

Vysoká škola ekonomická v Praze

Fakulta informatiky a statistiky

Katedra informačních technologií

Student : Zdeněk Hejl
Vedoucí bakalářské práce : Ing. Libor Gála

TÉMA BAKALÁŘSKÉ PRÁCE

Analýza metod testování použitelnosti webových aplikací

ROK : 2007

Prohlášení

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

V Praze dne 11.05.2007

.....

podpis

Poděkování

Rád bych na tomto místě poděkoval všem, kteří mi s prací pomohli, především mému vedoucímu panu Ing. Liboru Gálvi.

Abstrakt

Hlavním cílem této práce je analýza metod testování použitelnosti webových aplikací a zařazení těchto metod do jednotlivých fází životního cyklu vývoje webové aplikace. Mimo tento hlavní cíl se práce snaží upozornit na nedostatečný důraz kladený na použitelnost při vývoji těchto aplikací.

Největším přínosem této práce je zpřístupnění přehledu metod pro testování použitelnosti českým tvůrcům webových aplikací. Metody jsou prezentovány v takové formě, která umožňuje rychlé zjištění základních informací a porovnávání jednotlivých metod. K usnadnění výběru správné metody jsou v práci obsaženy tři grafické přehledy, které znázorňují zařazení jednotlivých metod do fází životního cyklu vývoje aplikace, příbuznost metod a možnost vzdáleného testování.

Práce je strukturována do tří hlavních kapitol. První kapitola poskytuje úvod do problematiky použitelnosti, vymezuje specifika použitelnosti webových aplikací a určuje význam použitelnosti. Druhá kapitola navazuje na teoretické poznatky z první kapitoly a zaměřuje se na představení modelů životního cyklu vývoje aplikace, postavení použitelnosti v těchto modelech a upozorňuje na nedostatečné využívání principů použitelnosti při vývoji aplikací. Hlavní náplní třetí kapitoly je strukturovaný přehled metod testování použitelnosti doplněný grafickými přehledy.

Abstract

Principal aim of this thesis is to analyze web applications usability testing methods and to place these methods into the appropriate phases of web application development life cycle. Except this principal aim, thesis tries to alert to inadequate accent put on the usability during development of these applications.

A retrieval of usability testing methods list for Czech web application developers is the biggest contribution of this thesis. The methods are presented in such form that enables quick finding of basic information and methods comparison. To facilitate a choice of appropriate method, the thesis includes three graphic summaries that illustrate an enlistment of methods into the phases of web application development life cycle, the relationships between methods and a remote testing possibility.

The thesis is divided into three main chapters. First chapter provides an introduction to usability, defines specificity of web applications usability and sets a usability importance. Second chapter takes up theoretic outputs from first chapter and focuses on presentation of web application development life cycle models, on position of usability in these models and alerts to inadequate use of usability principles during applications development. A structured usability testing methods overview with additional graphic overviews is the main contents of third chapter.

Obsah

1. ÚVOD	1
1.1. VYMEZENÍ TÉMATU	1
1.2. CÍLE PRÁCE	1
1.3. PŘÍSTUP KE ZPRACOVÁNÍ PRÁCE	2
2. ÚVOD DO POUŽITELNOSTI	3
2.1. DEFINICE POUŽITELNOSTI	3
2.2. SLOŽKY POUŽITELNOSTI	4
2.2.1. NAUČITELNOST	4
2.2.2. EFEKTIVITA	6
2.2.3. ZAPAMATOVATELNOST	6
2.2.4. CHYBOVOST	6
2.2.5. USPOKOJENÍ	7
2.3. VÝZNAM POUŽITELNOSTI	7
2.4. POUŽITELNOST WEBOVÝCH APLIKACÍ	8
3. POUŽITELNOST A ŽIVOTNÍ CYKLUS VÝVOJE APLIKACE	9
3.1. MODELY ŽIVOTNÍHO CYKLU VÝVOJE APLIKACE	9
3.1.1. MODEL VELKÉHO TŘESKU	10
3.1.2. METODA TUNELU	11
3.1.3. MODEL „PROGRAMUJ A OPRAVUJ“	12
3.1.4. MODEL VODOPÁDU	13
3.1.5. V-MODEL	14
3.1.6. INKREMENTÁLNÍ MODEL	15
3.1.7. SPIRÁLOVÝ MODEL	15
3.2. VYUŽITÍ PRINCIPŮ POUŽITELNOSTI PŘI VÝVOJI APLIKACE	17
3.2.1. KDY APLIKOVAT POUŽITELNOST?	17
3.2.2. TESTOVÁNÍ POUŽITELNOSTI	18
4. METODY TESTOVÁNÍ POUŽITELNOSTI	19
4.1. SLEDOVANÉ CHARAKTERISTIKY	19

4.2. TESTOVÁNÍ S ÚČASTÍ KONCOVÝCH UŽIVATELŮ APLIKACE	21
4.2.1. CARD SORTING	21
4.2.2. AFFINITY DIAGRAMS	21
4.2.3. PLURALISTIC WALKTHROUGHS	22
4.2.4. THINK ALOUD PROTOCOL	23
4.2.5. QUESTION ASKING PROTOCOL	23
4.2.6. EVALUATE PROTOTYPE	24
4.2.7. DIAGNOSTIC EVALUATION	25
4.2.8. SELF-REPORTING LOGS	25
4.2.9. JOURNALED SESSIONS	26
4.2.10. SCREEN SNAPSHOTS	26
4.2.11. CO-DISCOVERY METHOD	26
4.2.12. CRITICAL INCIDENT TECHNIQUE	27
4.2.13. PERFORMANCE MEASUREMENT	28
4.2.14. ATTITUDE QUESTIONNAIRES	29
4.3. TESTOVÁNÍ BEZ ÚČASTI KONCOVÝCH UŽIVATELŮ APLIKACE	29
4.3.1. STANDARDS INSPECTION	29
4.3.2. FORMAL USABILITY INSPECTION	30
4.3.3. CONSISTENCY INSPECTION	30
4.3.4. FEATURE INSPECTION	31
4.3.5. COGNITIVE WALKTHROUGHS	31
4.3.6. EXPERT EVALUATION	32
4.3.7. HEURISTIC EVALUATION	32
4.4. GRAFICKÉ PŘEHLEDY	33
<u>5. ZÁVĚR</u>	37
<u>POUŽITÉ ZDROJE</u>	39
<u>TERMINOLOGICKÝ SLOVNÍK</u>	43

1. Úvod

1.1. Vymezení tématu

V současnosti existují desítky až stovky nejrůznějších přístupů k tvorbě aplikací. Liší se ve velkém množství charakteristik, avšak jedno má většina z nich společného – jsou zaměřeny na technické aspekty vývoje, přičemž zapomínají na koncového uživatele aplikace. Důsledkem toho vznikají aplikace, které jsou obtížně ovladatelné, jejich chování je nepředvídatelné a uživatel se v nich špatně orientuje. Tato situace se týká nejen klasických desktopových aplikací, ale i webových aplikací a stránek.

V minulosti vznikl jako reakce na tento technický přístup k výrobě a vývoji různých nástrojů a systémů obor zvaný použitelnost, který se zabývá jednoduchostí používání jakýchkoliv nástrojů a systémů. Použitelnost v počátcích zkoumala nástroje reálného života (např. telefony), posléze se začala zabývat počítačovými aplikacemi a nejnovějším přírůstkem do množiny zkoumaných nástrojů jsou webové aplikace a stránky.

Vzhledem k tomu, že přístup k Internetu má stále více lidí, je nutné, aby se noví uživatelé ve světě Internetu mohli rychle zorientovat. Velké množství webových prezentací jim ale vůbec tento vstup do nového prostředí neusnadňuje. V souvislosti s nastupujícím trendem orientace na zákazníka však některé firmy pochopily, že webové aplikace a stránky jsou tvořeny především pro zákazníka a snaží se upravit životní cyklus jejich vývoje, aby výstupem byla aplikace, kterou bude moci zákazník snadno ovládat a bude ji s radostí opakovaně využívat. Životní cyklus vývoje aplikace bývá kvůli tomu doplněn o metody použitelnosti a o testování použitelnosti.

1.2. Cíle práce

Cílem této práce je představení metod testování použitelnosti webových aplikací a jejich zařazení do jednotlivých fází životního cyklu vývoje aplikace. Struktura informací o jednotlivých metodách uvedená v této práci by měla pomoci při rozhodování o tom, kterou metodu využít pro testování použitelnosti v konkrétních případech. Vedlejším cílem této práce je upozornění na nedostatečný důraz kladený na použitelnost při vývoji aplikací.

1.3. Přístup ke zpracování práce

Přístup ke zpracování této práce je podřízen cíli práce tj. analýze metod testování použitelnosti webových aplikací. Samotný text práce (mimo úvodu a závěru) se skládá ze tří kapitol. Smyslem první kapitoly (Úvod do použitelnosti) je představení použitelnosti jako oboru zkoumajícího interakci mezi nástroji a jejich uživateli, určení významu použitelnosti a vymezení specifík použitelnosti webových aplikací. Tato kapitola poskytuje teoretické zázemí pro zbylé dvě kapitoly a vychází z názorů uznávaných odborníků v oblasti použitelnosti.

Druhá kapitola (Použitelnost a životní cyklus vývoje aplikace) se zaměřuje na představení nejčastěji užívaných modelů životního cyklu vývoje aplikace a na postavení použitelnosti v těchto modelech. Mimo jiné je zde poukázáno na nedostatečnou implementaci principů použitelnosti do vývoje aplikací.

Třetí kapitola (Metody testování použitelnost), věnovaná analýze metod testování použitelnosti webových aplikací, tvoří jádro této práce a je vystavěna na základech položených první a druhou kapitolou. Snaží se představit existující metody testování použitelnosti ve strukturované formě, která umožní rychlé zjištění základních informací o metodách, snadné porovnání jednotlivých metod a výběr vhodných metod pro provedení samotného testování. Podklady pro tuto analýzu pochází z velkého množství kvalitních internetových zdrojů, které by měly zaručovat vysokou vypovídací hodnotu.

2. Úvod do použitelnosti

2.1. Definice použitelnosti

Použitelnost = jednoduchost použití. Touto jednoduchou rovností je možné v obecné rovině definovat použitelnost. Jedná se o vlastnost vztahu mezi nástrojem (systémem) a jeho uživatelem, která udává míru, jak dobře je nástroj navržen pro efektivní a jednoduchou práci. Tuto obecnou definici je možné poté aplikovat na jednotlivé nástroje, kterými se použitelnost zabývá např. použitelnost telefonu vystihuje kvalitu návrhu telefonu, která spočívá v možnosti snadného splnění cíle tj. v uskutečnění telefonního hovoru.

Za největší autoritu v oboru použitelnosti lze považovat Jakoba Nielsena¹. Tento „král použitelnosti“, jak bývá občas nazýván, definuje použitelnost následovně:

„Použitelnost je atribut kvality, který určuje, jak jsou rozhraní jednoduchá na použití. Slovo použitelnost označuje také metody užívané pro zdokonalení jednoduchosti použití během výrobního procesu.“ [1]

Z dalších možných definic lze jmenovat:

„Vlastnost systému, programu, dokumentu nebo zařízení, která zajišťuje, že bude jednoduše pochopitelný a pohodlně použitelný uživatelem.“ [2]

„Efektivita, efektivnost a uspokojení, s kterými uživatelé mohou dosáhnout určených cílů v daném prostředí. Synonymem je jednoduchost použití.“ [3]

Vzhledem k tomu, že tato práce je zaměřena na použitelnost webových aplikací, je nutné ji definovat. Za jednu z možných definic lze považovat:

„(Použitelnost webových stránek) odkazuje na úroveň nebo stupeň přívětivosti webové stránky pro uživatele a na to, jak efektivní, efektní, tolerantní k chybám a jednoduchá na naučení webová stránka je.“ [4]

¹ <http://www.useit.com/jakob/>

Tato definice vychází z výše uvedených definic použitelnosti, které se zabývaly použitelností v obecném slova smyslu a zaměřuje se pouze na oblast webových stránek. Od této definice se odvíjí celé snažení v oblasti použitelnosti webových stránek, jelikož stanovuje cíle, kterých by mělo zavedení principů použitelnosti dosáhnout. V kontextu této práce určuje charakteristiky, jejichž dosažení je zkoumáno pomocí metod testování použitelnosti.

