



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA PODNIKATELSKÁ

FACULTY OF BUSINESS AND MANAGEMENT

## ÚSTAV INFORMATIKY

INSTITUTE OF INFORMATICS

# NÁVRH, VÝVOJ A INTEGRACE VYHLEDÁVAČE PRO AGREGACI INFORMACÍ Z RŮZNÝCH ZDROJŮ

DESIGN, DEVELOPMENT AND INTEGRATION OF SEARCH ENGINE FOR AGGREGATION OF  
INFORMATION FROM VARIOUS SOURCES

## BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

## AUTOR PRÁCE

AUTHOR

Danila Kiselev

## VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Bernard Neuwirth, Ph.D., MSc

BRNO 2021

# Zadání bakalářské práce

Ústav:	Ústav informatiky
Student:	<b>Danila Kiselev</b>
Studijní program:	Systémové inženýrství a informatika
Studijní obor:	Manažerská informatika
Vedoucí práce:	<b>Ing. Bernard Neuwirth, Ph.D., MSc</b>
Akademický rok:	2020/21

Ředitel ústavu Vám v souladu se zákonem č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů a se Studijním a zkušebním řádem VUT v Brně zadává bakalářskou práci s názvem:

## **Návrh, vývoj a integrace vyhledávače pro agregaci informací z různých zdrojů**

### **Charakteristika problematiky úkolu:**

Úvod  
Cíle práce, metody a postupy zpracování  
Teoretická východiska práce  
Analýza současného stavu  
Vlastní návrhy řešení  
Závěr  
Seznam použité literatury  
Přílohy

### **Cíle, kterých má být dosaženo:**

Cílem bakalářské práce je návrh a implementace části webové aplikace pro realizaci servisu vyhledávání informací ve firmě. Dílčím cílem práce je provedení analýzy současného stavu řešené problematiky ve firmě, z jejichž výsledků bude v navrhovaném řešení vycházeno. Součástí práce bude ekonomické zhodnocení navrhované varianty řešení.

### **Základní literární prameny:**

BASL, Josef a Roman BLAŽÍČEK. Podnikové informační systémy: Podnik v informační společnosti. 3. vyd. Praha: Grada, 2012. 323 s. ISBN 978-80-247-4307-3.

GÁLA, Libor, Jan POUR a Zuzana ŠEDIVÁ. Podniková informatika. 3. vyd. Praha: Grada Publishing, 2015. 240 s. ISBN 978-80-247-5457-4.

LEISS, Oliver a Jasmin SCHMIDT. PHP v praxi: pro začátečníky a mírně pokročilé. Praha: Grada, 2010. ISBN 978-80-247-3060-8.

PILGRIM, Mark. Ponořme se do HTML5. Praha: CZ.NIC, z.s.p.o., 2015. ISBN 978-80-905802-6-8.

SODOMKA, Petr a Hana KLČOVÁ. Informační systémy v podnikové praxi. 2. vyd. Brno: Computer Press, 2010. 504 s. ISBN 978-80-251-2878-7.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2020/21

V Brně dne 28.2.2021

L. S.

---

Mgr. Veronika Novotná, Ph.D.  
ředitel

---

doc. Ing. Vojtěch Bartoš, Ph.D.  
děkan

## **Abstrakt**

Bakalářská práce se zaměřuje na vylepšení informačního systému firmy Red Hat. V práci se budu zabývat usnadněním procesu vyhledání informací. Při práci se seznámím s jazykem Python a jeho moduly, navrhnou změny, které by měly mít za následek uspořádaní času specialistů tak, aby se mohli plně věnovat své primární práci, kterou je řešení problému zákazníků.

## **Abstract**

The bachelor thesis focuses on the improvement of information system in the company Red Hat. In this thesis I will be dealing with facilitating the process of information retrieval. During the work I will get acquainted with the Python language and its modules, I will propose changes that should result in saving the time of specialists so that they can fully devote themselves to their primary job, which is solving customer problems.

### Klíčové slova

Openshift, Red Hat, vyhledáváč, Python, Jinja2, HTML, agregator

### Keywords

Openshift, Red Hat, search engine, Python, Jinja2, HTML, aggregator

## Čestné prohlášení

Prohlašuji, že předložená diplomová práce je původní a zpracoval jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem ve své práci neporušil autorská práva (ve smyslu Zákona č. 121/2000 Sb., o právu autorském a o právech souvisejících s právem autorským).

V Brně dne 23. května 2021

.....

## Poděkování

Dovoluji si tímto srdečně poděkovat vedoucímu bakalářské práce Ing. Bernardu Neuwirthovi, Ph.D., MSc za odbornou pomoc, věcné rady a cenné připomínky při zpracovávání této bakalářské práce.

<b>Úvod</b>	<b>6</b>
<b>Cíle práce, metody a postupy zpracování</b>	<b>6</b>
<b>Teoretická východiska práce</b>	<b>7</b>
1.1. Informace	7
1.2. Informační systém	8
1.3. Metodika tvorby informačního systému	8
1.4. Analýza podnikových procesů	9
1.4.1. SWOT analýza	9
1.5. Vývojový diagram	10
1.6. GNU/Linux	11
1.6.1. Open-source	11
1.6.2. GNU	12
1.6.3. Jádro Linux	12
1.6.4. Příkazový řádek (bash)	13
1.6.5. Distribuce GNU/Linux	13
1.6.6. Ubuntu	15
1.6.7. Fedora	15
1.6.8. Red Hat Enterprise Linux	16
1.7. HTML	16
1.6. Jazyk programování	17
1.6.1. Definice	17
1.6.2. PHP	17
1.6.3. Python	18
1.6.3.1. Popis jazyku	18
1.6.3.2. Jinja2	19
1.6.3.3. Flask	19
<b>Analýza současného stavu</b>	<b>20</b>
2.1. Informace o firmě	20
2.2. Popis společnosti	20
2.3. Organizační struktura	21
2.5. SWOT analýza projektu	22
2.6. Analýza hlavního procesu týmu	23
2.7. Průzkum	28
2.8. Shrnutí výsledků analýz stávajícího stavu	32
<b>Vlastní návrhy řešení</b>	<b>32</b>
3.1. Popis návrhu	32

3.2. Iterace 1	33
3.3. Iterace 2	34
3.4. Forma a náplň	35
3.5. Volba technologií	35
3.6. Instalace	36
3.7. Popis souborů	37
3.8. Instrukce k použití	38
3.9. Zaškolení	40
3.10. Ekonomické zhodnocení	41
<b>Závěr</b>	<b>42</b>
<b>Seznam použitých zdrojů</b>	<b>42</b>
<b>Seznam obrázků</b>	<b>43</b>



## Úvod

Význam této bakalářské práce spočívá v navržení funkčního vyhledávacího agregačního systému pro tým podpory zákazníků projektu Openshift ve firmě Red Hat. S tímto problémem jsem se setkal sam jako pracovník tohoto projektu. Velkou část času práci nad případem specialista tráví vyhledáváním informací použitím několik různých servisů co vyvolává zmatek a komplikace.

Přitom množství případů a pracovníků rapidně stoupá. Na základě těchto informací věřím, že i malé zlepšení tohoto procesu přivede k zvýšení efektivity týmu.

## Cíle práce, metody a postupy zpracování

Hlavní cíl práce je zlepšení efektivity procesu v týmu prostřednictvím zrychlení vyhledávání informací a agregaci všech zdrojů na stejném místě.

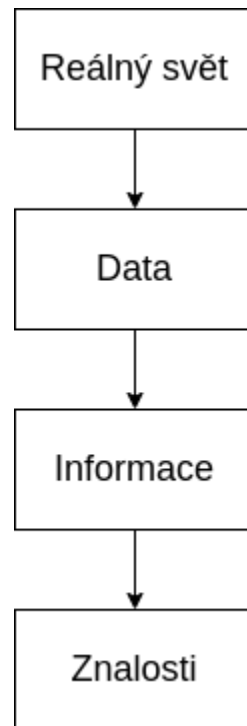
K dosažení cíle je třeba provést následující úkoly:

1. Analýza stávajícího stavu ve firmě.
2. Analýza potřeby navrhovaného řešení.
3. Získávání očekávání od systému a potřebných funkcí a vzhledu.
4. Zpracování podrobného návrhu řešení.
5. Spolupráce se všemi příslušnými členy firmy ohledně realizaci projektu.
6. Vyvoj první verzi programu s využitím jazyků Python a HTML.
7. Analýza hotového řešení a stanovení dalších kroků pro rozvoj projektu.

# 1. Teoretická východiska práce

## 1.1. Informace

Jsou možné různé definice pojmu informace. Jeden z nich je chápání informace jako jeden z elementů tzv. zpracovatelského řetězce.



**Zpracovatelský řetězec [1]**

Obrázek 1

V takovém řetězci jsou data z reálného světa převedena do elektronické podoby jako informace zachycující konkrétní událost, tyto informace jsou následně zpracovány pomocí pravidel a koncoví uživatelé z nich získávají potřebné znalosti. [1]

## 1.2. Informační systém

IS tvoří lidé, kteří pomocí dostupných technologických nástrojů a zavedené metodiky zpracovávají obchodní data a vytvářejí informační a znalostní základnu

organizace sloužící k řízení obchodních procesů, manažerskému rozhodování a řízení obchodní agendy [3].

### 1.3. Metodika tvorby informačního systému

Při navrhování informačního systému je důležité vědět, jací lidé jej budou používat, a pokud to podmínky dovolují, co nejvíce jej přizpůsobit jeho uživatelům. Protože informace poskytované systémem pomáhají nejen komunikaci uvnitř firmy, ale také obchodním procesům, které jsou důležité pro úspěšné fungování firmy. Správně fungující informační systém je proto nedílnou součástí každé úspěšné a prosperující společnosti. [2]

Hlavní požadavky na zavedení informačního systému jsou:

- Spolehlivost a výkonnost,
- Snadná obsluha a správa systému,
- Dosažitelná návratnost investice. [3]

### 1.4. Analýza podnikových procesů

Smyslem analýzy podnikových procesů je zjistit, jaký je současný stav řízení podniku v oblastech (prodej, nákup, výroba atd.), které má řešit plánovaná aplikace, kde jsou problémy v řízení a požadavky na jeho další rozvoj. Rozsah analýzy se podle řešené aplikace liší - od dílčí oblasti např. řízení prodeje nebo řízení vztahu k zákazníkům (CRM) až po analýzu celého podnikového řízení odpovídající zejména celopodnikovým aplikacím (ERP). [2]

#### 1.4.1. SWOT analýza

Při celkovém hodnocení dílčích údajů získaných z různých analýz vnějšího a vnitřního prostředí se používá analýza SWOT. Analýza SWOT má čtyři klíčové body, které jsou vyjádřeny počátečními písmeny příslušných anglických názvů:

- S - Strengths - Silné stránky,
- W - Weaknesses - Slabé stránky,
- O - Opportunities - Příležitosti,
- T - Threats - Hrozby.

