


MASARYKOVA UNIVERZITA  
FILOZOFICKÁ FAKULTA  
KABINET INFORMAČNÍCH STUDIÍ A KNIHOVNICTVÍ  
**Obor:** Informační studia a knihovnictví



fithesis/logo/mu/fithesis-phil.pdf

**Studie proveditelnosti  
implementace open source  
integrovaného knihovního  
systému Koha v knihovnách  
Masarykovy univerzity**

BAKALÁŘSKÁ DIPLOMOVÁ PRÁCE

**Martin Kravec**

Brno, jaro 2016

## Bibliografický záznam

KRAVEC, Martin. *Studie proveditelnosti implementace open source integrovaného knihovního systému Koha v knihovnách Masarykovy univerzity*. Brno: Masarykova univerzita, Filozofická fakulta, 2016. 40 s. Vedoucí diplomové práce Mgr. Michal Denár.

## Anotace

Cílem této práce je zejména provést studii proveditelnosti, která určí, zda je možná migrace z knihovního systému Aleph na open source integrovaný knihovní systém Koha v knihovnách Masarykovy univerzity. Celá studie je založena na šestiměsíčním průzkumu týkajícího se mapování knihovních procesů ve vybraných knihovnách univerzity, a to formou rozhovorů a stínování. Průzkum byl také prováděn i v síti 900 knihoven v Turecku. Studie v úvodu rozebírá analýzu trhu, na kterou pak navazuje finanční analýza propojená s řízením a minimalizací rizik. Součástí studie proveditelnosti je také evaluace efektivity a udržitelnosti projektu založena na SWOT analýze. Práce také pojednává o projektovém managementu a řízení lidských zdrojů a podrobněji rozebírá technické a technologické řešení projektu, přičemž ob-  
jasňuje aspekty vývoje open source softwaru.

## Klíčová slova

open source, Koha, Aleph, knihovní system

## Annotation

Cílem této práce je zejména provést studii proveditelnosti, která určí, zda je možná migrace z knihovního systému Aleph na open source integrovaný knihovní systém Koha v knihovnách Masarykovy univerzity. Celá studie je založena na šestiměsíčním průzkumu týkajícího se mapování knihovních procesů ve vybraných knihovnách univerzity, a to formou rozhovorů a stínování. Průzkum byl také prováděn i v síti 900 knihoven v Turecku. Studie v úvodu rozebírá analýzu trhu, na kterou pak navazuje finanční analýza propojená s řízením a minimalizací rizik. Součástí studie proveditelnosti je také evaluace efektivity a udržitelnosti projektu založena na SWOT analýze. Práce také pojednává o projektovém managementu a řízení lidských zdrojů a podrobněji rozebírá technické a technologické řešení projektu, přičemž ob-

jasňuje aspekty vývoje open source softwaru.

## Keywords

open source, Koha, Aleph, library system

## **Prohlášení**

Prohlašuji, že jsem předkládanou práci zpracoval samostatně a použil jen uvedené prameny a literaturu. Současně dávám svolení k tomu, aby elektronická verze této práce byla zpřístupněna přes informační systém Masarykovy univerzity.

Martin Kravec

## **Poděkování**

Chtěl bych poděkovat vedoucímu práce Mgr. Michalovi Denárovi za to, že mi ukázal, jak krásně by mohlo fungovat české knihovnictví, a také Mgr. Petře Žabičkové v roli konzultantky za to, že mi ukázala, jak je na tom české knihovnictví ve skutečnosti. Dále bych rád poděkoval RNDr. Michalovi Růžičkovi, který se chopil role konzultanta a poskytl nedocenitelné informace. A v neposlední řadě si zaslouží můj vděk také zaměstnanci univerzitních knihoven MU, kteří jako respondenti ochotně spolupracovali při provádění rozhovorů a stínování v rámci mého průzkumu mapování knihovnických procesů v knihovnách Masarykovy univerzity.

# Obsah

<b>1 Teoretická část</b>	<b>2</b>
1.1 <i>Projektové řízení jako takové</i>	2
1.1.1 Projekt	2
1.1.2 Proces řízení projektu	3
1.2 <i>Studie proveditelnosti obecně</i>	4
1.3 <i>Open source jako systém</i>	6
1.3.1 Open source software	6
1.3.2 Financování	7
1.3.3 Správa verzí	7
1.3.4 Systém pro sledování chyb	7
1.4 <i>Migrace na nový knihovní systém</i>	8
1.4.1 Migrace obecně	8
1.4.2 Proces migrace	9
1.4.3 Obecné požadavky na knihovní systém	9
<b>2 Praktická část</b>	<b>12</b>
2.1 <i>Výchozí stav, zdůvodnění realizace projektu a analýza jeho potřebnosti</i>	12
2.1.1 O projektu	12
2.1.2 Etapy studie proveditelnosti	13
2.2 <i>Management projektu, řízení lidských zdrojů a harmonogram projektu</i>	14
2.2.1 Řízení lidských zdrojů	14
2.2.2 Management projektu	15
2.2.3 Harmonogram projektu	15
2.3 <i>Analýza trhu</i>	15
2.3.1 Potřeby a požadavky akvizice	15
2.3.2 Potřeby a požadavky katalogizace	16
2.3.3 Potřeby a požadavky cirkulací	16
2.3.4 Potřeby a požadavky uživatelů	17
2.3.5 Systémové požadavky	17
2.3.6 Další požadavky	18
2.3.7 Současné problémy	18
2.3.8 Technické a technologické řešení projektu	22
2.3.9 Potřeby a požadavky cirkulací	24
2.3.10 Potřeby a požadavky uživatelů	25
2.3.11 Systémové požadavky	28
2.3.12 Potřeby a požadavky katalogizace	32
2.3.13 Potřeby a požadavky akvizice	34
2.4 <i>Dopad projektu na životní prostředí</i>	34

2.5	<i>Finanční plán a ekonomická analýza projektu . . . . .</i>	34
2.6	<i>Hodnocení efektivity a udržitelnosti projektu . . . . .</i>	35
2.7	<i>Analýza a řízení rizik . . . . .</i>	36
Rejstřík	<i>. . . . .</i>	41

## Úvod

Tato bakalářská diplomová práce se bude zabývat studií proveditelnosti implementace open source integrovaného knihovního systému Koha v knihovnách Masarykovy univerzity. V práci bude probíráno několik úhlů pohledu na celkovou problematiku, a to v kontextu studie proveditelnosti. V práci tedy rozebereme nejdřív teoretickou část knihovních systémů, co by měly umět a jaký je jejich cíl jako produktu softwarových firem, to znamená, že probereme technické zastřešení projektu a vysvětlíme veškeré pojmy s tím spojené. Teoretická část dále vysvětlí oblast projektového managementu a přiblíží problematiku migrací na nový knihovní systém. Celý projekt bude řešen jako harmonogram realizace spolu s finanční a ekonomickou analýzou. V rámci studie byl prováděn několikaměsíční průzkum v knihovnách Masarykovy univerzity a průzkum v síti 900 knihoven v Turecku. Při průzkumu byla realizována také SWOT analýza, na které je pak založeno celé hodnocení efektivity a udržitelnosti projektu, ve kterém nebude chybět analýza a řízení případných rizik, které s sebou projekt migrace nese. V plánu je také zkrácený výstup v angličtině ve formě článku, jelikož je to téma s globálním dopadem, které může pomoci i institucím v zahraničí, které řeší stejný nebo podobný problém.



# 1 Teoretická část

## 1.1 Projektové řízení jako takové

### 1.1.1 Projekt

Projektové řízení neboli projektový management se zabývá řízením projektů. Jeho cíle lze rozdělit na 3 části:

- splnění požadavků
- časový plán
- rozpočtové náklady

Tomuto rozdělení se také říká trojimperativ [Rosenau, Milton D, 2000, s. 5]. O každém projektu lze prohlásit, že je jedinečný, jelikož je prováděn jen jednou, pracují na něm jiní lidé a je časově ohraničen. Projekt je realizován pomocí lidských a materiálních zdrojů. To znamená, že projektový manager vlastně řídí lidi tak, aby byly efektivně využity materiální zdroje [Řeháček, Petr, 2013, s. 28]. Následující obrázek 1.1 znázorňuje kvalitu provedení v kontextu trojimperativu.

Projekt může být charakterizován jako hmotný či nehmotný projekt v závislosti na konečném produktu. V případě hmotného projektu by šlo například o hardware, u nehmotného o software. Produkt je tedy konečným výstupem každého projektu. V případě, kdy je zákazník mimo organizaci, která projekt realizuje, jsou projekty realizovány na základě uzavřených smluv. Výsledný produkt je tedy posuzován na základě požadavků zákazníka. Je-li zákazníkem stejná organizace, není potřeba uzavírat smlouvy. V tom případě je ale výstup projektu posuzován podle času dokončení a návratnosti investice [Rosenau, Milton D, 2000, s. 10-12]. V projektovém řízení se zákazník jako objednatel projektu označuje jako zadavatel projektu. Způsob provedení projektu také závisí na konkurenci. Situace, kdy je produkt na trhu jediný svého druhu, je úplně jiná, než u produktu, kterých je v oboru mnoho. [Rosenau, Milton D, 2000, s. 12] Projektovému řízení se také říká „řízení projektů“ nebo „řízení programu“ v závislosti na velikosti projektu. Obecně platí, že programy jsou větší než projekty a projekty větší než úkoly. [Rosenau, Milton D, 2000, s. 12] Úspěšnost projektu je závislá na kompetencích manažera (nebo leadera) projektu [Müller, Ralf, 2010, s. 1], které lze rozdělit na:

1. Managerské kompetence

- Efektivní řízení zdrojů
  - Podporování komunikace v týmu
  - Zmocnění (leader dává svým přímým podřízeným autonomii při řešení problémů, čímž přispívá k rozvoji jejich vlastní zodpovědnosti)
  - Rozvoj (leader nabádá ostatní, aby si brali stále náročnější úkoly a sám investuje svůj čas a úsilí do rozvoje jejich kompetencí)
  - Odhodlání (leader ukazuje své neochvějné odhodlání při dosahování cílů a realizaci projektu)
2. Intelektuální kompetence
- Kritické myšlení (sběr relevantních informací z různých zdrojů a hledání výhodných a nevýhodných řešení)
  - Vize a představivost (leader má jasnou vizi o budoucnosti projektu)
  - Strategická perspektiva (leader si je vědom problémů a jejich důsledků, nachází příležitosti a hrozby)
3. Emocionální kompetence
- Sebepoznání (leader je schopen ovládat své pocity)
  - Emocionální odolnost (leader je schopen udržovat stálý výkon ve všech situacích)
  - Intuitivnost
  - Motivace (leader musí svými činy působit motivačně)
  - Svědomitost (leader ukazuje svoji angažovanost k projektu a sám povzbuzuje ostatní členy pracovní skupiny)

### 1.1.2 Proces řízení projektu

Řízení projektu sestává z procesu pěti manažerských činností, a to [Řeháček, Petr, 2013, s. 22-23]:

- Definování projektových cílů
- Plánování splnění cílů, tedy trojimperativu
- Efektivní vedení lidských zdrojů (podřízení, dodavatelé)

- Monitorování - sledování odchylek od plánu či stavů jednotlivých fází projektu
- Ukončení - finalizace ve smyslu kontroly, zda produkt odpovídá definicím zadavatele, dokončení prací (například dokumentace)

Vzájemné závislosti v procesu řízení projektu znázorňuje obrázek 1.2. Celému cyklu však předchází tzv. předprojektová fáze, do které patří zejména studie proveditelnosti, o které pojednává další kapitola.