2.2. Složky použitelnosti

Použitelnost je velice komplexním oborem, který v sobě obsahuje prvky psychologie, sociologie, statistiky, softwarového inženýrství, informační architektury, testování kvality, grafického designu, marketingu a mnoha dalších. Není možné ovšem zkoumat použitelnost čistě z pohledu jednoho oboru, jelikož zde existují velmi silné vazby. Příkladem může být vzhled webové stránky, jehož působení na uživatele je ovlivněno mimo jiné informační architekturou, grafickým designem či psychologickými aspekty. Díky tomu je nutné nahlížet na použitelnost z několika různých úhlů, které reflektují tuto multioborovost a vzájemnou provázanost, přičemž tyto úhly pohledu jsou označovány jako složky použitelnosti.

J. Nielsen tvrdí [1], že použitelnost se skládá z pěti složek:

- naučitelnost (learnability);
- efektivita (efficiency);
- zapamatovatelnost (memorability);
- chybovost (errors);
- uspokojení (satisfaction).

Tyto složky podrobně rozebírá Tristan Louis¹ na svém blogu [5].

2.2.1. Naučitelnost

Naučitelnost je klíčovým prvkem použitelnosti. Určuje, jak je pro uživatele jednoduché vykonat základní úkony, jestliže se setkává se systémem poprvé. Koncept naučitelnosti je možné rozdělit na pět, vzájemně silně provázaných, částí:

¹ <http://www.tnl.net/who/>

- familiarita (familiarity);
- konzistence (consistency);
- zobecnitelnost (generalizability);
- předvídatelnost (predictability);
- jednoduchost (simplicity).

Familiarita je postavena na předpokladu, že uživatelé očekávají od určitého druhu systému určitý vzhled a určité chování. Nejsou-li tyto jejich očekávání naplněna, mají uživatelé problémy s ovládáním systému a raději budou hledat jiný systém, který bude odpovídat jejich představám. Příkladem může být umístění navigačního menu na webových stránkách do spodní části obrazovky. Uživatelé jsou zvyklí na menu v horní či levé části obrazovky a jakmile ho tam nenajdou, myslí si, že menu neexistuje a okamžitě opouští stránky.

Uživatelé také očekávají, že systém bude konzistentní tj. bude se v podobných situacích chovat podobně. Za nekonzistenci lze považovat používání různých označení pro tlačítka v podobných dialogových oknech, jako například v jednom okně používat „OK“ a „Zrušit“ a ve druhém „Ano“ a „Storno“.

Zobecnitelnost je koncept velice blízký familiaritě. Cílem je vytvořit systém, který bude vypadat podobně jako systémy sloužící ke stejnému účelu, a proto nebude pro uživatele těžké začít používat tento systém, i když se s ním předtím nikdy nesetkali. Jestliže například uživatel uvidí na webové stránce pole označené „Vyhledat“, automaticky si uvědomí, že toto pole slouží pro vyhledávání na daném webu. Kdyby toto pole bylo označeno jako „Podrobně prozkoumej web“ nebo by nebylo označeno vůbec, uživatele vůbec nenapadne, že se jedná o vyhledávací pole.

Předvídatelnost je taková vlastnost systému, která zajišťuje, že uživatel bude vědět v každém okamžiku svého pobytu v systému, co se stane, když provede určitou akci. Jestliže například uživatel objednává pomocí elektronického obchodu zboží, očekává, že po provedení posledního kroku objednávky mu bude poskytnut souhrnný přehled, ve kterém bude uvedeno množství a ceny za zboží. Nezobrazí-li se mu tento přehled, má pocit, že k žádné objednávce nedošlo.

Poslední částí naučitelnosti je jednoduchost. Čím jednodušší je použito rozhraní pro práci se systémem, tím snadněji se se systémem pracuje, protože noví nebo neznalí uživatelé nejsou zmateni velkým množstvím možností. Příkladem může být používání zkratk „RSS“ nebo „XML“ pro označení souborů, které obsahují přehled novinek na webových stránkách, jelikož ne každý uživatel má ponětí o tom, co tyto zkratky znamenají a co se stane po kliknutí na ně.

2.2.2. Efektivita

Jakmile je uživatel schopen orientovat se v systému a provádět základní úkony, získává na důležitosti další složka použitelnosti – efektivita. Efektivitou je myšlena rychlost, s jakou je uživatel schopen plnit úkoly, poté co je seznámen se základním rozhraním systému. Efektivita práce se systémem je ovlivněna:

- kvalitou chybových zpráv;
- přítomností zpětné vazby pro uživatele;
- dodržováním zvyklostí;
- konzistencí chování systému;
- kvalitou navigace.

Pozitivní vliv na efektivitu má zavedení zkratk a zástupců pro často opakované činnosti např. použití klávesových zkratk CTRL-C pro kopírování a CTRL-V pro vložení.

2.2.3. Zapamatovatelnost

Zapamatovatelnost je dalším krokem, který je nutné učinit na cestě k dokonalé použitelnosti. Určuje, jak jednoduché je pro uživatele ovládat systém, jestliže se systémem delší dobu nepracoval. Zapamatovatelnost je důležitým aspektem, protože existuje pouze málo systémů, se kterými se uživatelé setkávají každodenně. Důležitým prvkem, který podporuje zapamatovatelnost, je využívání intuitivních grafických symbolů. Naopak, jestliže se systém chová nekonzistentně, uživatel nemá jistotu, co způsobí jeho činnost a raději systém přestane používat.

2.2.4. Chybovost

Chybovost poukazuje na množství chyb, které v systému vznikají, na jejich závažnost a na jednoduchost, s jakou je možné opustit chybový stav. Chybovost má jako jediná charakteristika použitelnosti kromě dopadu na spokojenost uživatele také dopad na prostředky

uživatele – může dojít ke zničení či poškození uživatelských nástrojů či dat, a proto je minimalizace míry chybovosti jedním z nejdůležitějších úkolů při návrhu systému.

Vzhledem k tomu, že žádný systém není bezchybný, je nutné do systému implementovat mechanismy, které činí chybové stavy pro uživatele snadno rozpoznatelné a nabízí jednoduchý způsob, jak chybový stav opustit. Jedním z těchto mechanismů jsou chybové zprávy. Jejich správné použití umožňuje uživateli identifikovat chybový stav, zjistit příčinu vzniku chyby a v neposlední řadě nalézt způsob, jak chybový stav opustit. Základní struktura chybové zprávy je následující:

- text „chyba“ nebo ikona znázorňující vznik chyby;
- jednoduchý popis chyby;
- způsob, jakým je možné chybový stav opustit.

2.2.5. Uspokojení

Uspokojení uživatele z používání systému je poslední důležitou charakteristikou použitelnosti. Vyjadřuje míru spokojenosti uživatele, která doprovází jeho interakci se systémem. Splnění tohoto konceptu je završením celého snažení o dokonalou použitelnost. Jestliže systém splňuje předchozí čtyři charakteristiky použitelnosti, je velice pravděpodobné, že bude přinášet uživateli i uspokojení z jeho používání.

2.3. Význam použitelnosti

Význam použitelnosti je obrovský, ačkoliv byla dlouhou dobu podceňována nebo o ní tvůrci systémů vůbec nevěděli. Důvodem pro přehlížení použitelnosti byly především dodatečné náklady, které jsou spojeny se zavedením principů použitelnosti do tvorby nových systémů. Tento důvod je však naprosto neopodstatněný, jelikož výzkumné studie ukazují pravý opak tj. systémy vytvářené bez aplikace principů použitelnosti jsou ve konečném důsledku mnohem dražší než ty, kde byl kladen na použitelnost patřičný důraz. Podle zprávy IBM [6] každý dolar investovaný do použitelnosti se v budoucnu vrátí jako 10 – 100 dolarů. Tyto čísla vychází z průzkumů, které zjistily, že náklady spojené s řešením problémů s použitelností ve fázi implementace resp. po vypuštění systému do světa jsou desetinásobné resp. stonásobné oproti investici do použitelnosti ve fázi návrhu systému.

Hlavním přínosem použitelnosti ovšem nejsou úspory nákladů, ale dosažení uspokojení uživatelů z používání systému, jelikož úspory nákladů při vytváření systému jsou pouze jednorázové, kdežto jsou-li uživatelé spokojeni se systémem, budou rádi opakovaně systém používat a navíc o něm budou referovat dalším lidem, kteří se mohou stát novými uživateli. Tito noví uživatelé poté budou kontaktovat další možné uživatele, množina uživatelů se bude stále zvětšovat a firmě, která systém vyvinula, se budou zvyšovat zisky, což je důvod, proč většina firem existuje. Dobrá použitelnost systému má tedy dlouhodobě pozitivní význam pro uživatele i pro tvůrce systému.

2.4. Použitelnost webových aplikací

Webové aplikace jsou jedním z možných systémů, kterými se zabývá obor použitelnost. Ačkoliv pro použitelnost webových aplikací platí veškeré výše uvedené charakteristiky, jsou zde i určitá specifika. Nejvýznamnější rozdíl oproti většině ostatních systémů tkví v odlišném postavení použitelnosti v procesu uživatelova rozhodování o použití či nepoužití systému.

Tento rozdíl lze ilustrovat na následujícím příkladu. Jestliže se rozhoduji o tom, jaké koupit auto, nejdříve si zjistím jeho kvantitativní charakteristiky – značka, typ karosérie, typ motoru, výkon, typ paliva – a až posléze zkoumám jeho kvalitativní charakteristiky zahrnující i použitelnost. U webových aplikací je přístup opačný. Jestliže se nemůžu na stránce orientovat, nemůžu rychle najít to, co potřebuji, odcházím, aniž bych se dále zajímal o nabízené funkce a informace. Ve světě Internetu je tento první dojem velice důležitý, protože není nic snazšího, než webovou stránku zavřít a otevřít stránku konkurence, neboť konkurence je jen „o klik vedle“.

Aby byly webové aplikace použitelné a neodrazovaly potenciální uživatele, je vhodné dodržovat určitá doporučení. Tato doporučení se týkají celkového rozvržení stránky, grafických prvků, navigačních prvků či práce s odkazy.

3. Použitelnost a životní cyklus vývoje aplikace

Mezi informatiky koluje jeden vtíp: „Do výroby kterých tří věcí je lepší nevidět? Jsou to zákony, salámy a software.“ Jestliže však chceme proniknout do tajů testování použitelnosti webových aplikací, musíme navzdory tomuto vtípu nejdříve pochopit, jakým způsobem samotné aplikace vznikají a jaké postavení má použitelnost v tomto procesu tvorby aplikací.

3.1. *Modely životního cyklu vývoje aplikace*

Vznik každé aplikace je jedinečným procesem, jelikož na tvorbě různých aplikací pracuje různý počet lidí s různými znalostmi a zkušenostmi, liší se použité metody a nástroje i celkový přístup k vývoji. Mezi jednotlivými procesy lze ovšem nalézt určité podobnosti, na jejichž základě začaly vznikat modely životního cyklu vývoje aplikace. Tyto modely se snaží abstrahovat od konkrétních činností a představit obecná schémata vývoje aplikací. Dopady použití konkrétních modelů životního cyklu při vývoji jsou posléze patrné na výsledné aplikaci, přičemž nelze většinou dopředu přesně určit, který model je pro danou aplikaci 100% správný.

Podle [7 str. 21] lze rozlišit čtyři základní modely, přičemž jiné modely jsou pouze jejich variantami a kombinacemi. Jedná se o:

- model velkého třesku;
- model „programuj a opravuj“;
- model vodopádu;
- spirálový model.

Tyto čtyři modely lze podle [8] ještě doplnit o:

- inkrementální model;
- V-model.

Český popularizátor objektového přístupu k tvorbě software, RNDr. Ilja Kraval¹, přidává ještě tzv. metodu tunelu. Této metodě věnuje převážnou část své průběžně aktualizované online knihy „Návrh IS pomocí OOP, UML a vzorů“ [9].

Detailnější popis výše uvedených modelů bude postupně uveden v závislosti na míře složitosti jednotlivých modelů od nejjednodušších (model velkého třesku, metoda tunelu, model „programuj a opravuj“) přes středně složitě (model vodopádu, V-model) až po nejsložitější (inkrementální model, spirálový model), přičemž dané charakteristiky modelů vychází z výše uvedených pramenů [7], [8], [9].

