

DIPLOMOVÁ PRÁCE

Optimalizace business procesů virtuální infrastruktury firmy Gemalto s.r.o.

Process optimization of virtual infrastructure for company Gemalto s.r.o.

STUDIJNÍ PROGRAM

Podnikání a komerční inženýrství v průmyslu

STUDIJNÍ OBOR

Podnikání a management v průmyslu

VEDOUCÍ PRÁCE

Ing. Jiří Kaiser, Ph.D.

SEDLÁČEK

ONDŘEJ

2018

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: Sedláček Jméno: Ondřej Osobní číslo: 372869
Fakulta/ústav: Masarykův ústav vyšších studií (MÚVS)
Zadávající katedra/ústav: Katedra managementu, MÚVS
Studijní program: Podnikání a komerční inženýrství v průmyslu
Studijní obor: Podnikání a management v průmyslu

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:
Optimalizace business procesů virtuální infrastruktury firmy Gemalto s.r.o.

Název diplomové práce anglicky:
Process optimization of virtual infrastructure for company Gemalto s.r.o.

Pokyny pro vypracování:

Cílem diplomové práce bude analýza a optimalizace business procesů spojených s provozem i správou virtuální infrastruktury. V průběhu práce bude vytvořen automatizační framework, který kladně přispěje k optimalizaci zkoumaných procesů. Zmiňované procesy souvisí s požadavky uživatelů na změny ve virtuální infrastruktuře (vytvoření virtuálního stroje, klonování, mazání, ...). Součástí bude také ekonomické posouzení používaných serverů a kalkulace investičně výhodnější varianty. Výstupem práce budou vymodelované procesy pomocí BPMN, a jejich následná distribuce mezi zaměstnance, čímž se přispěje k jejich oficiálnímu nastavení. Ekonomická analýza bude předložena managementu.

Seznam doporučené literatury:

NEUSTADT, Ila; ARLOW, Jim. UML 2 a unifikovaný proces vývoje aplikací. Computer Press, Albatros Media as, 2016.
MOLNÁR, Zdeněk. Podnikové informační systémy. Vyd. 2., přeprac. V Praze: České vysoké učení technické, 2009.
ŘEPA, Václav. Procesně řízená organizace. Grada Publishing as, 2012.


Jméno a pracoviště vedoucí(ho) diplomové práce:
Ing. Jiří Kaiser, Ph.D., Katedra managementu, MÚVS

Jméno a pracoviště konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: 8.2.2017 Termín odevzdání diplomové práce: 5.5.2017

Platnost zadání diplomové práce: tři semestry


Podpis vedoucí(ho) práce

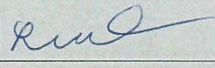

Podpis vedoucí(ho) ústavu/katedry


Podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

15.5.2017

Datum převzetí zadání


Podpis studenta(ky)

SEDLÁČEK, Ondřej. *Optimalizace business procesů virtuální infrastruktury firmy Gemalto s.r.o.* Praha: ČVUT 2017. Diplomová práce. České vysoké učení technické v Praze, Masarykův ústav vyšších studií.



**MASARYKŮV ÚSTAV
VYŠŠÍCH STUDIÍ
ČVUT V PRAZE**

Čestné prohlášení autora práce

Prohlašuji, že jsem svou diplomovou práci vypracoval samostatně. Dále prohlašuji, že jsem všechny použité zdroje správně a úplně citoval a uvádím je v přiloženém seznamu použité literatury.

Nemám závažný důvod proti zpřístupnění této závěrečné práce v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) v platném znění.

V Praze dne 10. 1. 2018

.....
Podpis autora práce

Poděkování

Rád bych zde poděkoval svému zaměstnavateli za ohromnou shovívavost, která umožnila vznik této práce. Zároveň děkuji mému vedoucímu diplomové práce Ing. Jirímu Kaiserovi, Ph.D. za odborné rady, připomínky a trpělivost, kterou se mnou měl při řešení dané problematiky.

Abstrakt

Tato práce řeší optimalizaci podnikových procesů spojených se správou virtualizační infrastruktury. Praktická část je zaměřena na popis teoretických základů, potřebných pro orientaci v dané problematice. Praktická část obsahuje analýzu a optimalizaci zkoumaných procesů společně s ekonomickým zhodnocením v případě vytvoření automatizačního frameworku. Vzhledem ke kladnému ekonomickému posouzení byl i automatizační framework vytvořen.