## 1.2 Studie proveditelnosti obecně

Hlavním účelem studie proveditelnosti (feasibility study) je zhodnotit možné varianty, které mohou nastat při provádění projektu, a posoudit realizovatelnost a následnou životaschopnost zvoleného řešení. Studie proveditelnosti zpřesňuje vlastnosti projektu a to hlavně specifikaci cíle, potřebné finanční, materiální a lidské zdroje, časový harmonogram, přínosy a rizika spojená s realizací projektu. Jelikož studie pojednává o finančních, technických, manažerských i ekonomických aspektech projektu, nazývá se také studií ekonomicko-technickou[Fotr, Jiří, 1995, s. 19]. Vyhodnocení studie tedy vyúsťuje do rozhodnutí o zamítnutí či přijetí projektu a případně jeho následné realizaci. [Fotr, Jiří, 1995, s. 19-20]

*„Významné je, aby studie co nejlépe popisovala, variantně řešila, optimalizovala a hodnotila investiční projekt se všemi z něj vyplývajícími specifiky.“[Sieber, Patrik, 2004, s. 8]*

Celková osnova studie proveditelnosti by měla vypadat přibližně takto:[Sieber, Patrik, 2004, s. 8-14]

1. **Obsah** - struktura kapitol
2. **Úvodní informace** - účel zpracování studie proveditelnosti
3. **Stručné vyhodnocení projektu** - závěry studie v rozsahu 1-2 stran, zhodnocení finanční efektivity projektu
4. **Stručný popis podstaty projektu a jeho etap** - komplexně pojednává o hlavních rysech projektu (název, smysl, zaměření projektu, výsledný produkt a problémy, které produkt řeší, lokalizace a etapy projektu)

5. **Analýzy trhu, odhad poptávky, marketingová strategie a marketingový mix** - marketingové aspekty (potřeby finálních uživatelů produktu, konkurenceschopnost produktu)
6. **Management projektu a řízení lidských zdrojů** - plánování, organizace a management procesů a lidských zdrojů projektu
7. **Technické a technologické řešení projektu** - výhody a nevýhody zvolených technologií, materiálové a energetické toky, technická rizika, náklady na údržbu, správu a provoz.
8. **Dopad projektu na životní prostředí** - kladné i negativní vlivy jednotlivých etap realizace projektu
9. **Zajištění dlouhodobého majetku** - výše investičních nákladů, struktura dlouhodobého majetku
10. **Řízení pracovního kapitálu (oběžný majetek)** - velikost a struktura oběžného majetku
11. **Finanční plán a analýza projektu** - celkové zobrazení předchozích bodů
12. **Hodnocení efektivity a udržitelnosti projektu** - evaluace projektu na základě zadaných kritérií, finanční toky a doba návratnosti investic
13. **Řízení rizik (citlivostní analýza)** - výčet zdrojů rizik projektu, opatření
14. **Harmonogram projektu** - časový harmonogram jednotlivých etap projektu. Začátky a konce jednotlivých činností
15. **Podrobné závěrečné hodnocení projektu** - komplexní vyjádření k realizovatelnosti projektu

Vyhodnocení finanční rentability projektu vybranými ukazateli se provádí tak, že se nejdříve určí jednorázové náklady na investice, odhadnou se budoucí výnosy z investice, následně se určí náklady na kapitál a nakonec se vypočítá současná hodnota očekávaných výnosů. Na to se pak různě aplikují metody ekonomického hodnocení investice.[PODEŠVOVÁ, Veronika, 2010] [cit. 2016-04-01], s. 17-18]

### 1.3 Open source jako systém

Open source systém lze chápat ve dvou rovinách, a to z pohledu teorie systémů jako systém, který je založen na otevřeném svobodném kódu a slouží pro využívání, zpracování a zprostředkování informací, a také z pohledu systematičnosti vývoje, kam patří správa verzí, bug trackery, financování, licencování, vytváření balíčků či spolupráce s komunitou.[Cejpek, Jiří, 2005, s. 40]

#### 1.3.1 Open source software

Pojem „open source software“ bývá častokrát nesprávně zaměňován s pojmem „free software“, je proto důležité vymezit si rozdíly v těchto pojmech.

**Open source software** (OSS) je software, jehož zdrojový kód může být kýmkoliv upravován nebo vylepšován. Open source licence totiž povoluje legální přístup, možnost modifikace či další distribuce tohoto softwaru. Nemalá část komerčního softwaru vychází právě z open source projektů.

**Free software** je software poskytovaný zdarma, což však neznamená, že má uživatel licenci k přístupu, modifikaci či k další distribuci tohoto softwaru. Je definován spíše v kontextu svobody než ceny.

Opakem free softwaru je software, jehož zdrojový kód může být modifikován pouze jeho autorem či organizací, která ho vydala. Tomu se říká **closed source software** neboli **proprietární software** (od slova „property“, což znamená, že zdrojový kód softwaru je majetkem svých autorů a jenom oni mohou legálně kopírovat či modifikovat tento software). Příkladem takového softwaru je například Microsoft Office.

Všeobecně lze OSS licence rozdělit do tří kategorií a to:[Gaff, B. M. 2012, s. 9-10]

- restriktivní neboli **copyleft** licence (omezující)
- středně restriktivní neboli **copycenter** licence
- **permisivní** licence (tolerantní, shovívavé)

Teoreticky však existuje nekonečné množství těchto licencí, jelikož si autoři open source softwaru mohou vybrat jakoukoliv licenci nebo si vytvořit licenci vlastní. Na základě výše řečeného lze tvrdit, že open source software je podmnožinou free softwaru, avšak free software nemusí být open source.

### 1.3.2 Financování

Financování vývoje open source softwaru probíhá častokrát neformálně z prostředků firem.

*„Firmy si začali uvědomovat výhody, které z open source softwaru plynou, a stále častěji se přímým způsobem podílejí na jeho vývoji.“* [Fogel, Karl, [2012], s. 129]

Karl Fogel (autor bestselleru *Tvorba open source softwaru* z osvětové edice CZ.NIC a autor svobodného verzovacího systému Subversion) dodává, že organizace, které mají své softwarové potřeby, zbytečně duplikují své snahy, když nakupují software se stejnou funkcionalitou nebo když ho dokonce vytvářejí samy znovu. Když by se připojili k open source vývoji, náklady na vývoj by se rozdělily mezi všechny další participanty a z prospěchu by tak čerpali všichni. Tento proces se navíc zdá být prospěšným nejen pro neziskové organizace, ale i pro výdělečné činnosti, kde jako příklady uvádí <http://www.openadaptor.org/> a <http://koha.org/>. [Fogel, Karl, [2012], s. 130-132] V komerční sféře to už ale kvůli licencím možné není.

### 1.3.3 Správa verzí

Systém pro správu verzí (version control system) je jistým standardem při vývoji open source softwaru. Verzovací systém je nástroj, který primárně spravuje vydávání nových verzí (release management), odděluje stabilní kód od testovacího, respektive produkčního kód od vývojového, pomocí takzvaných větví. Mimo jiné pomáhá při opravách chyb a také komunikaci mezi vývojáři. [Chacon, Scott, c2009, s. 17-21, 59-64] Mezi nejznámější verzovací systémy patří Git, Subversion (SVN) a Mercurial.

### 1.3.4 Systém pro sledování chyb

Spolu se systémy na správu verzí bývají propojené (ale mohou fungovat samostatně) i systémy na sledování chyb, neboli bug trackery<sup>1</sup>. Tento software slouží jako nástroj pro zaznamenávání „problémů“, u kterých lze definovat počáteční a koncový stav. Slouží nejen pro zaznamenávání chyb, ale také pro požadavky na novou funkcionalitu či vylepšení již stávající. Tyto problémy mohou procházet množstvím předdefinovaných stavů jako například „investigating“, „being worked on“, „near completion“ a podobně.

1. Pojem bug vystihující skrytou chybu vznikl na Harvardské univerzitě, kdy byl v jednom z prvních počítačů nalezen hmyz (štěnice).

Tyto záznamy problému lze různě kategorizovat (například frontend, backend), přidávat jim zadavatele, řešitele, přidávat komentáře a mnoho dalšího. Mezi nejznámější patří Bugzilla, Jira a Buggenie.[Fogel, Karl, [2012], s. 99-104]

## 1.4 Migrace na nový knihovní systém

### 1.4.1 Migrace obecně

Migrací se rozumí proces, při kterém dochází k přesunu aplikací integrovaného knihovního systému (dále jen ILS - Integrated Library System) z jednoho ILS na jiný, který lépe vystihuje potřeby knihovny. V minulosti se při migraci nahlíželo spíše na vývojové požadavky pro cirkulaci, katalogizaci, akvizici a podobně. Dnes se hodně přihlíží také k službám, praxi a potřebám a očekáváním uživatele.[Bilal, Dania, 2014, s. 151-152]

Důvody k migraci mohou být různé:[Bilal, Dania, 2014, s. 152]

- stávající knihovní systém je tradičního charakteru ve smyslu jeho uživatelského rozhraní a funkcí; knihovna se snaží modernizovat, vylepšit uživatelské rozhraní jak po stránce architektury, tak po stránce přívětivosti, chce nabídnout uživatelům discovery služby, které jsou založené na robustních modulech, které lze snadno přizpůsobit
- dodavatel nebo podpora systému je neuspokojivá
- závislost na dodavateli (vendor lock-in)
- systém nebo jeho moduly (neprůhledný finanční model) jsou drahé, návratnost investic je neuspokojivá
- otázka cloudu vs. lokálního úložiště
- stávající systém je zastaralý a dodavatel systému již tento produkt nepodporuje
- knihovna je součástí konsorcia, které rozhodlo o migraci
- kapacitní omezení, zpracování velkého množství dat
- provozní spolehlivost
- systémová omezení (provoz a bezpečnost)

- nedostatečná podpora standardů a nových technologií (MARC, RDA, Z39.50, NCIP<sup>2</sup>, PSGI/Plack<sup>3</sup>)
- omezené možnosti integrace (napojení na národní souborný katalog, národní autority, digitální knihovny)

Opakem nedostatečné funkcionality bývá jako další důvod k migraci také příliš komplikovaný systém či personál, který s ním reálně neumí zacházet. Hlavním důvodem pro výběr open source systému bývá hlavně cena. Volba konkrétního open source systému je pak ovlivněna jeho funkcionalitou.