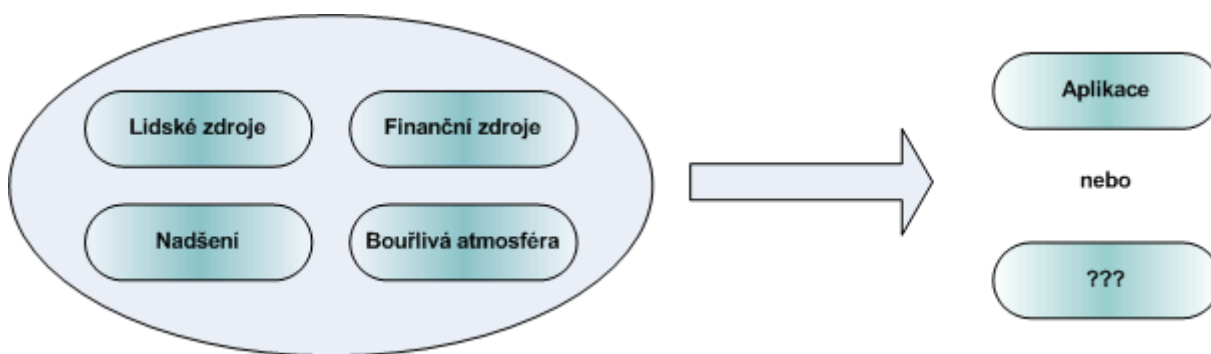
3.1.1. Model velkého třesku

Tento model, jak již název napovídá, má souvislost s jednou z teorií vzniku vesmíru s tzv. „velkým třeskem“. Základem této teorie je myšlenka, že veškerá hmota byla před miliardami let natěsnána v jednom nekonečně malém a nekonečně hmotném bodě, přičemž při velkém třesku tento bod explodoval a vzniklá hmota se začala rozpínat a utvářet vesmír.

Velký třesk jako model životního cyklu vývoje aplikace je založen na podobné myšlence. Podobně jako u vesmírné teorie zde dochází k nakumulování hmoty a energie, což je v tomto případě představováno nashromážděním lidských a finančních zdrojů a nadšení a bouřlivé atmosféry na pracovišti, přičemž vzájemnou neuspořádanou interakcí jednotlivých složek (explozí) dochází k procesu tvorby aplikace.

Jak již z výše uvedeného vyplývá, jedná se o velice chaotický a zároveň také velice jednoduchý model (viz obr. 3.1). Vzhledem k tomu, že činnosti jako např. plánování, organizování a rozvrhování prací jsou v tomto modelu téměř neupotřebitelné a veškeré úsilí je věnováno samotnému vývoji aplikace, je možné tento model použít pouze u specifických případů. Model je vhodné využít tehdy, existují-li velice nejasné vstupní požadavky na aplikaci a zároveň také neexistuje pevné datum ukončení vývoje aplikace. Nevýhodou však je vysoká nejistota zdárného dokončení aplikace vzhledem k neexistenci jakékoliv organizace práce.

¹<http://www.objects.cz>



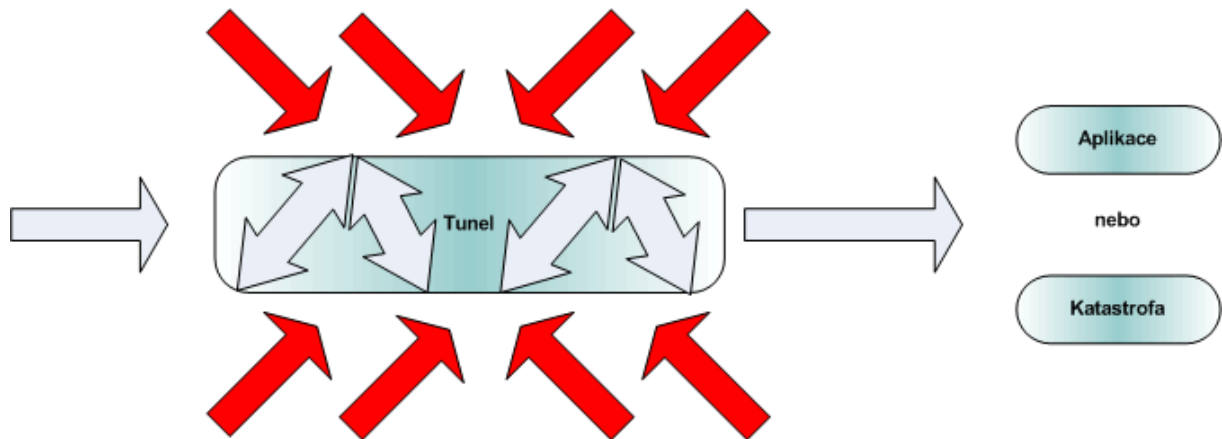
Obr. 3.1: Schéma modelu velkého třesku

3.1.2. Metoda tunelu

Tento model životního cyklu vývoje aplikace patří k ukázkovým příkladům toho, jak by se aplikace vyvíjet neměly a proč by měl být pro vývoj aplikací použit jiný přístup. I. Kraval staví jako základ tohoto jiného přístupu tyto čtyři pilíře [9 str. 9]:

- použití různých úrovní abstrakce v různých etapách vývoje aplikace;
- objektový přístup;
- modelování v jazyce UML;
- použití návrhových vzorů.

Metoda tunelu jako model životního cyklu je analogií k průchodu temným klikatým tunelem. Na počátku vývoje aplikace se obrazně vstupuje do temného tunelu vývoje, přičemž není jasný směr, kam se bude dále projekt ubírat. Další vývoj je vyjádřen tápáním v tunelu a narážením do stěn, které určují, že tudy cesta nepovede (viz obr. 3.2). Cílem tápání je spatření světla na konci tunelu znamenajícího úspěch tj. zdárné dokončení aplikace. I. Kraval k tomu dodává: „Bohužel velmi často nastane případ, kdy se nadějně světýlko na konci tunelu promění ve světla protijedoucího vlaku a celý projekt skončí katastrofickým scénářem.“ [9 str. 2]



Obr. 3.2: Schéma metody tunely

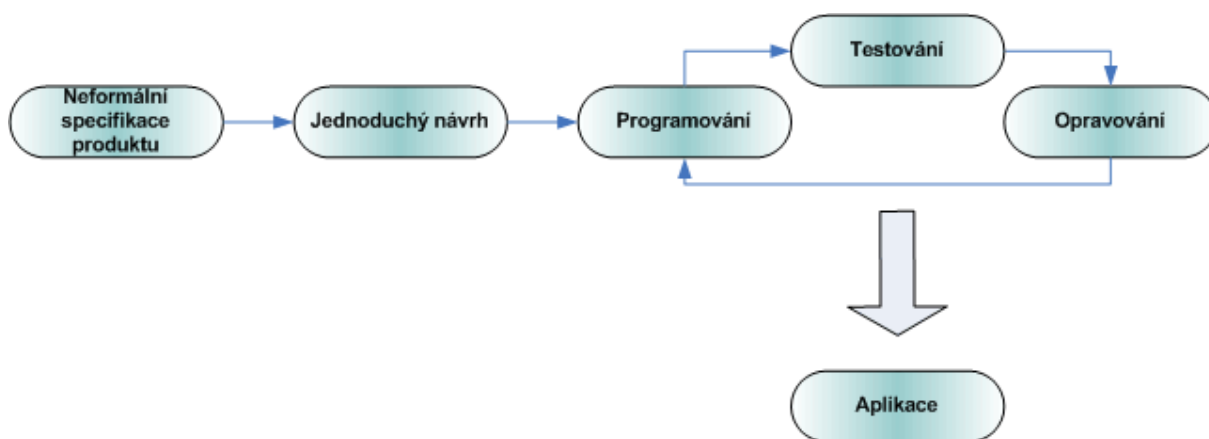
Použití metody tunelu má téměř vždy velké negativní důsledky, přičemž z těch nejdůležitějších lze jmenovat [9 str. 2-5]:

- zvýšené náklady;
- pozdní dodávka systému;
- naprostý nesoulad mezi funkcionalitou výsledné aplikace a úvodními požadavky;
- neexistence jednotné koncepce vývoje a s tím související nemožnost opakování dosažených výsledků;
- nemožnost odhadnout další kroky vývoje v jakékoliv fázi projektu.

3.1.3. Model „programuj a opravuj“

Model „programuj a opravuj“ je vylepšenou verzí modelu velkého třesku. Oproti modelu velkého třesku je na začátku vývoje vytvořena hrubá představa výsledné aplikace a jednoduchý návrh. Další část vývoje lze poté charakterizovat jako neustálý cyklus „vytvoření-oprava“ (viz obr. 3.3), přičemž neexistuje žádný přesně definovaný cílový stav. Cyklus je přerušen a aplikace dokončena, jakmile dojde k rozhodnutí, že stávající stav je již dostatečný.

Tento model klade, podobně jako model velkého třesku, minimální důraz na plánování a dokumentaci, a proto je vhodný na jednorázové projekty jako např. prototypy aplikací nebo demoverze, přičemž umožňuje okamžité předvedení výsledků v daném čase.



Obr. 3.3: Schéma modelu „programuj a opravuj“

3.1.4. Model vodopádu

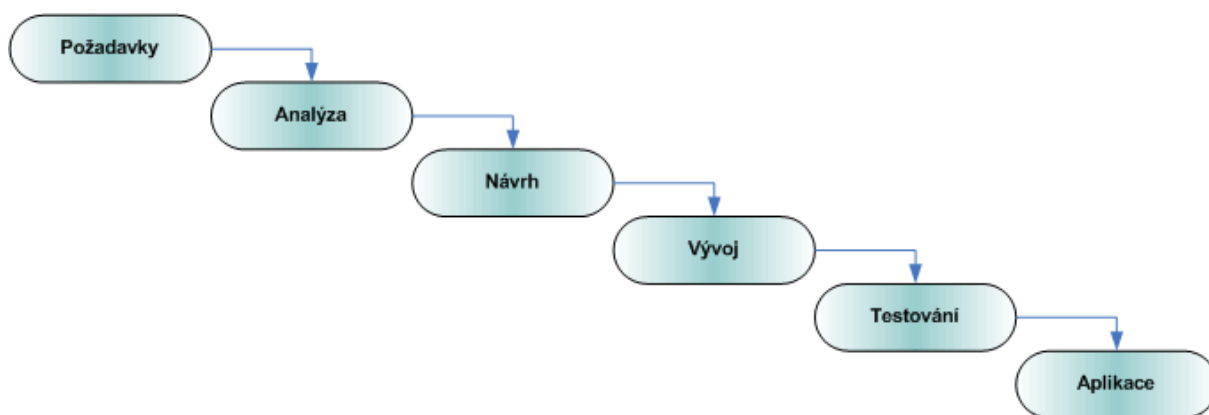
Model vodopádu je jedním z nejznámějších modelů životního cyklu vývoje aplikace. Podobně jako u některých předchozích modelů je název inspirován analogií z reálného světa - v tomto případě vodopádem. Vodopád vyjadřuje skutečnost, že vývoj aplikace je rozdělen do několika částí, přičemž další část může začít až skončí předchozí tj. dochází k přelévání z jedné části do druhé. Přechod z jedné části do druhé je možný pouze tehdy, jestliže je daná část již 100% hotová. Jsou-li pochybnosti o dokonalém splnění určité části, zůstává vývoj v dané části a nepřelévá se dál. Tento model je vhodný především u jednorázových, krátkodobých, dobře specifikovaných projektů, u kterých nedochází ke změnám v požadavcích během vývoje.

Z nastíněného principu vyplývá několik vlastností tohoto modelu [7 str. 30]:

- jednotlivé části jsou samostatné a nepřekrývají se;
- neexistuje cesta zpět do předchozí části;
- jednotlivé části mají specifikovány své vstupy a výstupy.

Model vodopádu¹ je značně rozšířeným způsobem vývoje aplikací, což dokládá i vznik několika různých variací. Snahou různých variant je především odbourání jednosměrnosti tohoto modelu. Jako příklad lze uvést inkrementální model, který je popsán dále v odstavci 3.1.6. Nejobecnější pohled na model vodopádu nabízí obrázek 3.4.

