu zobrazeny vpravo vedle odpovídajících položek. Součástí výzkumu je zjišťování, kolik uživatelů použije ve webové aplikaci klávesové zkratky, které běžně vyvolávají akci prohlížeče.

Tabulka 4: Klávesové zkratky druhého prototypu

Klávesová zkratka	Funkce
Ctrl+S	Uložit
Ctrl+P	Tisk
Ctrl+Z	Zpět
Ctrl+F	Hledat
Ctrl+H	Najít a nahradit
(Ctrl+N)	Nový dokument – Tato klávesová zkratka však není přístupná ve všech majoritních prohlížečích, proto ani není v rozhraní zobrazena jako podporovaná.

Zdroj: autor

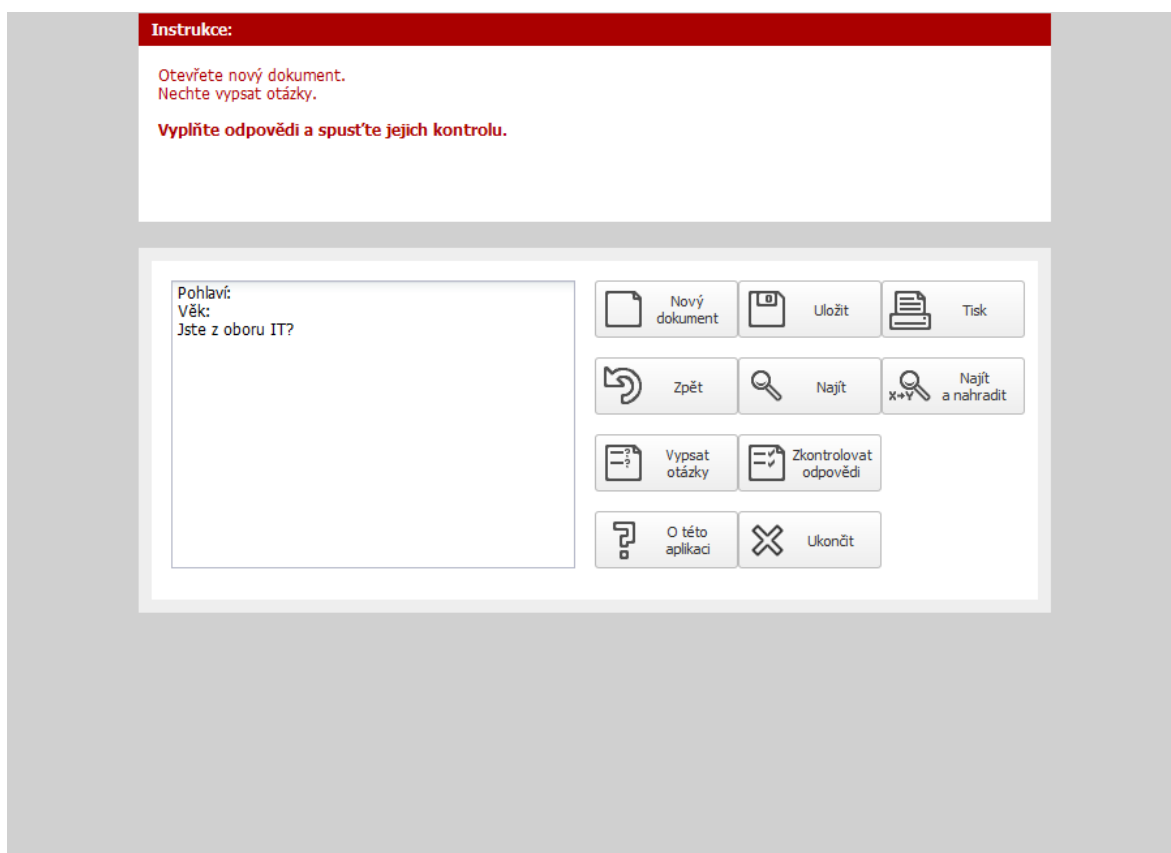
Druhý prototyp zkoumá adaptaci uživatele na idiom známý z desktopového prostředí. Uživatel tedy disponuje potřebnými návyky, ale musí je použít v novém kontextu.

3.1.4 Třetí prototyp – „dotykový“

Třetí prototyp neodpovídá běžným konvencím ani webové stránky, ani desktopové aplikace. Celkový vzhled aplikace přesněji nevychází z žádného idiomu, ale částečně připomíná uživatelské rozhraní optimalizované pro dotykové ovládání. Toto rozhraní by nemělo uživatele nijak zaskočit. Ačkoliv celkový vzhled neodkazuje na nic známého, jednotlivé prvky jsou známy dobře. Práce s tímto prostředím není založena na uživatelské znalosti nebo znovuoobjevování, nýbrž na

minimalizaci obtížnosti učení. Tím doplňuje výsledky z uživatelských rozhraní dodržujících platformní konvence. Pro uživatele by měla být o něco složitější orientace v nabídce příkazů, protože jejich dělení do skupin záměrně není tak výrazné, jako v předchozích dvou prototypech. Příkazy, patřící logicky k sobě, jsou vždy na jednom řádku a řádky jsou odděleny mezerou, ne však velkou. Nejsou použity rámečky, nadpisy ani barevné či jiné grafické rozdělení kategorií. Uživateli tedy není pomáháno s celkovou orientací v nabídce příkazů, naopak je ale usnadněno hledání příkazů jednotlivě. Jsou použity velké a výrazné ikony, z nichž většina je zcela ustálená, a akce, které v daný moment nejsou dostupné, jsou zašedlé (disabled). To jsou sice v druhém prototypu také, nicméně v aplikačním menu není tato indikace tak vizuálně výrazná.

Obrázek 5: Třetí prototyp



Zdroj: autor

Třetí prototyp měří výsledky v případě učení nového oproti využití známého v případě prvních dvou prototypů.

3.2 Technické zpracování

Z charakteru a cílů výzkumu vyplývají určité požadavky, které musí jeho technologické zpracování splňovat. Základní rozdíly v nárocích na testovací software ve webovém prostředí vyplývají z absence plné kontroly nad daným počítačem a z použití architektury klient-server. Tím vzniká potřeba klást velký důraz na konzistenci a robustnost aplikace.

Konzistence testovacího prostředí na odlišných platformách a na různých strojích je bezpodmínečnou nutností pro získání objektivních výsledků. Je nutné ošetřit případné rozdíly v interpretaci různými webovými prohlížeči, aby byly vlastnosti testovaných prototypů ve všech prostředích stejné. Aby bylo možné sbírat porovnatelná časová data, nesmí docházet k ovlivnění rozdíly v reakční době aplikace.

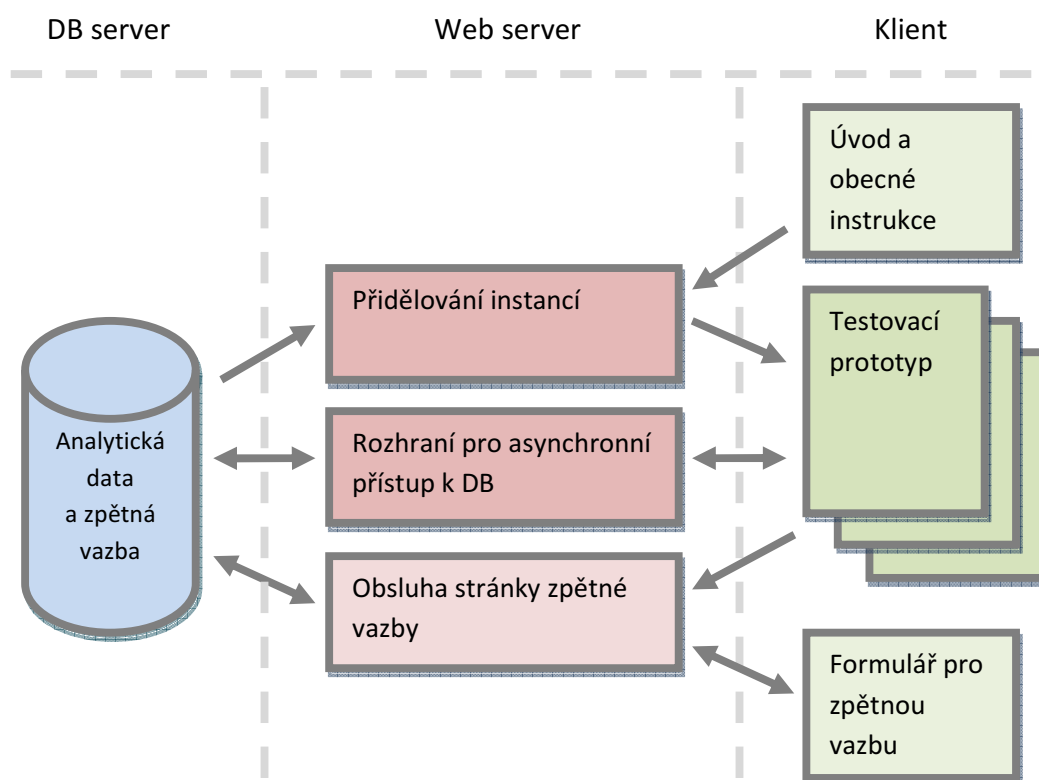
Dalším klíčovým požadavkem je robustnost, v aplikaci je potřeba minimalizovat možnost narušení testování jakýmkoliv technickým problémem, stejně jako zamezit ztrátě dat například při nenadálém ukončení aplikace.