#### 1.4.2 Proces migrace

Úspěšnost migrace závisí na mnoha faktorech. Na začátku je nutné pochopit celý migrační proces a rozplánovat ho na několik etap, kde se mezi první etapy řadí čištění dat. Je nutné pochopit datová schémata záznamů ve všech modulech stávajícího systému, aby bylo možné určit, jak by mohly být tato schémata převedena do nového ILS. Je potřeba vědět, v jakých formátech či stavech se extrahovaná data nacházejí, a také mít spolehlivou komunikaci s poskytovatelem či podporou systému. Při extrakci dat musí být jasné, jak lze data interpretovat, což znamená napsat extrakční skripty tak, aby skutečně odpovídali nové datové struktuře ILS.[Singh, Vandana, n.d., s. 43-48] Při splnění těchto bodů se přechází k samotné migraci dat. Tuto migraci je vhodné provádět opakovaně s různým množstvím dat a importovaná data okamžitě testovat na nějakých tomu určených testovacích stránkách. Po samotném převodu dat je na místě realizace několika testovacích strategií.<sup>32-35; 61-72</sup> Denar2015thesis Celkový proces migrace si vyžaduje i svůj „člověkočas“, na který je potřeba být připraven.

Celkový proces migrace vyžaduje svůj čas jak na přípravu, tak na realizaci. V přípravě nesmí chybět seznam problémů, kterými procházely jiné instituce, které podobnou migrací provedly. Také je vhodné mít případnou záložní „roll back“ strategii v případě neúspěšného pokusu migrace.

#### 1.4.3 Obecné požadavky na knihovní systém

Z pohledu správce lze definovat jako požadavky na knihovní systém následující. Systém musí v rámci modularity poskytovat: [Bilal, Dania, c2014, s. 8-12]

2. NISO Circulation Interchange Protocol (National Information Standards Organization Circulation Interchange Protocol) .

3. PSGI/Plack - <http://plackperl.org/>



### Modul pro cirkulace

Tento modul by měl minimálně nabízet funkcionalitu pro kontrolování a vypůjčování knihovního fondu, podporovat pokuty a poplatky, notifikace, rezervace a objednávky, správu čtenářského účtu, správu statistik.

### Akviziční modul

Tento modul by měl spravovat objednávání jednotek do knihovního fondu. To zahrnuje objednávání, příjem, fakturace, reklamace, alokování fondu či sledování prodejců.

### Modul pro seriály

Modul slouží pro správu ročníků, periodik a novin. Může zahrnovat zrušení předplatného, přiřazování opožděných čísel časopisů, alokování fondu, sledování prodejců nebo přechíslování seriálů. V modulu lze prohledávat fond podle různých parametrů (například ISSN<sup>4</sup>, nebo názvu).

### Modul pro meziknihovní výpůjční službu (MVS)

Modul pro MVS je určen na půjčování a vypůjčování jednotek mezi knihovnami. Není-li titul dostupný v rámci knihovny, může knihovna na žádost uživatele požádat o meziknihovní výpůjčku. Tyto výpůjčky jsou založeny na smlouvách, které mezi sebou knihovny uzavírají. Ne všechny fondy musí být možné vypůjčit.

### Autoritní modul

Tento modul může být součástí katalogizačního modulu a jeho hlavní funkcionalitou je tvorba a úprava hesel (jména autorů, titulů, seriálů, předmětů ve smyslu témat) pro bibliografické záznamy. Propojuje autoritní heslo s autoritním záznamem pomocí standardizovaného seznamu hesel.

### Katalogizační modul

Modul je určen pro zkatalogizování dokumentů a uložení metadat s následným zařazením do knihovního fondu. Momentálně se využívá formát MARC21, kterého zápis byl donedávna definován pravidly AACR2 (anglo-americká katalogizační pravidla verze 2).[Dilhofová, Adéla, 2013, s. 26-27]

---

4. International Standard Serial Number

Tyto pravidla jsou dnes nahrazena novým standardem RDA (Resource Description and Access), který je založen na konceptu FRBR (Functional Requirements for Bibliographic Record). To je taky důvod, proč jsou některé záznamy zapsané jinak. Je potřeba říct, že RDA ve všeobecnosti nereprezentuje ani pravidla, ani formát, ale spíš strukturovanou množinu myšlenek ve smyslu toho, jak by měl vypadat bibliografický záznam na to, aby byli splněny potřeby uživatele. RDA se stávají fakticky závaznými pravidly v ČR až ve chvíli, kdy chce instituce například dotaci z VISKu<sup>5</sup>, kdy musí přispívat do Souborného katalogu České republiky, protože musí dodržovat standardy a pravidla pro to daná a podobně.

Celý knihovní systém by měl interně podporovat správu digitálních informačních artefaktů, jako elektronických knih, elektronických periodik a další, a také mít možnost vytvářet nové multimediální materiály (video, DVD, Blu-Ray a podobně).

---

5. Dotační program Ministerstva kultury ČR - Veřejné informační služby knihoven (VISK).

## 2 Praktická část

### 2.1 Výchozí stav, zdůvodnění realizace projektu a analýza jeho potřeby

#### 2.1.1 O projektu

Aktuálnost problematiky tkví v tom, že společnost Ex Libris, která vyvíjí knihovní systém Aleph, přestává tomuto produktu poskytovat podporu a zároveň vyvíjí produkt jiný, jenž však funguje výhradně v cloudu<sup>1</sup>. Jelikož knihovny Masarykovy univerzity používají jako knihovní systém právě Aleph je potřeba situaci zhodnotit z různých pohledů.

Mimo fakt, že data již nebudou ukládána lokálně, tj. data již nebudou uložena na strojích patřících provozovateli<sup>2</sup>, ale v cloudu (navíc pravděpodobně mimo jurisdikci EU)[Breeding, Marshall, 2012, s. 16-17], to znamená, že knihovny a další instituce využívající knihovních systémů (televize, archivy) budou mít menší možnosti úprav a rozšíření své instance systému. Tyto problémy by mohla vyřešit právě migrace na open source knihovní systém. Jedním z kandidátů je open source integrovaný knihovní systém Koha, který tato studie proveditelnosti rozebírá.

Dalším kandidátem, uvaží-li se velikost knihoven MU a open source systémy, které prošli několikaletým vývojem, by byl systém Evergreen. Ten je však orientován na anglosaský svět, což může přinášet různé problémy způsobené omezenou podporou pro abecedy s diakritikou. Jelikož nedisponují balíčky, je problematický i proces aktualizací. Systém nebyl vybrán pro studii také pro nedostupnost technické podpory v ČR a relativně náročnou instalaci systému. Systém navíc nepodporuje stahování autoritních záznamů protokolem Z39.50. Navíc má Evergreen desktopového klienta, což činí správu náročnější. Další důvod podporující migraci na open source je fakt, že knihovny platí nezanedbatelné částky za podporu stávajícího systému, což je oproti open source systémům markantní rozdíl, který lze investovat do smysluplnějších projektů spojených s cíli knihoven jakožto vzdělávacími, informačními a kulturními institucemi. Investovat do open source projektů je dnes běžnou praxí i v komerční sféře, jelikož si firmy začali uvědomovat benefity, které díky těmto projektům čerpají od komunity zpátky. [Fogel, Karl, [2012], s. 129-132]

---

1. Cloud je jistá metafora pro Internet, která znamená ukládání a zpřístupňování dat a programů na Internet místo na lokálním stroji.

2. Možné politické a autorsko-právní problémy.

Mezi cílové skupiny tohoto projektu migrace ale nepatří jenom systémoví knihovníci, správci a vedení knihovny, ale také zaměstnanci, kteří potřebují splnit/poskytnout informační služby pomocí práce s klientem knihovního systému, a v neposlední řadě i uživatelé knihoven, kterým se musí dostat kvalitní služby poskytované danou institucí nehledě na uživateli handicap.

Tato práce tedy pojednává o problematice v rámci kontextu každé z výše zmíněných cílových skupin. Projekt v této studii proveditelnosti lze do jisté míry využít i v dalších institucích v zahraničí, které se zabývají migrací na jiný open source knihovní systém, avšak konkrétní „case study“ je lokalizován na knihovny Masarykovy univerzity, podléhající zákonům platným v České republice, a to konkrétně knihovnímu zákonu č. 257/2001 Sb. o knihovnách a podmínkách provozování veřejných knihovnických a informačních služeb, na který navazuje vyhláška č. 88/2002 Sb., zákon č. 106/1999 Sb., o svobodném přístupu k informacím, zákon č. 101/2000 Sb., o ochraně osobních údajů a o změně některých zákonů a v neposlední řadě i autorský zákon č. 121/2000 Sb.

### 2.1.2 Etapy studie proveditelnosti

Celou studii proveditelnosti lze rozdělit na několik etap. Tou první je průzkum, který je nutný k nalezení informací týkajících se následujících problémů:

- jaké procesy probíhají v sledovaných knihovnách
- jaké procesy probíhají mezi těmito knihovnami
- jak tyto procesy souvisí s externími službami
- jak tyto procesy interagují s technologickým okolím
- jak jednotlivé procesy řeší personál knihoven
- na jaké problémy naráží personál knihoven při konkrétních procesech
- jak by chtěl personál knihoven zlepšit již stávající procesy
- jak personál knihoven spravuje technologie v jednotlivých knihovnách
- jak jednotlivé knihovny spolupracují s Ústavem výpočetní techniky a Knihovnicko-informačním centrem univerzity

- jak je řešena metodologická podpora při věcné a jmenné katalogizaci

Druhou etapou studie je zjištění, jak lze získané informace aplikovat v novém systému, což je vlastně výstupem neboli produktem celé studie. Do další etapy patří rozpracování zjištěného do sekcí studie (viz. kapitola 1.2 Studie proveditelnosti), na což navazuje poslední etapa v této předprojektové fázi, a to evaluace všech sekcí studie ve formě podrobného závěru projektu.

## **2.2 Management projektu, řízení lidských zdrojů a harmonogram projektu**

### **2.2.1 Řízení lidských zdrojů**

Zásady dobrého managementu projektu jsou již obecně sepsány v kapitole 1.1 Projektové řízení jako takové. Při řízení lidských zdrojů na tomto projektu je potřeba brát v úvahu celkový časový harmonogram a mít detailně rozpracovaný plán realizace projektu. Přejde-li se k samotné realizaci, je potřeba určit skupiny lidí, kteří budou zodpovědní za určitou část realizace.

První skupinu by tvořil personál, který rozumí metadatům a standardům, minimálně MARC21 a RDA, který by za dohledu metodologické podpory ÚVT MU prováděl čištění dat. Je potřeba se soustředit zejména na umístění dat na správná místa, do správných polí a zapisovat data ve vhodném kódování, které podporuje i abecedy jiných jazyků. Při čištění by měly být odstraněny překlepy a nesprávné formáty stringů, viz standardy. Skupina 1 a skupina 2 by měly spolupracovat, protože množství dat k pročištění je nemalé, a to tak, že kde to půjde, bude skupina 2 vytvářet skripty pro hromadnou úpravu záznamů na základě pravidel daných skupinou 1.