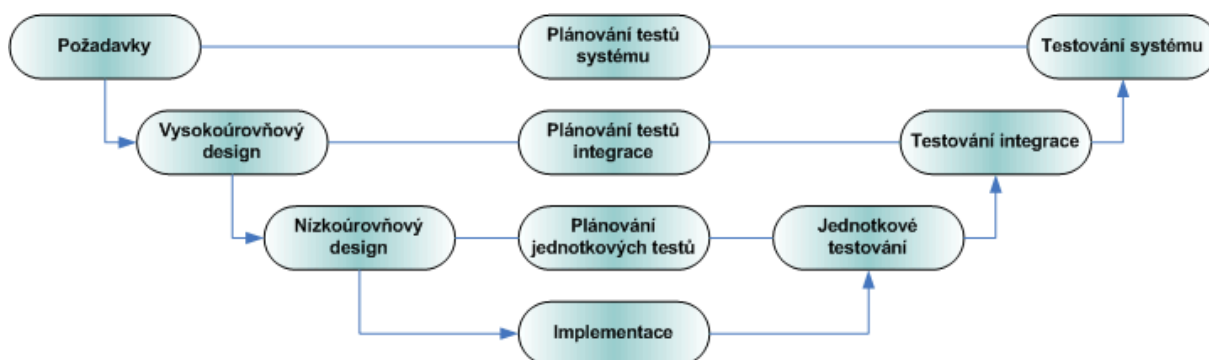
¹ Více informací o modelu vodopádu lze nalézt na http://en.wikipedia.org/wiki/Waterfall_model.



Obr. 3.4: Schéma modelu vodopádu

3.1.5. V-model

V-model je založen na stejném principu jako model vodopádu tj. jedná se o posloupnost jednotlivých kroků, které jsou samostatné a nepřekrývají se a přechod z jednoho kroku do druhého je možný pouze tehdy, jsou-li splněny všechny požadavky kladené na daný krok. Také tento model je pouze jednosměrný tj. jakmile je hotový jeden krok, přechází se na druhý a již není žádné cesty zpět.



Obr. 3.5: Schéma V-modelu

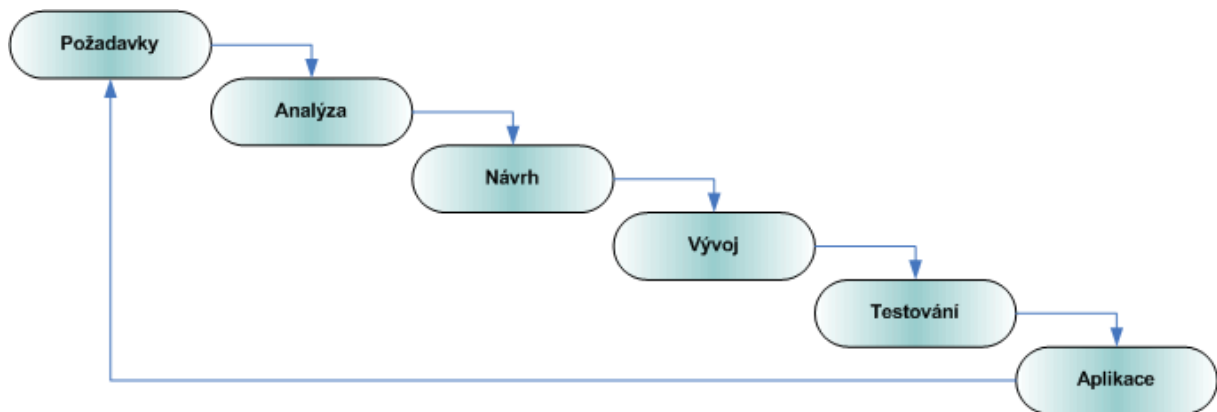
Rozdíl mezi V-modelem a modelem vodopádu spočívá v přístupu k testování (viz obr. 3.5). Zatímco u modelu vodopádu docházelo k jednorázovému testování na konci vývojového cyklu, u V-modelu se jedná o třístupňové testování se souvislostí s předimplementačními fázemi životního cyklu vývoje aplikace:

- ve fázi zpracování požadavků dochází k plánování testů systému se zaměřením na splnění funkcionality dané v požadavcích;
- ve fázi vysokoúrovňového designu dochází k plánování testů integrace se zaměřením na schopnost jednotlivých částí aplikace spolupracovat;

- ve fázi nízkoúrovňového designu dochází k plánování jednotkových testů se zaměřením na funkčnost jednotlivých funkcí a metod aplikace.

3.1.6. Inkrementální model

Inkrementální model je modifikací modelu vodopádu, přičemž eliminuje jeho největší nedostatek - jednosměrnost modelu. Inkrementální model je založen na rozdělení vývoje aplikace do menších částí, tzv. iterací, přičemž každá tato iterace prochází modelem vodopádu (viz obr. 3.6). Výhodou tohoto přístupu je dostupnost jádra funkční aplikace již v brzkém stadiu vývoje. V dalších stádiích vývoje je jádro aplikace dále rozšiřováno až do podoby závěrečné aplikace, přičemž po každé iteraci dostáváme funkční aplikaci.



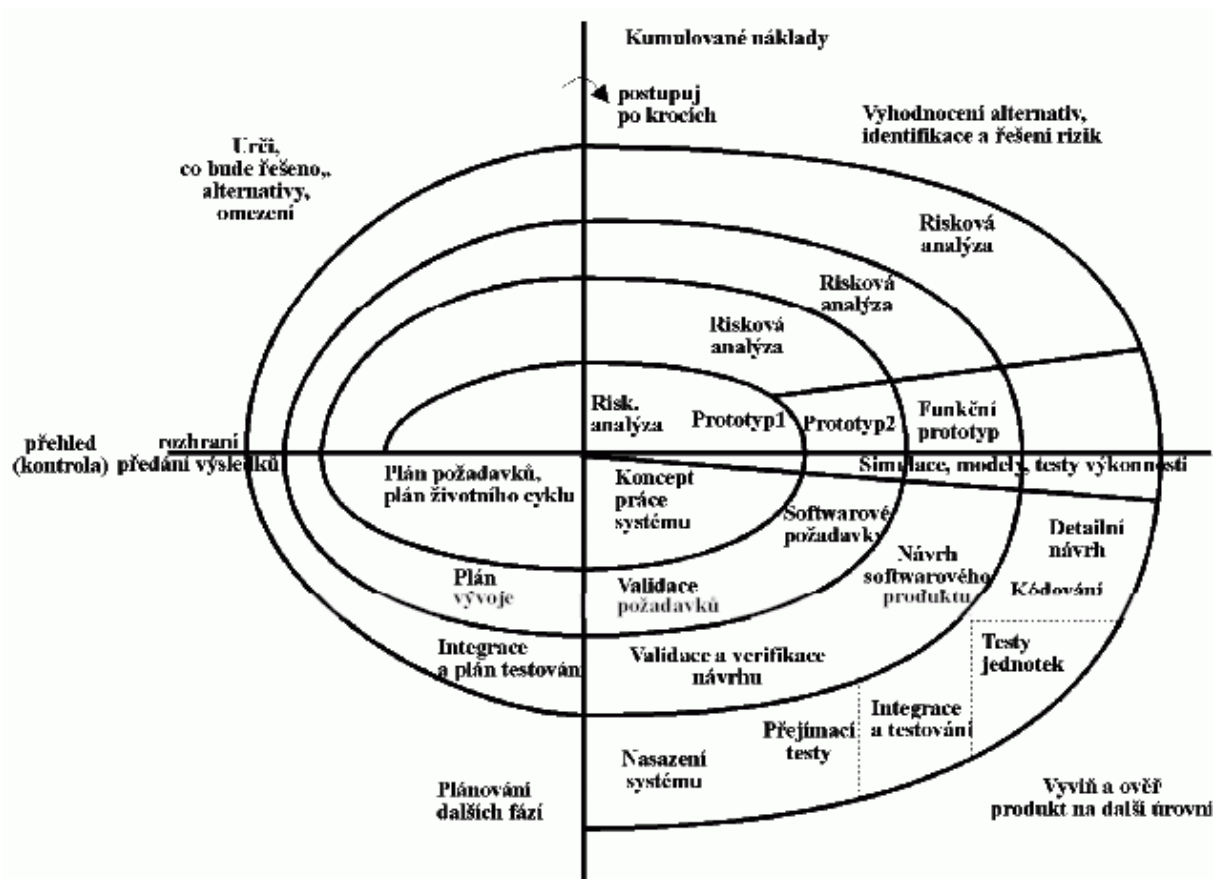
Obr. 3.6: Schéma inkrementálního modelu

3.1.7. Spirálový model

Spirálový model ční na vrcholu pyramidy dokonalosti zde uvedených modelů životního cyklu vývoje aplikace. Svoje postavení si vydobyl tím, že kombinuje pozitivní vlastnosti daných modelů a navíc k nim přidává ještě své vlastní. Historie tohoto modelu se začala psát již v roce 1986, ve kterém Barry Boehm¹ vydal dokument „A Spiral Model of Software Development and Enhancement“², v němž položil základy spirálového modelu.

¹ Profesor softwarového inženýrství na University of South California, <http://sunset.usc.edu/people/barry.html>

² <http://www.sce.carleton.ca/faculty/ajila/4106-5006/Spiral%20Model%20Boehm.pdf>



Obr. 3.7: Schéma spirálového modelu

[Zdroj: http://www.ft.muni.cz/~racek/vyuka/ananas/an_01_uvod.pdf]

Spirálový model je postaven na předpokladu, že na začátku životního cyklu vývoje aplikace nemáme veškeré informace potřebné k úspěšnému vytvoření aplikace. Tyto informace získáváme postupně v průběhu jednotlivých smyček spirály, iterací, z připomínek zákazníka. Pomocí těchto nových informací dochází ke zpřesňování požadavků a k přibližování se ke konečné podobě aplikace (viz obr. 3.7). Každý průchod spirálou obnáší šest kroků [7 str. 31]:

- určení cílů, alternativ a omezení;
- identifikace a řešení rizik;
- vyhodnocení alternativ;
- vývoj a testování aktuální iterace;
- plánování další iterace;
- rozhodování o postupu do další iterace.

3.2. Využití principů použitelnosti při vývoji aplikace

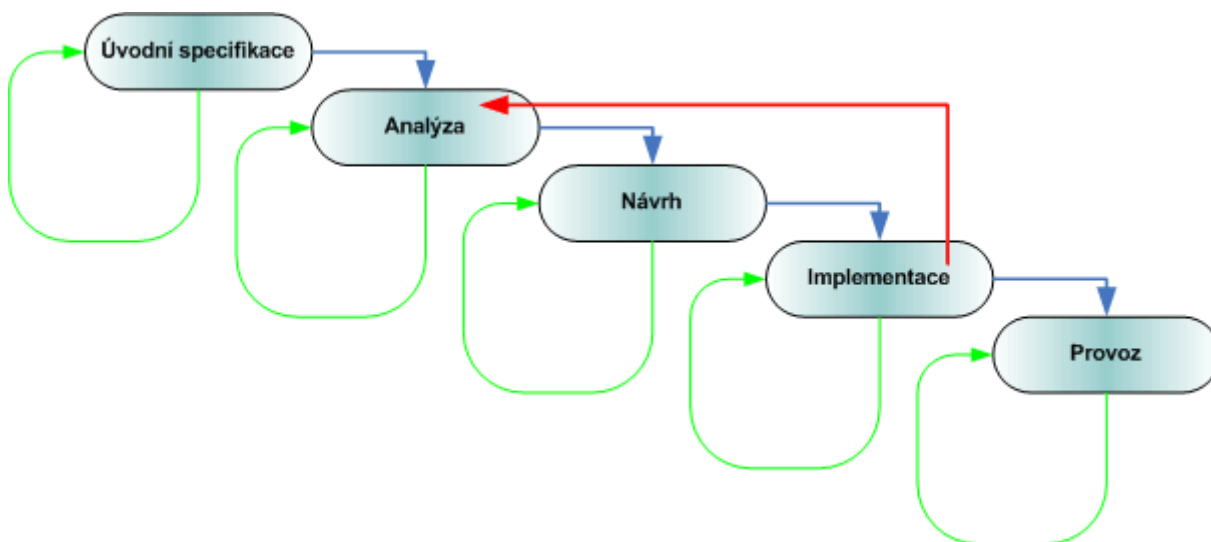
Proces vývoje aplikace lze v nejobecnější rovině rozložit na dvě paralelně existující linie:

- vývoj funkcionality;
- vývoj uživatelského rozhraní.

