

Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informačních technologií

Hodnocení kvality softwaru
Bakalářská práce

Autor: Adam Málek
Studijní obor: Informační management

Vedoucí práce: doc. Ing. Hana Tomášková, Ph.D.

Prohlášení:

Prohlašuji, že jsem bakalářskou práci zpracoval samostatně a s použitím uvedené literatury.

A handwritten signature in black ink, appearing to read 'Málek', written in a cursive style.

V Hradci Králové dne 11.11.2023

Adam Málek

Poděkování:

Děkuji vedoucímu bakalářské práce doc. Haně Tomáškové, Ing. Ph.D. za metodické vedení této práce a její rady.

Anotace

Tato bakalářská práce se zabývá problematikou hodnocením kvality softwaru a později analýzou metod pro hodnocení kvality softwaru, shrnutí jejich přínosů a negativ.

Práce se nejprve zabývá teoretickým základem hodnocení kvality softwaru a představuje různé přístupy, které jsou v této oblasti využívány. Dále jsou identifikovány a analyzovány základní modely a klíčové metody, které umožňují hodnotit technické i uživatelské aspekty kvality.

Praktická část této bakalářské práce provedla hodnocení a porovnání tří nejnámějších textových editorů: Microsoft Word, Google Dokumenty a LibreOffice Writer. Toto srovnání bylo provedeno s využitím modelu FURPS, který zahrnuje klíčové charakteristiky jako Funkčnost, Použitelnost, Spolehlivost, Výkonnost a Podporovatelnost.

V samotném závěru jsou reflektovány dosažené výsledky a nabízejí doporučení. Práce, tak přináší ucelený pohled na tuto problematiku.

Klíčová slova

Hodnocení, kvalita, softwarový produkt, metody, modely

Annotation

Title: Software quality evaluation

This bachelor thesis deals with the issue of software quality evaluation and later on the analysis of methods for software quality evaluation, a summary of their benefits and negatives.

The thesis first discusses the theoretical basis of software quality assessment and presents the different approaches that are used in this field. Then, the basic models and key methods that allow to evaluate both technical and user aspects of quality are identified and analyzed.

The practical part of this bachelor thesis evaluated and compared the three most well-known word processors: Microsoft Word, Google Docs and LibreOffice Writer. This comparison was done using the FURPS model, which includes key characteristics such as Functionality, Usability, Reliability, Performance and Supportability.

In the end, the results obtained are reflected and recommendations are offered. The bachelor thesis provides a comprehensive view of the subject.

Keywords

Evaluation, quality, software, methods, models

Obsah

1	Úvod.....	1
2	Cíl práce.....	2
3	Metodika zpracování.....	3
4	Software	4
4.1	Systémový software	4
4.2	Aplikační software	4
5	Kvalita software.....	5
5.1	Norma ISO/IEC 9126	6
5.2	Norma ISO/IEC 25010.....	7
5.3	McCallův model.....	9
5.4	Boehmův model.....	11
5.5	Dromeyho model.....	12
5.6	Model FURPS a FURPS+	13
6	Metody hodnocení kvality softwaru	16
6.1	Testování	16
6.1.1	Manuální testování.....	16
6.1.2	Automatizované testování	17
6.1.3	Jednotkové testování.....	18
6.1.4	Integrační testování.....	19
6.2	Statická analýza kódu	20
6.3	Metriky kvality	21
6.3.1	Proč měřit kvalitu?.....	22
6.3.2	Příklady metrik.....	23
7	Hodnocení a porovnání známých textových editorů	25
7.1	Představení testovaného software	25

7.1.1	Microsoft Word.....	25
7.1.2	Google Dokumenty	28
7.1.3	LibreOffice Writer	31
8	Vyhodnocení a porovnání textových editorů dle stanovených kritérií.....	34
9	Vyhodnocení softwaru dle typu uživatelů.....	36
9.1	Typ uživatele začátečník.....	36
9.2	Typ uživatele pokročilý.....	36
9.3	Typ uživatele expert.....	37
10	Shrnutí výsledků	40
11	Závěry a doporučení.....	41
12	Seznam použité literatury	42

Seznam obrázků

Obrázek 1: Model kvality podle ISO/IEC 25010 (autor podle ISO/IEC 25010)	9
Obrázek 2: Model FURPS (autor podle [5])	14

Seznam tabulek

Tabulka 1 – Vyhodnocení kritérií (zpracování vlastní)	35
---	----

1 Úvod

Kvalita softwaru má klíčový význam pro úspěch moderních aplikací a systémů. S nárůstem důležitosti softwarových produktů se stává nezbytným provádět úplné hodnocení kvality softwaru, které zajistí optimální výkonnost, spolehlivost a uživatelský zážitek. S tímto účelem je stále více organizací, vývojářských týmů, které jsou zaměřené na vytváření sofistikovaných metod a nástrojů pro hodnocení kvality softwaru.

Tato bakalářská práce se zabývá tématem hodnocení kvality softwaru jeho významu v dnešním vývojovém prostředí. Analyzuje různé modely, metody a techniky, které umožňují objektivně zhodnotit různé aspekty softwarového produktu. Práce se zaměřuje nejen na technické hledisko, ale i na uživatelskou perspektivu, ergonomii a další faktory, které ovlivňují celkovou kvalitu softwaru.

Cílem této bakalářské práce je poskytnout přehled o různých metodách, modelech a nástrojích pro hodnocení kvality softwaru, a to jak z technického, tak i uživatelského pohledu. Dále bude práce zkoumat výhody, nevýhody a případné využití jednotlivých metod.

Praktická část této bakalářské práce jsem se zaměřil na hodnocení a porovnání tří nejnámějších textových editorů: Microsoft Word, Google Dokumenty a LibreOffice Writer. Toto srovnání bylo provedeno s využitím modelu FURPS, který zahrnuje klíčové charakteristiky jako Funkčnost, Použitelnost, Spolehlivost, Výkonnost a Podporovatelnost.

2 Cíl práce

Cílem této práce je provést hodnocení základních metod, modelů a technik používaných pro hodnocení kvality softwaru. Dalším cílem je definovat a popsat základní metodiky a modely hodnocení kvality softwaru. Práce se zaměří na zkoumání jak technických, tak i uživatelských aspektů kvality a zhodnotí jejich výhody a nevýhody jednotlivých přístupů.

3 Metodika zpracování

Metodika zpracování bakalářské práce bude postavena na systematickém přístupu k analýze a hodnocení kvality softwaru. Následující kroky budou následovány k dosažení jednotlivých cílů práce:

1. **Rešerše a teorie** – Nejprve budou studovány relevantní zdroje týkající se hodnocení kvality softwaru. To poskytne teoretický základ a nadhled do existujících metodik a nástrojů.
2. **Identifikace metod a modelů** – Na základě rešerše budou identifikovány různé metody, modely a techniky používané pro hodnocení kvality softwaru. Popis výhod, nevýhod a oblast jejich využití.
3. **Aplikace metod** – Vybraná metoda bude aplikována na reálné softwarové produkty
4. **Zhodnocení výsledků** – Výsledky z hodnocení a porovnání textových editorů budou vyhodnoceny.
5. **Závěr**

4 Software

K úvodu práce je nutné si definovat, co vlastně pojem software znamená. Softwarem rozumíme veškeré programové vybavení počítače, které provádějí nějakou činnost.[1] V počítači lze software chápat jako vše, co není hardware neboli fyzické vybavení počítače.

V naprosté většině se můžeme setkat s rozdělením software na systémový software (například operační systém, firmware) a dále na aplikační software (programy).

4.1 Systémový software

Systémový software lze chápat jako speciální program, který zajišťuje komunikaci mezi hardwarem a aplikací. Jako systémový software lze tedy považovat operační systém [8] (například pro PC: Windows, Linux, pro mobilní zařízení: Android, iOS). V podstatě bez dříve jmenovaného systémového softwaru bychom nemohli aplikační software provozovat.