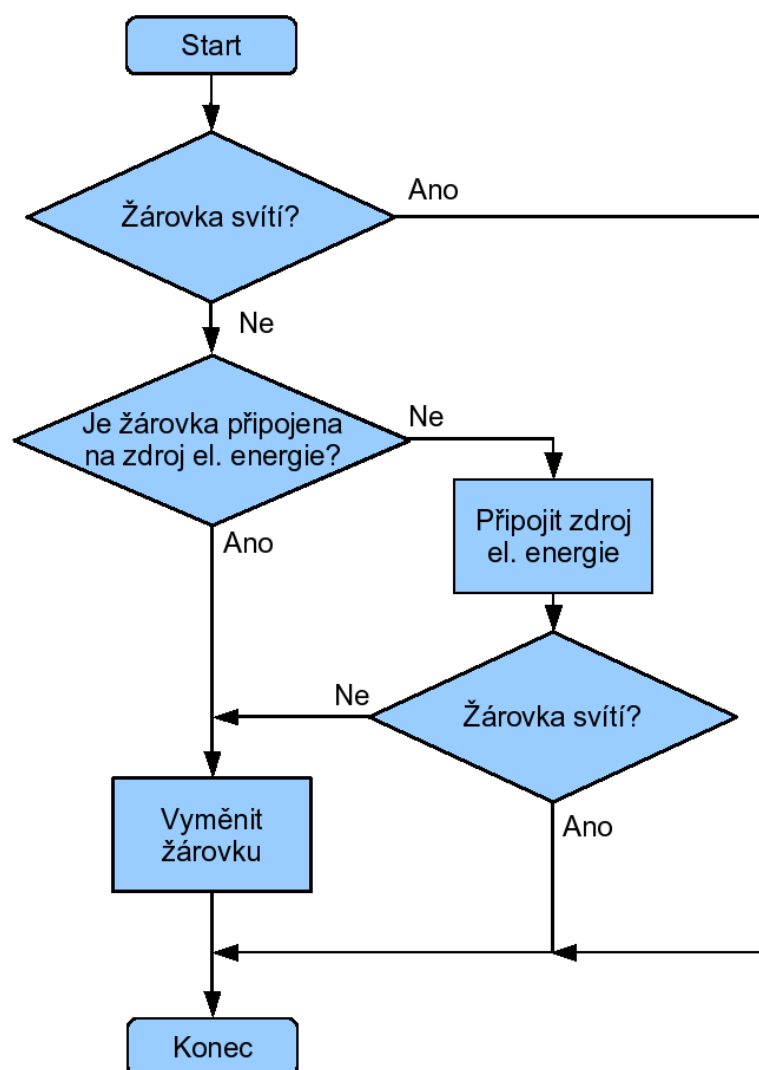
Analýza SWOT identifikuje základní faktory, které jsou pro danou oblast rozhodující. Úspěch podniku závisí na tom, jak podnik dokáže využívat nově vznikající příležitosti a své silné stránky a jak se mu daří eliminovat své slabé stránky a předcházet potenciálním hrozbám. Podle kombinace a porovnání těchto faktorů může podnik zvolit strategii a další kroky. [10]

### 1.5. Vývojový diagram

Vývojový diagram - běžný typ schémat (grafických modelů) popisujících algoritmy nebo procesy, v nichž jsou jednotlivé kroky znázorněny jako bloky různých tvarů spojené čarami označujícími směr posloupnosti.

Symbole vývojového diagramu:

- šipka** — určuje směr zpracování
  - svislé nebo vodorovné čáry
  - mohou se křížit nebo spojovat
  - směr dolů a doprava je prioritní, v tomto případě není nutné použít šipky
  - šipky se používají jenom v případě, že tento směr je jiný, nebo když je třeba směr toku informace zvýraznit, například při znázornění iterace
- obdélník s popisem** — definuje dílčí krok zpracování
- kosočtverec** — větvení postupu v závislosti na splnění podmínky, viz skok
- obdélník se zaoblenými rohy** — počátek nebo ukončení zpracování
- kruh** — spojka několika šipek



Obrázek 2

## 1.6. GNU/Linux

GNU/Linux je označení pro svobodný a otevřený počítačový operační systém, který je založený na linuxovém jádru.

### 1.6.1. Open-source

Otevřený software (anglicky open-source software) je počítačový software s otevřeným zdrojovým kódem. Otevřenost zde znamená jak technickou dostupnost kódu,

tak legální (ne)dostupnost – licenci software. Licence je velmi důležitá, jelikož specifikuje, jaká práva k otevřenému kódu získá a jak s ním může nakládat jeho uživatel. Některé licence (ku příkladu GNU) například dovolují uživatelům zdrojový kód upravovat a distribuovat (na rozdíl od proprietárního software).

### 1.6.2. GNU

Linux může vypadat různě v závislosti na distribuci, hardwaru a osobním vkusu, ale základy, na kterých jsou postavena všechna grafická a jiná rozhraní, zůstávají stejné. Systém Linux je založen na nástrojích GNU (Gnu's Not UNIX), které poskytují sadu standardních způsobů ovládání a používání systému.

Všechny nástroje GNU jsou open source, takže je lze nainstalovat na libovolný systém. Většina distribucí nabízí předkompilované balíčky většiny běžných nástrojů, například balíčky RPM v distribuci Red Hat a balíčky Debian (nazývané také deb nebo dpkg) v distribuci Debian, takže k instalaci balíčku do systému nemusíte být programátorem. Pokud si však rádi děláte věci sami, užijete si Linux o to lépe, že většina distribucí je dodávána s kompletní sadou vývojových nástrojů, které umožňují instalaci nového softwaru čistě ze zdrojových kódů. Toto nastavení vám také umožní instalovat software, i když neexistuje v předpřipravené podobě vhodné pro váš systém. [6]

### 1.6.3. Jádro Linux

Jádro Linuxu ("kostra" systému) není součástí projektu GNU, ale používá kód stejnou licenci jako software GNU. Velká většina nástrojů a vývojových nástrojů ("maso" systému), které nejsou specifické pro Linux, je převzata z projektu GNU. Protože každý použitelný systém musí obsahovat jak jádro a alespoň minimální sadu nástrojů, někteří lidé tvrdí, že takový systém by se měl nazývat GNU/Linux. [6]

#### 1.6.4. Příkazový řádek (bash)

V základu je shell jednoduše makroprocesor, který vykonává příkazy. Termín makro znamená funkci, při níž se text a symboly rozšiřují a vytvářejí větší výrazy.

Unixový shell je interpret příkazů i programovací jazyk. Jako interpret příkazů poskytuje shell uživatelské rozhraní k bohaté sadě nástrojů GNU. Na adrese funkce programovacího jazyka umožňují tyto nástroje kombinovat. Soubory obsahující příkazy lze vytvořit a oni tehdy se stávají příkazy. Tyto nové příkazy mají stejný status jako systémové příkazy v adresářích, jako je /bin, což umožňuje uživatelům nebo skupinám vytvořit vlastní prostředí pro automatizaci svých běžných úloh.

Shelly lze používat interaktivně nebo neinteraktivně. V interaktivním režimu přijímají vstup zadávaný z klávesnice. Při neinteraktivním spuštění shelly provádějí příkazy načtené ze souboru.

Shell umožňuje provádět příkazy GNU synchronně i asynchronně. Shell čeká na dokončení synchronních příkazů, než přijme další vstup; asynchronní příkazy se nadále vykonávají paralelně se shellem, zatímco ten čte a vykonává další příkazy. Konstrukce přesměrování umožňují jemné řízení vstupu a výstupu těchto příkazů. Kromě toho shell umožňuje kontrolu nad obsahem prostředí příkazů. [7]

#### 1.6.5. Distribuce GNU/Linux

Distribuce GNU/Linuxu je obecná definice operačních systémů využívajících jádro Linuxu, připravených k instalaci na uživatelské zařízení. Operační systém dodávaný jako součást distribuce se skládá z jádra Linuxu a obvykle obsahuje sadu knihoven a nástrojů vytvořených projektem GNU, grafický subsystém X Window System (poslední dobu i Wayland) a sadu aplikací, jako jsou editory dokumentů a tabulek, přehrávače médií, databázové systémy atd. Menší distribuce nemusí obsahovat subsystém X a mohou nabízet alternativy k nástrojům GNU, jako jsou musl, Busybox, uClibc nebo dietlibs, aby ušetřily náklady.



V současné době existuje více než šest set distribucí GNU/Linuxu; více než polovina z nich je udržována v aktuálním stavu pravidelnými aktualizacemi od vývojářů distribucí.

Vzhledem k tomu, že jádro a doprovodný software jsou většinou open source, lze určitou distribuci GNU/Linuxu nainstalovat na širokou škálu hardwaru, například na domácí počítač, server, notebook nebo netbook, smartphone nebo tablet. Některé distribuce GNU/Linuxu jsou navíc speciálně navrženy pro nasazení v prostředích s omezenými zdroji (například řešení založená na mikrokontrolérech nebo velmi starý hardware).

Existují distribuce vyvíjené jak s komerční podporou (Red Hat / Fedora, SLED / OpenSUSE, Ubuntu), tak výhradně dobrovolníky (Debian, Slackware, Gentoo, ArchLinux).

Základní principy zůstávají v každé distribuci stejné:

- Vždy existuje linuxové jádro (základní součást operačního systému Linux), které se skládá ze dvou částí.  
systému)
- výchozí software GNU (nástroje jako ls, rm atd.)
- obecný software, který se od linuxové distribuce očekává (textové editory atd.).