Klíčová slova: optimalizace procesů, podnikový proces, hodnocení investic, virtualizace, automatizace, framework, BPMN, VMware

Abstract

This master thesis deals with optimization of business processes related to managing a virtual infrastructure. The practical part is focused on the description of theoretic terms which are a prerequisite for understanding discussed problems. The practical part is focused on optimization of examined processes. There is also economical evaluation of the creation of the automation framework. The framework has been created due to the positive economic evaluation.

Key words: process optimization, business process, investment evaluation, virtualization, automation framework, BPMN, VMware

Obsah

1	Úvod.....	7
2	Teoretická část	9
2.1	Proces	9
2.2	Procesní řízení	10
2.3	Unified Modeling Language	11
2.3.1	Diagramy struktur	13
2.3.2	Diagramy chování	15
2.3.3	Diagramy interakce	19
2.4	BPMN.....	19
2.4.1	Aktivity	20
2.4.2	Události	23
2.4.3	Brány.....	27
2.4.4	Spojovací objekty.....	28
2.4.5	Bazény s plavecké dráhy.....	29
2.4.6	Artefakty	30
2.5	Hodnocení investic.....	30
2.5.1	Statické metody.....	30
2.5.2	Dynamické metody	31
2.6	Technický popis frameworku.....	32
2.6.1	PHP	32
2.6.2	HTML	32
2.6.3	Python	32
3	Praktická část	33
3.1	O firmě	33
3.2	Cílová skupina.....	34

3.3	Původní procesy ve virtuální infrastruktuře	35
3.3.1	Vytvoření nového stroje.....	35
3.3.2	Přejmenování stroje, změna vlastníka.....	37
3.3.3	Odstavení stroje	40
3.3.4	Klonování stroje.....	40
3.3.5	Změna výpočetních prostředků.....	41
3.3.6	Generování informací o stavu virtuální farmy a účtování	44
3.4	Nalezení neoptimálních procesů	45
3.4.1	Vytvoření nového stroje.....	45
3.4.2	Přidělení přístupu	47
3.4.3	Přejmenování stroje, změna vlastníka.....	47
3.4.4	Odstavení stroje	47
3.4.5	Klonování stroje.....	48
3.4.6	Změna výpočetních prostředků.....	48
3.4.7	Generování informací o stavu virtuální farmy a účtování	49
3.5	Navržení nových procesů	50
3.5.1	Vytvoření nového stroje.....	50
3.5.2	Přidělení přístupu	53
3.5.3	Přejmenování stroje, změna vlastníka.....	55
3.5.4	Odstavení stroje	57
3.5.5	Klonování stroje.....	57
3.5.6	Změna výpočetních prostředků.....	58
3.5.7	Generování informací o stavu virtuální farmy a účtování	61
3.5.8	Automatické zálohování	62
3.6	Požadavky na framework	64
3.7	Analýza nákladů pro vytvoření frameworku.....	66
3.7.1	Časová náročnost původních procesů	66

3.7.2	Časová náročnost optimalizovaných procesů	69
3.7.3	Vlastní náklady na zkoumané procesy.....	69
3.7.4	Odhad nákladů na implementaci jednotlivých procesů v rámci frameworku....	70
3.8	Zhodnocení investice.....	72
3.8.1	Diskontní míra	72
3.8.2	Cash flow	72
3.8.3	Čistá současná hodnota	75
4	Závěr	77
5	Citovaná literatura.....	79
6	Seznam obrázků	83
7	Seznam tabulek	85
8	Seznam souborů na přiloženém CD.....	86

1 Úvod

S nástupem informačních technologií se masivně rozšířilo i jejich využívání. Kromě složitých výpočetních operací počítače pomáhají ve zdravotnictví při operacích, v automobilovém průmyslu pro zlepšení bezpečnosti, nebo například při opakování podobných úkonů.

Právě automatizace stále stejných činností přináší v případě využití počítače úsporu nejen času, ale také ve formě finančních nákladů. Dle Davida Kirka se jedná přibližně o čtyřicet až sedmdesát pět procent nákladů, jejichž návratnost může být od několika měsíců i po několik let. [1]

Specifikem 21. století je turbulentní doba a splnění zadaných požadavků se očekává v co nejkratším čase. Ve světě výpočetní techniky se jedná o řády jednotek vteřin. Druhým specifikem doby je tlak na ekonomickou stránku veškerých činností. Eliminace nákladů a zlevnění procesu je společně s urychlením veškerých činností zcela klíčové pro většinu manažerských rozhodnutí.