Pro charakter a potřeby výzkumu nebyl vhodný žádný existující software pro prototypování. Popsané potřeby a omezení neumožnily ani vytvořit jednoduchou hypertextovou strukturu a sledovat ji běžnými nástroji pro webovou analytiku. Takové řešení by nenabízelo požadovaný rozsah funkcionality aplikace ani potřebnou detailnost měření, a dále by výsledky byly narušeny načítáním nových stránek. Bylo tedy nutné vyvinout řešení zcela nové.

3.2.1 Architektura:

Architektura je typická pro webovou aplikaci, třívrstvá: datová a aplikační vrstva jsou na serverové straně a prezentační vrstva na klientské straně. Oproti nejběžnějším řešením zde ovšem dochází k posunu obsahu vrstev směrem ke klientu. Za prvé, aplikace je pouze testovacím prostředím, v podstatě jí zcela chybí vlastní *business logika*. Za druhé, z důvodu popsané nutnosti vyhnout se dynamickému načítání, aby nedošlo k narušení měření času, je veškerá logika, odpovědná za chování celého prototypu, obsažena v prezentační vrstvě a běží na klientské straně.

Obrázek 6: Architektura testovací aplikace



Zdroj: autor

Aplikační vrstva zodpovídá za přidělení instance prototypu klientské straně, a to takovým způsobem, aby u každého ze tří prototypů proběhl stejný počet testování. Řešeno je tím způsobem, že jsou při přidělování instancí jednotlivé prototypy cyklicky střídány. Instance je poté

spuštěna na klientském stroji a běží co do funkcionality samostatně. Při každé větší uživatelské akci je poslán záznam aplikační vrstvě, která má v tuto chvíli pouze roli rozhraní pro předání dat datové vrstvě. Tyto přenosy jsou asynchronní, což znamená, že na rozdíl od běžného chování statických hypertextových stránek probíhají na pozadí, bez jakéhokoliv narušení chování či zobrazení prezentační vrstvy. Tím je zároveň zajištěno získání dat z testování i v případě nenadálého ukončení na klientské straně.

3.2.2 Výběr technologií:

Technicky nejnáročnějším ze tří prototypů je druhý, webové uživatelské rozhraní dle konvencí desktopové aplikace. Především jeho potřebám byl podřízen výběr technologie pro klientskou část aplikace. Požadavkem při výběru tak byla vybavenost nástroji pro tvorbu grafických uživatelských rozhraní odpovídajících desktopovým konvencím, a zároveň podpora asynchronního přenosu dat. Tyto podmínky velmi dobře plní prakticky jakýkoliv z větších javascriptových frameworků. Dalším kritériem proto, s přihlédnutím k minimální náročnosti vyvíjené aplikace, byla jednoduchost vývoje a nasazení. Jako objektivní měřítko byl brán běh frameworku na běžných hostingových technologiích PHP+MySQL, bez nutnosti překládat zdrojový kód. Zbytek tohoto hodnocení již je nutně subjektivní. Jako nejvhodnější byl takto vybrán framework Ext JS ve verzi 4.1. Jde o framework založený na HTML5, CSS a Javascriptu.

Pro serverovou část už byla hlavním kritériem jednoduchost nasazení, při absenci business logiky nevyvstala ani potřeba využití server-side frameworku. Zde byla zcela jasně zvolena běžná kombinace PHP a MySQL. Asynchronní komunikace mezi klientem a serverem probíhá ve formátu JSON.

3.3 Skupina účastníků testování

Účastníci byli osloveni prostřednictvím Internetu. Už z toho vyplývá, že disponují počítačovou gramotností přinejmenším na uživatelské úrovni. Pro orientační rozdělení skupiny testujících uživatelů dle počítačové gramotnosti na uživatele a profesionály sloužila otázka v dotazníku: „Jste z oboru IT?“

Osloveni byli převážně studenti a absolventi vysokých škol různého zaměření. To znamená, že jde oproti republikovému průměru o nadprůměrně vzdělanou skupinu.

Věk vzorku uživatelů odpovídá také věku studentů a čerstvých absolventů VŠ. Jde tedy o věkovou skupinu, u které lze jen na základě věku očekávat nadprůměrnou schopnost práce s počítačem.

4 Vyhodnocení výsledků výzkumu

4.1 Základní údaje o souboru

Celkem se testování účastnilo 123 uživatelů. V jednom případě došlo při interpretaci kódu na straně prohlížeče k chybě a bylo částečně narušeno chování aplikace, tento výsledek byl proto vyřazen. Relevantní výsledky byly získány od 122 účastníků. Po rozdělení na tři skupiny to znamená alespoň 40 účastníků na jeden prototyp. Úspěšně jich všechny úkoly dokončilo 109. Medián doby úspěšného splnění celého testování byl 76 sekund. Z účastníků bylo 47,86% žen, průměrný udaný věk byl 25,15 let a 25,64% uživatelů se přihlásilo k oboru informačních technologií. Poslední dvě hodnoty dokládají předpoklad, který byl vysloven při popisu skupiny účastníků: U této skupiny uživatelů je nutné počítat s nadprůměrnou počítačovou gramotností.

Vedle objektivních metrik bylo dále získáno 42 textových subjektivních hodnocení a zpětných vazeb.

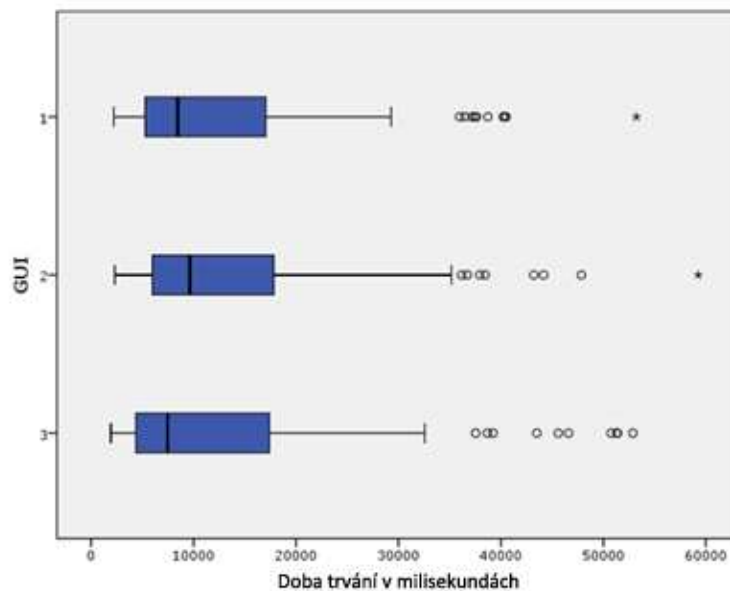
4.2 Míra účinnosti

První výkonnostní metrikou použitelnosti je účinnost, jejíž hodnotu lze vyjádřit jako inverzní funkci délky časů plnění jednotlivých úkolů.

Získané hodnoty časů plnění úkolů měly pochopitelně vysoký rozptyl. Uživatelům nebylo žádným způsobem sděleno, že je jeho čas zaznamenáván, nebo že by měl spěchat. V datech se tak objevují extrémně vysoké hodnoty. Tento způsob testování nevylučuje, že uživatelská pozornost v průběhu testování může být odvedena k jiné činnosti, čímž extrémně vysoké hodnoty časů zcela ztrácejí relevanci. Je potřeba s tímto charakterem dat počítat.

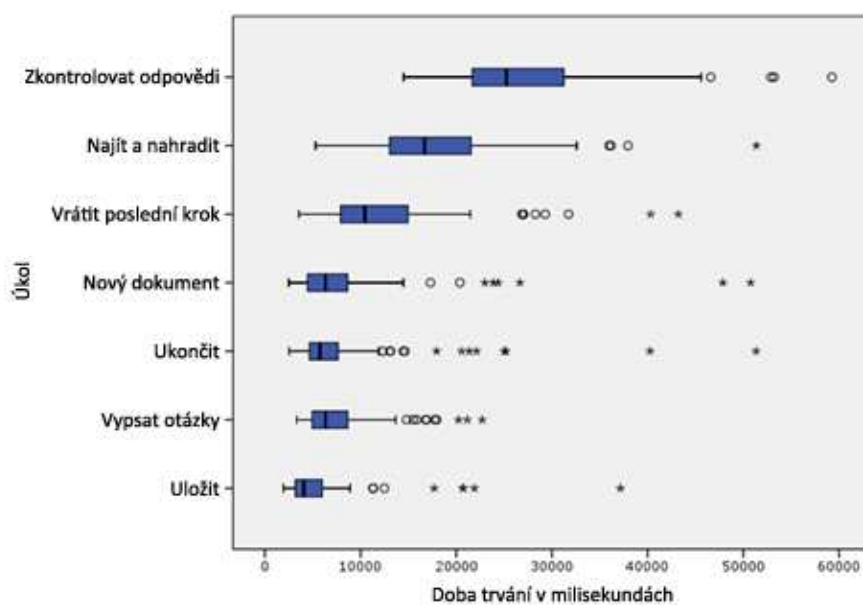
Souhrny všech změřených časů dělené podle typu GUI po odebrání tří nejvyšších hodnot jsou v grafu (Graf 3)