4.2 Aplikační software

Aplikačním softwarem tedy máme na mysli programy sloužící konkrétnímu účelu. Může se tedy jednat o programy jako [8]:

- textové editory, tabulkové procesory, grafické editory
- internetové prohlížeče
- programy pro ochranu počítače a pracování s daty
 - antivirové programy
 - správci souborů
- programy pro vzájemnou komunikaci s okolím
 - aplikace pro videohovory
 - e-mailový klienti
 - chatovací programy
- programy pro vzdálenou komunikaci s ostatními PC
- hry
- programy používané pro výuku

5 Kvalita software

Myšlenky a úvahy o podstatě kvality nejsou ničím novým, prokazatelně již v antickém Řecku se filozofové zabývali jejím definicí a pojetím a vnímáním. První, kdo tento pojem vůbec použil byl Aristotelův učitel Platón, datovaný do druhé poloviny 4. století před naším letopočtem. [2]

Co je to tedy kvalita? Pro každého z nás slovo kvalita představuje něco jiného. Obecně je velmi těžké tedy charakterizovat a vymezit kvalitu, protože neexistuje všeobecně platná a uznávaná definice tohoto pojmu. Kvalita určitého produktu pro dvě různé osoby může být naprosto odlišná. To stejné, když máme levný a drahý produkt stejné kategorie – oba mohou být považovány za kvalitní. Osoby, které mají omezený rozpočet si vyberou produkt, který představuje kompromis za danou cenu.[2]

Nemůžeme tedy brát levný produkt za kvalitní, pokud daný produkt bude mít vysokou poruchovost, horší spolehlivost nebo nutnost výměny určitých součástí důsledně nízké životnosti, což vede k budoucím dodatečným nákladům. [2]

Kvalitu může ovlivňovat mnoho faktorů například čas. Ukázkovým příkladem může být osoba, která na poslední chvíli shání určitý produkt, za něj bude ochoten zaplatit více peněz, jelikož je omezena časem. Dalším faktorem může být i sociální faktor. Osoba může preferovat daný výrobek na základě toho, že jej vlastní dostatečný počet lidí v jeho okolí.

Podle [1] ve své knize rozlišuje dva pohledy na kvalitu:

- 1) Laický pohled
- 2) Profesionální pohled

Laický pohled říká, že kvalitu nelze přesně definovat nebo měřit a je to spíše subjektivní dojem. Podle autorova pohledu je kvalita něco, co poznáme až v praxi, kdy toto poznání vychází z osobních pocitů. Jiný pohled spojuje kvalitu s luxusem a vkusem, což může vést k přesvědčení, že dražší produkty jsou automaticky

považovány za více kvalitní, bez ohledu na konkrétní kvalitativní faktory daného výrobku. [1]

Naopak, profesionální perspektiva na kvalitu se snaží o jasnou definici. Kan v tomto kontextu zdůrazňuje dva klíčové pojmy: „conformance to requirements“ a „fitness for use“. První termín česky, „shoda s požadavky“, vyjadřuje kvalitu v kontextu jasně stanovených požadavků, což znamená, že produkt nesmí nesplňovat žádný z nich. Druhý termín česky, „vhodnost pro použití“, zohledňuje potřeby zákazníků a očekávání, přičemž produkt může sloužit různým skupinám uživatelů a musí splňovat jejich specifická očekávání.[1]

Podle IEE (Institute of Electrical and Electronics Engineers) je kvalita softwaru definována takto:

„Kvalita softwaru je stupeň, do kterého software splňuje stanovené požadavky. Stupeň kvality může být měřen z hlediska souladu se specifikovanými požadavky, jak je stanoveno ve specifikaci produktu nebo jiném normativním dokumentu.“ [3]

Tato definice zdůrazňuje to, že kvalita softwaru je určena tím, do jaké míry odpovídá stanoveným požadavkům, které byly předem dané a specifikované v dokumentech jako je například specifikace produktu nebo technické požadavky.

5.1 Norma ISO/IEC 9126

Norma ISO/IEC 9126 představuje rámec pro model kvality softwaru. Tato norma se zaměřuje na identifikaci klíčových charakteristik, které jsou důležité pro celkovou kvalitu softwarového produktu. Zahrnuje šest klíčových charakteristik: Funkčnost, Spolehlivost, Použitelnost, Efektivita, Udržovatelnost a Přenositelnost, přičemž každá z těchto charakteristik obsahuje soubor podcharakteristik a atributů. Tyto charakteristiky jsou klíčové pro zajištění vysoké úrovně kvality v oblasti softwaru. [9]:

1. Funkcionalita

- Popisuje schopnost softwaru plnit základní úkoly a poskytovat požadované výstupy

2. Spolehlivost

- Spolehlivost softwaru vyjadřuje jeho schopnost poskytnout plnou funkčnost v daných podmínkách a v určitém časovém období. Součástí této charakteristiky je také schopnost systému se vzpamatovat z externích selhání, jako například výpadky proudu.

3. Použitelnost

- Použitelností se rozumí, jak snadno a rychle se nový uživatel naučí zacházet a ovládat daný software, tak že mu nebude dělat potíže ho ovládat

4. Efektivnost

- Efektivnost softwaru se týká schopnosti dosáhnout požadovaných výsledků nebo cílů s minimálním vynaložením zdrojů. Pro uživatele to znamená, že mohou rychle a účinně dosáhnout svých cílů při používání softwaru.

5. Udržovatelnost

- Udržovatelnost softwaru označuje schopnost být snadno spravován, aktualizován a rozšiřován po nasazení. Důležitý je dobře napsaný a srozumitelný produkční kód a schopnost rychle reagovat na změny a opravy.

6. Přenositelnost

- Tato charakteristika softwaru vyjadřuje jeho schopnost být úspěšně implementován na různých platformách nebo prostředích s minimálními úpravami.

5.2 Norma ISO/IEC 25010

Kvalitu softwaru také definuje a popisuje norma ISO/IEC 25010 (novější norma, která nahradila v roce 2011 starší normu ISO/IEC 9126), která poskytuje aktualizovaný rámec pro hodnocení kvality softwarových produktů. ISO/IEC 2510

je mezinárodní norma a definuje **model kvality pro používání** (5 charakteristik) a podle této normy **je model kvality softwaru** charakterizován do 8 charakteristik.

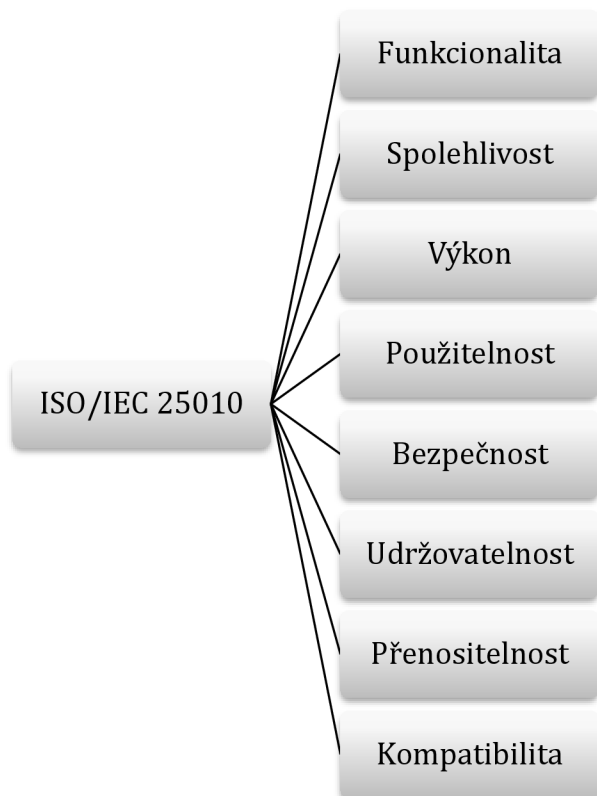
1. Model kvality pro použití:

- Účelnost
- Efektivnost
- Uspokojení:
 - Užitečnost
 - Důvěryhodnost
 - Potěšení z používání
 - Komfort při použití
- Bezpečnost:
 - Minimalizace ekonomického rizika
 - Minimalizace zdravotního a bezpečnostního rizika
- Pokrytí kontextu:
 - Úplnost pokrytí
 - Flexibilita

2. Model kvality softwaru, který je charakterizován namísto šesti – dle původní normy ISO/IEC 9126 – osm charakteristik kvality podle [4]:

- Funkčnost (functionality)
- Spolehlivost (reliability)
- Výkon (performance)
- Použitelnost (usability)
- Bezpečnost (security)
- Udržovatelnost (maintainability)
- Přenositelnost (portability)
- Kompatibilita (compatibility)

Každá z výše uvedených charakteristik má své vlastní kritéria, které umožňují objektivní měření a hodnocení, tak aby bylo možné posoudit do jaké míry software tyto požadavky splňuje.