To, co se liší mezi jednotlivými distribucemi, je obvykle:

- Instalační software (pro instalaci softwaru nebo operačního systému)
- Obecný software: (kancelářské aplikace, prog. jazyky, hry, webový software atd.)
- Dokumentace a příručky (kvalita, nedostatek, množství)
- Cena - zda za distribuci nezaplatíte nic, málo nebo hodně, závisí na tom, co od ní potřebujete, a na obchodním modelu, podle kterého distributor pracuje.
- Kvalita softwaru (chybový nebo nebugový software, nejnovější verze softwaru)
- Zda je aktuální, nebo ne
- Zda distributor nabízí dobrý kanál podpory, nebo ne
- Jak snadno se celkově používá. [9]

### 1.6.6. Ubuntu



Obrázek 3

Ubuntu je distribuce GNU/Linuxu založená na systému Debian GNU/Linux. Jeho hlavním vývojářem a sponzorem je společnost Canonical. Projekt je v současné době aktivně rozvíjen a podporován svobodnou komunitou.

Společnost Canonical tvrdí, že Ubuntu používá přibližně 20 milionů uživatelů po celém světě. Je na prvním místě v seznamu nejoblíbenějších distribucí GNU/Linuxu pro webové servery. Z hlediska počtu uživatelů, kteří navštívili DistroWatch.com (za poslední rok), je na 5. místě [8].

### 1.6.7. Fedora



Obrázek 4

Fedora (dříve Fedora Core) je linuxová distribuce sponzorovaná společností Red Hat, která obsahuje funkce, jež mají být v budoucnu použity v distribuci Red Hat Enterprise Linux. Cílem projektu je vytvořit kompletní operační systém ze svobodného softwaru, který bude vytvořen komunitou v duchu ekosystému Red Hat Linux. Verze jsou vydávány každých 6-8 měsíců podle veřejného harmonogramu.

Současné verze distribuce dostávají během období podpory průběžně aktualizace softwaru a knihoven, včetně nových stabilních verzí linuxového jádra. Fedora obsahuje software šířený pod různými svobodnými licencemi a snaží se být v čele open source technologií.

## 1.6.8. Red Hat Enterprise Linux

# Red Hat Enterprise Linux

Obrázek 5

Red Hat Enterprise Linux (neboli RHEL) je linuxová distribuce společnosti Red Hat.

Tato distribuce je určena pro firemní použití. Nové verze jsou vydávány v intervalu přibližně 3 let.

Hlavním rysem distribuční sady je komerční podpora na 10 let s možností prodloužení až na 13 let. Mnoho výrobců softwaru a hardwaru zařadilo RHEL mezi distribuce Linuxu, které podporují.

Verze Enterprise byla původně založena na systému Red Hat Linux, ale používala mnohem konzervativnější cyklus vydávání nových verzí. Později se pozornost přesunula na Fedoru.

Přibližně každá třetí verze Fedora Core se stává základem pro vydání verze Enterprise:

Fedora 19(23) → RHEL 7

Fedora 28 → RHEL 8

## 1.7. HTML

HTML je standardizovaný značkovací jazyk pro dokumenty na World Wide Webu. Většina webových stránek obsahuje popis značkovacího jazyka HTML (nebo XHTML). HTML je interpretováno prohlížeči a výsledný formátovaný text je zobrazen na monitoru počítače nebo mobilního zařízení.

Na celosvětové síti jsou stránky HTML obvykle odesílány do prohlížečů ze serveru prostřednictvím protokolu HTTP nebo HTTPS, a to v prostém textu nebo s použitím šifrování.

Je možné vložit do jazyka HTML kód JavaScriptu a ovládat tak chování a obsah webových stránek. Taky začlenění CSS do jazyka HTML popisuje vzhled a rozvržení stránky.

## 1.6. Jazyk programování

### 1.6.1. Definice

Programovací jazyk - formální jazyk určený k psaní počítačových programů. Programovací jazyk definuje soubor lexikálních, syntaktických a sémantických pravidel, která určují vzhled programu a činnosti, které bude vykonavatel (obvykle počítač) pod jeho kontrolou provádět.

Od vzniku prvních programovatelných strojů lidstvo vymyslelo více než osm tisíc programovacích jazyků (včetně esoterických, vizuálních a hraček). Jejich počet se každým rokem zvyšuje. Některé jazyky může používat jen malý počet jejich vývojářů, jiné se dostanou do povědomí milionů lidí. Profesionální programátoři mohou ovládat několik programovacích jazyků.

Programovací jazyk je určen k psaní počítačových programů, což je soubor pravidel, která umožňují počítači provádět určitý výpočetní proces, organizovat správu různých objektů apod. Programovací jazyk se od přirozených jazyků liší tím, že je určen k ovládní počítače, zatímco přirozené jazyky slouží především ke komunikaci mezi lidmi. Většina programovacích jazyků používá speciální konstrukce pro definici a manipulaci s datovými strukturami a řízení procesu výpočtu.

### 1.6.2. PHP

PHP (PHP: Hypertext Preprocessor) je univerzální skriptovací jazyk, který se intenzivně používá pro vývoj webových aplikací. V současné době je podporován

naprostou většinou poskytovatelů hostingu a je jedním z nejpoužívanějších jazyků pro tvorbu dynamických webových stránek.

Jazyk a jeho interpret (Zend Engine) vyvíjí skupina nadšenců v rámci open source projektu. Projekt je šířen pod vlastní licenci.

Syntaxe PHP je obdobná jako u jazyků C, Java a Perl, rozšířená o vlastní PHP vlastnosti jako např. o příkazy k integraci s databázemi. PHP dává vývojáři snadno naučitelné a zároveň mocné nástroje k vytvoření dynamických webových stránek.

PHP je použitelný jak pro UNIX, tak pro operační systém Windows. Velice oblíbené je použití PHP jako modulu na webovém serveru Apache, kromě toho je možná jeho realizace prostřednictvím CGI.

PHP nabízí možnost stáhnout určité informace ze serveru, aby je pak zobrazil v prohlížeči uživatele. Tyto možnosti jsou velice rozmanité, zahrnují např. synchronizaci času na homepage, interakci s uživatelem přes formuláře, nebo také komplexní operace s databázemi. [4]

### 1.6.3. Python

#### 1.6.3.1. Popis jazyku

Python je vysokoúrovňový univerzální programovací jazyk s dynamickým striktním typováním a automatickou správou paměti zaměřený na zvýšení produktivity vývojářů, čitelnosti a kvality kódu a zajištění přenositelnosti programů v něm napsaných. Jazyk je plně objektově orientovaný - vše jsou objekty. Neobvyklou vlastností jazyka je přidělování bloků kódu s mezerovitým odsazením. Syntaxe jádra jazyka je minimalistická, a proto je v praxi jen zřídka potřeba odkazovat na dokumentaci. Samotný jazyk je známý jako interpretovaný a používá se mimo jiné k psaní skriptů. Nevýhodou tohoto jazyka je často nižší rychlost a vyšší paměťová náročnost programů v něm napsaných ve srovnání s podobným kódem napsaným v kompilovaných jazycích, jako jsou C nebo C++.

Standardní knihovna obsahuje rozsáhlou sadu užitečných funkcí pro přenositelnost, od funkcí pro zpracování textu až po nástroje pro psaní síťových aplikací. Další funkce, jako je matematické modelování, manipulace s hardwarem,

webové aplikace nebo vývoj her, lze implementovat prostřednictvím velkého množství knihoven třetích stran, stejně jako integrací knihoven napsaných v jazycích C nebo C++, a samotný interpret jazyka Python lze integrovat do projektů napsaných v těchto jazycích. Existuje také specializované úložiště softwaru napsaného v jazyce Python - PyPI (také známý jako podle názvu klienta: pip). Tento repozitář poskytuje prostředky pro snadnou instalaci balíčků do operačního systému a stal se de facto standardem pro Python. V roce 2019 obsahoval více než 175 000 balíčků.

Python se stal jedním z nejoblíbenějších jazyků, který se používá při analýze dat, strojovém učení, vývoji DevOps a webových aplikací, ale i v dalších oblastech včetně vývoje her. Díky své srozumitelnosti, jednoduché syntaxi a absenci kompilace je tento jazyk vhodný pro výuku programování a umožňuje soustředit se na učení algoritmů, konceptů a paradigmat. Ladění a experimentování je výrazně usnadněno tím, že jazyk je interpretovatelný. Tento jazyk používá mnoho velkých společností, jako je Google nebo Facebook. Od dubna 2021 je Python třetím nejoblíbenějším programovacím jazykem v žebříčku TIOBE se skóre 11,03 %. "Jazykem roku" podle TIOBE Python byl vyhlášen v letech 2007, 2010, 2018 a 2020.

#### 1.6.3.2. Jinja2

Jinja je šablonovací engine pro programovací jazyk Python. Je podobný šablonovacímu enginu Django, ale poskytuje výrazy podobné jazyku Python a umožňuje spouštění šablon v sandboxu. Jedná se o textový šablonovací engine, takže jej lze použít k vytváření jakýchkoli značek i zdrojového kódu. Je licencován pod licencí BSD.

Šablonovací engine Jinja umožňuje přizpůsobit značky, filtry, testy a globální proměnné. Na rozdíl od šablonovacího enginu Django umožňuje Jinja volat funkce s argumenty na objektech.

#### 1.6.3.3. Flask

Flask je framework pro vytváření webových aplikací v programovacím jazyce Python pomocí sady nástrojů Werkzeug a šablonovacího enginu Jinja2. Patří do kategorie takzvaných mikroframeworků - minimalistických frameworků webových aplikací, které záměrně poskytují jen ty nejzákladnější funkce.

Verze 1.0 je podporována správcem balíčků PyPI a je kompatibilní s jazyky Python 2.7, Python 3.3 a vyššími.

## 2. Analýza současného stavu

### 2.1. Informace o firmě

NÁZEV: Red Hat Czech s.r.o.

IČO: 27690016

Z. KAPITÁL: 7.5 milionů Kč

ADRESA: Purkyňova 71/99, Královo Pole, 612 00 Brno

### 2.2. Popis společnosti

Tato část práce se bude již konkrétně věnovat vlastní společnosti, pro kterou budou později navrhována konkrétní řešení.