Většina současných přístupů k tvorbě aplikací, včetně zde uvedených modelů, klade důraz především na první linii – na vývoj funkcionality aplikace, aniž by si uvědomovala, že pro uživatele aplikace je mnohem důležitější rozhraní, se kterým pracuje, než to, jak je aplikace uvnitř naprogramovaná. Kvůli tomu je nutné implementovat do procesu vývoje aplikace principy použitelnosti.

3.2.1. Kdy aplikovat použitelnost?

Aplikování použitelnosti není jednorázovou záležitostí, jedná se činnost, který se táhne jako červená nit celým životním cyklem vývoje aplikace. Půžitelnost má svoje místo při úvodní specifikaci, během implementace i po vydání aplikace. Pro potřeby této práce bude dále používán model životního cyklu vývoje aplikace s důrazem na použitelnost znázorněný na obrázku 3.8, který jsem vytvořil speciálně pro účely této práce.



Obr. 3.8: Schéma modelu vývoje aplikace s důrazem na použitelnost

Schéma znázorňuje jednotlivé fáze vývoje aplikace s naznačenými vazbami. Modré šipky znázorňují logickou návaznost jednotlivých fází, červená šipka naznačuje možnost iterativního vývoje a zelené šipky označují cyklickou aplikaci principů použitelnosti v dané

fázi. Přesun na následující fázi je možný po splnění všech potřebných požadavků z hlediska technického i z hlediska použitelnosti.

Cílem úvodní specifikace je definování základních požadavků na aplikaci, určení zdrojů a stanovení harmonogramu. V navazující analýze dochází ke zpřesnění požadavků na aplikaci a vytváření základních modelů budoucí aplikace. Zatímco analýza určuje, co se má udělat, ve fázi návrhu je poté definováno, jak se to má udělat. Výstupem návrhu je podrobný podklad pro vytvoření aplikace tj. pro samotné naprogramování, které probíhá ve fázi implementace. Poté, co je aplikace dokončena, dostává se do poslední fáze, do fáze provozu.

3.2.2. Testování použitelnosti

S testováním se můžeme v různých formách setkat téměř v každém oboru lidské činnosti od výroby automobilů až po poskytování internetových služeb. V závislosti na oboru se různé přístupy k testování, jeho význam a použité metody. Cíl testování je však stále stejný: zkontrolovat, zda výstup splňuje kvantitativní i kvalitativní požadavky na něj kladené a zda se nevyskytují žádné chyby.

Jak již zde bylo zmíněno, investice do použitelnosti se vyplácí, jelikož se předchází možným problémům a dochází k významné úspoře budoucích nákladů díky produkovaní kvalitního výstupu. Snaha o zlepšení použitelnosti ovšem nemusí vždy vést k požadovanému cíli. K tomu, abychom se přesvědčili, jestli opravdu došlo ke zlepšení, je nutné použitelnost testovat.

Chceme-li testovat použitelnost, musíme nejdříve vědět kdy a jak testovat. Pohlédneme-li na modely životního cyklu vývoje aplikace, které byly uvedeny v podkapitole 3.1, mohli bychom snadno dojít k mylnému závěru, že někdy není potřeba testovat vůbec a jindy pouze jednou na konci vývoje aplikace před uvedením na trh. Problémem těchto modelů je však skutečnost, že testování je zde chápáno jako testování funkcionality, které nelze snadno zaměnit s testováním použitelnosti. Testování použitelnosti má totiž výrazně jiný charakter, který vychází z postavení použitelnosti v životním cyklu vývoje aplikace, které je prezentováno v odstavci 3.2.1. Na otázky „kdy a jak testovat?“ se snaží odpovědět kapitola 4, která přináší přehled metod pro testování použitelnosti a tvoří jádro této práce.

4. Metody testování použitelnosti

Nyní, když již víme, co to použitelnost je a jaké má postavení v životním cyklu vývoje aplikace, můžeme přistoupit k představení metod, které jsou využívány při testování použitelnosti. Metody, které zde budou dále uvedeny nejsou určeny pouze pro testování použitelnosti webových aplikací, ale lze je použít i pro testování desktopových aplikací či jakýchkoliv jiných systémů.

Vzhledem k tomu, že použitelnost není přesně specifikovaný obor, neexistuje také žádný závazný seznam metod testování použitelnosti. Každá firma či organizace může používat různé metody, modifikovat stávající metody či vymýšlet naprosto nové metody. Z tohoto důvodu je nemožné zachytit všechny existující metody, a proto zde budou představeny pouze ty metody, které jsou využívány nejčastěji či se velice často stávají základem pro tvorbu nových modifikací.

Ze stejného důvodu se také liší charakteristiky jedné a té samé metody u různých firem, organizací či podle názorů různých expertů. Zde uvedené charakteristiky jednotlivých metod vychází z analýzy velkého množství zdrojů, tudíž mají vysokou vypovídací hodnotu. Jakékoliv jiné interpretace jsou ovšem možné a nelze automaticky tvrdit, že jsou chybné.

4.1. Sledované charakteristiky

U všech metod byla sledována řada charakteristik. Výběr charakteristik byl inspirován návrhem systému PRS (Practical Review System), který prezentoval John S. Rhodes¹ ve svém článku [10] s názvem „A Proposal for Evaluating Usability Testing Methods: The Practical Review System (PRS)“. Systém PRS nebyl bohužel nikdy dokončen, jeho vývoj se zastavil ve fázi návrhu. Jeho cílem bylo vytvoření databáze metod testování použitelnosti. Díky využití některých charakteristik z PRS lze tuto práci považovat za naplnění původní myšlenky J. S. Rhodese.

Jednotlivé metody jsou prezentovány v přehledné tabulkové podobě (viz tab. 4.1), která umožňuje snadné a rychlé získání základních informací o metodách a vzájemné porovnání

¹ <http://webword.com/aboutjohn.html>

těchto metod. Metody jsou rozděleny do dvou skupin podle nutnosti mít přístup ke koncovým uživatelům pro zdárné provedení testování.

Alternativní anglické názvy	Další možné anglické názvy metody
České názvy	Používané české názvy pro metodu
Příbuzné metody	Metody, které mají podobné charakteristiky
Fáze životního cyklu	Fáze životního cyklu vývoje webové aplikace, ve kterých je nejvhodnější metodu použít
Cíle	Cíle, kterých se metoda snaží dosáhnout
Výhody	Výhody použití metody
Nevýhody	Nevýhody použití metody
Účastníci	Přehled účastníků nutných k provedení jednoho testování (účastníci uvedení v závorce jsou nepovinní)
Interakce mezi účastníky	Úroveň vzájemné interakce mezi jednotlivými účastníky testování – žádná, nízká, vysoká
Potřebné vybavení či prostředky	Speciální vybavení či prostředky pro provádění testování (za standardní vybavení jsou považovány papíry, psací potřeby, počítač, prototypy testované aplikace či hotová testovaná aplikace)
Vzdálené testování	Možnost vzdáleného testování tj. testování mimo společnost, která vyvíjí testovanou webovou aplikaci – ano (jedná se o metodu vzdáleného testování), možné (nejedná se o metodu vzdáleného testování, ale lze ji provádět vzdáleně), ne (nejedná se o metodu vzdáleného testování a nelze ji provádět vzdáleně)
Použité zdroje	Zdroje využité ke zpracování přehledu metody
Další informace	Odkazy na webové stránky (mimo použité zdroje) obsahující další doplňující informace k metodě např. samotný způsob provádění testování

Tab. 3.1: Šablona použitá pro popis metod

4.2. Testování s účastí koncových uživatelů aplikace

4.2.1. Card Sorting

Alternativní anglické názvy	/
České názvy	Třídění karet
Příbuzné metody	Affinity Diagrams
Fáze životního cyklu	Návrh
Cíle	Porovnání vývojáři navržené struktury menu a struktury menu navržené uživateli
Výhody	<ul style="list-style-type: none">• jednoduchá a levná metoda• získání uživatelského názoru na strukturu menu
Nevýhody	<ul style="list-style-type: none">• možný velký rozptyl výsledků• náročná analýza• zachycení pouze povrchových vlastností
Účastníci	Uživatel
Interakce mezi účastníky	/
Potřebné vybavení či prostředky	Fotoaparát
Vzdálené testování	Možné
Použité zdroje	[11], [12]
Další informace	Useit.com ¹ Boxes and Arrows ² Usability.gov ³

4.2.2. Affinity Diagrams

Alternativní anglické názvy	/
České názvy	Diagramy příbuznosti, afinitní diagramy, diagramy afinity
Příbuzné metody	Card Sorting
Fáze životního cyklu	Návrh
Cíle	Porovnání vývojáři navržené struktury menu a struktury menu navržené uživateli
Výhody	<ul style="list-style-type: none">• výstup je vytvořen po shodě většího počtu uživatelů• získání uživatelského názoru na strukturu menu

¹ <http://www.useit.com/alertbox/20040719.html>

² http://www.boxesandarrows.com/view/analyzing_card_sort_results_with_a_spreadsheet_template

³ <http://www.usability.gov/design/cardsort.html>

Nevýhody	<ul style="list-style-type: none"> • nutnost zapojení většího počtu účastníků • testování se může díky diskuzi velice protáhnout
Účastníci	Uživatelé
Interakce mezi účastníky	Vysoká
Potřebné vybavení či prostředky	Fotoaparát
Vzdálené testování	Ne
Použité zdroje	[13]
Další informace	Usability Method Toolbox ¹ Vanderbilt University ²

4.2.3. Pluralistic Walkthroughs

Alternativní anglické názvy	Group Walkthroughs
České názvy	Skupinové průchody
Příbuzné metody	Cognitive Walkthroughs
Fáze životního cyklu	Návrh
Cíle	Odhalení chyb v použitelnosti a nalezení možných řešení
Výhody	<ul style="list-style-type: none"> • více zúčastněných najde více chyb • účastníci jsou z více oborů tj. různé pohledy na webovou aplikaci • interakce mezi účastníky umožňuje nalezení většího množství chyb
Nevýhody	<ul style="list-style-type: none"> • nutnost zapojení většího počtu účastníků • testování se může díky diskuzi nad jednotlivými kroky velice protáhnout
Účastníci	Expert na použitelnost, návrháři, uživatelé
Interakce mezi účastníky	Vysoká
Potřebné vybavení či prostředky	Zařízení pro záznam videa
Vzdálené testování	Ne
Použité zdroje	[14], [15], [16]
Další informace	University of Texas ³

¹ <http://jthom.best.vwh.net/usability/affinity.htm>

² <http://mot.vuse.vanderbilt.edu/mt322/Affinity.htm>

³ <http://www.cs.utexas.edu/users/almstrum/cs370/elvisino/usaEval.html>

4.2.4. Think Aloud Protocol

Alternativní anglické názvy	/
České názvy	Myšlení nahlas
Příbuzné metody	Co-Discovery Method, Question Asking Protocol
Fáze životního cyklu	Návrh, implementace
Cíle	Porozumění uživatelskému modelu myšlení
Výhody	<ul style="list-style-type: none"> • rychlá a kvalitní zpětná vazba od uživatele • získávání dat z více zdrojů (sledování uživatele, poslouchání uživatele) • pozorovatel má možnost objasnit situaci, jestliže se uživatel dostane do potíží • testování může být flexibilně řízeno pozorovatelem
Nevýhody	<ul style="list-style-type: none"> • myšlení nahlas zpomaluje práci s webovou aplikací a uživatel si uvědomí více věcí než při normální práci a tím pádem může objevit méně chyb, než by objevil při běžné práci • myšlení nahlas může být pro některé uživatele nepřírozené a obtěžující • myšlení nahlas během celého testování je velice vyčerpávající
Účastníci	Pozorovatel, uživatel
Interakce mezi účastníky	Nízká
Potřebné vybavení či prostředky	Zařízení pro záznam zvuku či videa
Vzdálené testování	Ne
Použité zdroje	[17], [18], [19]
Další informace	Usability Home ¹

4.2.5. Question Asking Protocol

Alternativní anglické názvy	/
České názvy	/
Příbuzné metody	Think Aloud Protocol
Fáze životního cyklu	Návrh, implementace