**Obrázek 1: Model kvality podle ISO/IEC 25010
(autor podle ISO/IEC 25010)**

Právě jsme si definovali a popsali model kvality podle ISO/IEC 25010 viz. výše a v následujících kapitolách se seznámíme s dalšími modely kvality softwaru.

5.3 McCallův model

McCallův model je jeden z historických modelů, který se zabývá definováním a měřením kvality softwarových produktů.

Tento model byl představený v roce 1977 J. McCallem, tento model reflektuje jak pohled uživatelů, tak prioritu vývojářů. Byl vyvinut za účelem identifikace klíčových faktorů ovlivňujících kvalitu softwaru.

McCall definoval tři základní pohledy na kvalitu produktu a, které jsou následně charakterizovány pomocí faktorů kvality produktu. Celkem jedenáct faktorů jsou neměřitelné a obsahují jednotlivá kritéria. I když jsou samotné faktory orientovány spíše technicky, jsou popsány prostřednictvím otázek a charakteristik.

Pro názornou ukázkou je u jednoho z faktorů jsou vypsána hlavní kritéria, které jsou v nejnižší úrovni tohoto modelu.

Analyzovaný model se skládá z níže uvedeného seznamu [2]:

1. Revize produktu – proveditelnost změn

- Flexibilita
- Udržovatelnost
- Testovatelnost

2. Činnost produktu – provozní charakteristiky

- Správnost
 - sledovatelnost
 - úplnost
 - důslednost
- Spolehlivost
- Účinnost
- Integrita
- Použitelnost

3. Přechod produktu – schopnost přizpůsobení se novému prostředí

- Přenositelnost
- Znovupoužitelnost
- Interoperabilita

McCallův model přispěl k pochopení důležitých faktorů ovlivňující kvalitu softwaru a umožnil lepší řízení a hodnocení softwarových projektů. I přesto, že moderní přístupy k hodnocení kvality softwaru jsou rozvinutější a zahrnují více faktorů a metrik, McCallův model přinesl významný přínos v oblasti softwarového inženýrství.

5.4 Boehmův model

Boehmův model nebo také nazývaný Boehmova kvalitativní třídící metoda. Byl vyvinut B.W. Boehmem a poprvé představen v roce 1978, ve snaze definovat kvalitu softwaru kvantitativně jako množinu atributů a měřitelných kritérií.

Boehmův model obsahuje tři úrovně charakteristik. Nejvyšší úroveň představuje tři hlavní požadavky. Střední úroveň je tvořena skupinou sedmi jednotlivých charakteristik kvality, které v souhrnu představují kvality očekávané od systému. Na nejnižší úrovni se nachází základní kritéria kvality, za jejichž pomocí lze efektivněji definovat charakteristiky veškerých vyšších úrovní. [2]

Samotná struktura modelu je tvořena podobně jako u McCallova modelu viz. seznam níže. Stejně jako u předchozího modelu je pouze jedna charakteristika rozepsána až do nejnižší úrovně. Zajímavosti však je, že „přenositelnost“ autor vymezil jako nejvyšší charakteristiku a současně jako přímo měřitelný faktor, proto také neobsahuje střední úroveň.

Model se skládá z níže uvedeného seznamu [2]:

1. Udržovatelnost

- Srozumitelnost
 - konzistentnost
 - strukturovanost
 - čitelnost
 - stručnost
- Modifikovatelnost
- Testovatelnost

2. Vlastnosti při užití

- Spolehlivost
- Účinnost
- Použitelnost

3. Přenositelnost

Oproti McCallova modelu zásadní změnou v tomto Boehmovém modelu je chápání nejvyšší úrovně, neboť Boehm klade na nejvyšší úroveň právě požadavky koncových uživatelů. Naopak na nejnižší úrovni se zabývá technickými detaily. Díky tomu, lze říct, že Boehmův model je spíše orientován na vývojáře než na koncové uživatele.

V případě porovnání s McCallovým modelem mají oba modely společné to, že se snaží zjistit klíčové faktory, které ovlivňují kvalitu softwaru a poskytují hierarchický rámec pro hodnocení. V podstatě Boehmův model rozšiřuje McCallův model o další kategorie a je mnohem komplexnější (souhrnnější).

5.5 Dromeyho model

Konceptuální model kvality od R.G. Dromeyho byl představen v roce 1995 a je založen na kvalitě z pohledu produktu a jeho jednotlivých komponent. Podle Dromeyho tak „*vnitřní charakteristiky a vlastnosti produktu přispívají či přímo určují jeho vnější atributy kvality*“. Kvalita produktu v tomto modelu je určována jeho komponenty a jejich implementacemi. V případě nevhodně zvolené či implementované komponenty lze ovlivnit celkovou kvalitu produktu.

Dromeyho rámec kvality se skládá ze tří modelů [2]:

- 1. Model kvality implementace**
- 2. Model kvality požadavků**
- 3. Model kvality návrhu**

K jednotlivým modelům jsou pak definovány i jeho jednotlivé komponenty, které mohou být proměnné, příkazy, funkce, požadavky apod. [2]

Jednotlivé komponenty jsou pak definovány podle vlastností, které mají největší dopad na produktové charakteristiky. Následně posledním krokem je zhodnocení modelu pro identifikaci jeho nedostatků.

R.G. Dromey, autor tohoto modelu, vymezil čtyři kategorie vlastností s vlivem na kvalitu komponent:

1. **Vlastnosti týkající se správnosti** – mají vliv na základní chování produktu
2. **Interní vlastnosti** – souvisí se strukturou komponenty
3. **Kontextuální** – složení, vztahy komponent s důrazem na vnější vlivy
4. **Deskriptivní** – měří popisnost, srozumitelnost komponenty

Dromey vycházel z toho, že pokud budou všechny komponenty vykazovat uvedené vlastnosti ovlivňující kvalitu, pak očekává, že produkt jako celek bude vykazovat charakteristiky jako je udržitelnost, znovupoužití, funkčnost apod. Jeho model, tak nebyl příliš pozitivně přijímán, neboť nezahrnoval metriky, což mohlo zmanipulovat validaci¹ a následnou verifikaci² produktu.

5.6 Model *FURPS* a *FURPS+*

Posledním modelem, který bude v této práci představen se nazývá model *FURPS*. Model *FURPS* popsali v roce 1987 Robert Grady a Deborah Cashwell a následně v roce 1992 byl představen ve společnosti Hewlett-Packard. [5]

Zkratka *FURPS* podle [2] je složena z prvních písmen anglických slov, které znamenají:

- **Funkčnost (Functionality)** – schopnosti a funkce softwaru, které byly definovány ve specifikacích

¹ Validace – proces, který ověřuje, zda produkt vyhovuje potřebám uživatele

² Verifikace – proces, který zajišťuje, že výstupy jednotlivých fází vývoje odpovídají vstupům – specifikacím odvozeným z požadavků

- **Použitelnost (Usability)** – zaměřením na to, jak snadno a efektivně mohou uživatelé používat software
- **Spolehlivost (Reliability)** – schopnosti softwaru pracovat bez chyb a narušení, měří se doba bezporuchového provozu, schopnost obnovení provozu
- **Výkonnost (Performance)** – zahrnuje aspekty jako odezva, rychlost softwaru za specifikovaných podmínek, účinnost
- **Podporovatelnost (Supportability)** – zahrnuje vlastnosti jako udržovatelnost, testovatelnost, rozšiřitelnost



Obrázek 2: Model FURPS (autor podle [5])

Model FURPS poskytuje rámec pro strukturované a komplexní hodnocení kvality softwaru, v dnešní době se většinou setkáváme v rozšířené podobě s názvem **FURPS+**, kde je model rozšířen o další kategorie, které jsou důležité pro komplexní hodnocení kvality softwaru. Tyto další kategorie jsou označovány symbolem „+“ v názvu modelu. [2]

Model FURPS+ obsahuje základní kategorie modelu FURPS a je rozšířen o následující kategorie [2]:

Omezení návrhu (Design constraints) – závislost na specifických hardwarových či softwarových platformách a systémech

Požadavky na implementaci (Implementation requirements) – aspekty implementace zohledňují způsob vývoje systému (programovací jazyky, technologie, nástroje, návrhové vzory, soulad s normami, ...)