Red Hat, Inc. je americká mezinárodní softwarová společnost, která poskytuje podnikům softwarové produkty s otevřeným zdrojovým kódem. Společnost Red Hat, která byla založena v roce 1993, má své firemní sídlo v Raleighu v Severní Karolíně a má další kanceláře po celém světě.

Společnost Red Hat je do značné míry spojena s podnikovým operačním systémem Red Hat Enterprise Linux. Po akvizici dodavatele podnikového middlewaru s otevřeným zdrojovým kódem JBoss nabízí Red Hat také podnikový virtualizační produkt Red Hat Virtualization (RHV). Red Hat poskytuje úložiště, platformy operačních systémů, middleware, aplikace, produkty pro správu a podporu, školení a poradenské služby.

Red Hat vytváří, udržuje a přispívá k mnoha svobodným softwarovým produktům. Získala několik proprietárních kódových základů softwarových produktů prostřednictvím podnikových fúzí a akvizic a tento software vydala pod licenci open

source. Red Hat je po Intelu druhým největším korporátním přispěvatelem do linuxového jádra verze 4.14.

28. října 2018 společnost IBM oznámila svůj záměr získat Red Hat za 34 miliard dolarů. Akvizice byla ukončena 9. července 2019.

Red Hat Czech s.r.o. je výzkumná a vývojová pobočka společnosti Red Hat se sídlem v Brně v České republice. Dceřiná společnost byla založena v roce 2006 a má 1180 zaměstnanců (2019). Red Hat se rozhodl vstoupit do České republiky v roce 2006 kvůli popularitě open-source v zemi.

V roce 2016 společnost Red Hat Czech vykázala tržby ve výši 1 002 mil. Kč (FY 2016) a čistý příjem 123 mil. CZK (FY 2016), s aktivy 420 mil. Kč (FY 2016) | 325 mil. Kč (FY 2015).

Skupina byla v roce 2010 jmenována „Nejprogresivnějším zaměstnavatelem roku“ v České republice a „Nejlepší zaměstnavatel v České republice“ pro velké společnosti v roce 2011 společností Aon.

### 2.3. Organizační struktura

Red Hat má stovky různých produktů, ale v této práci budu věnovat Vaši pozornost především jednomu z neúspěšnějších - Openshift, a hlavně jeho podpoře, ve které já pracuji.

Proces	Role
Podpora zákazníků, generace článků, naplnění seminářů a t.d.	Associate Technical Support Engineer
	Principal Technical Support Engineer



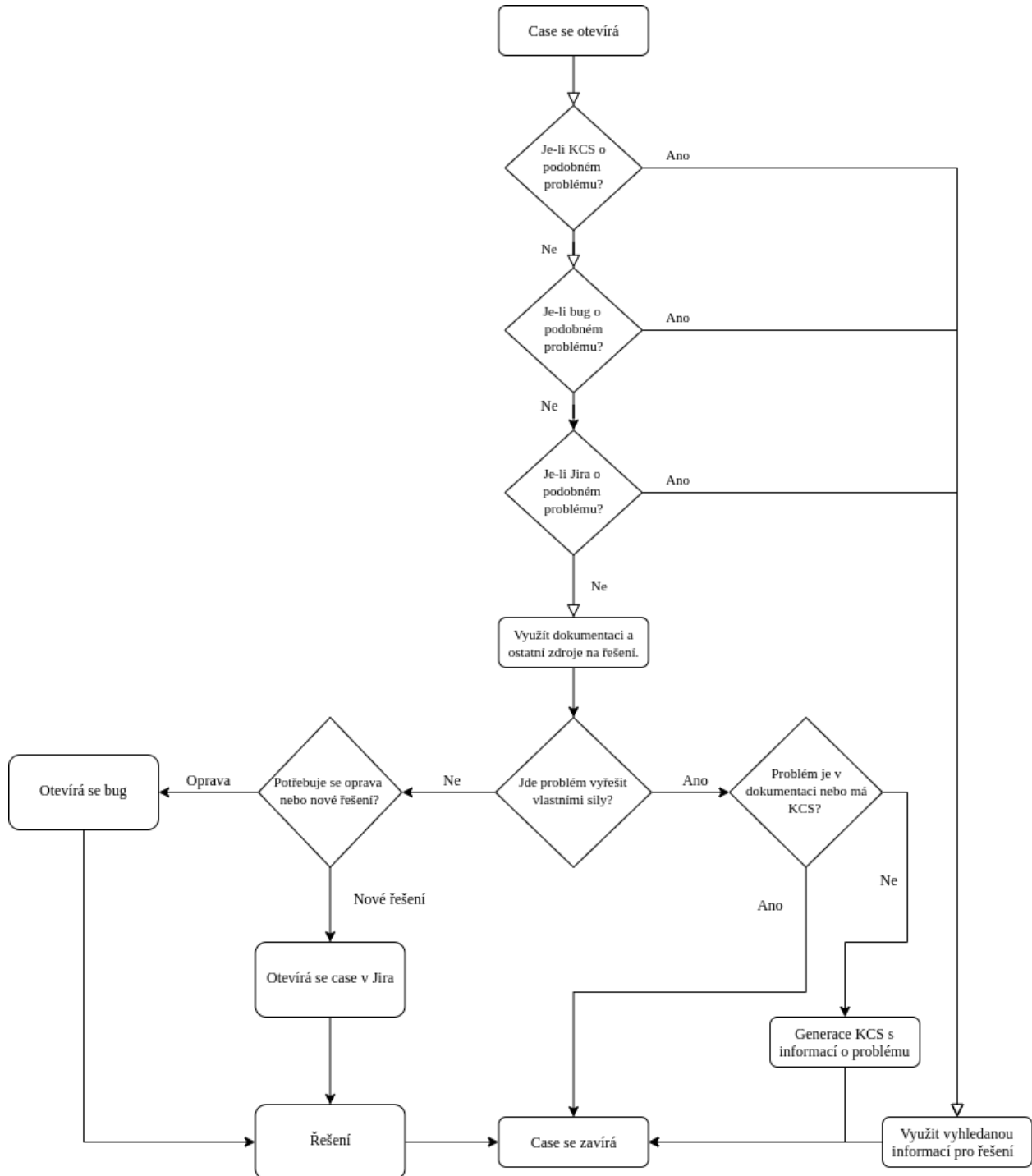
	Senior Technical Support Engineer
Řízení toku procesů, blahobyt zaměstnanců, organizace seminářů, kontrola efektivity	Associate Manager
	Principal Manager
	Senior Manager
Přímá podpora zákazníků (t.j. zaměstnanec zodpovědný za podporu jediný firmy)	Technical Account Management (TAM)

## 2.5. SWOT analýza projektu

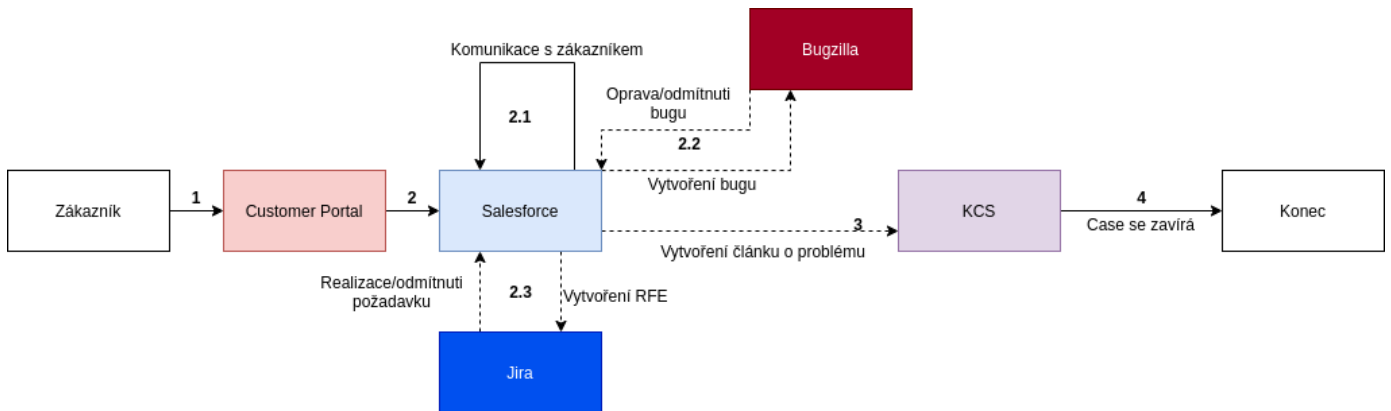
Silné stránky	Slabé stránky
<ul style="list-style-type: none"> <li>- Velká neustále rostoucí funkčnost produktu.</li> <li>- Profesionální tým.</li> </ul>	<ul style="list-style-type: none"> <li>- Těžký nábor specialistů kvůli příliš rychlému rozvoji: každý nový pracovník musí strávit hodně času studiu velkého počtu komponentů.</li> <li>- Nízká efektivita některých procesů</li> </ul>
Příležitosti	Hrozby
<ul style="list-style-type: none"> <li>- Větší znalost produktu na trhu pro snazší přijetí odborníků.</li> </ul>	<ul style="list-style-type: none"> <li>- Odchod zákazníků k konkurenčním bezplatným produktům.</li> </ul>

Z této analýzy vidíme, že největší problém pro firmu je nedostatek specialistů kvůli rychlému růstu a nízká efektivita procesů, která snižuje efektivitu.

## 2.6. Analýza hlavního procesu týmu



Obrázek 6



Obrázek 7

Během životu případu jsou několik kroků:

1. Zákazník vytvoří případ v zákaznickém portálu, kde uvádí hlavní problém, prioritu (1 - extrémní ohrožení toku firmy, 2 - velké, 3 - střední, 4 - nízké), případně přiloží nějaké dokumenty (logy, screenshoty a podobné).
2. Případ přichází do interního systému Salesforce, kde ho vezme jeden z pracovníků podpory.
  1. V prostředí Salesforce jde komunikace s zákazníkem.
  2. Při potřebě pracovník vytvoří bug, který bude řešit tým vývojarů.
  3. Pokud zákazník přeje realizaci nějakého napadu v systému - vytvoří se RFE (request for enhancement).
3. Po vyřešení problému může pracovník vytvořit článek v systému KCS (interní články s navody na řešení problémů)
4. Případ se uzavírá.