Vzhledem k těmto faktům, tedy za účelem snížení množství času, který administrátor virtuální infrastruktury firmy měsíčně tráví na požadovaných úkolech, je cílem práce zmapování současných procesů, které jsou spojené se správou serverové farmy zmíněné společnosti a následně provedení jejich optimalizace společně s ekonomickým zhodnocením implementace.

Obsahem první části práce bude vysvětlení nejdůležitějších pojmů a termínů spojených s problematikou mapování a optimalizace podnikových procesů. Zároveň dojde v teoretické části k popsání ekonomických ukazatelů, které se využijí k následnému hodnocení a stručnému popisu programovacích jazyků a technických komponent, v rámci implementace použitých.

V praktické části práce bude zkoumáno, zda je optimalizace současných procesů proveditelná a zároveň bude provedeno ekonomické zhodnocení za použití ekonomických ukazatelů. Tento výstup bude předložen managementu a až po jeho schválení se přistoupí k provedení optimalizace procesů.

Za účelem optimalizace bude naprogramován automatizační framework, který nahradí administrátora při plnění většiny úkonů, nebo mu práci výrazně ulehčí. Díky tomu se participace administrátora v jednotlivých procesech eliminuje na nezbytné minimum, případně zcela vypustí.

Samotná konkrétní implementace nebude v práci více rozebírána, jelikož se jedná o velmi technickou problematiku, která je ekonomickému pohledu na podnikové procesy velmi vzdálená a jedná se pouze o nástroj, který bude k optimalizaci použit.

Součástí optimalizace bude i vyhodnocení možnosti nákupu nových serverů, které by zvýšily množství spuštěných virtuálních počítačů a tím pádem i vlastní cash flow daného firemního týmu.

2 Teoretická část

V této kapitole jsou popsány veškeré teoretické základy, potřebné pro pochopení praktické části práce. Jsou zde jasně nadefinovány pojmy i postupy, používané níže při analýze a optimalizaci všech zkoumaných business procesů, stejně jako ekonomických ukazatelů.

2.1 Proces

Podnikový proces je dle definice „objektivně přirozená posloupnost činností, konaných s úmyslem dosažení daného cíle v objektivitě a daných podmínkách“. [2] Vlastností každého procesu je následně existence vstupů a výstupů, vlastníka, zdrojů i nákladů, času potřebného k realizaci, informačního zabezpečení a vnitřní organizační struktury. [3] Procesy se dělí na hlavní a podpůrné.

Nejdůležitější činnosti organizace jsou definovány v rámci hlavních procesů, které vytváří zisk či užitek pro zákazníka. Jejich identifikace je snadná, jelikož prostupují celou organizací. Začínají potřebou zákazníka a končí službou či výrobkem. Každá služba i produkt v rámci organizace musí mít hlavní proces.

Opakem jsou podpůrné procesy, které nepřinášejí napřímo žádný zisk, ale jsou důležité pro úspěšné vykonávání procesů hlavních. Jedná se například o nákup surovin, dopravu, interní ekonomické procesy, zkrátka všechny procesy, které nejsou nijak přímo spojeny se zákazníkem, produktem či poskytovanou službou.

Další možné dělení se zabývá pohledem přístupu, který se aplikuje v případě identifikace jednotlivých procesů. Jedná se o funkční přístup, projektový přístup a procesní přístup. [4]

Nejstarším přístupem je funkční přístup, který vztahuje veškeré aktivity k organizačnímu útvaru. Každá skupina organizace je tak oddělenou jednotkou a stará se pouze o aktivity vykonávané v rámci sebe sama.

Pokročilejším řízením je projektový přístup, ve kterém se zkoumají unikátní a neopakovatelné procesy spojené s konkrétním projektem. Úkol musí být pro tento přístup jasně definován a při plnění nesmí tým překročit náklady, musí dodržet termíny a zároveň dodržovat dobré pracovní vztahy s organizačním okolím. [5]

Posledním, v dnešní době nejvyužívanějším přístupem, je procesní přístup, neboli procesní řízení. V rámci tohoto typu se řízení organizace soustředí na klíčové procesy, které

jsou opakovatelné a prostupují skrze celou organizaci. Jelikož je tento přístup použit v rámci práce, níže je uveden detailnější rozbor.