Omezení rozhraní (Interface constraints) – popisuje interakci mezi softwarem a jinými systémy, v jaké formě a jakým způsobem bude probíhat výměna dat, nebo v jakém formátu

Požadavky na fyzické vlastnosti (Physical requirements) – tyto aspekty se týkají požadavků na hardware, například výkon, paměť, použité zobrazovací zařízení a další technické parametry

Model FURPS+ poskytuje daleko komplexnější pohled na hodnocení kvality softwaru a jeho hlavním benefitem je, že tento model zahrnuje širší spektrum faktorů a aspektů a díky tomu může lépe pokrýt různé stránky kvality softwaru a potřeby různých typů projektů a organizací.

6 Metody hodnocení kvality softwaru

Existuje mnoho různých metod a přístupů pro hodnocení kvality softwaru, které se mohou lišit v závislosti na cílech projektu, typu softwaru a dostupných zdrojích. Proto v této bakalářské práci jsou představené nejběžnější metody hodnocení kvality softwaru

6.1 Testování

Pojem testování v souvislosti s hodnocením kvality softwaru lze definovat jako proces, který slouží k ověření a validaci softwaru s cílem zjistit, zda splňuje stanovené požadavky, funkce a očekávání. Hlavním cílem testování je zjistit chyby, nedostatky a problémy v softwaru a identifikovat, zda je produkt funkční, spolehlivý a uživatelsky přívětivý.

Podle [6] vyplývá, že: „testování zkoumá softwarový produkt, čímž získává informace o kvalitě, kterou chápeme jako stupeň naplnění požadavků a přání uživatelů“. Čímž poukazuje na to, že obsahem testování je sběr a analýza informací.

6.1.1 Manuální testování

Manuální testování je metodou hodnocení kvality softwaru. Tato metoda je založena na fyzické interakci testera³ se softwarovým produktem v reálném čase. Testeři manuálně provádějí různé testovací scénáře a interakce se softwarovým produktem. [6]

Během manuálního testování testeři používají software jako běžní uživatelé a provádějí různé úkoly, scénáře a kroky, které mají ověřit správnost a funkčnost

³ Tester – osoba, která je zodpovědná především za provádění testů podle připravených scénářů, jejich cílem je ověřit a hodnotit funkčnost, mají za úkol identifikovat chyby, nedostatky a problémy v softwaru [6]

softwarového produktu. Testují se různé funkce, uživatelská rozhraní, vstupy a výstupy, a také se simulují různé podmínky a situace, které by mohli nastat uživatelům, kteří budou software používat.

Výhody:

- Rychlá zpětná vazba
- Intuitivnost – testeři mohou odhalit problémy, které by jiné metody nemusely odhalit
- Uživatelský pohled – testeři mohou napodobit uživatelské chování a reakce na software
- Komplexní testování – testeři můžou provádět různé akce a různé kombinace scénářů

Nevýhody:

- Časová náročnost – manuální testování může být pomalé a náročné zejména pro opakované testování
- Subjektivita – celkový test může být ovlivněn názory testera a tím pádem i to, že člověk může chybovat

Manuální testování se široce využívá ve všech fázích softwarového vývoje.

6.1.2 Automatizované testování

V jednoduchosti lze říci, že automatizované testování je takové testování, které je prováděno softwarem. Automatizované testování porovnává očekávané a skutečné výsledky testů a umí vytvořit podrobné zprávy. Cílem automatizace tedy není nahradit manuální testování, ale snížit počet manuálních ověření kvality. Jakákoli úprava aplikace vyžaduje její znovu otestování a proto, automatizované testy se jeví jako nejvhodnější volba. Zavedení takových testů vede k úspoře peněz a času. Pro automatizované testování lze použít nástroj nebo program na opakování jednoduché operace. [7]

Automatizovaným testem může být třeba validace e-mailové adresy nebo validace města a jeho PSČ, po načtení nechám aplikaci otestovat dané vstupy.

Výhody:

- Nachází více chyb než člověk
- Lze použít opakovaně
- Rychlejší a efektivnější
- Lze pokrýt širokou škálu testů, které by pro manuální testování mohly být náročné a zdlouhavé

Nevýhody:

- Nemůže nahradit všechny testy – automatizované testování nemůže otestovat například uživatelskou přívětivost s rozhráním
- Jakmile se software změní, je potřeba upravit testovací skripty
- Nákladnost – některé nástroje jsou finančně náročné

Automatizované testování je vhodné zejména pro větší a náročnější projekty, kde se software často mění a vyvíjí. Využívá se především tam, kde je potřeba rychlé, systematické a opakovatelné provádění testů.

6.1.3 Jednotkové testování

Jednotkové testování (angl. Unit testing) provádí nejčastěji programátor sám, aby otestoval jednotlivé „stavební bloky“ programu – jednotky, které lze chápat jako nejmenší testovatelné části programu. Jednotky mohou být funkce, procedury či metody nebo třídy.

Cílem testování jednotek je prokázat, že každá jednotka pracuje správně a vykonává to, co by měla dle specifikací a požadavků. Testy jsou vytvářeny programátorem, který testuje, zda jednotka reaguje správně na různé vstupy a vrací očekávané výstupy.

Výhody:

- Rychlé – testy se provádějí rychle, protože testujeme malý kousek kódu
- Detekce chyb – odhaluje chyby v raném vývoji softwaru, což následně zjednodušuje a zlevňuje jejich opravu

Nevýhody:

- Časová náročnost – vytváření testů pro každou jednotku může být časově náročné, zvláště když jde o složitý projekt
- Při změně kódu je potřeba změnit i samotný test
- Potřeba znalosti kódu – pokud tester nebo jiný vývojář nemá dostatečné znalosti, mohou být testy nedostatečné nebo špatně napsané

6.1.4 Integroční testování

Jednotky (moduly) můžeme považovat za základní prvky, které jejich propojením vzniká systém. Pokud všechny jednotky pracují a fungují správně je potřeba je integrovat do systému jako celku.

Do fáze integračního testování lze vstoupit pouze za předpokladu, že máme k dispozici minimálně dva moduly, které dohromady poskytují určitou službu. [6] Integrace probíhá tak, že postupně se do systému přidávají jednotkově otestované moduly, které nesmí narušovat celkovou stabilitu tohoto systému a musí fungovat správně. Cílem tohoto testování je zajistit, že jednotlivé moduly (části softwaru) spolupracují, tak jak mají a případné chyby jsou identifikovány a řešeny. [6]

Při provádění samotné systémové integrace můžeme postupovat různými způsoby, které si zkusíme rozebrat a krátce popsat [6]:

- Způsob tzv. „velký třesk“ (angl. Big bang)– v tomto způsobu jsou moduly integrovány zároveň v jednom kroku, což je velmi snadné. Podstatnou nevýhodu můžeme najít v tom, že pokud se objeví problém nejspíše jeho

nalezení bude daleko komplikovanější a složitější, neboť se může nacházet kdekoli. Tento způsob doporučuji použít spíše pro méně složité systémy.

- Přístup shora-dolů (angl. Top-down) – pokud zvolíme tuto metodu integrace postupujeme hierarchicky nejvyšším modulem (části kódu) a přidáváme postupně nižší moduly.
- Přístup zdola-nahoru (angl. Bottom up) – jedná se o opačný postup než u zmíněného výše tzv. integrace začíná od nejnižších modulů a postupně se přidávají nejvyšší moduly. Je důležité zmínit, že systém nebude fungovat jako celek, dokud nebude přidán nejvyšší modul.
- Kombinovaný přístup (combined) – tento přístup se může nazývat také jako tzv. „sendvičový přístup“. V podstatě kombinuje výše zmíněné metody na třech úrovních – na nejvyšší jsou integrovány hlavní moduly přístupem shora-dolů a na spodní úrovni jsou moduly integrovány přístupem zdola-nahoru a zbývající moduly jsou umístěny do prostřední úrovně.