Graf 3: Souhrny všech časů podle typu GUI



Zdroj: autor

Graf 4: Rozdělení časů podle jednotlivých úkolů



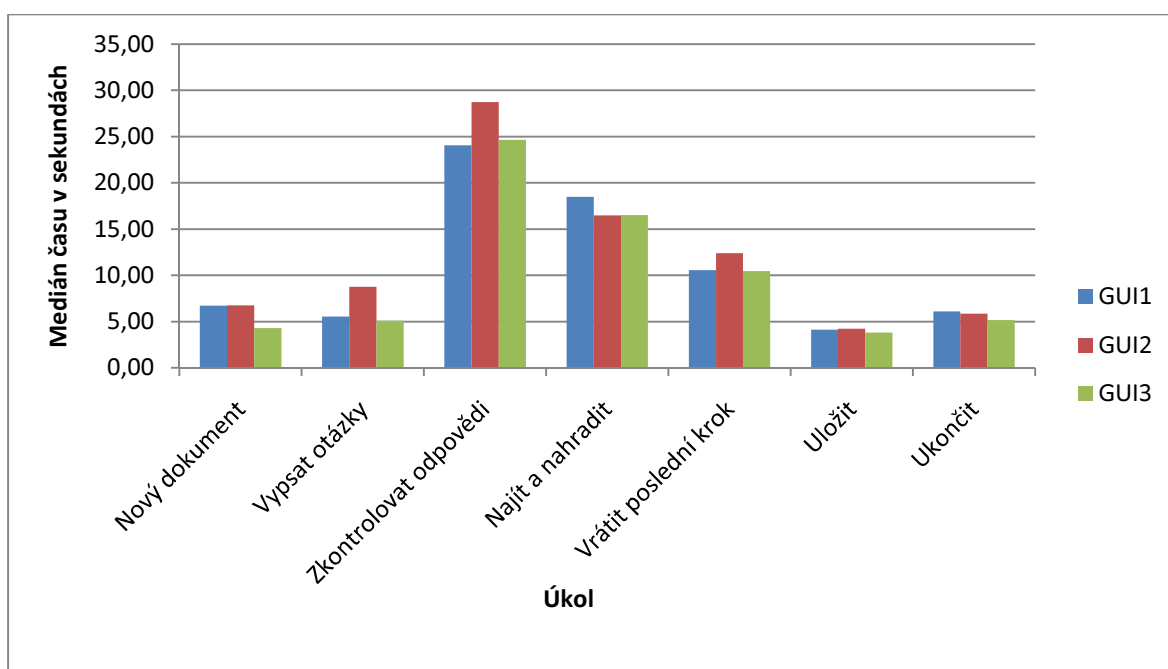
Zdroj: autor

Rozdělíme-li však doby trvání podle jednotlivých úkolů, ukážou se hodnoty časů jednotlivých uživatelů velmi konzistentní. Časy pro všechny tři prototypy pohromadě, rozdělené podle úkolů, jsou v grafu (Graf 4)

Cílem měření časů bylo porovnat délky trvání jednotlivých úkolů v závislosti na vybraném prototypu. Tím je porovnávána hodnota jedné ze tří hlavních výkonnostních metrik použitelnosti – Účinnost.

Dosažené časy jednotlivých úkolů se pro žádný z prototypů nelišil zcela výrazně, jak ukazuje graf mediánů časů (Graf 5)

Graf 5: Mediány časů řešení úkolů



Zdroj: autor

Tabulka 5: Statistické významnosti rozdílů účinnosti (vyšší je lepší)

Varianta rozhraní	Nový dokument	Vypsat otázky	Zkontrolovat odpovědi	Najít a nahradit	Vrátit poslední krok	Uložit	Ukončit
GUI1	-	+++	+	0	0	0	0
GUI2	-	---	-	0	0	0	0
GUI3	+	+++	+	0	0	0	0

Zdroj: autor

Pro testování významnosti rozdílů časů plnění jednotlivých úkolů na odlišných prototypích byl použit Kruskalův-Wallisův test, což je neparametrický statistický test vhodný pro tento případ, kdy se rozložení značně liší od normálního. Nulovou hypotézou je, že časy všech verzí prototypů mají stejné rozložení.

4.2.1 Nový dokument

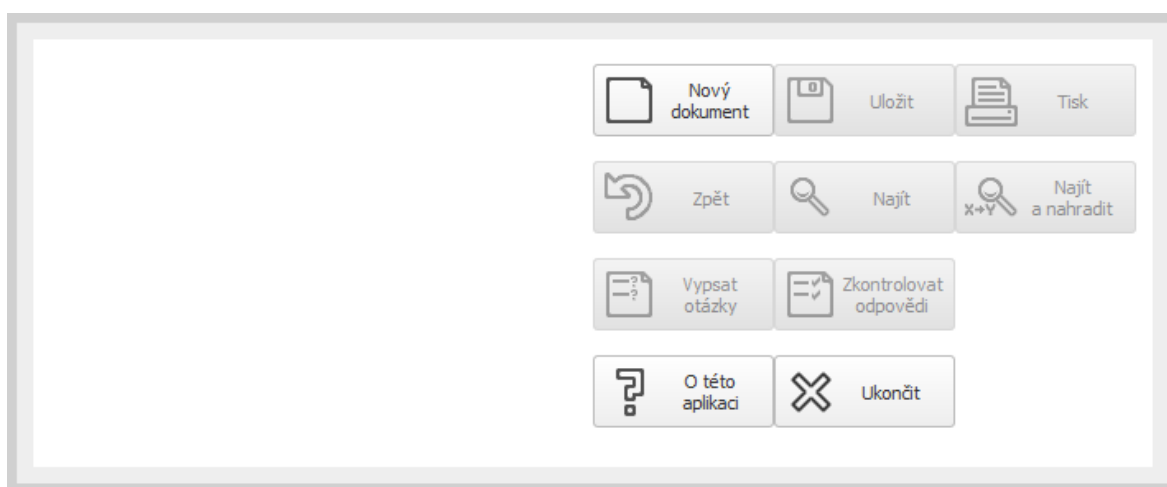
První úkol má v měření účinnosti zvláštní postavení, protože změřený čas zahrnuje dobu potřebnou pro zorientování se v uživatelském prostředí. Srovnání výsledků prvního úkolu a úkolů následujících dává představu o rozdílné výkonnosti při učení v novém prostředí oproti využití znalosti idiomu předloženého prostředí.

Někteří uživatelé strávili prvotním prozkoumáváním prostředí značné množství času. U hodnot z tohoto úkolu tím byl průměr posunut ze všech úkolů nejvýš nad medián.

Jak ukazuje graf mediánů (Graf 5), při tomto úkolu bylo dosaženo rychlejších časů na třetí variantě GUI. Kruksalův-Wallisův test prokázal tento rozdíl za významný na hladině významnosti 0,05. Pokud není uvedeno jinak, je ve všech následujících testech použita tato hladina významnosti.

Tento výsledek je překvapující, u třetího prototypu byla očekávána delší doba potřebná pro učení než u prvních dvou. Výsledek je patrně ovlivněn výrazným umístěním ovládacího prvku pro odpovídající akci v levém horním rohu ovládacího panelu, čímž je často prvním ovládacím prvkem, na který uživatel při zkoumání uživatelského rozhraní narazí. Tento prvek je ještě dále zvýrazněn (Obrázek 7), protože sedm následujících ovládacích prvků je po spuštění prototypu zašedlých, což indikuje, že je nelze v danou chvíli použít. Také má výraznou standardní ikonu.

Obrázek 7: Zvýraznění tlačítka nový dokument při prvním úkolu



Zdroj: autor

Přesto je medián trvání tohoto úkolu 4,28 sekundy pozoruhodně nízkou hodnotou, o dvě a půl sekundy nižší, než jsou mediány v případě prvních dvou prototypů.

4.2.2 Vypsat otázky

Při druhém úkolu bylo sledováno nalezení funkce, která je pro uživatele nová. Svoji roli stále může hrát učení se novému prostředí.

Kruksalův-Wallisův test prokázal významný rozdíl, uživatelské rozhraní druhého prototypu dosáhlo signifikantně delších časů, než ostatní dvě.

Zde již nejde o tak velké překvapení, ani znalost idiomu aplikace nepomohla při nalezení neznámé funkce. Podstatné ovšem je, že se potvrzuje nečekaně dobrý výsledek třetího prototypu, který mohl být v prvním úkolu popsán způsobem zkreslený. V tomto úkolu už není umístění hledaného ovládacího prvku tolik nápadné a přesto nebyla prokázána významná odlišnost od prvního prototypu a druhý prototyp je i tentokrát statisticky významně pomalejší. (Tabulka 5)

4.2.3 Vyplnit dotazník a zkontrolovat odpovědi

Tento úkol byl časově nejnáročnější, uživatelé při něm museli vyplnit tři údaje do dotazníku. Pokud údaje nezapsali nebo je zapsali způsobem, který neprošel kontrolou, byli vyzváni k jejich opravě a opětovnému spuštění kontroly, přičemž čas se počítá až do úspěšného splnění kontroly.

Testována je schopnost uživatelů nalézt ovládací prvek akce tematicky blízké akci předchozí, čemuž odpovídá i umístění prvků v bezprostřední blízkosti ve všech třech variantách GUI.

Stejně jako v předchozím úkolu, i zde Kruksalův-Wallisův test prokázal významný rozdíl, kdy uživatelské rozhraní druhého prototypu dosáhlo signifikantně delších časů, než ostatní dvě. Tento výsledek je třeba interpretovat opatrně, protože úkol agreguje dvě akce, z čehož vyplnění dotazníku je časově náročnější a přestože by mělo trvat na všech

variantách uživatelského rozhraní stejně, může znamenat riziko nepřesnosti.

4.2.4 Najít a nahradit

Úkol, při které měl uživatel nahradit všechny výskyty daného písmene jiným písmenem, slouží k otestování, jakým způsobem uživatel tento úkol splní. Čas zde není tolik sledovanou metrikou, nicméně mediány ukazují, že první prototyp je o dvě sekundy pomalejší, než druhý a třetí, což by odpovídalo předpokladům. Nepodařilo se však prokázat, že tento rozdíl je statisticky významný.