¹ <http://www.usabilityhome.com/ThinkAlo.htm>

Cíle	Porozumění uživatelskému modelu myšlení
Výhody	Uživatel mnohem snadněji prezentuje své pocity z užívání webové aplikace, jsou-li mu kladeny správné otázky
Nevýhody	Zachyceny pouze problémy, na které se tazatel přímo zeptá
Účastníci	Tazatel, uživatel
Interakce mezi účastníky	Vysoká
Potřebné vybavení či prostředky	Zařízení pro záznam zvuku či videa
Vzdálené testování	Ne
Použité zdroje	[20], [21]
Další informace	/

4.2.6. Evaluate Prototype

Alternativní anglické názvy	Prototype Testing
České názvy	Testování prototypů
Příbuzné metody	/
Fáze životního cyklu	Návrh, implementace
Cíle	Odhalení chyb v použitelnosti návrhu v brzkých fázích vývoje
Výhody	<ul style="list-style-type: none"> • úspora nákladů díky odhalení chyb v počátečních fázích vývoje • zjištění očekávání uživatelů a dojmů ze webové aplikace
Nevýhody	/
Účastníci	Pozorovatel, (návrhář), uživatel
Interakce mezi účastníky	Žádná (mezi pozorovatelem a uživatelem), vysoká (mezi návrhářem a uživatelem při použití papírových prototypů)
Potřebné vybavení či prostředky	Zařízení pro záznam videa
Vzdálené testování	Ne
Použité zdroje	[22]
Další informace	Usability Methods Toolbox ¹ Usability.gov ² Usability BoK ³

¹ <http://jthom.best.vwh.net/usability/prototyp.htm>

² <http://www.usability.gov/design/prototyping.html>

³ <http://www.usabilitybok.org/methods/p312>

4.2.7. Diagnostic Evaluation

Alternativní anglické názvy	/
České názvy	Diagnostické hodnocení
Příbuzné metody	Performance Measurement
Fáze životního cyklu	Návrh, implementace
Cíle	Zjištění s čím a proč mají uživatelé problémy při používání webové aplikace
Výhody	<ul style="list-style-type: none"> identifikuje závažné chyby v použitelnosti umožňuje porozumět, proč mají uživatelé problémy s používáním systému poskytuje hrubé určení efektivnosti, efektivity a uživatelského uspokojení
Nevýhody	/
Účastníci	Pozorovatel, uživatel
Interakce mezi účastníky	Žádná (jsou-li měřeny metriky), vysoká (nejsou-li měřeny metriky)
Potřebné vybavení či prostředky	Zařízení pro záznam videa
Vzdálené testování	Ne
Použité zdroje	[23]
Další informace	/

4.2.8. Self-Reporting Logs

Alternativní anglické názvy	/
České názvy	/
Příbuzné metody	Journalized Sessions
Fáze životního cyklu	Návrh, implementace
Cíle	Získání obecných dojmů a postřehů z užívání webové aplikace od velkého množství uživatelů
Výhody	Levné testování velkého množství uživatelů
Nevýhody	<ul style="list-style-type: none"> nepřítomnost pozorovatele poskytuje pouze textová data
Účastníci	Uživatel
Interakce mezi účastníky	/
Potřebné vybavení či prostředky	/
Vzdálené testování	Ano
Použité zdroje	[24]
Další informace	/

4.2.9. Journalled Sessions

Alternativní anglické názvy	/
České názvy	/
Příbuzné metody	Self-Reporting Logs
Fáze životního cyklu	Implementace
Cíle	Získání detailních informací o užívání webové aplikace od velkého množství uživatelů
Výhody	Levné testování velkého množství uživatelů
Nevýhody	Nepřítomnost pozorovatele
Účastníci	Uživatel
Interakce mezi účastníky	/
Potřebné vybavení či prostředky	Aplikace pro zaznamenávání činností uživatele
Vzdálené testování	Ano
Použité zdroje	[25]
Další informace	/

4.2.10. Screen Snapshots

Alternativní anglické názvy	Screenshots, Screen Capture
České názvy	Snímání obrazovky
Příbuzné metody	Journalled Sessions
Fáze životního cyklu	Implementace
Cíle	Poskytnutí grafických podkladů pro ostatní formy vzdáleného testování
Výhody	Levné testování velkého množství uživatelů
Nevýhody	Nepřítomnost pozorovatele
Účastníci	Uživatel
Interakce mezi účastníky	/
Potřebné vybavení či prostředky	Aplikace pro tvorbu snímků obrazovky
Vzdálené testování	Ano
Použité zdroje	[26]
Další informace	/

4.2.11. Co-Discovery Method

Alternativní anglické názvy	Co-Discovery Protocol, Constructive Interaction
České názvy	/
Příbuzné metody	Think Aloud Protocol
Fáze životního cyklu	Implementace

Cíle	Porozumění uživatelskému modelu myšlení ve skupině
Výhody	<ul style="list-style-type: none"> • uživatelé se cítí přirozeněji než u Thinking Aloud Protocolu (TAP) a díky tomu sami od sebe mluví nahlas • odhalení problémů, které by nebyly odhaleny pomocí TAP • větší množství zpětné vazby než u TAP
Nevýhody	<ul style="list-style-type: none"> • náročnější sledovat dva uživatele než jednoho • nákladnější než TAP
Účastníci	Pozorovatel, 2 uživatelé
Interakce mezi účastníky	Vysoká (mezi uživateli), nízká (mezi uživateli a pozorovatelem)
Potřebné vybavení či prostředky	Zařízení pro záznam zvuku či videa
Vzdálené testování	Ne
Použité zdroje	[27], [28]
Další informace	Usability Home ¹

4.2.12. Critical Incident Technique

Alternativní anglické názvy	Critical Incident Analysis, CIT
České názvy	Analýza kritických případů, Analýza kritických událostí
Příbuzné metody	/
Fáze životního cyklu	Implementace, provoz
Cíle	Určení událostí a stavů, které měly významný vliv na dokončení úkolu
Výhody	<ul style="list-style-type: none"> • nalezení chyb v uživatelském rozhraní, ke kterým nedochází často, ale mohou mít velký vliv na konečný výsledek činností • informace jsou sdělovány z pohledu uživatele a jeho vlastními slovy • levné zjištění obsahově bohatých informací

¹ <http://www.usabilityhome.com/Codiscov.htm>

Nevýhody	<ul style="list-style-type: none"> • zaměření na poměrně málo časté chyby, ne na komplexnější problémy • uživatelé si nemusí dobře pamatovat výjimečné chyby • nezájem uživatelů podrobně vysvětlovat okolnosti vzniku chyby
Účastníci	Tazatelé, pracovníci vývoje, uživatelé
Interakce mezi účastníky	Vysoká (při využití interview), žádná (při využití dotazníku)
Potřebné vybavení či prostředky	/
Vzdálené testování	Ne
Použité zdroje	[29], [30]
Další informace	University College Cork ¹

4.2.13. Performance Measurement

Alternativní anglické názvy	Performance Testing
České názvy	Měření výkonu, Výkonnostní testování
Příbuzné metody	Diagnostic Evaluation
Fáze životního cyklu	Návrh, implementace, provoz
Cíle	Odhalení chyb v použitelnosti a stanovení hodnot metrik
Výhody	<ul style="list-style-type: none"> • identifikuje závažné chyby v použitelnosti • poskytuje hodnoty metrik pro určení efektivnosti, efektivity a uživatelského uspokojení • nalezení problémů, které nejsou schopné odhalit kvalitativní metody • možnost využití výsledků ke stanovení cílů a milníků dalšího vývoje
Nevýhody	Zaměření pouze na kvantitativní charakteristiky
Účastníci	Pozorovatel, uživatel
Interakce mezi účastníky	Žádná
Potřebné vybavení či prostředky	Zařízení pro záznam videa
Vzdálené testování	Ne
Použité zdroje	[31], [32], [33]
Další informace	/

¹ <http://www.ucc.ie/hfrg/emmus/methods/cit.html>

4.2.14. Attitude Questionnaires

Alternativní anglické názvy	Subjective Assessment, Subjective Evaluation
České názvy	Subjektivní hodnocení
Příbuzné metody	/
Fáze životního cyklu	Návrh, implementace, provoz
Cíle	Zjištění uživatelského uspokojení z používání produktu
Výhody	<ul style="list-style-type: none"> • poskytuje přehled o kladných i záporných charakteristikách webové aplikace • zjištění uživatelské spokojenosti se systémem
Nevýhody	/
Účastníci	Uživatel
Interakce mezi účastníky	/
Potřebné vybavení či prostředky	/
Vzdálené testování	Možné
Použité zdroje	[34]
Další informace	University College Cork ¹ SUMI Questionnaire ² University of Maryland ³

4.3. Testování bez účasti koncových uživatelů aplikace

4.3.1. Standards Inspection

Alternativní anglické názvy	Guideline Checklists
České názvy	Kontrola dodržování standardů, Kontrola dodržování pravidel pro tvorbu uživatelského rozhraní
Příbuzné metody	/
Fáze životního cyklu	Návrh
Cíle	Zjištění míry shody s průmyslovými a projektovými standardy
Výhody	Zajištění dodržování všeobecně uznávaných principů tvorby uživatelského rozhraní
Nevýhody	/
Účastníci	Expert na použitelnost

¹ <http://www.ucc.ie/hfrg/resources/qfaq1.html>

² <http://sumi.ucc.ie/>

³ <http://www.lap.umd.edu/QUIS/>

Interakce mezi účastníky	/
Potřebné vybavení či prostředky	/
Vzdálené testování	Možné
Použité zdroje	[35], [36]
Další informace	ISII ¹

4.3.2. Formal Usability Inspection

Alternativní anglické názvy	/
České názvy	Formální kontrola použitelnosti
Příbuzné metody	/
Fáze životního cyklu	Návrh
Cíle	Zhodnocení uživatelského rozhraní z pohledu uživatele
Výhody	<ul style="list-style-type: none"> • konstruktivní přístup k hodnocení použitelnosti • více zúčastněných najde více chyb
Nevýhody	Jedná se pouze o snahu nahlédnout na webovou aplikaci očima uživatele, ne o plnohodnotné uživatelské testování
Účastníci	4 – 8 expertů na použitelnost
Interakce mezi účastníky	Žádná
Potřebné vybavení či prostředky	/
Vzdálené testování	Možné
Použité zdroje	[37], [38], [39]
Další informace	/

4.3.3. Consistency Inspection

Alternativní anglické názvy	/
České názvy	Kontrola konzistence
Příbuzné metody	/
Fáze životního cyklu	Návrh
Cíle	Odhalení nekonzistentních prvků webové aplikace
Výhody	<ul style="list-style-type: none"> • úspora nákladů díky odhalení chyb v počátečních fázích vývoje • více zúčastněných najde více chyb

¹ http://www.isii.com/ui_design.html

Nevýhody	<ul style="list-style-type: none"> • nutnost zapojení většího počtu účastníků • testování se může díky diskuzi nad jednotlivými kroky velice protáhnout
Účastníci	3- 4 experti na použitelnost
Interakce mezi účastníky	Vysoká
Potřebné vybavení či prostředky	Zařízení pro záznam videa
Vzdálené testování	Ne
Použité zdroje	[40], [41]
Další informace	/

4.3.4. Feature Inspection

Alternativní anglické názvy	/
České názvy	Zkoumání prvků
Příbuzné metody	/
Fáze životního cyklu	Implementace
Cíle	Zjištění, zda jsou prvky uživatelského rozhraní pochopitelné, užitečné a dostupné ve chvíli, kdy je uživatel potřebuje
Výhody	Kvalitativní přístup k hodnocení použitelnosti
Nevýhody	Nezahrnuje uživatelský pohled
Účastníci	Expert na použitelnost
Interakce mezi účastníky	/
Potřebné vybavení či prostředky	/
Vzdálené testování	Možné
Použité zdroje	[42], [43], [44]
Další informace	/

4.3.5. Cognitive Walkthroughs

Alternativní anglické názvy	/
České názvy	Kognitivní průchody
Příbuzné metody	Pluralistic Walkthroughs
Fáze životního cyklu	Návrh, implementace, provoz
Cíle	Nalezení průchodu webovou aplikací kvůli splnění určitého úkolu a identifikace slabých míst z pohledu uživatele
Výhody	Zjištění snadnosti naučitelnosti webové aplikace