Výhody:

- Ověření interakcí jednotlivých modulů
- Identifikace chyb na úrovni integrace – odhaluje chyby a problémy, které mohou vzniknout, když spolu jednotlivé moduly začnou komunikovat

Nevýhody:

- Složitost – integrace může být náročná
- Závislost na modulech – nekompletní moduly, může způsobit, že integrační testování nebude plně možné provést

6.2 Statická analýza kódu

Statická analýza kódu je metoda pro hodnocení kvality softwaru, která se zaměřuje na zkoumání zdrojového kódu softwaru bez jeho aktivního spuštění. Hlavním cílem této metody je identifikovat syntaktické chyby, nedostatky v kódu, špatné vzory nebo zda je kód v souladu se standardy. [6]

Statická analýza kódu zahrnuje několik postupů a technik jako například **kontrolu formátování** – zajišťuje, že je kód napsaný s danými standardy pro formátování, což zlepšuje jeho čitelnost a udržitelnost. Dalším postupem je **hledání chyb a nedostatků**, zde je důležitá identifikace logických chyb, nepřesností v kódu, které by mohly vést k nefunkčnímu chování nebo chybám za běhu. Jak již bylo zmíněno statická analýza kódu zahrnuje **detekci špatných programovacích vzorů**, které by mohly vést k problémům v budoucnosti, jako je nesprávné použití paměti nebo neefektivní postupy. [7]

Výhody:

- Brzká detekce chyb
- Snížení nákladů – opravy chyb již v raných fázích vývoje jsou často levnější než opravy po dokončení softwaru
- Zlepšení kvality – díky detekci syntaktických chyb, logických problémů, které by mohly ovlivnit výkon a stabilitu

Nevýhody:

- Časově náročné – u větších projektů může provedení statické analýzy kódu trvat delší dobu
- Obtížné na integraci do vývojového procesu -> nutné úpravy a přizpůsobení

Statická analýza kódu umožňuje odhalit problémy již v rané fázi vývoje, což může zvýšit efektivitu oprav a snižuje náklady na údržbu. Identifikuje nedostatky dříve, než by mohly způsobit problémy za běhu aplikace. [6]

6.3 Metriky kvality

Metriky kvality jsou kvantitativní měření a ukazatele, které se používají k hodnocení různých faktorů kvality softwaru například pro měření výkonu se zaměřují na

rychlost a odezvu softwaru. Pro údržbu se metriky zaměřují na čitelnost a strukturu kódu. [6]

Metriky jsou většinou vyjádřeny číselnou hodnotou, kterou lze porovnat nebo sledovat v průběhu času. Metriky kvality slouží k tomu, abychom dostali objektivní pohled na různé charakteristiky softwaru. Umožňují měřit různé vlastnosti a poskytovat data, která mohou být analyzována pro zlepšení kvality a zároveň mohou pomoci s identifikací případných problémů. [6]

6.3.1 Proč měřit kvalitu?

Vzhledem k tomu, že softwarové projekty jsou často spojovány s velkými investicemi a riziky, je nutné, aby byly projektová rizika kontrolována a minimalizována. Jelikož testování je proces, který se zabývá zjišťováním informací o kvalitě a jejich zpracování a předávání. Měřením kvality se také zabývá samotný testovací tým, který má na starosti nejen měření kvality, ale i například zkoumání samotného produktu. [6]

Měřením se zabýváme z několika důvodů:

1. Prvním důvodem může být zjištění aktuálního stavu produktu
2. Sledování postupu oproti plánu
3. Motivace pracovníků k lepším výsledkům
4. Data pro navržení a obhájení změny procesu
5. Hlídaná (měřená) oblast je vždy pod kontrolou

Díky měření kvality můžeme zajistit spolehlivost a stabilitu softwaru nebo pomůže identifikovat oblasti, ve kterých se software zpomaluje a kde dochází k únikům paměti – pomáhá optimalizovat výkon. Dalším aspektem měření kvality může být, že měření poskytuje data, která mohou podpořit rozhodování týkající se vývoje nebo nasazení softwaru.

6.3.2 Příklady metrik

Samotných metrik kvality je mnoho a závisí na specifických charakteristikách a cílech softwarového projektu, proto zde uvedu příklady metrik kvality v různých oblastech:

Oblast výkon a efektivita

1. Rychlost odezvy – metrika, která měří dobu, za kterou aplikace reaguje na interakce zadané uživatelem
2. Rychlost načítání – metrika, která měří čas, který potřebuje aplikace ke spuštění a načtení
3. Spotřeba paměti – metrika, která měří kolik aplikace spotřebuje paměti pro svůj běh

Oblast spolehlivost

1. Doba mezi poruchami – měří průměrný čas mezi výpadky aplikace
2. Chybovost – měří počet chyb v aplikaci za určité období

Oblast uživatelský zážitek (UX)

1. Čas strávený na stránce – tato metrika měří průměrný čas, který uživatel strávil na jednotlivých stránkách
2. Odezva uživatelského rozhraní – měří rychlost, s jakou uživatelské rozhraní reaguje na interakce

Oblast testování

1. Pokrytí kódu testy – metrika, která měří kolik procent kódu bylo pokryto testy
2. Počet testovacích případů – měří počet jednotlivých testů

Tyto příklady metrik ilustrují různé faktory, které můžeme měřit při hodnocení kvality softwaru. Výběr správných metrik závisí na konkrétním kontextu projektu a na cílech, které chceme dosáhnout. [7]

Výhody:

- Objektivita dat, která lze použít pro identifikaci problémů
- Monitorování vývoje – metriky umožňují sledovat změny a vývoj softwaru v průběhu času, to může být užitečně pro určení, zda se software zlepšil nebo zhoršil
- Různorodost metrik – můžeme měřit různé charakteristiky softwaru jako je výkon, spolehlivost a další

Nevýhody:

- Závislost na správném výběru metrik – výběr správných metrik je důležitý, nesprávně zvolené metriky mohou vést k nepřesným výsledkům
- Omezený pohled – díky tomu, že metriky jsou vyjádřeny číselně, tak nemohou zachytit uživatelský zážitek nebo kreativitu

Souhrnně lze říci, že metriky můžeme považovat za cenný nástroj pro objektivní hodnocení kvality softwaru, ale je nutné si uvědomit, že by neměly být jedinou metodou hodnocení a je ideální je používat ve spojení s dalšími technikami.

7 Hodnocení a porovnání známých textových editorů

V rámci praktické části této bakalářské práce jsem se rozhodl provést hodnocení a porovnání tří nejznámějších softwarových nástrojů pro zpracování textu: Microsoft Word, LibreOffice Writer a Google Dokumenty. Tato volba byla motivována jejich významným vlivem na pracovní a akademické prostředí, stejně jako v běžném domácím použití. Dále zohledníme rozdílné potřeby tří typů uživatelů: začátečník, pokročilý a expert. Každému z těchto uživatelů si přiřadíme kritéria, která jsou nejvhodnější a která s největší pravděpodobností uživatel preferuje.

Hodnocení těchto softwarových programů bude provedeno pomocí modelu FURPS (Functionality, Usability, Reliability, Performance, Supportability), který poskytuje rámec pro zhodnocení kvality softwaru. Každý z těchto faktorů bude analyzován a porovnán s cílem poskytnout užitečné informace pro potenciální uživatele a rozhodování v oblasti výběru textového editoru.

Nejprve vyhodnotíme a porovnáme textové editory. A poté tuto analýzu provedeme s ohledem na tři různé typy uživatelů: začátečník, pokročilý a expert. Cílem je porovnat, jak tyto editory splňují potřeby a očekávání každého typu uživatele. Na základě těchto informací můžeme poskytnout doporučení ohledně toho, který editor by mohl být nejvhodnější pro konkrétní skupinu uživatelů nebo pro konkrétní použití.