Druhou skupinu by tvořili lidé, kteří umí psát dotazy na databázi či pracovat s API stávajícího systému. Tato skupina musí vyčištěná data připravit ve smyslu napsání exportních skriptů. Těmto vyexportovaným datům je nutné rozumět. Budou-li správně interpretována, bude možné je správně nainportovat do nového systému. Třetí skupinu lidí musí tvořit systémoví knihovníci a čtvrtou knihovníci na referenčních a výpůjčních pultech. Tito lidé budou systémovým knihovníkům definovat chování, které má systém splňovat, zatímco systémoví knihovníci budou tyto požadavky konfigurovat. Čtvrtá skupina by také měla vytvořit testovací scénáře a snažit se otestovat nový systém, respektive správnost dat, správné interpretování dat, funkčnost poskytovaných funkcí.

### 2.2.2 Management projektu

Při řízení lidských zdrojů je také potřeba myslet na vývojový tým, tedy skupinu 2, která bude komunikovat s komunitou a konzultovat případné problémy s knihovnami, které již procesem prošly. Tato skupina lidí by měla být v kontaktu s předchozí skupinou a v případě potřeb dopisovat chybějící funkcionalitu, či upravovat tu stávající k potřebám knihovny.

Projektový manager by měl udržovat stálou komunikaci mezi těmito skupinami a zajistit, aby každý chápal, co obnáší proces migrace a proč k němu vlastně dochází. Musí být vidět, že vedení projektu je přesvědčeno o přínosu migrace a že si stojí za svým rozhodnutím.

### 2.2.3 Harmonogram projektu

Celková realizace projektu vyžaduje svůj čas a je potřeba na to vyčlenit lidské a materiální zdroje. Lidské zdroje jsou omezeny pracovním časem, jelikož musí řešit i běžnou knihovnickou práci, materiální zdroje zase vyžadují přípravu serverů, jak těch testovacích, určených pro nový systém, tak i těch produkčních pro export dat. Je tedy potřeba zajistit dostatečně velké úložiště dat pro nový systém, výkon a zachovat bezpečnost dat (licence, zákony). Celkově lze rozdělit pracovní skupiny dle následující tabulky.

## 2.3 Analýza trhu

Momentálně se Masarykova univerzita skládá z devíti univerzitních knihoven, které sestávají z vyše 110 dílčích knihoven. Dohromady spravují cca 2 000 000 knihovních jednotek a obsluhují cca 50 000 uživatelů. MU nemá žádnou ústřední knihovnu.

Při provádění průzkumu v knihovnách MU vzešlo několik potřeb a požadavků. Tyto požadavky jsou jak hlavní kritéria, které musí nový systém splňovat, tak i potřeby, které je potřeba splnit v rámci kvalitního designu služeb pro uživatele. K těmto požadavkům na systém se řadí i obecné požadavky na knihovní systém, definované v kapitole 1.4.3 Obecné požadavky na knihovní systém.

### 2.3.1 Potřeby a požadavky akvizice

- evidovat adresy dodavatelů, IČ, DIČ
- možnost veřejně spouštět soutěže pro jednotlivé knihovny

- možnost dohledat, kolik knih bylo koupeno a z jakých zakázek (každá katedra má jiné zakázky)

### 2.3.2 Potřeby a požadavky katalogizace

- ILS musí podporovat protokol na výměnu dat OAI-PMH<sup>3</sup>
- provázání s NKP - stahování národních autorit ČR pomocí FTP nebo OAI-PMH
- podpora ILS v katalogizaci elektronických zdrojů (momentálně v Alephu katalogizují jako knihy)
- možnost hromadných úprav
- stahování záznamů Z39.50<sup>4</sup>
- podpora RDA
- možnost uzamknout přístup k editovací části záznamu či celé záznamy pro personál s omezenými právy
- ILS musí podporovat práci s rejstříky
- napojení na authority, vybrat autoritní heslo

### 2.3.3 Potřeby a požadavky cirkulací

- možnost zjistit, zda je uživatel student, jestli má registraci nebo jestli je neaktivní student a kde studoval
- možnost ověření souhlasu uživatele v INETu
- možnost udělovat globální nebo lokální blokace (v Alephu globální a lokální karty)
- možnost dohledávat diplomové práce v ISu
- podpora více výpůjčních protokolů
- notifikační systém (email, SMS)
- možnost propojit ILS se selfchecky

---

3. The Open Archives Initiative Protocol for Metadata Harvesting.

4. Mezinárodní standard definující protokol pro hledání a výměnu dat mezi dvěma stroji.

- dohledávat uživatele (propojit Informační systém IS MU, INET MU (ekonomicko-správní informační systém Masarykovy univerzity a databáze/databázi uživatelů v novém systému)

#### 2.3.4 Potřeby a požadavky uživatelů

- veřejně dostupný katalog (OPAC) pro uživatele knihovny s možností správy uživatelského účtu a dalších
- 
- možných služeb s nativní podporou discovery služeb nebo využití Centrálního portálu knihoven
- napojení OPACu na obálky knih přes API
- možnost platit pokuty, dobíjet kredit, propojení se systémem SUPO (systém úhrad pohledávek za osobami)
- možnost COD (Copy on Demand) služeb v OPACu
- možnost opravovat překlepy ve vyhledávacích dotazech a nabízet alternativy
- sjednotit uživatelské rozhraní pro vyhledávání (momentálně Aleph OPAC, Discovery service a někde uvažují o vývoji vlastního OPAC)

#### 2.3.5 Systémové požadavky

- možnost napojení systému na externí technologie a služby
- napojení na open source federativní systém pro propojování identit Shibboleth
- ekonomický systém (akviziční modul)
- personální systém
- robustnost a vysoká provozní spolehlivost systému
- flexibilita systému, členění systému do složitějších hierarchických struktur
- otevřenost systému



- podpora velkého počtu knihovnických jednotek, uživatelů, poboček, podpoboček

Momentálně lze definovat vztahy jednotlivých systémů takto:

### 2.3.6 Další požadavky

- možnost zjišťovat půjčovnost dokumentu v rámci jednotlivých knihoven, ne pouze celkově
- přehledné uživatelské konto ze strany knihovníka
- mít k dispozici API pro napojení vlastních či vytvoření nových aplikací
- mít k dispozici dokumentaci systému v českém jazyce
- zdokumentovaná konfigurace systému

### 2.3.7 Současné problémy

Při provádění průzkumu bylo zjištěno několik nedostatků neboli problémů, se kterými se personál knihoven musel potýkat. Jedná se o nejaktuálnější zdrojový kód, tedy ke dni psaní této práce to byl Aleph verze 22. V drtivě většině byl problém nalézt cestu k cíli v rámci architektury systému. Byly to většinou problémy uživatelského rozhraní, kdy jim například při provádění některých hromadných akcí vyhodil systém chybu 45krát, nastal-li 45krát problém. Toto okno bylo potřeba 45krát zavřít. Další problém byl zjistit půjčovnost dokumentu v rámci jedné knihovny.

Následující výčet reprezentuje vlastní aplikace, které jsou založeny na API Alephu nebo využívají SQL dotazy na databázi Alephu. Tyto aplikace vznikly buď chybějící funkcí v Alephu, nadbytečným množstvím informací poskytnutých systémem (informační přetížení, redundantní informace) nebo nepřesnou interpretací, kterou si knihovny musely ke svým účelům upravit.

#### Aplikace KIC

Aplikace knihovnicko-informačního centra ústavu výpočetní techniky Masarykovy univerzity, které jsou k dispozici všem knihovnám MU.

Aplikace pro knihovníky:

- **Souhlas** – Real-time ověření udělení elektronického souhlasu s provozními řády knihoven MU v systému INET, kde je to primárně ukládáno, tzn. s obejitím případných zpoždění/chyb synchronizace při přenosu do knihovního systému
- **Průkazka** (starší a nová verze) – Aplikace pro registraci externích čtenářů a vydání čipové karty externího čtenáře. Průvodce celým procesem pro knihovníky, protože to integruje interakci s externím systémem, kde je pro externisty (tj. lidi „z ulice“ bez jiného vztahu k MU) potřeba založit účet v univerzitním systému správy identit, aby se byli schopni přihlásit na počítače v knihovně a do svého konta v Aleph OPACu.
- **Signatury** – aplikace konvertující textový seznam čárových kódů knih na PDF s barevnými štítky se signaturami (stahují se z databáze k daným čárovým kódům, barví se dle nějakých pravidel) k vytištění a nalepení na knihy
- **Sbírký** – aplikace pro konfiguraci sbírek, tj. kategorií, do kterých se dají jednotky řadit. Generuje se z konfiguračního souboru Alephu na serveru, pokud knihovník udělá změnu, vygenerují se z toho na serveru nové konfigurační soubory, kterými se původní konfigurace přepíše a provedou se další akce, aby se aktivovala.
- **Závazek / seznam výpůjček** – Přehled závazků (případně včetně historie výpůjček) a stavu registrací osoby napříč všemi knihovnami MU, aby se dalo zjistit, jestli nemá čtenář jinde blokace apod.
- **Report** – Aplikace na e-mailování různých informací. Uvnitř nakonfigurováno cca 20 různých přehledových sestav (např. seznam dlužníků, záznamy bez jednotek, různé typy chyb v bibliografických záznamech apod.), které si může každý pracovník naklikat s parametry, které ho zajímají (dlužníci v konkrétní knihovně s dluhem větším než 500 Kč apod.) a případně si je nechat i v pravidelných intervalech s těmito parametry nechat posílat na svou e-mailovou adresu.
- **Items** – Konvertor textové seznamu čárových kódů -> Čárový kód, název, sbírka, signatura, rok, poznámka, poznámka OPAC, interní poznámka, materiál
- **Záznamy** – Pro zadané sysno vygeneruje HTML výpis záznamu a la zobrazení v knihovním klientu.

- **Katalogizace** – Přehled úprav záznamů katalogizátory (pro rozsah datumů)
- **Půjčování** – Kdo co kdy půjčoval pro rozsah datumů
- **Seznam** – Tisk přírůstkového / úbytkového seznamu pro vybranou knihovnu a rozsah čárových kódů nebo datumů
- **Evidence** – Už se prakticky nepoužívá, vnitřní evidence půjčování několika interně půjčených knih v rámci ÚVT
- **Eprace** – Již nepoužívaná jednoúčelová aplikace. Dříve se používalo pro evidenci zpracování diplomových prací, když se digitalizovali pro elektronickou evidenci. Teď už se přijímají přímo elektronicky.
- **Toplist** – Seznam nejčastěji půjčovaných knih s možností volby knihovny, rozsahu datumů, počtu příček a volitelným filtrem dle sbírky, do které jednotky patří
- **Dlužníci** – Report o dlužnících (s možností filtrace dle knihovny, datumů a minimální dlužné částky) ve formátu přímo vložitelném do ekonomického systému Magion
- **Výpůjčky** – Čtenáři s nevrácenými výpůjčkami v prodlení s možností filtrace dle knihovny a druhů čtenářů (studenti, doktorandi, zaměstnanci apod.)
- **Selfcheck** – report o využívání selfchecků s filtrem dle knihovny a rozsahů datumů
- **E-prezenčka** – Evidence zpracování pro ePrezenčku
- **Smazání** – Seznam smazaných čtenářských účtů s datem vymazání
- **FAQ** – Různé návody, pokyny ke katalogizaci apod. pro knihovníky (Pamatuje si, co kdo už viděl, takže nepřečtené/aktualizované věci jsou při první návštěvě knihovnickovi zvýrazněny.)