Nevýhody	<ul style="list-style-type: none"> • nutnost zapojení většího počtu účastníků • testování se může díky diskusi nad jednotlivými kroky velice protáhnout
Účastníci	1 – 4 experti na použitelnost
Interakce mezi účastníky	Vysoká
Potřebné vybavení či prostředky	Zařízení pro záznam videa
Vzdálené testování	Ne
Použité zdroje	[45], [46], [47]
Další informace	Usability Home ¹

4.3.6. Expert Evaluation

Alternativní anglické názvy	Expert Review
České názvy	Expertní testování
Příbuzné metody	Heuristic Evaluation
Fáze životního cyklu	Návrh, implementace, provoz
Cíle	Odhalení pozitivních i negativních vlastností webové aplikace z hlediska použitelnosti
Výhody	<ul style="list-style-type: none"> • rychlá a levná zpětná vazba pro návrháře • určení problematických míst uživatelského rozhraní • odhalení pozitivních i negativních vlastností
Nevýhody	Nutnost zapojení většího počtu účastníků
Účastníci	5 expertů na použitelnost
Interakce mezi účastníky	Žádná
Potřebné vybavení či prostředky	/
Vzdálené testování	Možné
Použité zdroje	[48], [49]
Další informace	/

4.3.7. Heuristic Evaluation

Alternativní anglické názvy	Heuristic Review
České názvy	Heuristické testování
Příbuzné metody	Expert Evaluation
Fáze životního cyklu	Návrh, implementace, provoz
Cíle	Odhalení nedostatků v použitelnosti

¹ <http://www.usabilityhome.com/CognWalk.htm>

Výhody	<ul style="list-style-type: none"> rychlá a levná zpětná vazba pro návrháře určení problematických míst uživatelského rozhraní
Nevýhody	<ul style="list-style-type: none"> nutnost zapojení většího počtu účastníků pouze kritické zhodnocení bez určení pozitivních rysů neodhalení všech důležitých chyb
Účastníci	5 hodnotitelů se základními znalostmi použitelnosti
Interakce mezi účastníky	Žádná
Potřebné vybavení či prostředky	/
Vzdálené testování	Možné
Použité zdroje	[48], [49], [50]
Další informace	Usability BoK ¹ Usability.gov ² Usability Home ³

4.4. Grafické přehledy

Cílem této kapitoly je vizualizace některých charakteristik zde uvedených metod. Tyto grafické přehledy usnadňují prvotní orientaci v oblasti metod testování použitelnosti webových aplikací a mohou sloužit jako rozcestníky při rozhodování o tom, kterou metodu použít při testování použitelnosti.

Na prvním obrázku (viz obr. 4.1) lze nalézt přehled metod s určením, ve které fázi životního cyklu vývoje aplikace je nejvhodnější tuto metodu aplikovat. Tento obrázek poskytuje základní přehled o metodách testování použitelnosti dostupných ve fázi návrhu, implementace a provozu. Fáze *úvodní specifikace* a *analýza* nejsou do přehledu zahrnuty, jelikož se v těchto fázích neuplatňují žádné metody testování, protože neexistuje žádná webová aplikace či její prototyp, které by mohly být testovány.

Druhý obrázek (viz obr. 4.2) vystihuje vazby, které mezi jednotlivými metodami existují. Vychází z příbuznosti metod, která je uváděna v tabulce charakteristik u jednotlivých metod.

¹ <http://www.usabilitybok.org/methods/p275>

² <http://www.usability.gov/methods/heuristicEval.html>

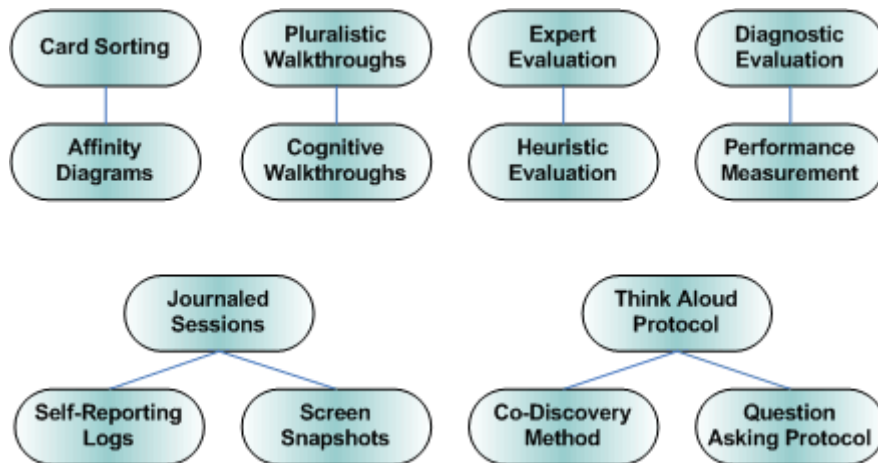
³ <http://www.usabilityhome.com/Heuristi.htm>

Na tomto obrázku jsou zobrazeny pouze ty metody, ke kterým se váží nějaké vazby. Čtenář díky tomuto schématu může rychle nalézt alternativy k metodám, které již zná nebo které si vyhledal na obrázku 4.1.

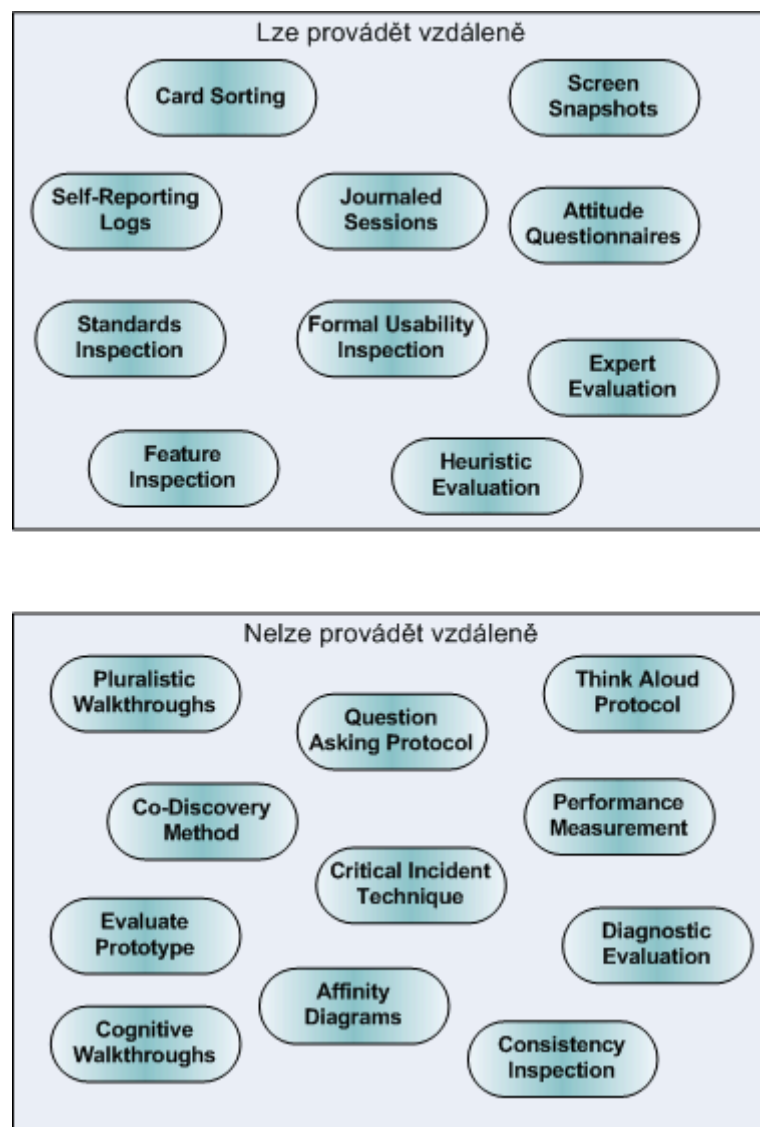
Vzhledem k tomu, že ne vždy je subjekt vyvíjející webovou aplikaci schopný zajistit prostory a vybavení k provedení testování, je důležité znát, které metody testování použitelnosti lze provádět i vzdáleně. Jednoduché rozdělení těchto metod lze nalézt na obrázku 4.3.

	Návrh	Implementace	Provoz
Card Sorting	Progress bar		
Affinity Diagrams	Progress bar		
Pluralistic Walkthroughs	Progress bar		
Standards Inspection	Progress bar		
Formal Usability Inspection	Progress bar		
Consistency Inspection	Progress bar		
Think Aloud Protocol	Progress bar		
Question Asking Protocol	Progress bar		
Evaluate Prototype	Progress bar		
Diagnostic Evaluation	Progress bar		
Self-Reporting Logs	Progress bar		
Performance Measurement	Progress bar		
Attitude Questionnaires	Progress bar		
Cognitive Walkthroughs	Progress bar		
Expert Evaluation	Progress bar		
Heuristic Evaluation	Progress bar		
Journalled Sessions		Progress bar	
Screen Snapshots		Progress bar	
Co-Discovery Method		Progress bar	
Feature Inspection		Progress bar	
Critical Incident Technique		Progress bar	

Obr. 4.1: Přehled metod z hlediska postavení v životním cyklu vývoje webové aplikace



Obr. 4.2: Příbuzné metody



Obr. 4.3: Rozdělení metod dle možnosti, zda mohou být prováděny vzdáleně

5. Závěr

Použitelnost je faktorem, který ve velké míře ovlivňuje kvalitu webových aplikací. Jestliže se uživatel v aplikaci rychle orientuje, jednoduše se mu s ní pracuje a aplikace vykonává to, co předpokládá, bude uživatel rád tuto aplikaci využívat a doporučí ji i jiným uživatelům. Bohužel v současnosti není použitelnosti přikládána taková důležitost, jakou by si zasloužila, a to ke škodě uživatelů i samotných tvůrců webových aplikací.

Firmy, organizace i jednotlivci vyvíjející webové aplikace si musí uvědomit, že opomíjení použitelnosti jim způsobuje vysoké dodatečné náklady. Někteří tvůrci o použitelnosti vůbec neví a jiní jí úmyslně vynechávají z procesu vývoje, jelikož si myslí, že neaplikováním principů použitelnosti ušetří zdroje a jsou schopní vyvíjet aplikace s menšími náklady. To je bohužel mylná představa, jelikož náklady na budoucí úpravy aplikace z důvodu špatné použitelnosti jsou deseti- až stonásobné než investice do použitelnosti v průběhu vývoje.

Tato práce ukázala, že nejčastěji využívané modely životního cyklu vývoje aplikace v sobě nemají implementovány principy použitelnosti a jsou zaměřeny čistě na vývoj funkcionality aplikací na úkor vývoje uživatelského rozhraní. Přitom pro uživatele je mnohem důležitější právě uživatelské rozhraní, s kterým pracuje, než vnitřní uspořádání aplikace a způsob jejího naprogramování. Z tohoto důvodu je zavedení principů použitelnosti do vývoje velmi podstatné.

Z práce také vyplývá, že není důležité pouze snažit se o zavedení principů použitelnosti, ale že je důležité také testovat jejich dopad na celkovou použitelnost webové aplikace, protože samotné zavedení není zárukou zvýšení použitelnosti. Analýza metod testování použitelnosti tvoří jádro práce. Provedení této analýzy ovšem nebylo vůbec jednoduché, jelikož o této problematice existuje velmi málo (vesměs zahraničních) zdrojů a navíc tyto zdroje uvádí často odlišné informace. Rozhoduje-li se tedy tvůrce webové aplikace o tom, kterou metodu použít pro testování použitelnosti, může být z dostupných informací zmaten, popřípadě tyto informace vůbec nenalezne.

Hlavním přínosem této práce je právě zpřístupnění přehledu metod pro testování použitelnosti pro české vývojáře webových aplikací. Díky zde prezentovaným informacím mohou tvůrci

aplikací jednoduše podle několika kritérií rozhodnout, která metoda je pro ně nejvhodnější a kdy tuto metodu použít v průběhu životního cyklu vývoje aplikace. Informace o metodách jsou představeny v takové struktuře, které je maximálně přehledná a umožňuje rychlé porovnání jednotlivých metod. Tyto informace jsou doplněny grafickými přehledy, které zachycují:

- vhodnost použití konkrétní metody v určité fázi životního cyklu vývoje aplikace;
- příbuznost jednotlivých metod;
- možnost vzdáleného testování.