7.1 Představení testovaného software

7.1.1 Microsoft Word

Microsoft Word je nejznámějším textovým editorem v rámci kancelářských balíčků a nezbytným nástrojem pro tvorbu dokumentů ve firemním, školském i osobním prostředí. Jeho výjimečná funkčnost, spolehlivost a pokročilé možnosti formátování ho staví do čela průmyslu. Microsoft Word je software, který umožňuje vytvářet, upravovat dokumenty, formátovat texty, vkládat obrázky či jiné vytváření specifického obsahu. [10]

Dle [11] si zde uvedeme silné a slabé stránky tohoto editoru:

Silné stránky:

1. Funkčnost

- **Pokročilé formátování:**
 - Široké možnosti formátování textu, odstavců a dokumentů
 - Podpora pro vkládání a úpravu obrázků, grafů a dalších prvků
- **Nástroje pro spolupráci:**
 - Integrované možnosti spolupráce v reálném čase. Snadné komentování a sledování revizí dokumentů
- **Vkládání obsahu:**
 - Jednoduché vložení odkazů, obrázků, grafů
- **Tabulkový procesor:**
 - Pokročilé funkce pro práci s tabulkami a výpočty. Automatická aktualizace tabulek podle změn v datech

2. Použitelnost

- **Uživatelské rozhraní:**
 - Přehledné a intuitivní rozhraní s rychlým přístupem ke všem funkcím. Snadné ovládání a nastavení
- **Integrace s Office:**
 - Jednotné nastavení a design s ostatními aplikacemi v balíku Office. Synchronizace s dalšími službami Microsoft
- **Personalizace:**
 - Možnost nastavení uživatelských preferencí. Vytváření vlastních šablon a stylů pro rychlou práci.

3. Spolehlivost

- **Stabilita:**
 - Stabilní výkon a minimální chyby během používání (pravidelné aktualizace softwaru)

- **Aktivní podpora:**
 - Kvalitní technická podpora od společnosti Microsoft. Rozsáhlá komunita a fóra pro řešení problémů.
- **Zotavení aplikace po pádu:**
 - Automatické ukládání dokumentů

4. Výkonnost

- **Rychlost:**
 - Rychlá odezva na uživatelské příkazy. Minimální doba spuštění aplikace.
- **Automatizace:**
 - Podpora vytváření různých maker a automatizací
 - Možnost vytváření a využívání vlastních šablon
 - Dobře optimalizovaný kód pro efektivní využití systémových prostředků

5. Podpora

- **Pravidelné aktualizace**

Slabé stránky:

1. Funkčnost

- **Licenční náklady:**
 - Plná verze softwaru je zpoplatněná a může omezit dostupnost pro uživatele s menším rozpočtem
- **Příliš mnoho možností:**
 - Pro začátečníky může být rozhraní komplikované vzhledem k velkému množství funkcí s přeplněným menu

2. Použitelnost

- **Velikost souborů:**
 - Dokumenty mohou mít velkou velikost, což může zpomalit sdílení a ukládání při omezeném internetovém připojení.

- **Omezená personalizace na mobilním zařízení**

3. Spolehlivost

- **Nekompatibilita s alternativními editory:**

- Při otevření souboru z jiného editoru se může vyskytnout chyba v podobě špatného formátování nebo ztrátou některých z funkcí

4. Výkonnost

- **Náročnost na hardware:**

- Pro plné využití všech funkcí může být vyžadován výkonnější hardware, což může být problém na starších zařízeních.
- Při otevření velkých a rozsáhlých dokumentů se může snížit výkon aplikace

5. Podpora

- **Kompatibilita s novějšími verzemi:**

- Nové verze mohou mít odlišné uživatelské rozhraní

- **Závislost na aktualizacích:**

- Některé funkce jsou pouze dostupné jen na nejaktualnější verzi software

7.1.2 Google Dokumenty

Google Dokumenty je online textový editor poskytovaný společností Google jako součást jejich kancelářského balíku služeb Google Workspace (dříve G Suite). Jeho přínosy spočívají ve spolupráci v reálném čase, přístupu k dokumentům z libovolného zařízení s připojením k internetu a jednoduchém sdílení s ostatními uživateli. [12]

Dle [13] jsou vyjmenované silné a slabé stránky v návaznosti na modelu FURPS.

Silné stránky Google Dokumentů:

1. Funkčnost

- **Online sharing:**
 - Více uživatelů mohou pracovat na jednom dokumentu současně
- **Automatické ukládání:**
 - Díky tomu, že dokument je uložen v cloudu, je zde minimální pravděpodobnost ztráty dat
- **Velký výběr šablon:**
 - Nabízí mnoho šablon pro rychlé vytvoření dokumentů s profesionálním vzhledem.

2. Použitelnost

- **Jednoduché rozhraní:**
 - Uživatelsky přívětivé převážně těm, kteří již používali program z ekosystému Google
- **Snadné sdílení:**
 - Možnost jednoduchého sdílení pomocí e-mailových pozvánek nebo odkazů

3. Spolehlivost

- **Dostupnost z jakéhokoli zařízení:**
 - Ke svým dokumentům můžete přistupovat z libovolného zařízení, stačí se pouze přihlásit do svého Google účtu
- **Aktualizace změn v reálném čase**
 - Umožňuje sledovat změny provedené ostatními uživateli v reálném čase.
- **Jednoduchá obnova verzí**

4. Výkonnost

- **Využití méně výkonu díky cloudovým službám**
- **Možnost získání oznámení:**

- Upozornění na změny a komentáře mohou být odeslány přes e-mail nebo přímo do aplikace

5. Podpora

- **Automatické aktualizace**
 - Aktualizace jsou prováděny automaticky bez nutnosti uživatelského zásahu
- **Dlouhodobá možnost uložení na Cloudu:**
 - Možnost uložení svých dokumentů na dlouhou dobu

Slabé stránky:

1. Funkčnost

- **Omezený výběr pokročilých funkcí:**
 - Můžou chybět některé pokročilé funkce – využití maker, pokročilé formátování

2. Použitelnost

- **Omezená personalizace prostředí:**
 - Může mít omezené možnosti personalizace ve srovnání s některými desktopovými textovými editory – uspořádání karet a funkcí

3. Spolehlivost

- **Nekompatibilita s textovými editory:**
 - Omezená kompatibilita s jinými alternativními textovými editory – soubor se otevře, ale s odlišným formátováním. Některé funkce také nemusejí být přeneseny z jiného textového editoru

4. Výkonnost

- **Závislost na internetovém připojení:**
 - Během rychlejšího vkládání obsahu se můžou vyskytnout záseky
 - Nelze přistoupit k souborům, pokud uživatel není online – offline přístup je možné nastavit skrz nainstalování aplikace

- **Omezené možnosti editace obrázků:**
 - Pro pokročilejší úpravy obrázků je nutno použít software třetích stran

5. Podpora

- **Nutnost vlastnit Google účet:**
 - Používání Google Dokumentů vyžaduje aktivní Google účet, což může být problém pro ty, kteří preferují jiné služby.

7.1.3 LibreOffice Writer

LibreOffice Writer je svobodný a otevřený textový procesor, který poskytuje širokou škálu funkcí pro tvorbu a formátování dokumentů. Tento software je součástí kancelářského balíčku LibreOffice, který je k dispozici zdarma a nabízí uživatelům možnost pracovat s textem efektivně a profesionálně. [14]

Díky své otevřené povaze je LibreOffice Writer aktivně vyvíjen komunitou dobrovolníků, což znamená pravidelné aktualizace a zdokonalování. Tento software je ideální pro ty, kteří hledají kvalitní alternativu k placeným textovým procesorům s ohledem na flexibilitu a bezplatnou dostupnost. [14]

Dle [15] jsem vyzdvihl silné a slabé stránky tohoto textového editoru za pomoci modelu FURPS.

Silné stránky:

1. Funkčnost
 - **Kompatibilita s různými formáty:**
 - Tento textový editor podporuje širokou škálu formátů, včetně formátů používaných v Microsoft Word
 - **Rozšířené nástroje formátování**
 - Nabízí pokročilé možnosti formátování textu, stylů a odstavců.
 - **Vložení obrázků a grafiky:**
 - Umožňuje snadné vkládání obrázků a jejich úpravu

2. Použitelnost

- **Jednoduché rozhraní:**
 - Uživatelsky velmi přívětivé rozhraní s intuitivním ovládáním
- **Průvodce:**
 - Obsahuje průvodce, který slouží pro usnadnění orientace a používání jednotlivých funkcí

3. Spolehlivost

- **Ukládání na lokální disk:**
 - Dokumenty lze ukládat na lokálním zařízení nebo v cloudu dle uživatelské preference
- **Vývoj ze strany uživatelů:**
 - LibreOffice je aktivně vyvíjen uživatelskou komunitou, díky tomu, že se jedná o Open-source software

4. Výkonnost

- **Automatická korektura:**
 - Nabízí nástroje pro automatickou kontrolu pravopisu
- **Zdarma open-source software:**
 - LibreOffice má otevřený zdrojový kód a zároveň je zdarma
- **Možnost importování a převodu jakéhokoliv typu dokumentu z jiného textového editoru**