Nástroje pro konvertování záznamů:

- používají hlavně katalogizátoři. Jedná se o vyhledávací formuláře nad různými MARC dumpy<sup>5</sup> bibliografických záznamů (často i různých období apod.) s možností vyhledávání dle různých kritérií (včetně

5. Výpisy, exporty.

možnosti regulárních výrazů apod.). Vyhledává se Perlem na textovém dumpem MARC záznamů, takže se tam dají udělat různá poměrně komplexní vyhledávání dle složitých podmínek.

- aplikace pro kontrolu bibliografických záznamů před odesláním do Souborného katalogu v Národní knihovně

Systémové nástroje:

- náhled dat z databáze
- odkazy na stažení instalační soubory knihovního klienta (aby je měl správce k dispozici v aktuální verzi, když se v knihovně reinstaluje nějaký počítač) a TeamView klienta, když je někomu potřeba vzdáleně pomoci přímo na jeho počítači

Aplikace Fakulty sociálních studií

- aplikace pro soutěž akvizic

Aplikace KUK (knihovny univerzitního kampusu)

- **MVS** – správa požadavků
- **Dlouhodobé výpůjčky** – v OPACu může uživatel požádat o dočasnou výpůjčku knihy, kterou má aktuálně půjčenou vyučující dlouhodobě.
- **Dlouhodobé, které letos končí** – aby bylo možné upozornit včas vyučující, prodloužit výpůjčku apod
- **Hromadná objednávka** – vytisknout více titulů do jednoho objednávkového formuláře v designu MU
- **Rozpočty** – přehled rozpočtů aktuálního roku, stav čerpání financí a zobrazení navázaných titulů/jednotek.
- **Limity nákupu** – jeden z rozpočtů má určeny limity pro jednotlivá pracoviště (sbírky) – každé pracoviště má zobrazen stav čerpání svého limitu a nakoupené jednotky.
- **Poptávky** – velký systém pro sdružené poptávání sad titulů, sběr cenových nabídek a následný tisk objednávek po ukončení soutěže.

- **Dílčí knihovny** – MU má cca 90 dílčích knihoven (kateder). Zobrazení aktuálních přírůstků, celkového stavu fondu, evidence údajů o jednotlivých sbírkách pro zobrazení na webu knihovny
- **Dopis hříšníkům** – na základě seznamu UČO vygenerovat jednotlivé dopisy v komunikačních jazycích uživatelů (cz/en). Z Alephu se přebírá seznam dlužených knih a rozpis aktuální dlužené částky Kč
- **Dary** – musí evidovat osobní údaje lidí, kteří darují knihovně knihu.
- **Kontrola chyb** – skript se sadou sql dotazů zobrazí očividné chyby – např jednotka bez čárového kódu, se špatně uvedenou cenou, špatným datem nabytí apod.
- **Kontrola řady čárových kódů** – při zpracování knih se občas ztratí čárový kód, je potřeba ho dohledat a obsadit (jeho číslo je zároveň přírůstkové číslo)
- **Žebříčky/rezervace** - přehled počtu rezervovaných/vypůjčených dokumentů v daném roce podle sbírky jednotky, fakultní příslušnosti čtenáře
- **Přehled fondů pro potřeby akreditací** - vygenerování přehledu počtu svazků knih/časopisů podle majetku fakulty a kolik z toho je ve volných výběrech
- **Výpůjční statistiky** - generování přehledu počtu vypůjčených/vracených dokumentů po hodinách v jednotlivých dnech + počet čtenářů, kteří v daný den si něco půjčili/vrátili
- **Přehled knih pro zahraniční studenty**

### 2.3.8 Technické a technologické řešení projektu

Veškeré požadavky vycházejí z šestiměsíčního zkoumání fakultních knihoven MU, které bylo prováděno jako moderované rozhovory s jednou až třemi osobami zároveň a stínování každé z nich. Rozhovorů se účastnili nejen zaměstnanci, ale i vedoucí knihoven.

Každý okruh knihovních procesů byl mapován samostatně, nejdříve se zkoumaly služby u referenčních pultů, pak akvizice, katalogizace a automatizace. Následně byly zkoumány zahraniční knihovny a instituce, které

migrací prošli formou rešerší a osobní či e-mailové komunikace v rámci komunitního maillistu<sup>6</sup>.

Tak jako každý knihovní systém, i Koha splňuje základní funkcionalitu jednotlivých modulů definovaných v kapitole 1.4.3 Obecné požadavky na knihovní systém.

Při prováděném výzkumu bylo zjištěno několik problémů a menších či větších nedostatků. Tyto nedostatky se snaží tato studie vyřešit open source řešeními. Studie počítá se systémem Koha ze stejných důvodů jako jiné knihovny (s různými komerčními knihovními systémy), které už migrací prošly. Mezi hlavní důvody lze zařadit:

- výrazné jednodušší instalace a aktualizace systému oproti open source systému Evergreen
- instalační balíčky pro Debian a Ubuntu
- možnost využívat autoritní záznamy při katalogizaci
- stabilní vývoj, pravidelný cyklus verzí
- aktivita komunity
- jasná budoucí vize projektu
- podpora autorit
- napojení na discovery VuFind a elektronické informační zdroje (například EBSCO)
- výborné schopnosti akvizičního modulu
- přehledný katalogizační editor s možností rozšíření
- rozsáhlé možnosti notifikací stavů čtenářům e-mailem a SMS
- pestré možnosti API včetně jeho rozšiřování

Mezi zajímavá zjištění lze zařadit, že drtivá většina knihoven některé moduly, jako je třeba akviziční modul, nepoužívají a nahrazují jej buďto vlastními aplikacemi nebo výše zmíněnými alternativami.

Modul cirkulací s historií výpůjček byl v jednom případě vyměněn za fyzické lístečky s historií výpůjčky dané jednotky umístěnou v deskách jednotky.

---

6. E-mailový chat komunity. .

Potřeby a požadavky MU zpracované v této kapitole je možné rozpracovat detailněji, což však přesahuje rozsah bakalářské práce a může tak být tématem pro další navazující práci. Tento výčet požadavků a jejich řešení však dostatečně reflektuje potřebné informace k tomu, aby bylo možné sestavit plnohodnotný závěr studie proveditelnosti.

### 2.3.9 Potřeby a požadavky cirkulací

Modul cirkulací je v Koha velmi přívětivý a flexibilní. Uživatele nebo výpůjčku lze dohledat načtením čárového kódu nebo zadáním názvu či kódu jednotky, v případě uživatele zadáním jména. Výstupem této akce jsou nejdůležitější údaje o hledaném, například datum vypůjčení dané jednotky, datum pro vrácení jednotky, možnost prodloužit jednu nebo více výpůjček, místo půjčení, typ jednotky a podobně. U vrácení stačí načíst čárové kódy, promíjet upomínky nebo vracet knihy z biblioboxů.

Koha umí uživatele předem notifikovat pomocí emailu nebo SMS, kde správa šablon různých zpráv je samozřejmostí. Těmto notifikacím lze různě nastavovat například periodicitu a další vlastnosti.

#### Napojení na selfchecky

Koha podporuje protokol SIP2, díky kterému je možné napojit systémy selfchecků, RFID technologie, kopírky / tiskárny či Zebra tiskárny pro tisk štítků.

#### Podpora více výpůjčních protokolů

Jelikož má Koha možnost vytvářet pobočky, má i podporu více výpůjčních protokolů. V systému Koha je možné nastavovat vazby mezi jednotlivými knihovnami / pobočkami, tudíž je možné nastavit, kdo má na co práva a kam „vidí“. Je tedy možné pobočky propojit nebo naopak rozdělit.

#### Lokální a globální karty Alephu

Tato funkcionality vyžaduje hlubší šetření. Momentálně se databáze uživatelů sdílí mezi všemi v rámci instance a jde blokovat jen globálně a to buď na základě poplatků nebo nevrácení. V případě, kdy by v rámci instance neměla knihovna vidět do uživatelské databáze jiné knihovny, je potřeba chování systému upravit. Bylo by tedy potřeba vytvořit testovací instanci, která by představovala novou strukturu systému a na ní odladit potřebné požadavky.

### MVS (Meziknihovní výpůjční služba)

Meziknihovní výpůjční služba funguje v Koha jako běžná výpůjčka, ale pro speciální typ uživatele - knihovnu. Knihovny jsou speciální kategorie čtenářů s odpovídajícími parametry, jako cena za nula a pravidly pro výpůjčky. Takže se provede klasická výpůjčka, která se následně odešle poštou. Lze vygenerovat report, který přehledně zobrazí, jaká knihovna od kdy má daný dokument. Příchozí MVS je řešena specifickým typem dokumentu, který má nastaven kratší dobu výpůjčky, navěšeno hlášení při návratu a specifickou katalogizační šablonu. Takže dokument je přijat, rychle zkatalogizován, do jednoho pole je zapsáno, od koho je, a půjčen čtenáři. Opět je k dispozici report, který vypisuje, jaké dokumenty v rámci příchozích MVS jsou k dispozici, kdo je má, atd. Toto fungování nemusí zcela vyhovovat MU a celou agentu bude pravděpodobně potřeba upravit nebo použít MVS modul<sup>7</sup>.

#### 2.3.10 Potřeby a požadavky uživatelů

Celý problém lze rozdělit na dvě části, vnitřní a vnější. Tou vnitřní částí je rozhraní systému, se kterým měli potíže zaměstnanci. Tento problém lze v systému Koha vyřešit velmi elegantně a prakticky bez bariér. Jelikož je rozhraní systému webové, lze ho upravovat pomocí technologií HTML5 a CSS3 spolu s JavaScriptovou knihovnou jQuery, která dodá celému rozhraní uživatelskou přívětivost (efekty místo statického loadingu, asynchronní načítání obsahu, hezké zobrazování notifikací, slideUp efekty místo klasického smazání části obsahu a mnoho dalšího). Právě JavaScript je jednoduché řešení, po kterém knihovny sahají, potřebují-li si přizpůsobit nějaké řešení svým potřebám.[DENÁR, Michal, 2015 [cit. 2016-04-16], s. 80-82] Do každé stránky rozhraní lze vkládat libovolný javascript, takže upravit lze prakticky všechno. Díky jQuery lze tedy definovat vlastní události a přiřazovat k nim nové či pozměněné akce nebo měnit či přidávat efekty.[*jQuery*, 2010, s. 3-9]

Vnější část problému je uživatelské rozhraní, tedy OPAC<sup>8</sup>, se kterým uživatelé neumí pořádně pracovat. Při provádění výzkumu měl autor studie možnost vyzkoušet si práci referenčního knihovníka na Filozofické fakultě Masarykovy univerzity, z čehož vyplynulo, že uživatelé OPACu opakovaně nebyli schopni dohledat informace, které potřebovali, a žádali o pomoc referenční knihovníky. Většinou to byly problémy jako nedohledatelná kniha na regálu, nemožnost inteligentně dohledat knihu na nějaké téma v ruském jazyce a podobně.