Jak můžeme vidět používá se hodně různých zdrojů informace, celkem to jsou:

Interní články (tzv. KCS);

Dokumentace;

Bugzilla;

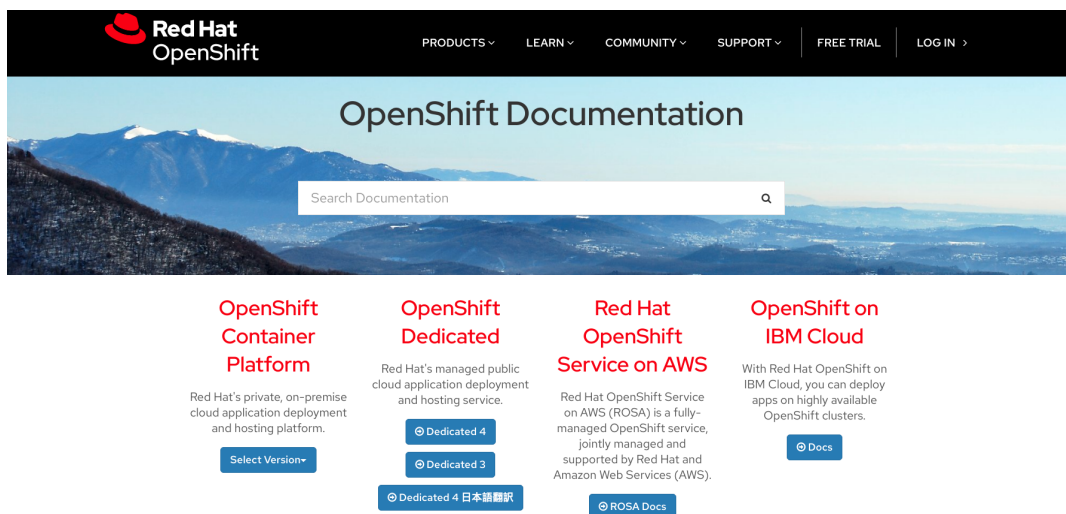
Jira;

Salesforce (portal komunikaci s zákazníci).

Každý z těchto systémů má svou stránku a vyhledávač, což nutí pracovníků utrácet hodně času na vyhledávání informace.

Například, vyhledávání v dokumentaci (nejsnadnější část hledání) potřebuje několik kroků:

1. Zajít na stránku docs.openshift.com.

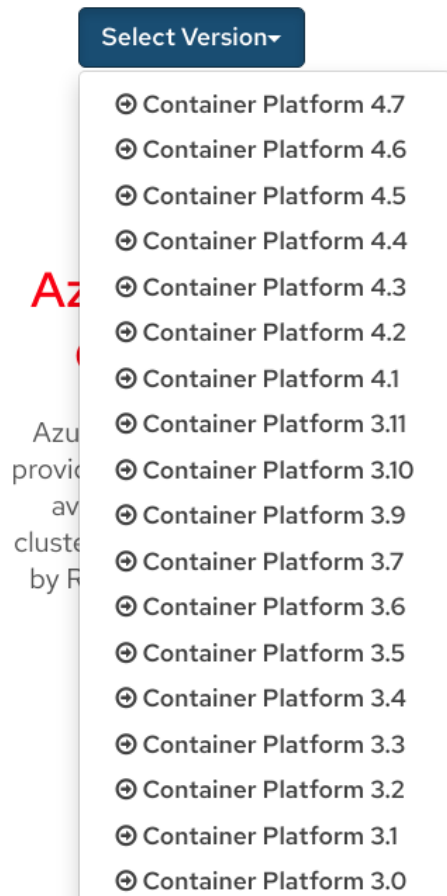


Obrázek 8

2. Vybrat verzi a zmáčknout ji.

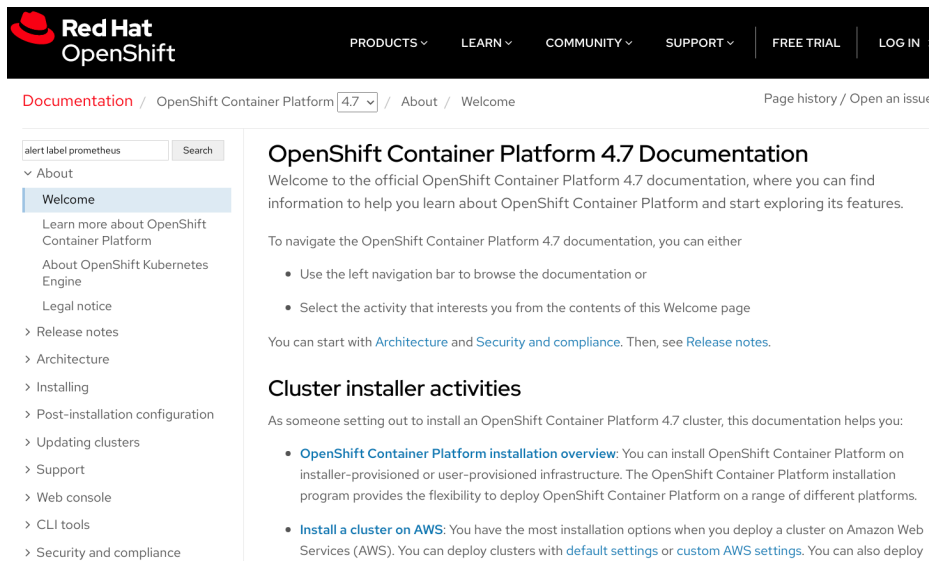
## OpenShift Container Platform

Red Hat's private, on-premise  
cloud application deployment  
and hosting platform.



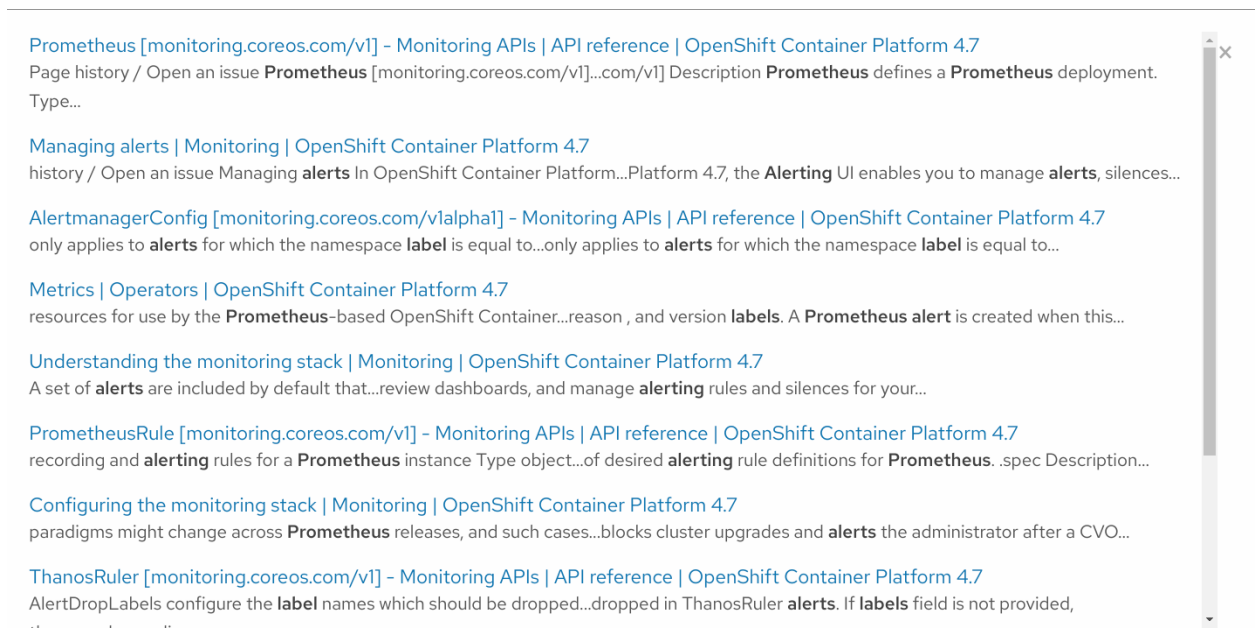
Obrázek 9

### 3. Napsat klíčová slova.



Obrázek 10

### 4. Jenom teď obdržet výsledek.



Obrázek 11

I když to nevypadá jako dlouhý postup: v čase když potřebujete rychle uspořádat informací z různých zdrojů i malá pauza může zhoršit produktivitu.

Podle vlastního experimentu získat informací (jeden požadavek) ze všech 4 zdrojů trvá 4 minuty a 21 krok. Je to čas na získání informací bez její analýzy.

Cílem projektu bude vytvoření vyhledávače pro agregaci informací z různých zdrojů tak aby celý proces vyhledávání trval méně než minutu a 3 kroky.

Tento systém bude užitečný především pro zaměstnanci s roli “Technical Support Engineer” (všech úrovní) a “Technical Account Manager”.

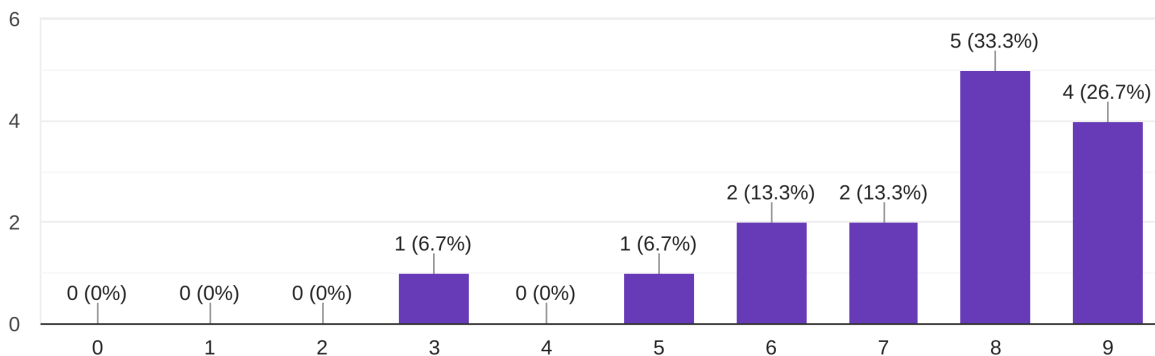
## 2.7. Průzkum

Pro lepší představu očekávání od projektu, pro zjištění hlavních problémů a pro pochopení produktivity (kolik času projekt je schopen ušetřit) provedl jsem průzkum části svého týmu. Z celkových 30 pracovníků k průzkumu přispělo 15, což považuji za dostatečné číslo, jelikož většina respondentů jsou té neaktivnější specialisty. Celkem jsem zeptal 6 otázek: 2 na zjištění spokojenosti s aktuálním systémem, 2 ohledně práci s vyhledáváním a 2 ohledně očekávané podoby projektu.