4.2.5 Vrátit poslední krok

Tento úkol byl opět zaměřen na testování způsobu řešení a čas zde není klíčovou metrikou, nicméně předpoklady byly nakloněny nejkratšímu času u desktopového GUI 2, protože tam je ze všech ostatních prototypů akce uživatelům nejlépe známá. To se nepotvrdilo, medián pro druhé uživatelské rozhraní vyšel dokonce o necelé dvě sekundy vyšší, než ostatní dva. Tento rozdíl se však opět nepodařilo prokázat za statisticky významný.

4.2.6 Uložit

Poslední dva úkoly patřily k těm jednoduchým, dobře známým. Navíc se v tu chvíli již uživatel v prostředí prototypu dobře orientoval. Pochopitelně zde byly dosahovány velmi krátké časy.

Pro úkol *uložit dokument*, testující jeden z nejznámějších počítačových příkazů vůbec, byl předpokládán dobrý výsledek v druhém prototypu. Jde o akci, kterou by měl provádět uživatel v desktopovém prostředí zcela automaticky. To se však nestalo z důvodů, které budou popsány níže. Žádný rozdíl zde nebyl statisticky významný.

4.2.7 Ukončit

Při akci *ukončení* platil opět předpoklad lehké výhody na straně druhého, desktopového uživatelského rozhraní, protože to šlo ukončit mimo menu také velmi dobře naučeným způsobem, křížkem v pravém horním rohu. V ostatních případech byl příkaz pro ukončení také dobře rozpoznatelný, umístěním v případě prvního prototypu nebo výraznou ikonou v případě třetího. Všechny časy však byly velmi rychlé a u žádné z variant se nepodařilo prokázat statisticky významně odlišný výsledek.

4.3 Míra chybovosti a míra úspěšného dokončení testu

Vedle *účinnosti* je další výkonnostní metrikou použitelnosti *přesnost*. Ta vychází z množství chyb, které uživatel udělá při práci s produktem.

Při tomto testování byly chyby rozděleny na závažné a malé. Závažné chyby jsou takové, které naruší práci a uživatel musí opakovat několik kroků, aby se dostal zpět tam, kde chybu udělal. Uživatelské rozhraní by nemělo dopustit, aby uživatel svojí chybou snadno ztratil data nebo způsobil nějakou nevratnou škodu. To dodržují i všechny tři testované prototypy a před krokem, který by znamenal takto charakterizovanou závažnou chybu, požaduje v dialogovém okně potvrzení. Jsou to dva případy, tlačítko *nový dokument*, které smaže aktuální obsah textového pole, a *vypsát otázky*, které přepíše aktuální text otázkami. Pokud uživatel použije tyto akce jindy, než je k tomu v instrukcích vyzván, riskuje ztrátu vyplněných odpovědí, a proto je požadováno potvrzení akce. Přesto došlo během testování k šesti případům závažných chyb. Tito uživatelé často brzy po takové chybě testování předčasně ukončili.

Malé chyby jsou zde definovány jako takové, které nemají vliv na další průběh práce. Jde například o stisknutí tlačítka *hledat* místo *najít a nahradit*, vyvolání kontroly, když není požadována, a podobně.

Tabulka 6: Přesnost podle jednotlivých prototypů

Sledovaná hodnota	Varianta GUI		
	1	2	3
Počet uživatelů	40	42	40
Četnost opuštění testování	5	2	6
Četnost závažných chyb	4	1	1
Četnost malých chyb	17	11	8
Statistická významnost rozdílu přesnosti (méně chyb je lépe)	-	+	+

Zdroj: autor

Tabulka 6 ukazuje počty závažných a malých chyb v jednotlivých variantách uživatelského rozhraní testovací aplikace. Z těchto hodnot vyplývá výrazně vyšší počet chyb v prvním prototypu, zejména závažných. Tyto hodnoty byly otestovány znaménkovým schématem a Chi-Square testem dobré shody.

Přestože na pohled rozdíl počtů závažných chyb vypadá dramaticky, nepodařilo se ani znaménkovým schématem, ani Chi-Square testem prokázat jeho statistickou významnost.

U počtů malých chyb byly zaznamenány vyšší četnosti, ze kterých již lze vyvodit závěry. Znaménkové schéma potvrdilo významnost odchylky počtu chyb v prvním prototypu. V Chi-Square testu vyšla signifikance 0,051, tedy na hranici přijatelnosti. Proto byly provedeny doplňující testy. Při porovnání výsledků prvního prototypu se sečtenými hodnotami druhého a třetího už Chi-Square test i Fisherův test na hladině významnosti 0,05 potvrzují vyšší počet chyb v prvním prototypu.

Při sečtení výskytů malých a závažných chyb a srovnání prvního prototypu se součtem dalších dvou je výsledek Chi-Square testu významný na hladině 0,01. Počet všech chyb v prvním prototypu je tedy prokázán jako významně vyšší, než v prototypch 2 a 3.

Další sledovanou hodnotou je poměr úspěšných dokončení a předčasných opuštění testování. Opuštění může být vyvoláno uživatelskou frustrací, čímž se stává jedním z ukazatelů subjektivního uspokojení uživatele. V tomto smyslu patří mezi nejdůležitější údaje webové analytiky. Zde ale není vzorek dostatečně rozsáhlý a statistickou významnost rozdílu mezi počty předčasných opuštění jednotlivých prototypů se nepodařilo prokázat.

4.4 Alternativní postupy řešení

Pro zkoumání adaptace odpovídajících uživatelských návyků pro rozdílná prostředí byly zařazeny tři úkoly, které měly více možností řešení. Šlo o úkoly *najít a nahradit*, *zpět* a *ukončit*. Smyslem těchto úkolů bylo porovnání, nakolik uživatele přiměje *look and feel* daného prototypu využít postup, který zná z takového prostředí.

4.4.1 Najít a nahradit

Najít a nahradit je funkce v prostředí desktopových aplikací, zejména v idiomu textového editoru, velmi běžná. Předpokládaným výsledkem bylo, že v desktopové variantě prototypu tuto funkci využije více uživatelů, než v hypertextové, kde taková funkce není očekávaná.

Jak je vidět v tabulce (Tabulka 7), výsledky odpovídají předpokladu a jsou velmi výrazné. Statistický test potvrdil významnost rozdílu mezi prvním a oběma dalšími prototypy.

Tabulka 7: Způsoby řešení úkolu *najít a nahradit*

Způsob řešení úkolu	Varianta GUI		
	1	2	3
Funkce <i>najít a nahradit</i>	9	22	21
	25%	55%	58,3%
Ruční přepsání	27	18	15
	75%	45%	41,7%

Zdroj: autor

Poněkud překvapivý je vysoký počet užití funkce *najít a nahradit* ve třetím prototypu, jehož prostředí nevychází ze známého idiomu. To znamená že v tomto prostředí uživatel nevyužil stávající znalosti, ale funkci bez problémů objevil. Přitom neexistuje standardní vyjádření této funkce ikonou, takže ta není tolik výrazná, jako některé ostatní. Ikona tedy tento výsledek zřejmě zásadně neovlivnila. Stejně tak pozice ovládacího prvku pro funkci *najít a nahradit* je spíše skrytá, na konci druhé řady, takže také nepomáhala snadnému nalezení a využití příkazu.

Obě alternativní varianty řešení úkolu *najít a nahradit* vykazovaly podobné časy řešení.

4.4.2 Zpět

Úkol vrátit poslední úpravu měl, jak již bylo popsáno, tři možné postupy řešení. Bylo možné použít funkci aplikace *zpět* (undo), nebo opět jednu z variant jako v předchozím případě, funkci *najít a nahradit* nebo přepsat písmena ručně. Předpoklad byl podobný, jako v předchozím případě. Akce *zpět* je uživatelům z desktopového prostředí velmi dobře známá,

proto by měli tuto akci umět použít nejčastěji v druhém, „desktopovém“ prototypu.

Tabulka 8: Způsoby řešení úkolu *Vrátit poslední krok*

Způsob řešení úkolu	Varianta GUI		
	1	2	3
Funkce zpět: četnost	4	15	15
Procentuální vyjádření	11,4%	37,5%	44,1%
Medián času	8,3 s	7,2 s	8 s
Funkce najít a nahradit: četnost	5	6	6
Procentuální vyjádření	14,3%	15%	17,7%
medián času	15, 8 s	16,6 s	12,8 s
Ruční přepsání: četnost	26	19	13
Procentuální vyjádření	74,3%	47,5%	38,2%
Medián času	10,6 s	15 s	11,1 s

Zdroj: autor

Ve výsledcích (Tabulka 8) je výrazně vidět návaznost na předchozí úkol. Většina uživatelů, kteří přepsali písmena ručně jednou, je stejným způsobem přepsala i zpět. Někteří uživatelé, rovnoměrně zastoupení u odlišných prototypů, postupovali přísně logicky a použili reverzně funkci, kterou použili naposled, tedy *najít a nahradit*. Akci *zpět* dle očekávání použili uživatelé druhého prototypu statisticky významně více, než uživatelé prvního. I u tohoto úkolu jsou však velmi zajímavé výsledky třetího prototypu, kde bylo dokonce ručního přepisování nejméně, přestože rozdíl druhého a třetího prototypu není statisticky významný. Zde mohla uživatelům pomoci výrazná standardní ikona a umístění ovládacího prvku na výhodné, dobře viditelné pozici.