7. ILL modul Koha - [https://github.com/PTFS-Europe/koha/tree/ill\\_master](https://github.com/PTFS-Europe/koha/tree/ill_master)

8. Online Public Access Catalog



Problém s jazyky byl ten, že OPAC nabízel například 3 jazyky z určitého MARC pole, zatímco se reálně prohledávalo pole jiné. V době psaní této práce existovala někde v rozhraní tabulka se zkratkami jazyka, které bylo možné použít pro prohledávání. Uživatelé, kteří o tabulce nevěděli, byli nuceni zvolit jeden z jazyků, spustit vyhledávání a v URL změnit zkratku jazyku za jinou, kterou si tipli, že je správná pro daný jazyk.

Tyto problémy lze v systému Koha vyřešit zaprvé správným zaindexováním dat a zadruhé uživatelsky přívětivým OPACem, kterým se osvědčil být VuFind vyvíjený jako open source na Villanova University.[Coufalová, Jindřiška, 2009, s. 12] Tento OPAC používá drtivá většina knihoven v České republice provozující systém Koha (zbytek knihoven je maximálně nenáročných a zůstali na interním Koha OPACu). VuFind umožňuje uživatelům vyhledávat v celém fondu knihovny a nabízí i doplňkové služby a to celé přehledně a jednoduše. VuFind je modulární, o čemž svědčí i fakt, že je na něm postaven český Centrální portál knihoven<sup>9</sup>, který poskytuje přístup k fondům a službám všech zapojených knihovnických i neknihovnických institucí v rámci republiky. Centrální portál knihoven je vyvíjen jako open source a jeho moduly jsou dostupné v repozitáři na Githubu<sup>10</sup>. Lze se tak inspirovat řešením, které prošlo uživatelským testováním a je založeno na uživatelsky přívětivém moderním designu.

VuFind je navíc discovery systém, který lze v rámci Masarykovy univerzity využít k zapojení externích elektronických informačních zdrojů, což OPAC Kohy neumožňuje. VuFind je také stavěný na práci se Solrem, což je databázový systém vhodný pro práci s Big Data<sup>11</sup>. Podle Alfreda Serafiniho<sup>12</sup> je to vhodný nástroj pro prohledávání velkého množství read-only metadat, kde jako příklady využití uvádí právě VuFind a Internet Archive<sup>13</sup>.

Idea rozhraní VuFind je založena na teorii, že uživatele nezajímá cesta, jakou se k dokumentu dostane, ale v jaké formě je k dispozici a jak je možné ho získat v rámci instituce.[Coufalová, Jindřiška, 2009, s. 20] Lze tedy tvrdit, že katalog, který slouží knihovníkům, nemůže být stejný jako OPAC[Schmidt, Aaron, 2012, s. 5], jelikož ten má být designován pro uživatele, kteří s ním interagují. OPAC je veřejnou částí knihovnického systému, tedy vstupní bránou uživatele, která musí být uživatelsky přívětivá, musí tedy splnit potřeby uživatele, které lze vydefinovat pomocí Maslowovy pyramidy webde-

9. Centrální portál Knihoven (CPK) - <https://knihovny.cz/>

10. CPK repozitář <https://github.com/moravianlibrary/VuFind-2.x>

11. Více informací o BigData k nalezení v - BigData, ViktorMayer citace

12. Freelance software konzultant, autor publikace Apache Solr Beginner's Guide (ISBN 978-1-78216-252-0).

13. Internet Archive - <http://www.archive.org/>

signu[Řezáč, Jan, 2014, s. 156-179], kam patří důvěryhodnost, radost z používání, vytvoření vazby, přesvědčivost, použitelnost, přístupnost, dostupnost, naležitelnost a smysluplnost.

Tak jako Koha, i VuFind je postaven na webových technologiích, což dává designerům a softwarovým architektům obrovské možnosti v rámci přizpůsobení veřejného rozhraní. Pomocí HTML5 lze rozhraní zpřístupnit i zrakově či jinak handicapovaným uživatelům, kterým se tak OPAC zpřístupní pro jejich speciální čtecí zařízení. [Hogan, Brian P, 2011, s. 19-27]

Zajímavostí je také budoucnost těchto technologií, která vnáší do oboru knihovnictví sémantičnost v rámci webového rozhraní. Díky HTML5 a sémantickým technologiím, jako je například mikroformát RDF (a také samotná podstata sémantiky v HTML5), je možné dávat jednotlivým částem webu význam, což z něho dělá lépe prohledávatelný web na Internetu, na což vzápětí navazuje i chytré vyhledávání na webu za využití umělé inteligence.[Fay, Robin M, 2012, s. 70-77] Takto lze OPAC připravit na budoucnost webu 3.0, respektive webu 4.0.

VuFind je navíc založen na myšlence responsivního designu a využívá webového frameworku Bootstrap 3, je tedy responzivní ve smyslu přístupnosti na zařízeních s různou uhlopříčkou a rozlišením, takže je stejně použitelný jak na desktopech, tak i na tabletech či smartphonech.

### Služba Copy on Demand (COD)

Tato služba přímo v Koha ani ve VuFindu neexistuje, avšak je natolik triviální, že její napsání nezabere víc jak pár hodin. Tuto funkci lze také řešit externí službou, která vygeneruje odkaz (tlačítko), který stačí jednoduše embedovat<sup>14</sup> do šablony OPACu.

### Obálky knih

Nejen obálky knih, ale i naskenované obsahy a indexy lze zpřístupnit ve VuFindu pomocí API ObálekKnih<sup>15</sup> ve verzi 3. Příklad implementace ve VuFindu lze vidět na Centrálním portálu knihoven, respektive v repozitáři<sup>16</sup> na Gthubu.

14. Vložit.

15. API ObálkyKnih.cz - [https://obalkyknih.cz/doc/Dokumentace\\_API\\_OKCZ\\_3.1.pdf](https://obalkyknih.cz/doc/Dokumentace_API_OKCZ_3.1.pdf).

16. Implementace ObálekKnih v CPK <https://github.com/moravianlibrary/VuFind-2.x/tree/master/themes/obalkyknih-api-v3-bootstrap3>

Automatická oprava překlepů a nabídka alternativ při neúspěšném hledání

Tuto funkcionalitu lze zapnout vyhledávací komponentou `spellcheck`.

### 2.3.11 Systémové požadavky

Následující část rozebírá hlavní systémové požadavky na knihovní systém Masarykovy univerzity.

Velký počet knihovních jednotek, uživatelů, poboček, podpoboček

V této chvíli je potřeba, aby nový systém zvládl cca 2 miliony knihovních jednotek, cca 50 000 aktuálně registrovaných uživatelů.

Toto kritérium by nemělo být problémem, jelikož např. síť 900 aktivních knihoven v Turecku (z celkového počtu 1 126) obsluhuje 13 700 000 knihovních jednotek a 1 220 000 aktivních uživatelů v jedné instalaci systému Koha. Používají na to 14 produkčních serverů, 4 testovací, MariaDB a Galera Cluster se čtyřmi nody, 8 Apache serverů a jeden Zebra server.

Koha je navíc škálovatelná<sup>17</sup>, lze tedy tvrdit, že počet obslužených jednotek a uživatelů je záležitost technická. Toto škálování (horizontální i vertikální)[Liu, C. Y. 2014, s. 1-4] je možné provádět například pomocí open source produktů OpenStack a OpenShift společnosti RedHat.

OpenStack řeší infrastrukturu (infrastructure as a service - IaaS), to znamená poskytování bootovatelného virtuálního stroje, networkingu, datového úložiště a dalších.[Breeding, Marshall, 2012, s. 14-16] OpenShift jako platforma (platform as a service - PaaS) zase řeší rychlý deploy LAMP<sup>18</sup> aplikací.

#### Flexibilita

Jedním ze základních požadavků na knihovní systém v knihovnách Masarykovy univerzity je vysoká potřeba vše členit do složitějších hierarchických struktur (uživatelé, katalogy, pobočky). Velké univerzity nemají typicky jednu knihovnu, ale celou síť navzájem propojených knihoven, kterou musí být systém schopný obsluhovat jednotně a současně podporovat jejich odlišnosti. Masarykova univerzita má 10 fakultních knihoven, které momentálně operují nad společným systémem/katalogem, fungují však do značné míry

17. Schopnost systému pružně reagovat na vzrůstající nebo klesající nároky uživatelů na výkon systému.

18. Linux, Apache, MySQL/MariaDb, PHP/Perl/Python.

samostatně. Mají odlišné zejména otevírací doby, statusy knih, výpůjční doby a pokuty, akvizice a fakturace jednotek. Společnou však mají centrální bázi uživatelů s možností lokální registrace na pobočkách.

Důležitá je také podpora konsorcií. Celá Masarykova univerzita se chová svým způsobem jako konsorcium. Kromě toho ale může nastat případ, kdy instalovaný systém bude potřeba provozovat skutečně pro více různých nezávislých institucí - konsorcium.

Velká flexibilita je potřebná prakticky ve všem, například při propojování záznamů při zpracování specifických typů dokumentů (přítisky, pří vazky, monografická čísla a podobně).

Při současném řešení je nutné řešit vícero samostatně prohledávatelných knihovních databází (s možností dalšího členění do dílčích knihoven a studoven) a souborný katalog pro celou univerzitu.

Z pohledu systémové vrstvy existují při instalaci systému Koha tři možnosti:

- samostatná instalace pro každou knihovnu
- jedna instalace a instance pro každou knihovnu
- jedna instalace a knihovny jako jednotlivá oddělení (některá mohou být nezávislá - nevidí zbytek)

Samostatná instalace je nezávislá ve všech směrech. Instalace pro jednotlivé knihovny mohou zjednodušit správu, protože vše běží na jednom serveru. To znamená, že aktualizace a správa systému se dělá jen jednou. Navíc je využita jedna databáze, takže existuje teoretická možnost automatizované synchronizace subdatabází jednotlivých instancí.[Breeding, Marshall, 2012, s. 19-21]

Jednou z možností, které by nejvíce vyhovovaly stávající architektuře systému Aleph a systémům, které je potřeba zapojit, se jeví schéma na obrázku 3. Finální návrh struktur instalace však vyjde až z kompletních informací ohledně rozdělení jednotlivých knihoven a poboček a požadavků na jejich samostatnost a začlenění do skupin a hierarchie.

### Robustnost a vysoká provozní spolehlivost systému

Tak robustní systém jako ILS pro MU může být náchylný k výpadkům, spravuje-li velké množství dat a dotazů. Tomu je potřeba zabránit nebo to alespoň omezit na minimum. V případě výpadku sítě či napájení je nutné systém obnovit ze zálohy, aniž by došlo k poruše konzistence či integrity dat.