2.2 Procesní řízení

Tento pojem se začal vyvíjet již okolo roku 1920, kdy Frederic Winslow Tylor nazýval podobné aktivity pojmem „analýza metod a procedur“. [6] Dalším významným krokem k formování termínu přispěli J. Champy a M. Hammer, kteří na začátku devadesátých let podněcovali k přechodu na nový způsob řízení organizací [7] a zdůrazňovali nutnost změny přístupu, jako zcela elementární. Termín je velmi vhodně definován v rámci souboru ITIL¹, jako „soubor činností týkajících se plánování a sledování výkonnosti, především realizačních firemních procesů“. [8] Nejedná se tedy pouze o správné zvolení přístupu k vedení firmy, ale také o sledování těchto procesů a jejich případnou optimalizaci. Procesní řízení se velmi často označuje zkratkou BPM (Business Process Management).

Zcela zásadním znakem procesního přístupu je správné identifikování hlavních činností, které jsou potřebné k satisfakci zákaznickova požadavku. Kromě správné identifikace je klíčová následná systematická optimalizace a kontrola správného nastavení daných procesů. [3]

Pro úspěšnou identifikaci je nutné pohlížet na veškeré organizační procesy globálně, nereflektovat zavedenou organizační strukturu a hledat souvislosti ve všech prováděných aktivitách. Při analyzování procesu je nutné identifikovat nejen nepotřebné a redundantní aktivity, které jsou v organizaci prováděny, ale také určit neefektivní procesy a ty následně optimalizovat. Míra optimalizace závisí na finančních i časových možnostech podniku.

V případě zásadní optimalizace je možné se dostat až do tzv. „reengineeringu procesu“, který se vyznačuje zcela kompletní výměnou procesu za naprosto nový. Rozhodování o rozsahu změn je vždy závislé na konkrétním analytikovi, který danou optimalizaci provádí, a na míře součinnosti uživatelů, na procesu participujících.

Opakující se procesy jsou nastaveny na určitý rozptyl proměnlivosti vstupů, na který jsou schopny do jisté míry reagovat. Občas existují určité případy, při kterých mohou být výsledky procesů ovlivněny, jelikož nezapadají do širě akceptovatelné změny. Pokud se zmíněný problém vyskytuje častěji, je potřeba procesy dále optimalizovat a rozsah akceptovatelných změn zvětšit. [9]

¹ ITIL = Information Technology Infrastructure Library

Existuje velké množství způsobů, kterými je možné procesy modelovat. V rámci této práce bylo použito modelování za pomoci metodiky UML, konkrétně BPMN, která je detailně popsána v následujících kapitolách.

2.3 Unified Modeling Language

UML je univerzální modelovací jazyk, který se používá k modelování procesů. Do roku 1997, kdy se stal průmyslovým standardem, se jednalo pouze o metodiku. V současnosti se jedná se o standardizovaný zápis, který se využívá při modelování, specifikaci systémů či vizualizaci převážně počítačových systémů. Umožňuje zobrazit všechny existující procesy díky sjednocení dříve vyžívaných modelovacích technik. Sjednocuje osvědčené postupy z oblasti softwarového inženýrství, a za pomoci vizuálního jazyku, který reprezentuje, slouží pouze jako zobrazovací nástroj. [10]

Od roku 1970 se snažili softwaroví vývojáři nalézt objektově orientovaný modelovací jazyk, kterým by vhodně dokázali zobrazit implementované procesy. Stovky uživatelů naráželi na problém neexistence komplexního modelovacího jazyku, který by zcela uspokojil veškeré jejich potřeby. Začátkem devadesátých let se začaly objevovat upravené verze doposud používaných popisů, případně docházelo ke spojování různých přístupů.

S vývojem unifikovaného UML jazyka začali Grady Booch a Jim Rumbaugh ve společnosti Rational Software corporation, kteří začali optimalizovat Booch a OMT (Object Modeling Technique) metody. Na podzim roku 1995 se přidal Ivar Jacobson se svou firmou Objectory a vytvořili OOSE (Object-Oriented Software Engineering) metody. [11]

Motivace k vytvoření unifikovaného jazyka byly pro zmíněné autory tři následující. První byla nutnost sjednocení navzájem různě se vyvíjející metody OMT a OOSE. Dávalo tedy smysl nezávislé vývoje sjednotit a eliminovat případné rozdíly, které by mohly uživatele zmást.