5. Podpora

- **Komunitní vývoj softwaru**
 - Otevřený zdrojový kód s podporou aktivní komunity vývojářů
- **Pravidelné aktualizace softwaru**

Slabé stránky:

1. Funkčnost

- **Podpora profesionálních nástrojů:**
 - Některé složitější nástroje jako například makra a jiné nástroje nejsou dostupné nebo podporovány
- **Podpora spolupráce a sdílení:**

- LibreOffice nenabízí online spolupráce na jednom dokumentu v reálném čase. Nenabízí cloudové ukládání
2. Použitelnost
- **Uživatelské rozhraní se může zdát zastaralé:**
 - LibreOffice nabízí jednoduché uživatelské rozhraní, avšak může působit zastaralým dojmem
 - **Menší výběr ze šablon či doplňků**
 - LibreOffice nemá tak rozšířený výběr ze šablon či instalaci doplňků
3. Spolehlivost
- **Omezenější online spolupráce:**
 - Funkce pro online spolupráci nemusí být tak robustní jako u některých cloudových platforem
4. Výkonnost
- **Absence cloudové integrace:**
 - Integrace s cloudovými úložišti nemusí být tak plynulá jako u jiných textových editorů
5. Podpora
- **Podpora na platformách:**
 - Uživatelé operačního systému ChromeOS bohužel nemohou nainstalovat tento editor a pracovat s ním
 - **Ekosystém:**
 - Nemá tak rozsáhlý ekosystém doplňků a šablon jako některé komerční alternativy

8 Vyhodnocení a porovnání textových editorů dle stanovených kritérií

K porovnání hodnocení třech nejnámějších textových editorů – Microsoft Word, Google Dokumenty a LibreOffice Writer, jsem shrnul do tabulky níže.

První sloupec tabulky obsahuje charakteristiky modelu FURPS se sledovaným kritériem. Váhy každé kategorie jsou nastaveny podle důležitosti dané kategorie (Funkčnost 25 %, Použitelnost 20 %, Spolehlivost 20 %, Výkonnost 15 %, Podpora 10 %, Cena 10 %)

Bodování:

Každá vlastnost je hodnocena na stupnici od 1 do 5, kde číslo 1 reprezentuje nízkou kvalitu a číslo 5 vysokou kvalitu.

Celkové skóre:

Celkové skóre každého softwaru je spočítáno jako vážený průměr všech hodnocených vlastností (body * váha).

Kategorie	Váha	MS Word	Google Dokumenty	LibreOffice Writer
Funkčnost (Formátování, Nástroje)	25 %	5	4	3
Použitelnost (Ovládání, Rozhraní)	20 %	5	4	4
Spolehlivost (Stabilita, Aktualizace)	20 %	4	4	4
Výkonnost (Rychlost, Sdílení)	15 %	4	4	3

Podpora (podpora, ekosystém)	10 %	5	4	3
Cena	10 %	3	5	5
Celkové skóre	100 %	4,45	4,1	3,6

Tabulka 1 – Vyhodnocení kritérií (zpracování vlastní)

Podle celkového skóre bylo zjištěno, že Microsoft Word vede před Google Dokumenty a LibreOffice Writer. Toto pořadí ukazuje na výraznou konkurenceschopnost Microsoft Word v rámci sledovaných kritérií FURPS.

Microsoft Word (4,45):

- Microsoft Word vyniká ve funkčnosti, spolehlivosti a výkonu.
- Uživatelské rozhraní a schopnost uživatelů naučit se jej jsou silné stránky Wordu.
- Celkově se jedná o rozsáhlý a výkonný balík funkcí.

Google Dokumenty (4,1):

- Google Dokumenty jsou konkurenceschopné, především ve funkčnosti a použitelnosti.
- Synchronizace a bezplatnost jsou výrazné výhody.
- Rychlost a podpora na ostatních zařízeních jsou silné stránky.

LibreOffice Writer (3,6):

- LibreOffice Writer má nižší skóre, především v oblasti funkčnosti a výkonu.
- Je stále konkurenceschopný, zejména pro nenáročného uživatele preferující otevřený software.
- Nízké skóre může být způsobeno menší škálou funkcí ve srovnání s ostatními.

9 Vyhodnocení softwaru dle typu uživatelů

Dalším cílem bakalářské práce bylo vyhodnocení software vzhledem k typu uživatele. Jaký uživatel by byl nejvhodnější k danému textovému editoru.

Nejprve si stanovíme kritéria a definujeme si uživatele.

9.1 Typ uživatele začátečník

Pro definování uživatele začátečník, bereme v potaz, že tento typ uživatele preferuje jednoduché ovládání a co nejsnadnější zacházení se softwarem. Dále očekáváme, že tento typ uživatele je naprosto nenáročný a textový editor používá pouze pro psaní textu bez složitějších funkcí jako jsou makra a složité formátování.

1. Funkčnost
 - Jednoduché a přehledné nástroje
 - Snadné vkládání obrázků a jednoduchá manipulace
2. Použitelnost
 - Snadné ovládání
 - Intuitivní prostředí
3. Spolehlivost
 - Stabilní aplikace
 - Snadné obnovení v případě výpadku
4. Výkonnost
 - Rychlé možnosti otevírání a ukládání souborů
 - Synchronizace a sdílení s ostatními zařízeními
5. Podpora
 - Aktualizace a údržba bez složitých aktualizací

9.2 Typ uživatele pokročilý

Pokud chceme definovat typ uživatele pokročilý. Budeme brát v úvahu, že tento uživatel potřebuje využití pokročilejších nástrojů, vytváření šablon, možnost

personalizovat aplikaci, nástroje pro správu velkého množství dat a tabulek s rozšířenějším nastavení software.

1. Funkčnost

- Pokročilejší nástroje pro stylizaci a formátování textu
- Vkládání obrázků a tabulek

2. Použitelnost

- Používání pokročilých funkcí
- Personalizace aplikace

3. Spolehlivost

- Stabilní aplikace při pokročilejších operacích
- Schopnost obnovy dat při výpadku

4. Výkonnost

- Rychlé načítání velkých souborů
- Export do používaných formátů
- Synchronizace s ostatními zařízeními

5. Podpora

- Pokročilé nastavení aplikace a možností

9.3 Typ uživatele expert

Tento náročný typ uživatele potřebuje využít rozsáhlé možnosti pro práci s makry a automatizacemi. Používá profesionální nástroje pro formátování a úpravy obsahu. Preferuje maximální rychlost načítání obsahu bez ztráty dat v případě výpadku.

1. Funkčnost

- Rozsáhlé možnosti formátování,
- Práce s makry
- Detailní kontrola nad složitými tabulkami (např. kontingenční tabulky)
- Využití filtrů

2. Použitelnost

- Personalizace aplikace
- Plná kontrola nad všemi aspekty programu
- Vytváření vlastních šablon

3. Spolehlivost

- Stabilní aplikace
- Schopnost automatického ukládání na cloudové úložiště

4. Výkonnost

- Rychlé načítání rozsáhlých souborů
- Synchronizace dokumentů do ostatních zařízeních

5. Podpora

- Detailní přizpůsobení aplikace, přidání doplňků

Hodnocení a volba textového editoru závisí na různých faktorech, jako jsou potřeby uživatelů, jejich úroveň zkušeností a požadavky na funkcionalitu. Po pečlivém zvážení stanovených kritérií s využitím modelu FURPS a specifických preferencí každé skupiny uživatelů jsme dospěli k závěru, že pro každou úroveň uživatele existuje optimální volba.

Začátečník:

Pro začátečníky se jeví jako nejlepší volba **LibreOffice Writer** nebo v kombinaci s **Google Dokumenty**. Tento textový editor nabízí jednoduché a přehledné uživatelské rozhraní, které umožňuje snadný a rychlý start bez komplikovaných nastavení. Bezplatná dostupnost je dalším zásadním faktorem, což usnadňuje novým uživatelům přístup k plnohodnotnému nástroji pro práci s textem.

Pokročilý:

Pro uživatele na pokročilé úrovni se jako optimální volba jeví **Microsoft Word**. Tento textový editor poskytuje rozsáhlé možnosti formátování, pokročilé nástroje pro správu obsahu a šablony, které usnadňují práci s rozsáhlými projekty. Profesionální vzhled, sdílení obsahu mezi zařízeními a využití ukládání do

cloudových úložišť jsou klíčovými faktory pro pokročilé uživatele, kteří vyžadují více než základní editaci textu.