Systém musí fungovat non-stop. Při problémech způsobených nějakou vstupní chybou nesmí tyto chyby způsobit výpadek systému nebo porušit integritu data tyto chyby/výjimky musí být korektně odchyťovány a ošetřovány. S provozní spolehlivostí také souvisí výkon. Důležitým aspektem je jak rychlá odezva, tak rychlé zpracování velkého množství dat. O škálování systému se již psalo v kapitole 2.2.1 Velký počet knihovnických jednotek, uživatelů, poboček, podpoboček, avšak je potřeba myslet i na to, že neustále zvyšování nároků na výpočetní výkon také zvyšuje nároky na koncové stanice uživatelů.

Koha také plně podporuje technologii PSGI/Plack, která slibuje zrychlení většiny každodenních úkonů. PSGI je rozhraní mezi perlovskou aplikací a serverem, zatímco samotný Plack je perlovským modulem a nástrojem, který obsahuje PSGI middleware, helpery a adaptéry pro web server.

### Otevřenost systému

Systém musí být technicky připraven na to, aby mohl být napojen na další systémy využívané v rámci Masarykovy univerzity, tj. personální, ekonomický, studijní.

Celkově fakt, že Koha je open source systém, dává případnému napojování na další systémy velké možnosti, jelikož je možné do kódu zasahovat a přizpůsobovat si tak knihovní systém svým potřebám, aniž by knihovna musela žádat výrobce o novou funkcionalitu.

V případě, že jiná knihovna již tento krok podstoupila, je možné v rámci komunity tento kód implementovat i do forku (derivátu repozitáře)[Chacon, Scott, c2009, s. 29] té dané knihovny, tudíž není potřeba znovu vyvíjet software s ekvivalentní funkcionalitou nebo nesmyslně platit za již zaplacenou věc. Knihovna samozřejmě nemusí mít vlastní fork a může přispívat do upstreamu (zdrojový repozitář), odkud si Kohu vzala, tudíž sdílí novou funkcionalitu s komunitou.

V případě, že organizace má vlastní vývojáře, je rychlost vývoje OSS rychlá a relativně bezproblémová a navíc projde kód kontrolou nejen formální (kód samotný), ale i věcnou (zakomponování do zbytku systému). OSS reaguje na požadavky uživatelů, kdežto proprietární software nabízí funkcionalitu na základě své strategie nebo obchodního modelu. Některé funkce tak díky vysokým nákladům nemusí být nikdy implementovány, výrobcům se zkrátka nevyplatí. Pokud se mu tyto části funkcionality mají vyplatit vyvíjet, nasadí podle toho i jejich cenu. Výsledkem jsou služby, které si některé knihovny nemohou dovolit, což má vliv na jejich začlenění do kooperativních projektů nebo na kvalitu a rozsah poskytovaných služeb.

## Bezpečnost systému

Bezpečnost systému lze uvažovat nejméně ze tří úhlů pohledu. První úroveň uvažuje bezpečnost v rámci přístupu do systému a jeho částí (autentizace). Druhá úroveň uvažuje bezpečnost týkající se pravomocí jednotlivých skupin uživatelů (autorizace). U těchto úrovní se dá mluvit jako o prevenci zneužití či poškození dat, ať už úmyslného nebo ne. Třetí úroveň lze uvažovat jako logovací, neboli zaznamenávající chování v systému. Veškerá bezpečnost se týká aplikační vrstvy ISO/OSI[Kurose, James F, 2014, s. 80-82] modelu počítačové sítě.

V Koha se připravuje balíčková podpora LetsEncrypt, takže bude poměrně snadné přejít na HTTPS. Pravidelně se řeší bezpečnostní aktualizace (minimálně 1x/měsíc). Hesla jsou šifrována pomocí Bcryptu.

## Shibboleth

Napojení na open source federativní systém pro propojování identit Shibboleth je také možné a konfigurace je dostupná na wiki stránkách<sup>19</sup> mezinárodní komunity.

## Dokumentace

Jedním z požadavků byla dostupná plná dokumentace systému. Pro systém Koha je k dispozici:

- dokumentace systému<sup>20</sup>
- dokumentace k veškerým API a podporovaným protokolům<sup>21</sup>
- český manuál<sup>22</sup>

## Externí aplikace knihoven

Následující externí aplikace knihoven lze řešit přímo knihovním systémem.

- **Dlouhodobé výpůjčky, které letos končí (KUK)** - Tohle by měla zvládnout pravidla výpůjček. Ta se definují od obecných ke specialitám.

19. Konfigurace Shibbolethu pro Koha - [https://wiki.koha-community.org/wiki/Shibboleth\\_Configuration](https://wiki.koha-community.org/wiki/Shibboleth_Configuration)

20. Dokumentace Koha - <https://koha-community.org/documentation/>

21. API a protokoly Koha - [https://wiki.koha-community.org/wiki/APIs\\_and\\_protocols\\_supported\\_by\\_Koha](https://wiki.koha-community.org/wiki/APIs_and_protocols_supported_by_Koha)

22. Český manuál Koha - <http://www.knihovni-system-koha.cz/index.php/navody-a-vzdelavani/cesky-manual-systemu-koha>

Buď podle typu dokumentu nebo kategorie čtenáře, nebo knihovny. Upozornění pak chodí automaticky.

- **Hromadná objednávka** (KUK) - možnost vytisknout více titulů do jednoho objednávkového formuláře. V tomto případě stačí nadefinovat design MU.
- **Rozpočty** - přehled rozpočtů, čerpání financí (KUK)
- **Limity nákupů** (KUK)
- **Přehled fondů pro potřeby akreditací** - vygenerování přehledu počtu svazků knih/časopisů podle majetku fakulty a kolik z toho je ve volných výběrech (KUK). V systému Koha lze řešit jako statistiky na základě metadat.
- **Kontrola řady čárových kódů** – při zpracování knih se občas ztratí čárový kód, je potřeba ho dohledat a obsadit (jeho číslo je zároveň přírůstkové číslo) (KUK)
- **Kontrola chyb** (KUK) - v Koha existuje celá řada chybových kontrol, které lze spouštět ručně nebo je automatizovat.
- **Dopis hříšníků** (KUK)

### 2.3.12 Potřeby a požadavky katalogizace

Katalogizační modul Kohy samozřejmě podporuje správu digitálních informačních artefaktů a má i možnost vytvářet nové multimediální materiály. Modul disponuje přehledným katalogizačním editorem, který lze snadno editovat. Záznamy lze dotahovat přes Z39.50/SRU<sup>23</sup>.

Je možné vytvářet vlastní MARC šablony pro různé typy dokumentů. Položky, které se často opakují či dlouhé seznamy možností je možné definovat do tzv. ověřených hodnot. Tyto ověřené hodnoty se pak zobrazí v šabloně v podobě roletek, ze kterých je možné vybírat.

Národní autority se v Koha dotahují přes Z39.50 (v knihovnách MU pomocí FTP<sup>24</sup>). OAI harvester je pro Kohu ve fázi implementace.

Editor lze rozdělit do záložek 0-9, které definují rozsah polí, viz. obrázek x.y. Tyto záložky lze slovně pojmenovat nebo rozhraní zobrazovat jako celý záznam.

23. Search/Retrieve via URL.

24. File Transfer Protocol - protokol pro transport dat. .

### Hromadné úpravy

Záznamy lze editovat také hromadně pomocí nástroje „Hromadné úpravy jednotek“, kde se nadefinují modifikační šablony, viz obrázek x.x.

Co se katalogizace elektronických zdrojů týče, elektronické zdroje se katalogizují jako knihy, jen se použije jiná šablona. Rozdíl v záznamu je primárně v kódovaných údajích.

### Podpora RDA

Hezkým příkladem, jak lze upravovat frontend za využití Javascriptu, je úprava bibliografických a autoritních šablon pro podporu RDA, kde jsou jednoduše vložena chybějící pole a možnost zobrazovat hodnoty, které lze do polí vložit.

### Stahování bibliografických záznamů přes Z39.50

Koha podporuje protokol Z39.50 jako klient i jako server, to znamená, že dokáže stahovat záznamy například z Národní knihovny, Jednotné informační brány, Souborného katalogu, Moravské zemské knihovny a podobně, ale také povoluje jiným knihovnám stahovat své záznamy (například v rámci konsorcia).

### Rejstříky

Koha umí pracovat s čímkoliv, co má povahu autority (korporace, věcné téma, místo, žánr) a také umí vytvořit řízený slovník čehokoli, co má pak povahu ověřených hodnot, které lze provázat s katalogizační šablonou.

### Omezení práv

V Kohu lze udělovat práva pro to, který personál má mít přístup k editaci jednotek a který k editaci záznamů. To znamená že někteří zaměstnanci nemohou editovat záznam, ale mohou jenom změnit například čárový kód, a podobně. Slouží k tomu komponenty `SubfieldsToAllowForRestrictedEditing` a `SubfieldsToUseWhenPrefill`.

### Autoritní hesla

Výběr autoritních hesel je pro Kohu samozřejmostí. Možnost dohledat autoritu zobrazuje obrázek x.g.



### 2.3.13 Potřeby a požadavky akvizice

Modul akvizice disponuje běžnými funkcemi, které jsou pro procesy akvizice v knihovných potřeba, to znamená, že Koha dokáže vytvářet dodavatele, aplikovat slevy, rozlišovat ceny s DPH a bez DPH a podobně. Tyto údaje tedy pomáhají při příjmu objednávky, kdy lze jednoduše zkontrolovat fakturu. Objednávku lze vyvářet jednoduše, a to vytvořením jednoho nebo více košíku, kde se pro každého dodavatele navolí jednotlivé položky objednávky. Záznamy těchto položek lze doplnit buď z vlastní databáze, nebo dotáhnout přes Z39.50 například z Česká národní bibliografie nebo Souborného katalogu.

Jedním z požadavků bylo, aby nový systém uměl evidovat adresy dodavatelů, IČ a DIČ, což pro Kohu není problém.

V systému Koha je také možné dohledat, kolik knih bylo koupeno a z jakých zakázek. Ke každému dodavateli jsou přiřazené objednávky. Když se uzavřou, je možné je exportovat do tabulkového formátu nebo PDF. Ve vygenerované tabulce lze označovat, které položky objednávky v balíku dorazily, a tedy vidět, které položky nebyly pokryty. Takto se vytvoří seznam dokladů, ke kterým se přiloží naskenovaná faktura. Částky se odepíší z rozpočtů, které mohou mít různé složky a fondy. Koha sama sleduje vykrytí, má jednoduché predikce plnění a upozorňuje, když se přibližuje k vyčerpání částky nebo procentuální části. Příklad objednávky lze vidět na obrázku x.x.x.

## 2.4 Dopad projektu na životní prostředí

Systém Koha je napsán v perlu, neupovídaném stručném skriptovacím jazyce [Wall, Larry, 1997, s. v-ix], což má za následek také menší celkovou velikost zdrojového kódu. To vede k menší náročnosti na zpracování na straně serverů a menší energetickou zátěž, tudíž lze tvrdit, že více šetří životní prostředí. Jak tvrdí motto jQuery, který využívá Koha i VuFind: „*Write less, do more*“.