Druhým důvodem bylo přinesení stability do objektového modelování z pohledu sémantiky a notace, čímž by umožnili modelovacím programům využití jedné společné rodiny jazyků a tím pádem se vývojáři mohli zaměřit na rozvoj více použitelných funkcionalit, než na jazyk samotný. Posledním důvodem byl fakt, že společná spolupráce jim umožnila převzít získané zkušenosti od ostatních a díky tomu vylepšit doposud vytvořené metody. [12]

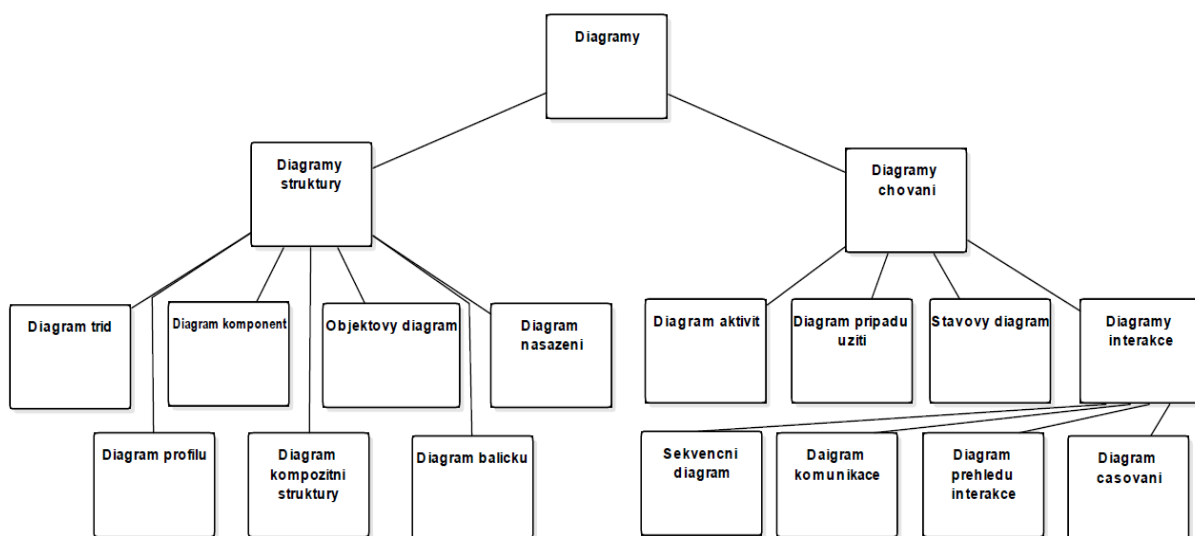
Společné úsilí se blížilo ke konci v říjnu roku 1996 ve verzi UML 0.91, kdy autoři dostávali připomínky od širší komunity, které byly postupně zapracovávány, i když bylo jasné, že bude třeba další přepracování. V lednu roku 1997 společnost IBM společně s ObjectTime,

Platinum Technology, Ptech, Taskon, Reich Technologies a Softeam zaslaly žádosti do OMG (Object Management Group) ohledně přijetí standardů a na základě těchto podkladů byla na podzim roku 1997 vytvořena standardizace UML 1.1. [13]

UML prochází neustálým vývojem a od roku 2004 se využívá po zásadních vylepšeních v oblasti konzistence, diagramů a zpětné kompatibility, ve verzích navazujících na „UML 2.0“. Poslední používanou verzí je UML 2.5, které organizace OMG schválila v roce 2015 a je volně přístupné na internetových stránkách. [14]

Modelovací jazyk je unifikovaný skrze historické metody a notace, životní cyklus vývoje, programovací jazyky a platformy, vývojové procesy i rozličné vnitřní koncepce. Díky tomu jej adaptovali veškeré vývojové nástroje, které využívají počítačovou podporu (CASE)².

UML modely jsou tvořeny ve verzi 2.0 z různých diagramů, obsahující objekty a vzájemné vazby. Existují různé typy diagramů, které jsou určeny k popisu struktury, či chování daného systému s jistou mírou abstrakce a zobecnění. Dělí se na dvě základní skupiny – diagramy struktur a diagramy chování (behaviorální). V rámci chování existuje ještě vlastní skupina diagramů interakce. Přehledné rozdělení diagramů je na Obr. 1.