### 1. Jak často hledáte informaci v několika zdrojích?

How often do you search information at several sources (KCS, BZ, etc..)?

15 responses



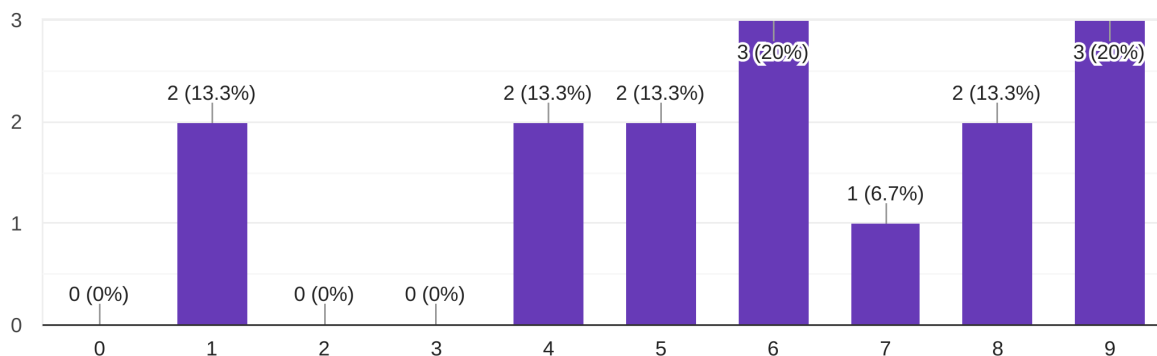
Obrázek 12

Jak můžeme vidět, drtivá většina zaměstnanců (60%) využívají systém na maximalni úrovni, co ukazuje jak je důležitý tento proces pro práce týmu.

## 2. Jak moc se během své práce potýkáte s hledáním?

How much do you struggle with searching during your work?

15 responses



Obrázek 13

Na tomto příkladu je jasně vidět, že většina odborníků je nespokojena se současným stavem systému vyhledávání ve firmě a ten jim brání v efektivní práci na případech.

## 3. Kolik času (přibližně) trávíte hledáním informací pro váš případ?

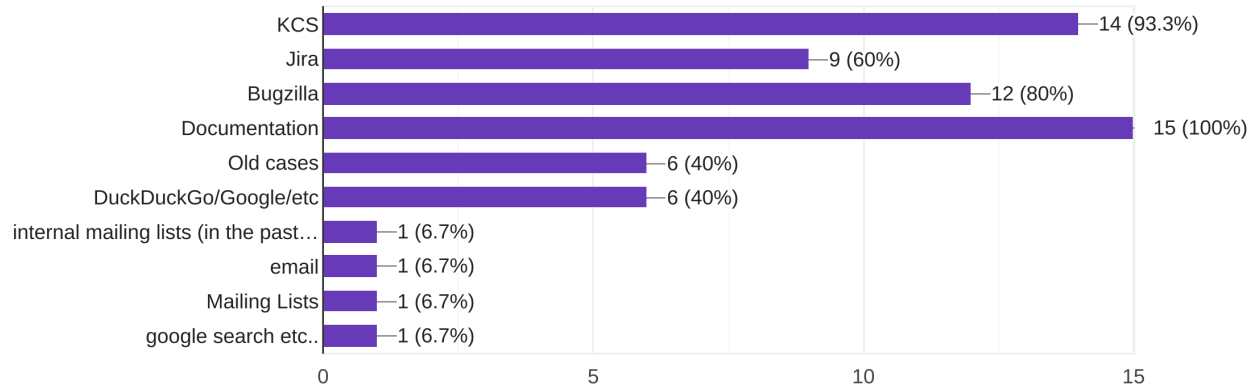
Otazka byla otevřená, většina odpovědi se shodili v tom, že každý case potřebuje cca 30 minut vyhledávání. Každý měsíc se zaregistruje okolo 1000 případů a toto číslo se roste. I malé zlepšení efektivity může výrazně pomoci zvýšení efektivity práci specialistů, počet kterých se taky rapidně zvyšuje.



#### 4. Jaké zdroje často používáte?

What sources do you use often?

15 responses



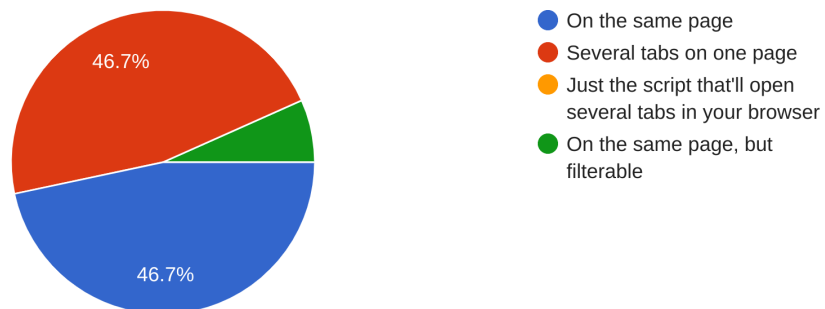
Obrázek 14

Jak je vidět z toho grafu - nejpoužívanější zdroje jsou dokumentace, články KCS, Bugzilla a Jira. Proto budu věnovat pozornost v této práci především jim.

Pro rozhodnutí o formě a naplnění systému poprosil jsem tým o názor.

Do you want the search results from several pages to be

15 responses



Obrázek 15

## 5. Chcete, aby byly výsledky vyhledávání z několika stránek:

Na stejné stránce (46.7%)

Několik karet na jedné stránce (46.7%)

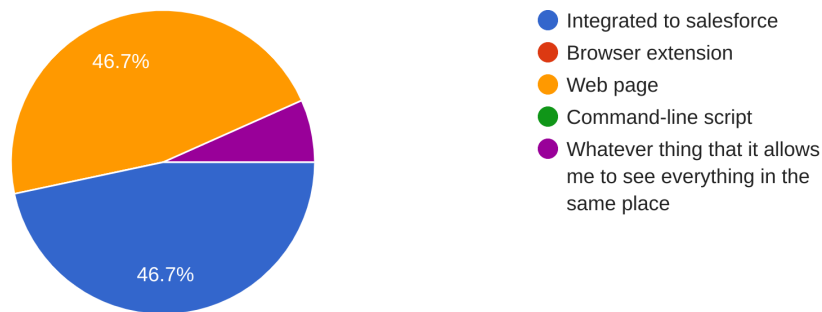
Pouze skript, který se otevře několik karet ve vašem prohlížeči (0%)

Na stejné stránce, ale filtrovatelný (6.6%)

Všichni chtějí realizaci projektu na jedné stránce. Liší se jenom požadovaný vzhled.

What is the best way for you to see your search results?

15 responses



Obrázek 16

## 6. Jaký je nejlepší způsob, jak zobrazit výsledky vyhledávání?

Integrované do Salesforce (46.7%)

Rozšíření prohlížeče (0%)

Webová stránka (46.7%)

Skript příkazového řádku (0%)

Cokoliv, co mi umožní vidět všechno na stejném místě (6.6%)

Většina respondentů chce webovou stránku nebo integraci do zákaznického portálu.

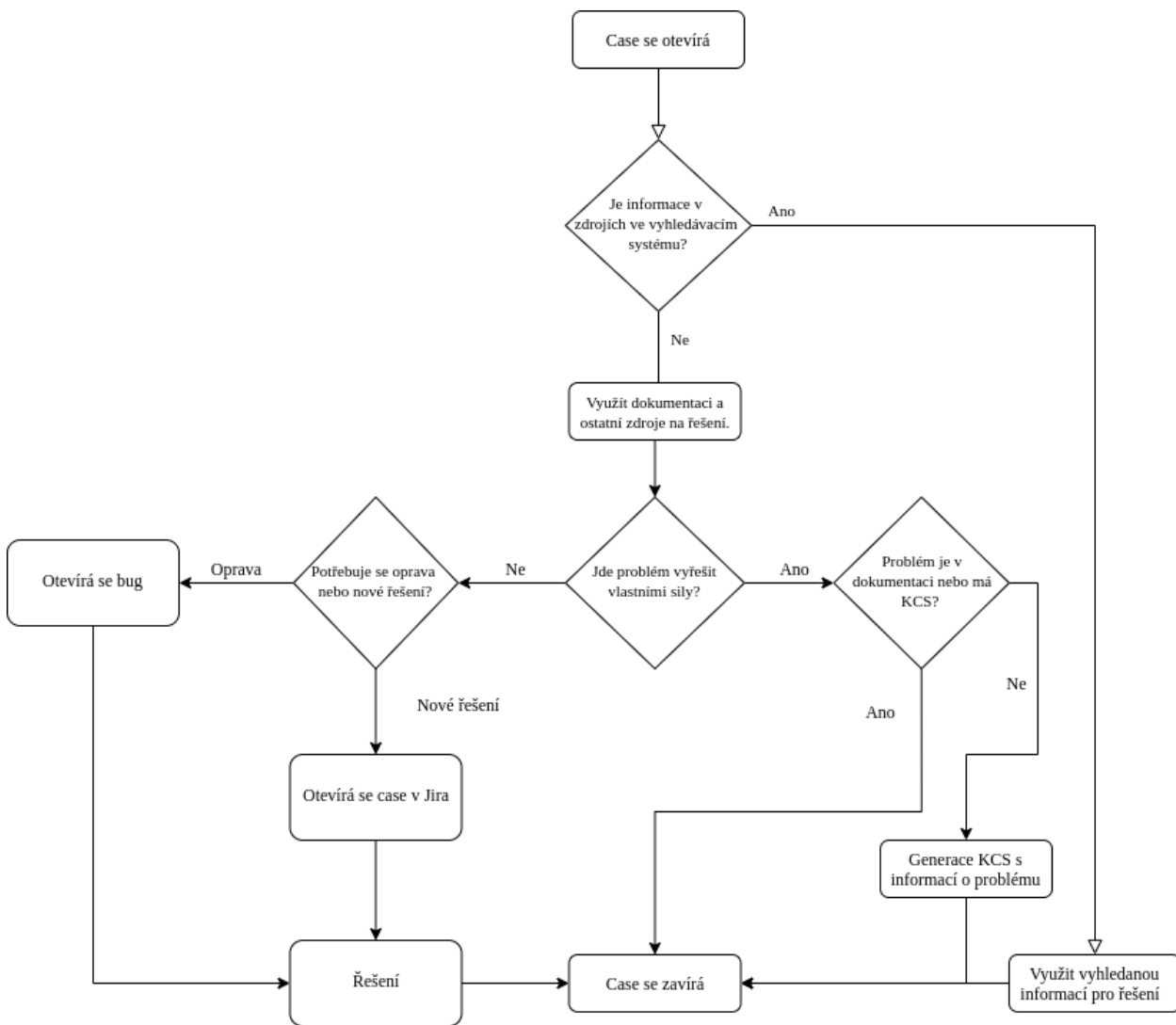
## 2.8. Shrnutí výsledků analýz stávajícího stavu