Uživatelé, kteří pro splnění tohoto úkolu použili tlačítko zpět, dosáhli v součtu všech variant uživatelského rozhraní statisticky signifikantně kratších časů. Výrazné rozdíly v mediánech časů jednotlivých variant řešení na jednotlivých prototypch ukazuje tabulka (Tabulka 8).

Bylo také sledováno, jestli některý z účastníků zmáčkne tlačítko *zpět* prohlížeče. Nebyl zaznamenán ani jeden jeho stisk. Je možné, že častější užití tohoto příkazu by se ukázalo v počítačově méně gramotném vzorku uživatelů.

4.4.3 Ukončit

Úkol *ukončit* přinesl výsledky, které nemají hodnotu pro analýzu práce s prototypy, ale jsou velmi důležitým poznatkem o vlastnostech této metody výzkumu.

Tabulka 9: Způsoby ukončení

Ukončení	předčasné	vyžádané
zavření stránky v prohlížeči	11	2
funkce aplikace	2	107

Zdroj: autor

Četnosti v tabulce (Tabulka 9) ukazují zásadní rozdíl mezi způsobem zavření testovací aplikace u uživatelů, kteří dostali za úkol aplikaci ukončit, a uživatelů, kteří se rozhodli testování opustit a skutečně se chtěli dostat pryč. Z 13 předčasných opuštění testu pouze 2 proběhly příkazem k ukončení testovací aplikace a ve zbylých případech uživatelé rovnou zavírali stránku v prohlížeči. Při vyzvání uživatele k ukončení aplikace se poměr obrátil, pouze 2 ze 109 uživatelů zavřeli stránku přímo v prohlížeči, ostatní použili příkaz k ukončení v rámci aplikace.

Vlastně jde o krajní případ rozdílu mezi skutečným a simulovaným scénářem. Uživatelé si uvědomovali, že se nacházejí v testovacím prostředí a že se od nich očekává práce v rámci tohoto prostředí, tedy při instrukci k ukončení aplikace hledali tento příkaz právě v nabídce aplikace. Také se mohli například obávat, že při přímém zavření by se ztratila testovací data a podobně.

Podobný výsledek ukázaly i četnosti použití dvou alternativních možností ukončení druhého prototypu. Ten umožňoval zavření klasickým křížkem v pravém horním rohu aplikačního okna, nebo položkou v nabídce aplikačního menu. Předpokladem zde bylo, že zavření příkazem z menu je méně obvyklým způsobem. Přesto však 22 uživatelů druhého prototypu použilo položku v menu a pouze 18 zavřelo křížkem okno webové aplikace.

Výsledky tohoto úkolu jednoznačně ukazují, že při této metodě testování je kriticky důležité dbát na vhodnou formulaci instrukcí k úkolům, a především je nutné počítat s možným ovlivněním chování účastníků v testovacím prostředí oproti reálnému nasazení.

4.5 Klávesové zkratky

Druhý prototyp v souladu s idiomem textového editoru v desktopovém uživatelském rozhraní nabízel ovládání standardními klávesovými zkratkami. Poněkud překvapivým výsledkem bylo, že ze 124 vykonaných akcí, kdy mohly být použity, se tak stalo pouze dvakrát. Jedno použití bylo navíc neúspěšné, v případě, kdy se uživatel pokusil vyvolat funkci *najít a nahradit*, použil klávesovou zkratku Ctrl+F. V komentáři poté

vysvětlil, že v aplikaci MS Word je tato funkce sdružena právě pod touto klávesovou zkratkou s běžným hledáním.⁸

Druhý případ použití byl už úspěšný, šlo o uložení klávesovou zkratkou Ctrl+S.

Tento výsledek velmi výmluvně ukazuje, jak málo uživatelé přijímají klávesové zkratky v prostředí webu, přestože desktopový *look and feel* webové aplikace by je k tomu měl vybízet a možnost jejich využití byla vizuálně indikována.

4.6 Pohyb myši

V průběhu testování byl zaznamenáván pohyb myši. Vzhledem k charakteru výzkumu, kdy každý prototyp má jiné nároky na použití myši, není toto měření relevantní pro stanovení rozdílů mezi prototypy, mohlo však doplnit některé poznatky z jiných měření. Z těchto důvodů byla použita pouze velmi jednoduchá metoda.

V různých prototypech je na základě různého rozložení ovládacích prvků potřeba různé množství práce s myší, aniž by z toho bylo ovšem možné vyvozovat závěry o použitelnosti. Proto zde nelze srovnávat výsledky pohybu myši u prototypů navzájem. Je ovšem možné sledovat, jakým způsobem se liší použití myši u jednotlivých úkolů a zjistit případné rozdíly v uživatelských návycích.

Jediné významné výsledky se ukázaly v prvním prototypu, kde při úkolech *najít a nahradit* a *zpět* tři čtvrtiny uživatelů provedly úkoly

⁸ Kancelářský balík Microsoft Office má pro funkce *najít* a *najít a nahradit* jedno dialogové okno se dvěma korespondujícími záložkami. Podle stisku Ctrl+F nebo Ctrl+H nebo podle vybrané položky z menu je potom při zobrazení dialogového okna aktivní odpovídající záložka. Některé další aplikace obě funkce zcela sdružují pod označením *najít* a pouze pod klávesovou zkratkou Ctrl+F.

manuálním přepsáním, s minimálním použitím myši. Tento výsledek však vyšel již z analýzy alternativních řešení úkolů

Data z tohoto měření se neukázala jako statisticky hodnotná.

4.7 Subjektivní hodnocení a zpětná vazba

Po zavření testovacího prototypu se účastníci dostali k možnosti zápisu subjektivního hodnocení a další zpětné vazby. Subjektivní hodnocení bylo pouze textové, bez uzavřených otázek. Nejvíce pozitivních ohlasů bylo u druhého prototypu, nicméně relevantní hodnocení zanechalo příliš málo uživatelů na to, aby bylo možné provést statistické porovnání hodnocení jednotlivých prototypů. Zpětná vazba však v několika případech pomohla vysvětlit, čím byla způsobena chyba, kterou uživatel udělal.

4.8 Hodnoty dle jednotlivých prototypů

4.8.1 První prototyp – „hypertextový“

První prototyp, aplikující konvence statického hypertextu, svými výsledky z většiny naplnil předpoklady. Uživatelé se v prostředí dokázali orientovat a dosahované časy byly dobré. Překvapivě dobré byly časy i při úkolech, které uživatelé z prostředí webu nemají naučené, a kde byly předpoklady nakloněné spíše druhým dvěma prototypům. Nepřipravenost uživatelů na složitější funkcionalitu aplikace ve webovém prostředí se však projevila ve vyšší chybovosti. Podstatný je také významný rozdíl v přístupu k úkolům s alternativním řešením, kdy většina uživatelů nevyužila funkcionalitu aplikace a úkoly provedla ručně. Tyto výsledky potvrzují, že použité konvence uživatelského rozhraní ovlivnily chování uživatelů, přesněji řečeno aplikaci uživatelských návyků.

4.8.2 Druhý prototyp – „desktopový“

Druhý prototyp, představující idiom desktopového textového editoru ve webovém prostředí, přinesl očekávané výsledky jen částečně. Sice se na něj dokázali uživatelé dobře adaptovat, což dokazuje nízká míra chybovosti a pozitivní subjektivní hodnocení, byť statisticky neprůkazné, ale dosahované časy neprokázaly vyšší výkonnost. Využití aplikačního menu zřejmě zpomalilo práci s tímto prototypem, zejména v neznámých situacích. Tomu by odpovídaly výsledky časů pro akce, které v tomto idiomu nepatří k základním či standardním. Úkoly *generovat dotazník* a *vypsát otázky a zkontrolovat odpovědi* trvaly ve druhém prototypu nejdéle. To byly dva pro uživatele nové úkoly po sobě, a při plnění druhého z nich nepomohla ani tematická podobnost a blízkost umístění ovládacího prvku s prvním. Rychlost nebyla lepší než u ostatních prototypů ani při plnění úkolů, kde byla výhodou velmi dobrá uživatelova znalost prostředí. Ani vysoce počítačově gramotní uživatelé totiž ve webovém prostředí nevyužili klávesové zkratky, které by práci zrychlily.

Předpoklady druhý prototyp naplnil ve volbách variant řešení úkolů, kdy skutečně *look and feel* desktopové aplikace navedl uživatele k využití funkcí pro web nestandardních, uživatelé tedy aplikovali své uživatelské návyky z desktopových aplikací i v prostředí webu.

4.8.3 Třetí prototyp – „dotykový“

Třetí prototyp, nevycházející z konkrétních konvencí, ale připomínající uživatelské rozhraní dotykových zařízení, naopak překvapil velmi dobrými výsledky. Protože práce s tímto prototypem byla pro uživatele nová a nepomáhaly jim zde stávající uživatelské návyky ze složitějších rozhraní, byla očekávána nižší použitelnost. V této souvislosti byla očekávána zejména delší doba potřebná pro orientaci v uživatelském rozhraní, která by se projevila u prvního úkolu. Tento předpoklad byl

vyvrácen, když třetí prototyp při prvním úkolu dosáhl významně kratšího času, než další dva. I v dalších úkolech byly hodnoty výkonnostních metriky času i chybovosti dobré. Ve všech prvních třech úkolech byl prototyp 3 významně rychlejší, než prototyp 2. V dalších čtyřech úkolech nebyl prokazatelný rozdíl v rychlosti. Ačkoliv subjektivní hodnocení naznačovalo obtížnou orientaci v množství nevýrazně seřazených ovládacích prvků, kvantitativní hodnoty ukazují, že uživatelé neměli problém s nalezením funkcí aplikace.