## 2.5 Finanční plán a ekonomická analýza projektu

Při řešení finanční otázky projektu je potřeba myslet zejména na to, že finance ušetřené na koupi nebo podporu systému bude nutné vložit do mezd personálu, který bude migraci provádět. Tento projekt bude realizován několik měsíců a v každé etapě bude potřeba jiné skupiny lidí. Tyto lidské zdroje však budou v plném nasazení potřeba jen ze začátku, kdy se bude

migrovat a pak systém kustomizovat. Poté náklady na pracovní skupiny klesnou na potřebnou systémovou podporu, která se bude starat o chod a správu systému.

Ve chvílích, kdy bude potřeba psát nové funkce či upravovat ty stávající, je vhodné je konzultovat s komunitou. To z toho důvodu, že bude-li mít o funkcionalitu zájem více institucí, mohou se náklady rozdělit mezi zájemce v komunitě, a to jak finanční, tak i ty lidské.

## 2.6 Hodnocení efektivity a udržitelnosti projektu

Celé hodnocení efektivity a udržitelnosti projektu ve velké míře závisí na tom, jak se případný objednatel (v tomto případě MU) postaví k výsledkům SWOT analýzy a jak využije silné stránky a jak se vypořádá se slabými stránkami projektu. Následující SWOT analýza byla prováděná v horizontu 6 měsíců, který probíhal současně s průzkumem mapování knihovních procesů v MU. Hlavním předmětem analýzy byla migrace na open source systém Koha. Analýza byla prováděna jako instalace nové Kohy, vytvoření instancí a následného zjišťování externí podpory, rozhovorů s knihovníky a systémovými knihovníky MU, Ústavem výpočetní techniky a knihovnami, které migrací prošly buď samostatně nebo s využitím externí komerční podpory. Analýza je zpracována ke dni 07. 03. 2015 a její životnost se odhaduje na 5 let (v případě její další revize).

Udržitelnost projektu, který by byl zrealizován, je závislá na mnoha faktorech. Jedním a zároveň tím nejzásadnějším je způsob vývoje a udržování systému. Systém je totiž možné provozovat tak, že se vyklonuje<sup>25</sup> vzdálený Koha repozitář z upstreamu<sup>26</sup>, do kterého v budoucnu bude možné jednoduše stahovat<sup>27</sup> aktualizace z upstreamu. Druhá cesta je vytvořit kopii vzdáleného repozitáře a vývoj dělat v rámci svého forku. To sebou nese jak výhody, tak i nevýhody. Hlavní výhodou je, že změny, které se udělají ve forku, nebude nutné procházet zdlouhavým code review na straně Koha komunity, což má za následek i to, že MU nebude závislá na tom, zda novou funkcionalitu komunita zařadí do standardního balíčku v master větví<sup>28</sup> repozitáře. Tento krok však může MU využít, bude-li se jednat o nějakou všeobecnou univerzální funkcionalitu.

25. Pořídí kopie.

26. Zdrojový repozitář verzovacího systému.

27. Stahovat - pullovat nové verze dat z repozitáře. Opakem pullu je push, kdy se to repozitáře vkládají nové verze dat.

28. Hlavní větev v repozitáři verzovacího systému Git.

Bude-li to lokální záležitost, může to provozovat ve svém forku a o nic se nestarat. Tato varianta se jeví jako nejvhodnější, ale je potřeba počítat s tím, že bude-li chtít MU stahovat aktualizace z upstreamu, mohou nastat konflikty v kódu, které bude potřeba řešit ve smyslu slučování<sup>29</sup> dvou rozdílných verzí souboru. Třetí cesta je pak, že MU naklonuje Koha repozitář, který již nebude slučován s upstreamem, tedy úplně se odřízne od upstreamu. Toto řešení však není vhodné kvůli budoucí správě systému v případě, kdy by MU chtěla externí pomoc a verze Kohy a upstreamu by se hodně lišila.

## 2.7 Analýza a řízení rizik

V průběhu provádění všech etap a hlavně po celkovém ukončení realizace je potřeba mít možnost poskytnutí zpětné vazby a fungující komunikaci v celém konsorciu. V kterékoliv z etap realizace projektu mohou nastat problémy, které je vhodné řešit co nejdříve, aby nedošlo k dalším komplikacím. Celá realizace stojí a padá také na vedení a proto je nutné, aby v případě problémů vedení tyto problémy diskutovalo a řešilo.

Může nastat situace, kdy se změni směrování ústavu a open source již nebude hlavní myšlenkou dne. V tom případě se při nejhorším ztratí čas a úsilí věnované projektu (s tím spojené i mzdové náklady). Knihovny však nebudou jinak zasaženy a budou fungovat stávajícím způsobem. Je vhodné migraci, respektive implementaci nového systému, provádět na testovacích serverech a data importovat do testovacích databází, aby se zabránilo případným nechtěným kolizím.

Další riziko představuje fakt, že problémy, které MU měla při stávajícím systému, přetrvávají i po migraci na Kohu. V tom případě problém nebyl knihovní systém, ale špatná komunikace, nastavení nebo procesy.

Rizikem může být i to, že nový systém přinese mnoho změn, které se v tak konzervativním prostředí, jako je knihovnictví, těžko uplatňují. V tomto případě je dobré vymyslet plán zaškolení a dalšího vzdělávání, minimálně co se nového systému týká.

Může nastat situace, kdy se prodlouží podpora Alephu. Tím se ale nevyřeší problémy, které stávající systém má. Zruší-li se podpora Alephu, MU na něm stále může fungovat, avšak je potřeba myslet i na Aleph klienta, který již nebude optimalizován na nové verze operačních systémů, tudíž se stane nepoužitelným.

Finanční riziko zase představuje neušetření financí (alespoň na začátku) při migraci na open source. Důvodem může být využití investičních ná-

29. Situace, kdy je potřeba spojit 2 verze souboru, se nazývá merge.

kladů na mzdách pracovníků, kteří budou nový systém upravovat a nastavovat.

Riziko může také nastat v době, kdy budou knihovníci muset vývojářům a systémovým knihovníkům definovat, jak má která služba fungovat nebo jak se mají data interpretovat. Může se stát, že to ve skutečnosti nebudou vědět. Je proto potřeba pečlivě vybrat pracovní skupinu s nejvyšší odborností.

Další riziko je zhroucení projektu ve formě nevedených zaměstnanců. Nebudou-li zaměstnanci řízení, celý projekt se může časově i finančně protáhnout nebo se dokonce zhroutit.

## Závěr

Knihovní systém by měl být systémem, který pomáhá knihovníkům v poskytování kvalitních služeb svým uživatelům - čtenářům. Ten je však kvůli neustále se měnícím potřebám uživatele potřeba upravovat. Jednou z možností jak toho dosáhnout je pomocí využití open source přístupu. Migrace na open source knihovní systém s sebou přináší velké změny. V první řadě jsou to politické a právní otázky určující směr a možnosti, na které navazuje financování. Situace, kdy je správce knihovního systému bezmocně odkázán na podporu poskytovatele (vendor lock-in), se náhle změní, je-li možné přelévat investiční a mzdové náklady instituce. Jak píše RNDr. Miroslav Bartošek, CSc.: *„Knihovní systém nesmí mít podobu ‚černé skřínky‘, do které správce ‚nevidí‘ dostatečně hluboko a je s každým nestandardním zásahem a požadavkem odkázán na milost či nemilost dodavatele systému.“* Jsem přesvědčen, že migrace na open source knihovní systém Koha nejen v tomto pomůže. Je potřeba si ale uvědomit, že ne všechny problémy lze řešit knihovním systémem, ale že je potřeba upravovat samotné procesy, v kterých k problémům dochází. Nakonec je to vždy lidský faktor, který je na vstupu i na výstupu každého systému. Samotný proces migrace s sebou nese nejen výhody, ale také rizika, které je potřeba řídit a minimalizovat tak hrozby, které z něj vyplývají. Projekt je svým rozsahem a složitostí nutné rozdělit na menší, jednodušší části (etapy). V těchto etapách musí projektový manager řídit jak lidské, tak i materiální zdroje tak, aby byly co nejefektivněji využity. Je potřeba myslet na to, že projektový manager nebo leader je osoba, která bude spojovat jednotlivé pracovní skupiny projektu. Musí být také komunikačním centrem, kterému lze při realizaci projektu poskytovat zpětnou vazbu. Problémy byly, jsou a budou, avšak s přátelskou pomocí komunity a odhodlaným projektovým managerem je lze zvládnout. Jak práce dokazuje na příkladech zahraničních knihoven a univerzit, které již migraci podstoupili, přechod z komerčního systému Aleph na Kohu možný je, avšak je nutné sestavit podrobnější harmonogram realizace a detailně rozpracovat (hlavně sdílení uživatelů a MVS modul), jak se budou řešit jednotlivé kroky migrace, což je svým rozsahem možné rozpracovat v další diplomové práci. Výstupem teoretické části by pak mohl být harmonogram realizace projektu, zatímco praktická část by nabízela konkrétní, nejen softwarová řešení všech problémů, které je nutné vyřešit pro úspěšnou migraci knihoven MU na open source systém Koha.

## Appendix

V případě zájmu o migraci na systém Koha je možné využít pomoci komunity, ať už české<sup>30</sup> nebo mezinárodní<sup>31</sup>, nebo také komerční podpory<sup>32</sup>. Sledovat dění v české Koha komunitě lze také na facebookových stránkách<sup>33</sup> komunity a na wiki stránkách Githubu<sup>34</sup>. Otázky k problematice je možné klást do českého chatu<sup>35</sup> na Gitteru nebo se zeptat kterékoli knihovny<sup>36</sup>, která migraci podstoupila. Vhodné médium ke komunikaci problémů anebo jiných informací je také mailing list Koha.

---

30. Česká Koha komunita - <http://www.koha.cz/>

31. Mezinárodní Koha komunita - <https://koha-community.org/>

32. Komerční podpora Koha - <http://www.koha-v-knihovne.cz/>

33. Facebook Koha - <https://www.facebook.com/kohaCZ/>

34. Wiki stránky Koha - <https://github.com/open-source-knihovna/KohaCZ/wiki>

35. Koha e-mail Chat - <https://gitter.im/open-source-knihovna/KohaCZ>

36. Seznam knihoven, které v ČR používají systém Koha v běžném provozu - <https://github.com/open-source-knihovna/KohaCZ/wiki/Seznam-knihoven,-kter%C3%A9-v-%C4%8CR-pou%C5%BE%C3%ADvaj%C3%AD-syst%C3%A9m-Koha-v-b%C4%9B%C5%BE%C3%A9m-provozu>

Obrázek 1.1: Důsledky trojimperativu.

Obrázek 1.2: Vzájemné závislosti v procesu řízení projektu.

## Seznam tabulek



## Seznam obrázků

- 1.1 Důsledky trojimperativu. 40
- 1.2 Vzájemné závislosti v procesu řízení projektu. 40