Během analýzy jsem zjistil, že specialisty tráví 30 minut práci nad případem na vyhledání informací a proto potřebuji vyhledávací systém pro vyhledání informací, který musí být realizovaný jako webová stránka. Nejdůležitější zdroje jsou články KCS, Jira, Bugzilla a dokumentace.

## 3. Vlastní návrhy řešení

### 3.1. Popis návrhu

Proto navrhuji univerzální vyhledávač, ve kterém bude možné najít všechno co pracovník potřebuje. Pak tento systém může přejít do další fázi (generace článku a t.d. přes systém).



Obrázek 17

Vytváření všech příslušných nástrojů by se mohlo rychle stát velkým úkolem v závislosti na požadavcích vytvořených / popsanych níže. Výsledkem tohoto návrhu je proto rozdělit práci na menší zvládnutelné výstupy.

### 3.2. Iterace 1

Jako první krok by mělo být možné mít (nástroj pro konfiguraci a správu dat), který zobrazuje všechny produkty, verze a typy vydání (jako matici NxN). To by umožnilo nástroji orientovanému na uživatele umožnit uživatelům vybrat odpovídající

pole (jedno po druhém) a na základě toho mapování (vybrané) by nástroje měly směřovat / poskytnout odkaz na správné místo pro uložení typu problému.

Nepovinný požadavek:

Když uživatel klikne na odkaz, při vytvoření nového případu by mělo automaticky vyplnit správná (předem vybraná pole) jejich předáním do systému sledování problémů.

Poznámky:

Toto je pravděpodobně také něco, co by mohlo být poměrně snadno zabudováno do zákaznického portálu Salesforce, aby to pro spolupracovníky, partnery a zákazníky bylo docela snadné.

### 3.3. Iterace 2

Umožnění vytváření případu (v rámci nástroje).

Požadavek:

Při odesílání problému by měl nástroj prohledat stávající problémy, aby zjistil, zda je vaše odeslání podobné jiným problémům. Pokud je to možné, měl by nástroj načíst / použít šablony pro sledování problémů pro popis problému, aby se ujistil, že spolupracovníci poskytují požadované informace a podrobnosti.

Požadavek:

Nástroj by měl také (volitelně) propojit vytvořený případ s případem Salesforce, aby bylo možné snadno sledovat průběh.

Tedy vystavení čísla případu jako volitelného vstupu.

Požadavek:

Jakmile je problém vytvořen, měl by být uživateli představen (např. otevřená webová stránka, uveden odkaz na ID případu atd.).

Poznámka:

Tento „informační“ díl by měl nastínit „další kroky“ uživatelů, jako je vyplnění jakýchkoliv dalších informací (např. volba dílčí komponentu, komponentu, prioritu, úroveň zabezpečení atd.)

Poznámka:

Umožnit uživatelům zobrazit / zkontrolovat podobné problémy před odesláním problému. Nástroj by měl také vystavit verzi produktu, protože nástroje pro sledování chyb a funkcí se mohou v závislosti na verzi změnit.

V této práci však budu věnovat čas pouze vývoju vyhledávače.

### 3.4. Forma a náplň

Většina kolegů požadovalo realizace nebo na vlastní stránce nebo s integraci do zákaznického portálu. Stejně se rozdělili vztahy k formě: půlka chce výsledky na jedné stránce, druhá chce filtrace. Realizace filtrace a implementace do zákaznického portálu potřebuje mít už hotový systém.

Proto navrhuji aby vyhledávač byl realizovaný jako webová stránka a měl jednu stránku s výsledky. V dalších verzích programu je možná realizace i ostatních požadavku.

### 3.5. Volba technologii

Jako operační systém jsem zvolil distribuce Ubuntu 20.04 kvůli její pohodlnosti (většina potřebných balíčků jsou již nainstalované), stabilitě (má větší stabilitu proti například Fedoře), velkému repozitáři, široké komunitě (proto je na internetu hodně článků a řešení častých problémů) a vlastní zkušenosti s systémem. Projekt však bude možné překonfigurovat pro využití v jiných distribucích GNU/Linux změnou několika parametru kódu (například v Ubuntu certifikační autorita se nachází v `“/etc/ssl/certs/ca-certificates.crt”`, ale ve Fedoře (a většině jiných distribucích) v `“/etc/pki/tls/certs/ca-bundle.crt”`).

## Python vs PHP

	<b>Python</b>	<b>PHP</b>
Snadnost učení	Snadno se učí. I začátečníci raději začínají svou cestu s Pythonem.	Nebyl vyvinut jako univerzální jazyk. Ve srovnání s Pythonem je těžké se ho naučit.
Podpora komunity	Python se v současné době docela dobře vyrovnává PHP.	Nabízí vynikající podporu komunity, protože je na trhu již delší dobu.
Dokumentace	Dokumentace uvedená na oficiálních webových stránkách a dalších fórech.	PHP se zde také shoduje s Pythonem a je dobře zdokumentováno.
Velikost knihovny modulů	Python má výjimečně dobře rozvinutou podporu knihoven pro téměř všechny typy aplikací.	Ačkoli má PHP velkou knihovnu balíčků v úložišti balíčků PHP, není tak velká jako u Pythonu.
Integrace s databází	Integrace s databází není tak silná jako u PHP.	Poskytuje přístup k více než 20 různým databázím.
Rychlost	Python je rychlý.	PHP je třikrát rychlejší.
Frameworky	Django, Flask, Pylons, Pyramid.	Codeigniter, Zend, Laravel, Symfony.
Ladění	Python nabízí výkonný ladicí program PDB (Python Debugger).	PHP poskytuje balíček XDebug pro ladění.

Oba jazyky jsou velice podobné ve funkcionalitě. PHP je rychlejší, má velkou podporu ze strany komunity a lépe pracuje s databází. V tomto projektu ale této faktory nejsou tak důležité jako jsou snadnost psaní kódu a velká nabídka hotových modulů.

### 3.6. Instalace

Předpokládáme, že Ubuntu je již nainstalovaná. Pro další využití musíme nainstalovat potřebný software:

```
$ sudo apt install pip pipenv
```

Takto nainstalujeme paketový manažer pipenv a pip - správce modulů programovacího prostředí Python, který využijeme pro instalaci dalších potřebných modulů jazyku Python:

```
$ pip3 install jq
```

jq - jq je lehký a flexibilní procesor JSON pro příkazový řádek [11].

Podobným způsobem nainstalujeme balíčky:

jira - Modul Jira je opensource knihovna, která usnadňuje používání rozhraní Jira REST API v jazyce Python a již několik let se používá ve výrobě [12].

requests - umožňuje velmi snadno odesílat požadavky HTTP/1.1.

textwrap, jinja2 a flask

balík z korporativního repozitáře pro komunikaci z KCS.

Pro komunikaci s API Bugzilla potřebujeme nainstalovat modul “python-bugzilla” [[link](#)]. Stahujeme soubory z github a instalujeme:

```
$ sudo python3 setup.py install
```

Pote administrator programu musí vytvořit společné fiktivní účty ve všech systémech a nakonfigurovat pověření do programu.

Po těchto krocích systém je připravený k spuštění programu.

### 3.7. Popis souborů

V složce projektu jsou tyto soubory a složky:

app.py - program pro spuštění webového rozhraní.

jira\_get.py - program příkazového řádku pro hledání případu na platformě Jira.



main.py - jádro projektu, je zároveň samostatným programem příkazového řádku i poskytovatel informací pro webové rozhraní

\_\_pycache\_\_ - technická složka s cachem, který se generuje během vykonání programu.

templates:

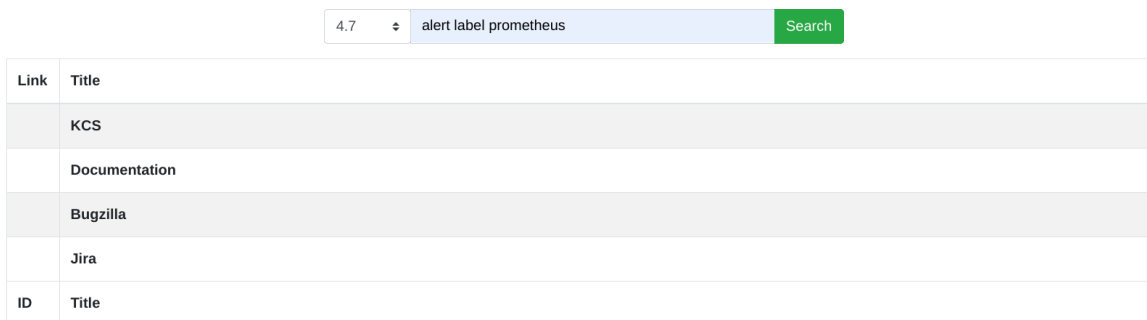
search.html - šablona webové stránky.