V úkolech s alternativními možnostmi řešení nebyl významný rozdíl v rozložení postupů řešení od druhého prototypu, to znamená, že uživatelé dokázali využít funkce aplikace, aniž by přítomnost těchto funkcí vyvodili ze stávajících znalostí. Funkce jednoduše našli. V relativně neznámém prostředí se chovali flexibilněji, než v hypertextovém prostředí prvního prototypu, kde převážná většina uživatelů tyto funkce nehledala, protože tam nejsou obvyklé. Pomoci mohly i výrazné ikony. Nepodařilo se prokázat, že by výkonnost při jednotlivých úkolech souvisela s výhodností pozice daného ovládacího prvku.

Třetí prototyp dosahoval výkonnosti stejné nebo vyšší než první a druhý prototyp. Uživatelům učení nového prostředí nedělalo prokazatelný problém, naopak reagovali flexibilněji.

4.9 Hodnoty dle počítačové gramotnosti

Korelační analýza ukázala, že účastníci, kteří se přihlásili k oboru IT, dosahovali v souladu s očekáváním kratších časů. Korelační koeficient zde byl -0,074. Tento korelační koeficient byl prokázán za statisticky významný. Na výsledcích dvou úkolů s alternativními způsoby řešení se výrazně projevil vliv počítačové gramotnosti na použitý postup. Účastníci, kteří se přihlásili k IT oboru, statisticky významně méně přistupovali na možnost přepsat písmena ručně. Zatímco souhrnně při obou úkolech na všech třech prototypy účastníci, kteří se nepřihlásili

k IT oboru, přepisovali ručně v 62,7 % případů, ti, co se k IT oboru přihlásili, přepisovali ručně pouze v 26,5 % případů.

Byla potvrzena významná souvislost mezi počítačovou gramotností a uživatelskými návyky.

4.10 Výsledky

H1: Typ uživatelského rozhraní má vliv na účinnost.

Ve třech ze sedmi úkolů se podařilo prokázat statisticky významný rozdíl mezi účinností (=rychlostí) jednotlivých prototypů. Hypotéza 1 byla potvrzena, typ uživatelského rozhraní tedy má vliv na účinnost.

H2: Typ uživatelského rozhraní má vliv na přesnost.

Byl prokázán statisticky významný rozdíl mezi chybovostí jednotlivých prototypů. Hypotéza 2 byla potvrzena, typ uživatelského rozhraní tedy má vliv na přesnost.

H3: Typ uživatelského rozhraní má vliv na užití odlišných uživatelských návyků.

Byl prokázán statisticky významný rozdíl ve způsobu řešení úkolů v rozdílných prostředích jednotlivých prototypů. Hypotéza 3 byla potvrzena, typ uživatelského rozhraní tedy má vliv na užití odlišných uživatelských návyků.

4.11 Zhodnocení a aplikace výsledků

Každý prototyp se zaměřoval na jinou schopnost uživatele:

- znalost konvencí v případě prvního prototypu
- užití známých konvencí v novém prostředí v případě druhého prototypu
- učení nového prostředí v případě třetího prototypu

V tak jednoduché aplikaci, jako byla testovací, ukázaly tyto tři přístupy překvapivé výsledky. Učení nového prostředí bylo velmi rychlé, prvotní orientace v prostředí byla dokonce rychlejší, než u prototypů založených na konvencích. Přitom přesnost byla také vysoká. Tento výsledek naznačuje, že pro použitelnost velmi jednoduchých aplikací nemají idiomy podstatný význam. Hlavní je pro uživatele znalost jednotlivých prvků, dále návrhových vzorů, pak teprve celých idiomů.

Stejně tak aplikační menu se neukázalo jako účinné řešení pro takto jednoduchou aplikaci. Uživatel sice tento vzor dobře zná a umí použít, pro rychlou orientaci v nabídce akcí ale neslouží dobře, protože ovládací prvky všech akcí nejsou vidět na první pohled.

Velmi pozitivní pro rychlou identifikaci ovládacího prvku se ukázaly výrazné standardní grafické ikony.

V hypertextovém prostředí uživatel neočekává žádnou funkcionalitu aplikace na klientské straně. Pokud tam taková funkcionalita je, málo uživatelů ji využije. V souladu s Tidwellovou (Tidwell, 2005) by akce měly být označeny tlačítky, hypertextové odkazy slouží pro navigaci.

Uživatel je na webu, zejména v novém prostředí, značně flexibilní. Viditelné ovládací prvky úspěšně nachází a používá i bez pomoci naučených konvencí. Výsledky tohoto výzkumu podporují doporučení McCarthyho (citace) nebát se aplikovat prvky na základě empirických výzkumů i za cenu odklonění se od konvencí.

Oproti flexibilitě v použití viditelných ovládacích prvků silně kontrastuje schopnost užití skrytých ovládacích prvků. Ty prokázaly výrazně vyšší závislost na platformních konvencích, protože uživatel se je učí velmi pomalu. Na webové platformě skryté ovládací prvky konvenční nejsou.

Protože klávesové zkratky efektivně slouží pro zrychlení práce zkušených uživatelů, ale ve webovém prostředí je uživatelé pro nekonzistenci používají minimálně, bylo by vhodné hledat způsob jejich implementace, který by se stal konvencí. Takový krok mohou udělat například vývojáři majoritních prohlížečů, pokud by omezili užití standardních klávesových zkratk na straně vlastního prohlížeče, a tím je uvolnili pro použití webových aplikací. Například klávesová zkratka Ctrl+S by nesloužila pro funkci prohlížeče uložení aktuální stránky, ale byla by volná pro případné použití webovou aplikací. Uživatelé by se tak nemuseli obávat, že vyvolají nechtěnou akci prohlížeče, a pravděpodobně by se naučili tuto klávesovou zkratku používat i v prostředí webových aplikací. V současné situaci je implementace této zkratky ve webovém prostředí velmi problematická, protože vede k nekonzistenci napříč webovými aplikacemi i nekonzistentnímu chování prohlížeče. Další možností by byla standardizace a vyšší rozšíření alternativního systému k desktopovým standardním klávesovým zkratkám, například už používaných jednoklávesových zkratk z prostředí textového editoru *vi*. To by však neřešilo užití desktopových uživatelských návyků ve webovém prostředí.

5 Závěr

Cíle práce byly splněny v takovém rozsahu, v jakém byly stanoveny. Podařilo se shromáždit dostatečně rozsáhlý soubor dat, aby bylo možné získat statisticky signifikantní výsledky.

Výzkumem se podařilo prokázat všechny tři stanovené hypotézy. Typ uživatelského rozhraní má vliv na účinnost, přesnost i užití odlišných uživatelských návyků.

Rozdíly v účinnosti se lišily od očekávání. Druhý, „desktopový“, prototyp, který využíval uživatelské znalosti tohoto typu prostředí, prokázal oproti ostatním prototypům významně nižší účinnost při úkolech, které nejsou v tomto prostředí zcela standardní. Nepotvrdil se ani předpoklad, že třetí prototyp, jehož prostředí nevycházelo ze známého idiomu, bude od uživatelů vyžadovat delší čas na orientaci v novém prostředí. Celkově byly míry účinnosti u odlišných prototypů velmi vyrovnané a u čtyř ze sedmi úkolů se rozdíl mezi prototypy nepodařilo prokázat.

Míry přesnosti potvrdily předpoklad dobrého výsledku druhého prototypu, ale překvapivě i u třetího prototypu. První, „hypertextový“, prototyp, zaznamenal statisticky významně nižší přesnost.

Jednotlivé výkonnostní metriky použitelnosti tedy prokázaly rozdíly mezi odlišnými prototypy, v takto jednoduché aplikaci však prototypy dodržující platformní konvence nevykázaly celkově vyšší použitelnost.

Nejhodnotnějším výsledkem je prokázání užití odlišných uživatelských návyků v různých prostředích jednotlivých prototypů. *Look and feel* grafického uživatelského rozhraní tedy dokázal navést uživatele k využití uživatelských návyků odpovídajících tomuto typu rozhraní.

Výsledky výzkumu jasně ukázaly rozdíl mezi vnímáním viditelných a skrytých ovládacích prvků. Uživatelé prokazují vysokou flexibilitu při adaptaci na různá prostředí s odlišným pojetím viditelných ovládacích prvků, naproti tomu se velmi obtížně učí používat skryté ovládací prvky.

5.1 Přínos práce

Práce se zabývá problematikou, která dosud nebyla přesně popsána a podložena kvantitativním výzkumem. Hlavním přínosem práce je statistické prokázání závislosti používaných uživatelských návyků na implementaci odlišných konvencí v uživatelském rozhraní webové aplikace. Zároveň byla prokázána nízká míra adaptace na skryté ovládací prvky.

Získané výsledky rozšiřují znalosti o chování uživatelů v závislosti na prostředí. Lze je využít přímo v praxi při návrhu uživatelských rozhraní nebo jako podklad pro další výzkum.

Přestože tento výzkum se zaměřoval na obecné otázky uživatelské adaptace na odlišná prostředí, použitá metodika je aplikovatelná pro širokou škálu dalšího zkoumání, včetně uživatelského chování v konkrétních scénářích při návrhu reálných aplikací.