Expert:

Pro expertní uživatele zůstává nejlepší volbou **Microsoft Word**. Tento software nejenže poskytuje nejvíce pokročilých nástrojů a možností pro formátování a automatizaci, ale také nabízí plnou integraci s dalšími aplikacemi v rámci balíku Microsoft Office. Pro profesionální práci a manipulaci s projekty nabízí Word optimální flexibilitu a škálovatelnost.

Celkově lze říci, že volba textového editoru závisí na konkrétních potřebách a preferencích uživatele. LibreOffice Writer je výbornou volbou pro ty, kdo začínají a chtějí solidní nástroj bez finanční investice.

Microsoft Word vyniká pro uživatele s pokročilými a expertními potřebami díky svým širokým funkcím a integrovaným možnostem v rámci balíku Office. Ve světě textových editorů existuje mnoho možností, a každý uživatel si může vybrat ten, který nejlépe vyhovuje jeho konkrétním požadavkům a preferencím.

10 Shrnutí výsledků

V rámci porovnání tří nejznámějších textových editorů, Microsoft Word, Google Dokumenty a LibreOffice Writer, jsme použili model FURPS k identifikaci silných a slabých stránek každého editoru. Následně jsme provedli hodnocení a porovnání pomocí rozsáhlé tabulky, kde se Microsoft Word umístil na prvním místě s celkovým skóre 4,45, následovaný Google Dokumenty s 4,1 a LibreOffice Writer s 3,6.

Definovali jsme tři typy uživatelů – začátečník, pokročilý a expert, a pro každého z nich jsme stanovili kritéria podle modelu FURPS. Na základě těchto kritérií jsme poté vyhodnotili nejvhodnější textový editor pro každou skupinu uživatelů. Začátečníkům se jeví jako optimální volba LibreOffice Writer, zatímco pokročilí a expertní uživatelé doporučují využívat Microsoft Word.

11 Závěry a doporučení

Každý textový editor od těchto společností má svoje pro a proti. Je tak pouze na uživateli, jestli preferuje spíše jednoduchý textový editor LibreOffice Writer, tedy editor, který využívá základní funkce a je tak dostatečný pro začátečníky či pokročilé uživatele. Nebo daný uživatel ocení i další programy v kancelářském balíku při zakoupení licence např. Microsoft Office a díky tomu sáhne spíše po textovém editoru Microsoft Word.

Na Microsoft Word musím ocenit jeho ekosystém s kancelářským balíkem Microsoft Office, který je mezi sebou skvěle provázaný. Musím zmínit i možnosti kolaborace mezi uživateli. Dále široký výběr možností formátování textů, tabulek či práci s pokročilejšími makry nebo možnost si textový editor upravit dle svých preferencí – stáhnout si doplňky, které usnadní práci s editorem.

Co se týče Google Dokumentů, zde bych vyzdvihl také možnost kolaborace a sdílení, které může usnadnit spolupráci a zvýšit efektivitu práce. Naopak pokud chceme použít Google Dokumenty v offline prostředí, tedy v prostředí bez připojení k internetu. Může být toto prostředí limitované, jelikož některé pokročilé funkce mohou být v tomto režimu omezené.

Posledním hodnoceným textovým editorem byl LibreOffice Writer, jeho open-source charakter a kompatibilita s různými formáty činí z LibreOffice Writer přitažlivý textový editor pro uživatele, kteří vyhledávají bezplatnou alternativu s bohatými možnostmi.

Výběr textového editoru závisí na potřebách a preferencích uživatele, ať už je začátečníkem, pokročilým uživatelem nebo expertem. Každý z těchto tří textových editorů má své specifické vlastnosti, které mohou vyhovovat různým uživatelským skupinám.

12 Seznam použité literatury

- [1] KAN, Stephen H. Metrics and models in software quality engineering. 2nd ed. Boston: Addison-Wesley, c2003. ISBN isbn0-201-72915-6.
- [2] ROUDENSKÝ, Petr. Kvalita softwaru: teorie a praxe. Upravené a rozšířené 2. vydání. Prostějov: Computer Media, 2018. ISBN 978-80-7402-322-4.
- [3] IEEE Standard For Software Quality Assurance Processes. 730-2014. USA: IEEE, 2014. ISBN 978-0-7381-9168-3..
- [4] ISO/IEC 25010: Systems and software engineering – Systems and software Quality Requirements and Evaluation. Ženeva, Švýcarsko: International Organization for Standardization, 2011
- [5] Towards Customized Smart Government Quality Model, International Journal of Software Engineering & Applications (IJSEA), Vol.9, No.2, Lincoln University College, Malajsie, 2021
- [6] ROUDENSKÝ, Petr a Anna HAVLÍČKOVÁ. Řízení kvality softwaru: průvodce testováním. Brno: Computer Press, 2013. ISBN 978-80-251-3816-8.
- [7] BUREŠ, Miroslav, Miroslav RENDA, Michal DOLEŽEL, Peter SVOBODA, Zdeněk GRÖSSL, Martin KOMÁREK, Ondřej MACEK a Radoslav MLYNÁŘ. Efektivní testování softwaru: klíčové otázky pro efektivitu testovacího procesu. Praha: Grada, 2016. Profesionál. ISBN 978-80-247- 5594-6.
- [8] Software. Správa sítě – slovník pojmů [online]. Praha, 2022 [cit. 2023-07-25]. Dostupné z: <https://www.sprava-site.eu/software/>
- [9] ISO/IEC 9126. Software engineering - Product quality: Part 1: Quality model. Ženeva, Švýcarsko: International Organization for Standardization, 2001 [cit. 2023-10-10]. Dostupné z: <https://www.iso.org/>
- [10] Microsoft Word. Alza.cz [online]. 2020 [cit. 2023-11-02]. Dostupné z: <https://www.alza.cz/word-navod>
- [11] Advantages and Disadvantages of MS Word. Javatpoint [online]. 2023 [cit. 2023-11-05]. Dostupné z: <https://www.javatpoint.com/advantages-and-disadvantages-of-ms-word>
- [12] Google Dokumenty je online textový editor poskytovaný společností Google jako součást jejich kancelářského balíku služeb Google Workspace (dříve G Suite). Jeho přínosy spočívají ve spolupráci v reálném čase, přístupu k dokumentům z libovolného zařízení s připojením k internetu a jednoduchém sdílení s ostatními uživateli. Online. Google Docs. 2023. Dostupné z: https://www.google.com/intl/cs_CZ/docs/about/. [cit. 2023-11-06].

- [13] Advantages and Disadvantages of Google Docs. Online. Profolus. 2023. Dostupné z: <https://www.profolus.com/topics/advantages-disadvantages-of-google-docs/>. [cit. 2023-11-06].
- [14] Writer. Online. LibreOffice The Document foundation. 2023. Dostupné z: <https://cs.libreoffice.org/discover/writer/>. [cit. 2023-11-06].
- [15] LibreOffice Review. Online. PCmag. 2023. Dostupné z: <https://www.pcmag.com/reviews/libreoffice>. [cit. 2023-11-08].

Zadání bakalářské práce

Autor: Adam Málek

Studium: I1900615

Studijní program: B0688A140001 Informační management

Studijní obor: Informační management

Název bakalářské práce: **Hodnocení kvality softwaru**

Název bakalářské práce AJ: Software quality evaluation

Cíl, metody, literatura, předpoklady:

Analýza metod pro hodnocení kvality softwaru, jejich porovnání, přínosy, negativa a využití.

- Roudenský, P. (2016). Kvalita softwaru: teorie a praxe. Computer Media.
- Bureš, M., Renda, M., & Doležel, M. (2016). Efektivní testování softwaru: klíčové otázky pro efektivitu testovacího procesu. Grada Publishing as.
- Sommerville, I. (2013). Softwarové inženýrství. Computer Press, Albatros Media as.

Literatura bude doporučena zadavatelem.

Zadávací pracoviště: Katedra informačních technologií,
Fakulta informatiky a managementu

Vedoucí práce: doc. Ing. Hana Tomášková, Ph.D

Datum zadání závěrečné práce: 2.8.2021