Společně s rozdělením metod na metody testování s přítomností koncových uživatelů a bez přítomnosti koncových uživatelů tvoří tyto grafické přehledy rozcestník, který je možné využít pro prvotní rozhodování o výběru testovacích metod. Poté, co si vývojář vybere pomocí rozcestníku metody, které by mohl využít, získá podrobnější informace v přehledu charakteristik uvedeném u každé metody, může využít zdrojů uvedených u této metody pro zjištění detailních informací o metodě např. o konkrétním způsobu provádění testování pomocí této metody.

Na výstupy této práce je možné navázat detailnějšími analýzami jednotlivých metod, vytvářením nových metod, aplikací těchto metod na konkrétních případech nebo zhodnocením ekonomického dopadu zavedení použitelnosti do procesu vývoje webových aplikací.

Použité zdroje

- [1] NIELSEN, Jakob. *Usability 101: Fundamentals and Definition - What, Why, How* [online]. 2003-08-25 [cit. 2007-04-14].
Dostupný z WWW: <<http://www.useit.com/alertbox/20030825.html>>.
- [2] Keevil & Associates. *Glossary* [online]. [1996-2000] [cit. 2007-04-14].
Dostupný z WWW: <<http://www3.sympatico.ca/bkeevil/tapuser/gloss.html>>.
- [3] SQAtester.com. *Glossary* [online]. c2000- [cit. 2007-04-14].
Dostupný z WWW: <<http://www.sqatester.com/glossary/>>.
- [4] Google. *Define: usability* [online]. c2007 [cit. 2007-04-15].
Dostupný z WWW: <<http://www.google.com/search?hl=cs&q=define%3A+usability&lr=>>>.
- [5] LOUIS, Tristan. *The TLN.net blog* [online]. c1994-2007 [cit. 2007-04-15].
Dostupný z WWW: <<http://www.tnl.net/blog/>>.
- [6] IBM. *Cost justifying ease of use* [online]. [2005] [cit. 2007-04-16].
Dostupný z WWW: <<http://www-03.ibm.com/easy/page/23>>.
- [7] PATTON, Ron. *Testování softwaru*. Libor Pácl; David Krásenský. 1. vyd. Praha : Computer Press, 2002. 313 s. ISBN 80-7226-636-5.
- [8] LEWALLEN, Raymond. *Software Development Life Cycle Models* [online]. 2005-06-13 [cit. 2006-12-24].
Dostupný z WWW: <<http://codebetter.com/blogs/raymond.lewallen/archive/2005/07/13/129114.aspx>>.
- [9] KRAVAL, Ilja. *Návrh IS pomocí OOP, UML a vzorů* [online]. [2006] , 2006-03-05 [cit. 2006-12-26].
Dostupný z WWW: <http://www.objects.cz/clanky/clanky_IS/NavrhIS.pdf>.
- [10] RHODES, John. A Proposal for Evaluating Usability Testing Methods: The Practical Review System (PRS) [online]. 2003-04-30 [cit. 2007-04-18]. Dostupný z WWW: <<http://webword.com/moving/prs.html>>.
- [11] MAURER, Donna, WARFEL, Todd. *Card sorting: a definitive guide* [online]. 2004-04-07 [cit. 2007-04-18].
Dostupný z WWW: <http://www.boxesandarrows.com/view/card_sorting_a_definitive_guide>.
- [12] UsabilityNet. *Card sorting* [online]. c2006 [cit. 2007-04-18]. Dostupný z WWW: <<http://www.usabilitynet.org/tools/cardsorting.htm>>.
- [13] UsabilityNet. *Affinity diagramming* [online]. c2006 [cit. 2007-04-18]. Dostupný z WWW: <<http://www.usabilitynet.org/tools/affinity.htm>>.
- [14] HOM, James. *Pluralistic Walkthroughs* [online]. c1996-2003 [cit. 2007-04-19]. Dostupný z WWW: <<http://jthom.best.vwh.net/usability/plural.htm>>.

- [15] UPA. *Pluralistic Usability Walkthrough* [online]. c2005 [cit. 2007-04-26]. Dostupný z WWW: <<http://www.usabilitybok.org/methods/p2049>>.
- [16] ZHIJUN, Zhang. *Pluralistic Walkthrough* [online]. [2000-2005] [cit. 2007-04-19]. Dostupný z WWW: <<http://www.usabilityhome.com/PlurWalk.htm>>.
- [17] HOM, James. *Think Aloud Protocol* [online]. c1996-2003 [cit. 2007-04-19]. Dostupný z WWW: <<http://jthom.best.vwh.net/usability/thnkalod.htm>>.
- [18] BEATON, Angus, et al. *HCI Lecture 5 - Think-Aloud Protocols* [online]. [1998] [cit. 2007-04-19]. Dostupný z WWW: <<http://www.psy.gla.ac.uk/~steve/HCI/cscln/trail1/Lecture5.html>>.
- [19] Human Factors Laboratory, National Institute for Aviation Research at Wichita State University. *Think-Aloud Protocol* [online]. 2004-05-11 [cit. 2007-04-19]. Dostupný z WWW: <http://www.niar.wichita.edu/humanfactors/toolbox/T_A%20Protocol.htm>.
- [20] HOM, James. *Question-asking Protocol* [online]. c1996-2003 [cit. 2007-04-20]. Dostupný z WWW: <<http://jthom.best.vwh.net/usability/questest.htm>>.
- [21] ZHIJUN, Zhang. *Question-asking Protocol* [online]. [2000-2005] [cit. 2007-04-20]. Dostupný z WWW: <<http://www.usabilityhome.com/Question.htm>>.
- [22] UsabilityNet. *Evaluate prototype* [online]. c2006 [cit. 2007-04-29]. Dostupný z WWW: <<http://www.usabilitynet.org/tools/evaluate.htm>>.
- [23] UsabilityNet. *Diagnostic evaluation* [online]. c2006 [cit. 2007-04-20]. Dostupný z WWW: <<http://www.usabilitynet.org/tools/diagnostic.htm>>.
- [24] HOM, James. *Self-Reporting Logs* [online]. c1996-2003 [cit. 2007-04-20]. Dostupný z WWW: <<http://jthom.best.vwh.net/usability/selfrept.htm>>.
- [25] HOM, James. *Journalled Sessions* [online]. c1996-2003 [cit. 2007-04-20]. Dostupný z WWW: <<http://jthom.best.vwh.net/usability/journals.htm>>.
- [26] HOM, James. *Screen Snapshots* [online]. c1996-2003 [cit. 2007-04-20]. Dostupný z WWW: <<http://jthom.best.vwh.net/usability/snapshot.htm>>.
- [27] Human Factors Laboratory, National Institute for Aviation Research at Wichita State University. *Co-Discovery User Evaluation* [online]. 2004-05-11 [cit. 2007-04-21]. Dostupný z WWW: <<http://www.niar.twsu.edu/humanfactors/toolbox/Co-Discovery.htm>>.
- [28] HOM, James. *Co-Discovery Method* [online]. c1996-2003 [cit. 2007-04-20]. Dostupný z WWW: <<http://jthom.best.vwh.net/usability/codiscvr.htm>>.

- [29] UsabilityNet. *Critical Incident Technique Analysis* [online]. c2006 [cit. 2007-04-21]. Dostupný z WWW: <<http://www.usabilitynet.org/tools/criticalincidents.htm>>.
- [30] UPA. *Critical Incident Technique* [online]. c2005 [cit. 2007-04-21]. Dostupný z WWW: <<http://www.usabilitybok.org/methods/p2052>>.
- [31] HOM, James. *Performance Measurement* [online]. c1996-2003 [cit. 2007-04-21]. Dostupný z WWW: <<http://jthom.best.vwh.net/usability/perfmeas.htm>>.
- [32] ZHIJUN, Zhang. *Performance Measurement* [online]. [2000-2005] [cit. 2007-04-21]. Dostupný z WWW: <<http://www.usabilityhome.com/PerfMeas.htm>>.
- [33] UsabilityNet. *Performance testing* [online]. c2006 [cit. 2007-04-21]. Dostupný z WWW: <<http://www.usabilitynet.org/tools/testing.htm>>.
- [34] UsabilityNet. *Subjective Assessment* [online]. c2006 [cit. 2007-04-21]. Dostupný z WWW: <<http://www.usabilitynet.org/tools/subjective.htm>>.
- [35] HOM, James. *Standards Inspection* [online]. c1996-2003 [cit. 2007-04-23]. Dostupný z WWW: <<http://jthom.best.vwh.net/usability/stdinsp.htm>>.
- [36] UsabilityNet. *Style guides* [online]. c2006 [cit. 2007-04-23]. Dostupný z WWW: <<http://www.usabilitynet.org/tools/style.htm>>.
- [37] HOM, James. *Formal Usability Inspection* [online]. c1996-2003 [cit. 2007-04-23]. Dostupný z WWW: <<http://jthom.best.vwh.net/usability/frmlinsp.htm>>.
- [38] GUNN, Cathy. *An Example of Formal Usability Inspections in Practice at Hewlett-Packard Company* [online]. [1995] [cit. 2007-04-23]. Dostupný z WWW: <http://acm.org/sigchi/chi95/proceedings/intpost/cg_bdy.htm>.
- [39] ALMSTRÖM, Peter. *Usability inspections* [online]. c2001 [cit. 2007-04-23]. Dostupný z WWW: <<http://www.affectus.se/publicerat/usabilityinsp/index.html>>.
- [40] HOM, James. *Consistency Inspections* [online]. c1996-2003 [cit. 2007-04-26]. Dostupný z WWW: <<http://jthom.best.vwh.net/usability/consist.htm>>.
- [41] CS 6693: *User interfaces and usability* [online]. 2006-04-22 [cit. 2007-04-26]. Dostupný z WWW: <<http://vip.cs.utsa.edu/classes/cs6693s2006/lectures/cs6693week13.html>>.
- [42] Foraker Design. *Usability Glossary: feature inspection* [online]. c2002-2005 [cit. 2007-04-29]. Dostupný z WWW: <http://www.usabilityfirst.com/glossary/term_512.txl>.
- [43] HOM, James. *Feature Inspection* [online]. c1996-2003 [cit. 2007-04-29]. Dostupný z WWW: <<http://jthom.best.vwh.net/usability/fttrinsp.htm>>.

- [44] ZHIJUN, Zhang. *Feature Inspection* [online]. [2000-2005] [cit. 2007-04-29]. Dostupný z WWW: <<http://www.usabilityhome.com/Feature1.htm>>.
- [45] LIEBEL, Gene. *How to Conduct a Cognitive Walkthrough* [online]. c2007 [cit. 2007-04-29]. Dostupný z WWW: <<http://usability.about.com/od/usabilitytesting/a/cognitive.htm>>.
- [46] Human Factors Laboratory, National Institute for Aviation Research at Wichita State University. *Cognitive Walkthroughs* [online]. 2004-05-11 [cit. 2007-04-29]. Dostupný z WWW: <<http://www.niar.twsu.edu/humanfactors/toolbox/CWalk.htm>>.
- [47] HOM, James. *Cognitive Walkthrough* [online]. c1996-2003 [cit. 2007-04-29]. Dostupný z WWW: <<http://jthom.best.vwh.net/usability/cognitiv.htm>>.
- [48] UsabilityNet. *Heuristic evaluation* [online]. c2006 [cit. 2007-04-30]. Dostupný z WWW: <<http://www.usabilitynet.org/tools/expertheuristic.htm>>.
- [49] NIELSEN, Jakob. *How to Conduct a Heuristic Evaluation* [online]. [2005] [cit. 2007-04-30]. Dostupný z WWW: <http://www.useit.com/papers/heuristic/heuristic_evaluation.html>.
- [50] HOM, James. *Heuristic evaluation* [online]. c1996-2003 [cit. 2007-04-30]. Dostupný z WWW: <<http://jthom.best.vwh.net/usability/heuristic.htm>>.

Terminologický slovník

Použitelnost

Použitelnost je vlastnost vztahu mezi nástrojem (systémem) a jeho uživatelem, která udává míru, jak dobře je nástroj navržen pro efektivní a jednoduchou práci.

Metoda testování použitelnosti

Souhrn technik a postupů, které slouží k ověření dosažené míry použitelnosti nástroje (systému).