Terminologický slovník

Termín	Význam
Best-practices	Soubor řešení, která jsou osvědčená, standardizovaná či prosazovaná.
Business logika	Část aplikace představující „výkonné jádro“.
Drag and drop	Přímá manipulace s virtuálními objekty myši. Stisknutí tlačítka myši s kurzorem na daném objektu, přetažení objektu na jiné místo a puštění tlačítka myši.
Drobečková navigace	Navigační prvek webových stránek, sloužící pro udržení orientace o umístění aktuální stránky ve struktuře celého webu a pro návrat na vyšší úroveň struktury. Drobečková navigace bývá umístěna přímo nad obsahem a má podobu například: Právě prohlížíte: Hlavní stránka fakulty / O fakultě / Katedry a pracoviště (příklad je z webové prezentace fis.vse.cz)
Eye tracking	Sledování pohledu oční kamerou slouží ke zkoumání, na jaké prvky vizuální stránky prostředí se uživatel zaměřuje, jakým způsobem probíhá hledání a podobně.
Grafické uživatelské rozhraní (GUI)	Typ uživatelského rozhraní založený na vizuálním zobrazení. K jeho ovládání slouží interaktivní grafické ovládací prvky.
Guidelines	Soubory doporučených postupů založené konvencích i empirických výzkumech. Zpravidla jde o rozsáhlé publikace významných odborníků v dané oblasti.
Human-computer Interaction (HCI)	Pole výzkumu zabývající se všemi možnostmi, jak mohou člověk a počítač vzájemně komunikovat a spolupracovat.
Idiom rozhraní	Idiom rozhraní (interface idiom) je známý a rozpoznatelný typ rozhraní se typickou sadou objektů, akcí a zobrazení.

Look and feel	Celkový dojem z produktu zahrnující vizuální aspekty i ovládací prvky.
Mouse tracking	Sledování pohybu kurzoru myši může sloužit jako jednoduchá náhrada za sledování oční kamerou. Velmi hrubě pomáhá indikovat objekt pozornosti uživatele nebo úsilí vynaložené na daný úkol.
Návrhový vzor	Zobecněný způsob řešení ukazuje zavedenou a osvědčenou možnost přístupu ke konkrétnímu problému návrhu.
Panel nástrojů	Panel nástrojů (toolbar) je návrhovým vzorem rozložení vizuálního rozhraní aplikace, představuje umístění tlačítek pro akce na jednom panelu.
Použitelnost	Míra, do jaké může být produkt nebo systém používán stanovenými uživateli pro dosažení stanovených cílů účinně, přesně a uspokojivě ve stanoveném kontextu použití. (ISO/IEC 25010:2011, 2011)
User Experience (UX)	Oblast zkoumání a vývoje zahrnující kompletně uživatelský prožitek z produktu, služby či systému.
Uživatelské rozhraní (UI)	Uživatelské rozhraní je souhrnem aspektů počítačového systému či programu, které může člověk jako uživatel vidět nebo jiným způsobem vnímat, a příkazy a mechanismy, kterými uživatel řídí činnost systému či programu a vkládá data (Howe, 2009)

Seznam literatury

ALBERT, William, Thomas TULLIS a Donna TEDESCO. 2010. *Beyond the Usability Lab: Conducting Large-Scale User Experience Studies*. S.l.: Morgan Kaufmann. ISBN 9780123748928.

ANSI/INCITS-354. *Common Industry Format for Usability Test Reports*. Washington, DC, USA: American National Standards Institute, 2001.

BARNUM, Carol M. 2010. *Usability Testing Essentials: Ready, Set...Test!*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. ISBN 012375092X, 9780123750921.

GALITZ, Wilbert O. 2002. *The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Techniques*. 2. New York: Wiley Computer Pub. ISBN 0471084646 9780471084648.

GUO, Qi a Eugene AGICHTEIN. 2010. Towards predicting web searcher gaze position from mouse movements. In: *Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems* [online]. New York, NY, USA: ACM, s. 3601–3606. CHI EA '10. ISBN 978-1-60558-930-5. Dostupné z: <http://doi.acm.org/10.1145/1753846.1754025>
Atlanta, Georgia, USA

HOWE, Denis. User interface. *The Free On-line Dictionary of Computing* [online]. 2009 [cit. 2009-05-28]. Dostupný z WWW: <Dictionary.com [http://dictionary.reference.com/browse/user interface](http://dictionary.reference.com/browse/user%20interface)>.

ISO/IEC 25010:2011. *Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models*. Geneva, Switzerland: ISO/IEC, 2011.

JONES, Derek.M. 2002. The 7±2 Urban Legend. In: *MISRA C Conference 2002*. S.l.: Knowledge Software,

KALBACH, James. *Designing Web Navigation*. [s.l.] : O'Reilly Media, 2007. 416 s.

MCCARTHY, John D., M. Angela SASSE a Jens RIEGELSBERGER. 2003. Could i have the menu please? an eye tracking study of design conventions. In: *In Proceedings of HCI2003*. S.l.: Springer-Verlag, s. 401–414

NIELSEN, Jakob. Interface Standards and Design Creativity. [online]. 1999 [vid. 26. květen 2012]. Dostupné z: <http://www.useit.com/alertbox/990822.html>

NIELSEN, Jakob. Usability Testing with 5 Users. [online]. 2000 [vid. 11.květen 2012]. Dostupné z: <http://www.useit.com/alertbox/20000319.html>

NIELSEN, Jakob. Usability 101: Definition and Fundamentals - What, Why, How. [online]. 2003 [vid. 19. březen 2012]. Dostupné z: <http://www.useit.com/alertbox/20030825.html>

NIELSEN, Jakob. *Web.design*. Praha : SoftPress, 2002. 382 s.

NIELSEN, Jakob. 10 Heuristics for User Interface Design. [online]. 2005 [vid. 24. březen 2012]. Dostupné z: http://www.useit.com/papers/heuristic/heuristic_list.html

NIELSEN, Jakob a Marie TAHIR. 2005. *Použitelnost domovských stránek*. Brno: Zoner Press. ISBN 80-86815-18-8.

PARUSH, Avi. 2001. Usability Design and Testing. *interactions*, roč. 8, č. 5, s. 13–17. doi 10.1145/382899.382902.

PARUSH, Avi, Yonit SHWARTS, Avy SHTUB a M. Jeya CHANDRA. 2005. The Impact of Visual Layout Factors on Performance in Web Pages: A Cross-Language Study. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, roč. 47, č. 1, s. 141–157. doi 10.1518/0018720053653785.

POWELL, Thomas A. 2008. *Ajax: The Complete Reference*. S.l.: McGraw-Hill Prof Med/Tech. ISBN 9780071492164.

SHACKEL, B a S. J RICHARDSON. 1991. *Human factors for informatics usability*. Cambridge; New York: Cambridge University Press. ISBN 0521365708 9780521365703.

TIDWELL, Jenifer. 2005. *Designing Interfaces*. 1. S.l.: O'Reilly Media. ISBN 978-0-596-00803-1.

TOUŠEK, Pavel. 2009. *Srovnání uživatelských rozhraní webových a desktopových aplikací a otázka jejich konvergence*. Praha, FIS VŠE v Praze.

Seznam tabulek, obrázků a grafů

Seznam tabulek

Tabulka 1: Strategie a nástroje pro tvorbu řešení uživatelských rozhraní	5
Tabulka 2: Tabulka Standardní klávesové zkratky	17
Tabulka 3: Úkoly při testování	30
Tabulka 4: Klávesové zkratky druhého prototypu	34
Tabulka 5: Statistické významnosti rozdílů účinnosti (vyšší je lepší)	43
Tabulka 6: Přesnost podle jednotlivých prototypů	48
Tabulka 7: Způsoby řešení úkolu <i>najít a nahradit</i>	50
Tabulka 8: Způsoby řešení úkolu <i>Vrátit poslední krok</i>	51
Tabulka 9: Způsoby ukončení	52

Seznam obrázků

Obrázek 1: Upozornění na klávesové ovládání na webu 9gag.com	20
Obrázek 2: Viditelné tlačítko kontextového menu v Google Docs	21
Obrázek 3: První prototyp	32
Obrázek 4: Druhý prototyp	33
Obrázek 5: Třetí prototyp	35
Obrázek 6: Architektura testovací aplikace	37
Obrázek 7: Zvýraznění tlačítka nový dokument při prvním úkolu	44

Seznam grafů

Graf 1: Efekt očekávání při první návštěvě stránky	13
Graf 2: Procento odhalených chyb v závislosti na počtu účastníků	22
Graf 3: Souhrny všech časů podle typu GUI	41
Graf 4: Rozdělení časů podle jednotlivých úkolů	41
Graf 5: Mediány časů řešení úkolů	42

Seznam příloh

Příloha 1: Vstupní stránka testování	70
Příloha 2: Tabulka korelací a jejich statistických významností	71
Příloha 3: Statistické testy rozdílů časů plnění úkolu <i>ukončit</i>	72
Příloha 4: Statistické testy rozdílů časů plnění úkolu <i>vypsát otázky</i>	73
Příloha 5: Statistické testy rozdílů časů plnění úkolu <i>zkontrolovat odpovědi</i>	74
Příloha 6: Statistické testy rozdílů časů plnění úkolu <i>nový dokument</i>	75
Příloha 7: Statistické testy rozdílů časů plnění úkolu <i>najít a nahradit</i>	76
Příloha 8: Statistické testy rozdílů časů plnění úkolu <i>uložit</i>	77
Příloha 9: Statistické testy rozdílů časů plnění úkolu <i>vrátit poslední krok</i>	78
Příloha 10: Statistické testy rozdílů chybovosti jednotlivých prototypů.....	79
Příloha 11: Statistické testy rozdílů ve způsobu ukončení	80

Příloha 1: Vstupní stránka testování

prototypovétestování.eu

Co je to „prototypové testování“?