### 3.8. Instrukce k použití

Jsou několik způsoby používání tohoto programu:

#### 3.8.1. Grafický.

Po spuštění na serveru programu “app.py” uživatel může se prostřednictvím webového prohlížeče připojit k IP serveru přes port :5000 a dostat se na stránku vyhledávače. Tady musí vybrat verzi Openshift, zadat příkaz a zmáčknout “Search”. Vyhledávání může zabrat čas, který záleží na příkazu a vytíženosti servisů.



Link	Title
	KCS
	Documentation
	Bugzilla
	Jira
ID	Title

Obrázek 18

Na konci vyhledání se objeví výsledky hledání. Každý zdroj je oddělený od jiných. Bugzilla a Jira mají v prvním sloupci identifikační číslo případu za kterým je odkaz na stránku s případem. Jelikož články KCS a dokumentaci identifikační číslo

nemají - zobrazí jím jen nápis “Link”. V případě toho, že vyhledávač nenajde nic, co by odpovídalo klíčovým slovům - zobrazí se nápis “Nothing found”.

Link	Title
	<b>KCS</b>
	Nothing found
	<b>Documentation</b>
<a href="#">Link</a>	Prometheus [monitoring.coreos.com/v1] - Monitoring APIs   API reference   OpenShift Container Platform 4.7
<a href="#">Link</a>	Managing alerts   Monitoring   OpenShift Container Platform 4.7
<a href="#">Link</a>	AlertmanagerConfig [monitoring.coreos.com/v1alpha1] - Monitoring APIs   API reference   OpenShift Container Platform 4.7
<a href="#">Link</a>	Metrics   Operators   OpenShift Container Platform 4.7
<a href="#">Link</a>	Understanding the monitoring stack   Monitoring   OpenShift Container Platform 4.7
<a href="#">Link</a>	PrometheusRule [monitoring.coreos.com/v1] - Monitoring APIs   API reference   OpenShift Container Platform 4.7
<a href="#">Link</a>	Configuring the monitoring stack   Monitoring   OpenShift Container Platform 4.7
<a href="#">Link</a>	ThanosRuler [monitoring.coreos.com/v1] - Monitoring APIs   API reference   OpenShift Container Platform 4.7
<a href="#">Link</a>	OpenShift Container Platform 4.7 release notes   Release notes   OpenShift Container Platform 4.7
<a href="#">Link</a>	Installing a cluster on any platform - Installing on any platform   Installing   OpenShift Container Platform 4.7
	<b>Bugzilla</b>
	Nothing found
	<b>Jira</b>
<a href="#">OSD-5872</a>	Route SRE alerts only when alert has label "prometheus = openshift-monitoring/k8s"
<a href="#">MON-885</a>	prometheus: Filter alerts by label in Prometheus

Obrázek 19

### 3.8.2. Prostředstvím příkazového řádku

#### a) Jira

Jelikož analýza obsahu Jiry může potřebovat víc instrumentu byl napsaný samostatný program vyhledávání informace na platformě Jira “jira\_get.py”, za pomoci kterého můžeme vyhledávat poli případu (klíč ‘-k’ pro uvedení identifikátora případu, klíč ‘--fields’ pro uvedení potřebných poli):

```
$ python3 jira_get.py -u dkiselev -p PASSWORD -k LOG-660 --fields  
"description" "project"
```

nebo i klasické vyhledávání podle popisu případu za pomoci klíče '-d':

```
$ python3 jira_get.py -u dkiselev -p PASSWORD -d "alert label  
prometheus elastic kibana"
```

#### b) Celý projekt

Hlavní program v příkazovém řádku má skoro stejný funkcional jako webové rozhraní. Stačí spustit program "main.py" a zadat potřebnou informaci:

```
$ python3 main.py  
Search query: alert label prometheus elastic kibana  
Version: 4.7  
Customer portal username (i.e. dkiselev): dkiselev  
Customer portal password:  
Jira username (i.e. dkiselev): dkiselev  
Jira password:
```

### 3.9. Zaškolení

Zaškolení uživatelů bude ve formátu prezentace a praktické demonstraci využity systému a jeho komponentů. Za týden zaměstnanci projdou otázníku:

- Jak jsou s systémem spokojení? (případné hlášení chyb)
- Co v systému chybí?
- Vyhovuje-li jim vzhled programu?
- Čemu v práci s programem nerozumí?

Na základě výsledků při potřebě bude provedena doplňující konzultace a oprava a další vyvoj projektu.

Taky bude potřeba zaškolit jednoho člověka pro potřebnou administraci programu aby vždy online byl aspoň jeden specialista s porozuměním principů fungování a možných scénářů chování programu a se znalostí způsobů obnovení fungování systému.

### 3.10. Ekonomické zhodnocení

Analýza týmového procesu mi zabrala celkem 4 hodiny.

Získávání přístupu ke všem systémům trvalo taky 4 hodiny mi a mého kolegy.

Nejvíce času jsem využil na vyvoj samotného programu: 16 hodin.

Zaškolení specialistů trvá 30 minut na člověka, t.j.  $0,5 * 30 = 15$  normohodin.

Celkové časové náklady jsou **39** normohodin.

Cena implementaci je v podstatě nulová, jelikož firma již disponuje velkým počtem serveru a tento program není náročný na počítačové zdroje.

Aby celé řešení se oplatilo musí toto řešení ušetřit týmu 39 hodin.

Každý měsíc se generuje 1000 nových případů. Každý případ průměrně potřebuje 4 vyhledávacích dotazů.

Pokud tento program šetří optimistických 3 minuty na každý dotaz - to znamená, že měsíčně ušetří  $3 * 4 * 1000 = 120000$  minut nebo 200 hodin.

I při pesimistické předpovědi v 1 minutu na dotaz projekt šetří 4000 minut nebo 67 hodin, co ukazuje, že tento projekt bude přinášet časovou usporu už minimálně za 3 týdny práce.

## 4. Závěr

Cílem této práce bylo zvýšení efektivity práce týmu prostřednictvím vývoje a implementace vyhledávače v týmu podpory projektu Openshift.

V analýze současného stavu (taky za pomoci průzkumu) jsem zjistil, že zaměstnanci tráví velkou část svého pracovního času na vyhledávání a potřebují vyhledávací agregační systém.

Poté jsem pomocí diagramů popsal chování a práci se systémem, navrhl pohledy a formuláře pro snadnější hledání.

Pak jsem s pomocí jazyků Python a HTML a modulu Flask a Jinja2 vytvořil funkční vyhledávač, který následně popsal, specifikoval potřebný software a představil instrukce k jeho instalaci a využití.

Tímto ovšem celková práce na systému nekončí, nyní musíme zlepšit systém tím, že to implementujeme do portálu Salesforce a přidáme možnost filtrování a výběru potřebných zdrojů jako to chtěla část kolegů.

## Seznam použitých zdrojů

[1] BASL, Josef a Roman BLAŽÍČEK. Podnikové informační systémy: Podnik v informační společnosti. 3. vyd. Praha: Grada, 2012. 323 s. ISBN 978-80-247-4307-3.

[2] GÁLA, Libor, Jan POUR a Zuzana ŠEDIVÁ. Podniková informatika. 3. vyd. Praha: Grada Publishing, 2015. 240 s. ISBN 978-80-247-5457-4.

[3] SODOMKA, Petr a Hana KLČOVÁ. Informační systémy v podnikové praxi. 2. vyd. Brno: Computer Press, 2010. 504 s. ISBN 978-80-251-2878-7.

[4] LEISS, Oliver a Jasmin SCHMIDT. PHP v praxi: pro začátečníky a mírně pokročilé. Praha: Grada, 2010. ISBN 978-80-247-3060-8.

[5] PILGRIM, Mark. Ponořme se do HTML5. Praha: CZ.NIC, z.s.p.o., 2015. ISBN 978-80-905802-6-8.

[6] GARRELS, Machtelt. Introduction to Linux. 2008 (dostupné z <https://tldp.org/LDP/intro-linux/intro-linux.pdf>).

[7] RAMEY, Chet a FOX, Brian. Bash Reference Manual. 2020 (dostupné z <https://www.gnu.org/software/bash/manual/bash.pdf>).

[8] <https://distrowatch.com/dwres.php?resource=popularity>

[9] The Ultimate Linux Newbie Guide (dostupný z <https://linuxnewbieguide.org/wp-content/uploads/2016/07/The-Ultimate-Linux-Newbie-Guide-eBook-Edition-January-2017.pdf>).

[10] ZAMAZALOVÁ, M. Marketing. 2., přeprac. a dopl. vyd. V Praze: C.H. Beck, 2010, xxiv, 499 s. Beckovy ekonomické učebnice. ISBN 978-80-7400-115-4

[11] Oficiální stránka projektu jq (dostupné z <https://stedolan.github.io/jq/>)

[12] Oficiální stránka projektu pycontribs/jira (dostupné z <https://github.com/pycontribs/jira>)

## Seznam obrázků

Obrázek 1 - Zpracovatelský řetězec

Obrázek 2 - Vývojový diagram

Obrázek 3 - Logo Ubuntu

Obrázek 4 - Logo Fedory

Obrázek 5 - Logo RHEL

Obrázek 6 - Vývojový diagram popisující život případu.

Obrázek 7 - Diagram popisující komunikaci mezi prvky během případu.

Obrázek 8 - Web dokumentaci pro Openshift.

Obrázek 9 - Volba verze Openshift.

Obrázek 10 - Web dokumentaci pro Openshift.

Obrázek 11 - Výsledky hledání.

Obrázek 12 - Diagram odpovědí na otázku “Jak často hledáte informaci v několika zdrojích?”

Obrázek 13 - Diagram odpovědí na otázku “Jak moc se během své práce potýkáte s hledáním?”

Obrázek 14 - Diagram odpovědí na otázku “Jaké zdroje často používáte?”

Obrázek 15 - Diagram odpovědí na otázku “Chcete, aby byly výsledky vyhledávání z několika stránek byli:”

Obrázek 16 - Diagram odpovědí na otázku “Jaký je nejlepší způsob, jak zobrazit výsledky vyhledávání?”

Obrázek 17 - Vývojový diagram popisující život případu po realizaci návrhu.

Obrázek 18 - Vzhled programu do vyhledávání.

Obrázek 19 - Vzhled programu s výsledky vyhledávání.