Prototypové testování je technika, používaná při vývoji uživatelského rozhraní. Její podstatou je studium chování uživatelů při práci s prototypy určité aplikace, což umožňuje odhalit chyby v návrhu a vybrat nejlepší řešení, aby se s výslednou aplikací pracovalo co nejjednodušší.

Co dělám já?

Já ve své diplomové práci zkoumám rozdíly mezi návyky uživatelů v různých prostředích. Nejde mi o konkrétní aplikaci, proto testované prototypy jako takové záměrně nemají žádný smysl. Já budu automaticky zaznamenávat Vaše kroky v testovacím prostředí a následně statisticky porovnávat reakce v různých situacích.

Co chci po Vás?

Vás chci požádat o účast na testování. Jednoduše splníte zadání úkolů v modelové aplikaci. Po spuštění se Vám zobrazí stránka se dvěma základními prvky: nahoře bude panel s instrukcemi a pod ním vlastní aplikace. Instrukcí se postupně zobrazí sedm. Všechny jsou velmi jednoduché a zabere Vám to asi dvě minuty. Vše je zcela anonymní.

Spuštěním aplikace souhlasíte s automatickým zaznamenáním Vaší činnosti v testovacím prostředí.

spustit

Pavel Toušek
pavel.tousek@gmail.com

Příloha 2: Tabulka korelací a jejich statistických významností

Kendall's tau_b	Stage Duration	Correlation Coefficient		Stage Duration	malá chyba	závažná chyba	předčasné ukončení	ZÁVAŽNÁ CHYBA A PŘEDČASNÉ UKONČENÍ	VĚK	IT	POHLAVÍ
		Sig. (2-tailed)	N								
	malá chyba	Correlation Coefficient	1231	-1,00	1,000						
		Sig. (2-tailed)	N								
	závažná chyba	Correlation Coefficient	1231	,078	,012	1,000					
		Sig. (2-tailed)	N								
	předčasné ukončení	Correlation Coefficient	1231	,079	,076	,007	1,000				
		Sig. (2-tailed)	N								
	ZÁVAŽNÁ CHYBA A PŘEDČASNÉ UKONČENÍ	Correlation Coefficient	1231	,022	,056	,559	,825	1,000			
		Sig. (2-tailed)	N								
	VĚK	Correlation Coefficient	1231	,008	,025	,039	,041	,057	1,000		
		Sig. (2-tailed)	N								
	IT	Correlation Coefficient	1231	,074	,013	,017	,004	,014	,155	1,000	
		Sig. (2-tailed)	N								
	POHLAVÍ	Correlation Coefficient	1221	,042	,044	,029	,027	,040	,226	,343	1,000
		Sig. (2-tailed)	N								
			1219		1219	1219	1219	1219	1186	1219	1219

Příloha 3: Statistické testy rozdílů časů plnění úkolu *ukončit*

Spec = end

Kruskal-Wallis Test

Ranksa			
GUI		N	Mean Rank
StageDuration	1	34	59,35
	2	40	55,69
	3	33	46,44
	Total	107	

Test Statisticsa,b,c

	StageDuration
Chi-square	3,089
df	2
Asymp. Sig.	,213

Median Test

Frequenciesa

		GUI		
		1	2	3
StageDuration	> Median	20	21	12
	<= Median	14	19	21

Test Statisticsb,c

	StageDuration
N	107
Median	5758,00
Chi-square	3,604
df	2
Asymp. Sig.	,165

Příloha 4: Statistické testy rozdílů časů plnění úkolu *vypsát otázky*

Spec = generate

Kruskal-Wallis Test

Ranksa			
GUI		N	Mean Rank
StageDuration	1	38	48,59
	2	41	84,12
	3	39	44,24
	Total	118	

Test Statisticsa,b,c	
	StageDuration
Chi-square	32,863
df	2
Asymp. Sig.	,000

Median Test

Frecuenciasa				
		GUI		
		1	2	3
StageDuration	> Median	11	35	13
	<= Median	27	6	26

Test Statisticsb,c	
	StageDuration
N	118
Median	6343,00
Chi-square	31,582
df	2
Asymp. Sig.	,000

Příloha 5: Statistické testy rozdílů časů plnění úkolu *zkontrolovat odpovědi*

Spec = checkok

Kruskal-Wallis Test

Ranksa			
GUI		N	Mean Rank
StageDuration	1	36	46,13
	2	40	66,34
	3	37	57,49
	Total	113	

Test Statisticsa,b,c	
	StageDuration
Chi-square	7,223
df	2
Asymp. Sig.	,027

Median Test

Frequenciasa				
		GUI		
		1	2	3
StageDuration	> Median	13	26	17
	<= Median	23	14	20

Test Statisticsb,c	
	StageDuration
N	113
Median	25283,00
Chi-square	6,613
df	2
Asymp. Sig.	,037

Příloha 6: Statistické testy rozdílů časů plnění úkolu *nový dokument*

Spec = new

Kruskal-Wallis Test

Ranksa			
GUI		N	Mean Rank
StageDuration	1	38	66,68
	2	41	69,02
	3	40	44,40
	Total	119	

Test Statisticsa,b,c

	StageDuration
Chi-square	12,413
df	2
Asymp. Sig.	,002

Median Test

Frequenciesa

		GUI		
		1	2	3
StageDuration	> Median	21	24	14
	<= Median	17	17	26

Test Statisticsb,c

	StageDuration
N	119
Median	6472,00
Chi-square	5,208
df	2
Asymp. Sig.	,074

Příloha 7: Statistické testy rozdílů časů plnění úkolu *najít a nahradit*

Spec = replacecomplete

Kruskal-Wallis Test

Ranksa			
GUI		N	Mean Rank
StageDuration	1	41	67,00
	2	46	64,00
	3	42	64,14
	Total	129	

Test Statisticsa,b,c

	StageDuration
Chi-square	,172
df	2
Asymp. Sig.	,917

Median Test

Frequenciesa

		GUI		
		1	2	3
StageDuration	> Median	23	21	20
	<= Median	18	25	22

Test Statisticsb,c

	StageDuration
N	129
Median	16692,00
Chi-square	1,045
df	2
Asymp. Sig.	,593

Příloha 8: Statistické testy rozdílů časů plnění úkolu *uložit*

Spec = save

Kruskal-Wallis Test

Ranksa			
GUI		N	Mean Rank
StageDuration	1	35	54,40
	2	40	61,18
	3	34	48,35
	Total	109	

Test Statisticsa,b,c

	StageDuration
Chi-square	3,043
df	2
Asymp. Sig.	,218

Median Test

Frequenciesa

		GUI		
		1	2	3
StageDuration	> Median	18	21	15
	<= Median	17	19	19

Test Statisticsb,c

	StageDuration
N	109
Median	4071,00
Chi-square	,590
df	2
Asymp. Sig.	,745

Příloha 9: Statistické testy rozdílů časů plnění úkolu *vrátit poslední krok*

Spec = undo

Kruskal-Wallis Test

Ranksa			
GUI		N	Mean Rank
StageDuration	1	30	47,07
	2	34	49,15
	3	28	42,68
	Total	92	

Test Statisticsa,b,c	
	StageDuration
Chi-square	,921
df	2
Asymp. Sig.	,631

Median Test

Frequenciasa				
		GUI		
		1	2	3
StageDuration	> Median	15	18	13
	<= Median	15	16	15

Test Statisticsb,c	
	StageDuration
N	92
Median	10446,00
Chi-square	,261
df	2
Asymp. Sig.	,878

Příloha 10: Statistické testy rozdílů chybovosti jednotlivých prototypů

GUI * malá chyba malá chyba Crosstabulation

			malá chyba		Total
			0	1	
Count	GUI	1	376	17	393
		2	425	11	436
		3	394	8	402
	Total		1195	36	1231
Sign Scheme	GUI	1	-	+	
		2	o	o	
		3	o	o	

Chi-Square Tests

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	4,202	2	,122
Likelihood Ratio	4,025	2	,134
Linear-by-Linear Association	3,799	1	,051
N of Valid Cases	1231		

GUI * závažná chyba závažná chyba Crosstabulation

			závažná chyba		Total
			0	1	
Count	GUI	1	389	4	393
		2	435	1	436
		3	401	1	402
	Total		1225	6	1231
Sign Scheme	GUI	1	o	o	
		2	o	o	
		3	o	o	

Chi-Square Tests

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	3,350	2	,187
Likelihood Ratio	3,054	2	,217
Linear-by-Linear Association	2,401	1	,121
N of Valid Cases	1231		

Příloha 11: Statistické testy rozdílů ve způsobu ukončení

GUI * předčasnéukončení předčasné ukončení Crosstabulation

			předčasné ukončení		Total
			0	1	
Count	GUI	1	388	5	393
		2	434	2	436
		3	396	6	402
	Total		1218	13	1231
Sign Scheme	GUI	1	o	o	
		2	o	o	
		3	o	o	

Chi-Square Tests

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	2,398	2	,302
Likelihood Ratio	2,704	2	,259
Linear-by-Linear Association	,099	1	,754
N of Valid Cases	1231		

GUI * ZÁVAŽNÁ CHYBA A PŘEDČASNÉ UKONČENÍ Crosstabulation

			ZÁVAŽNÁ CHYBA A PŘEDČASNÉ UKONČENÍ		Total
			0	1	
Count	GUI	1	384	9	393
		2	433	3	436
		3	395	7	402
	Total		1212	19	1231
Sign Scheme	GUI	1	o	o	
		2	o	o	
		3	o	o	

Chi-Square Tests

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	3,644	2	,162
Likelihood Ratio	3,999	2	,135
Linear-by-Linear Association	,378	1	,538
N of Valid Cases	1231		