Obr. 1: Přehled UML diagramů

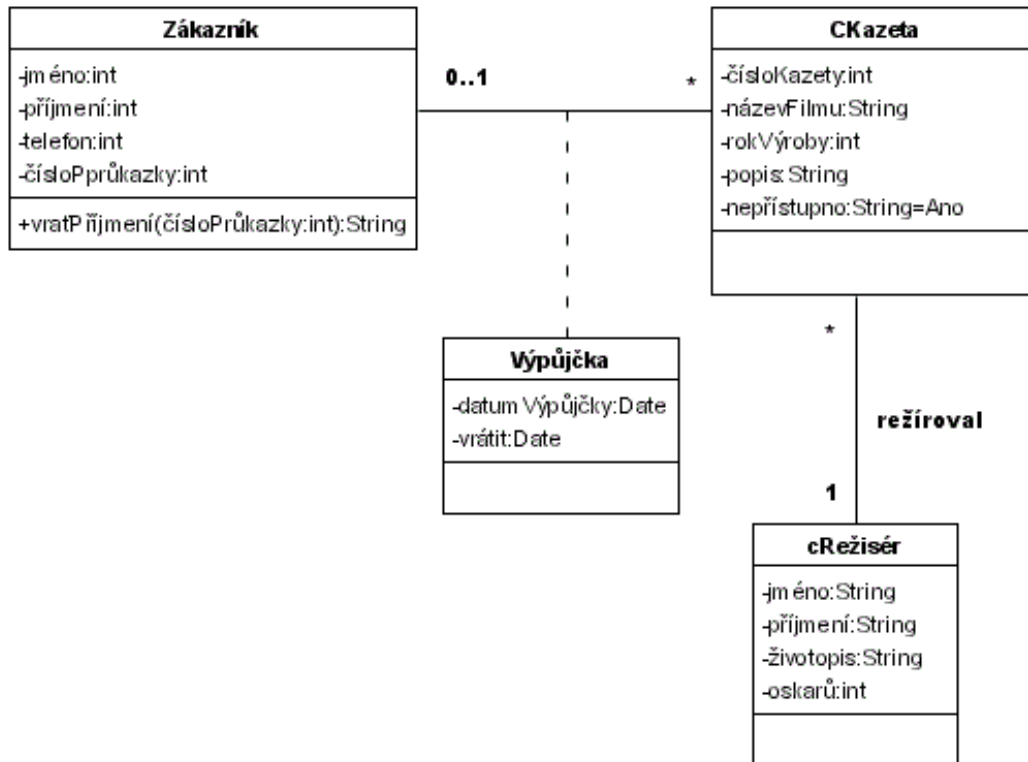
Diagramy chování zobrazují procesy uvnitř systému. Tento typ diagramů bude použit níže v praktické části práce pro popis jednotlivých procesů vykonávaných v rámci virtualizační infrastruktury. Pro lepší pochopení je níže uveden popis vybrané množiny výše uvedených diagramů.

² Computer Aided Systems Engineering

2.3.1 Diagramy struktur

2.3.1.1 Diagram Tříd

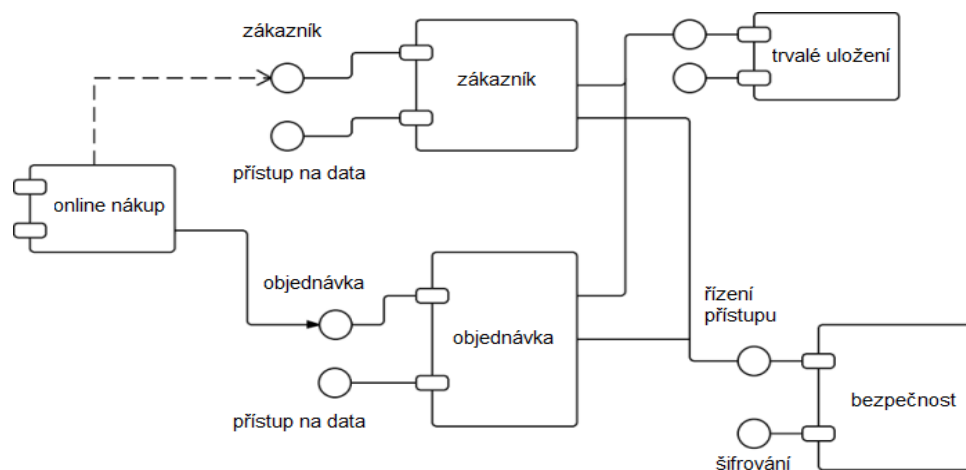
Obsahuje popis statických struktur implementovaného systému. Jsou zde zobrazeny a detailně popsány všechny objekty včetně vzájemných relací.



Obr. 2: Příklad diagramu tříd [15]

2.3.1.2 Diagram komponent

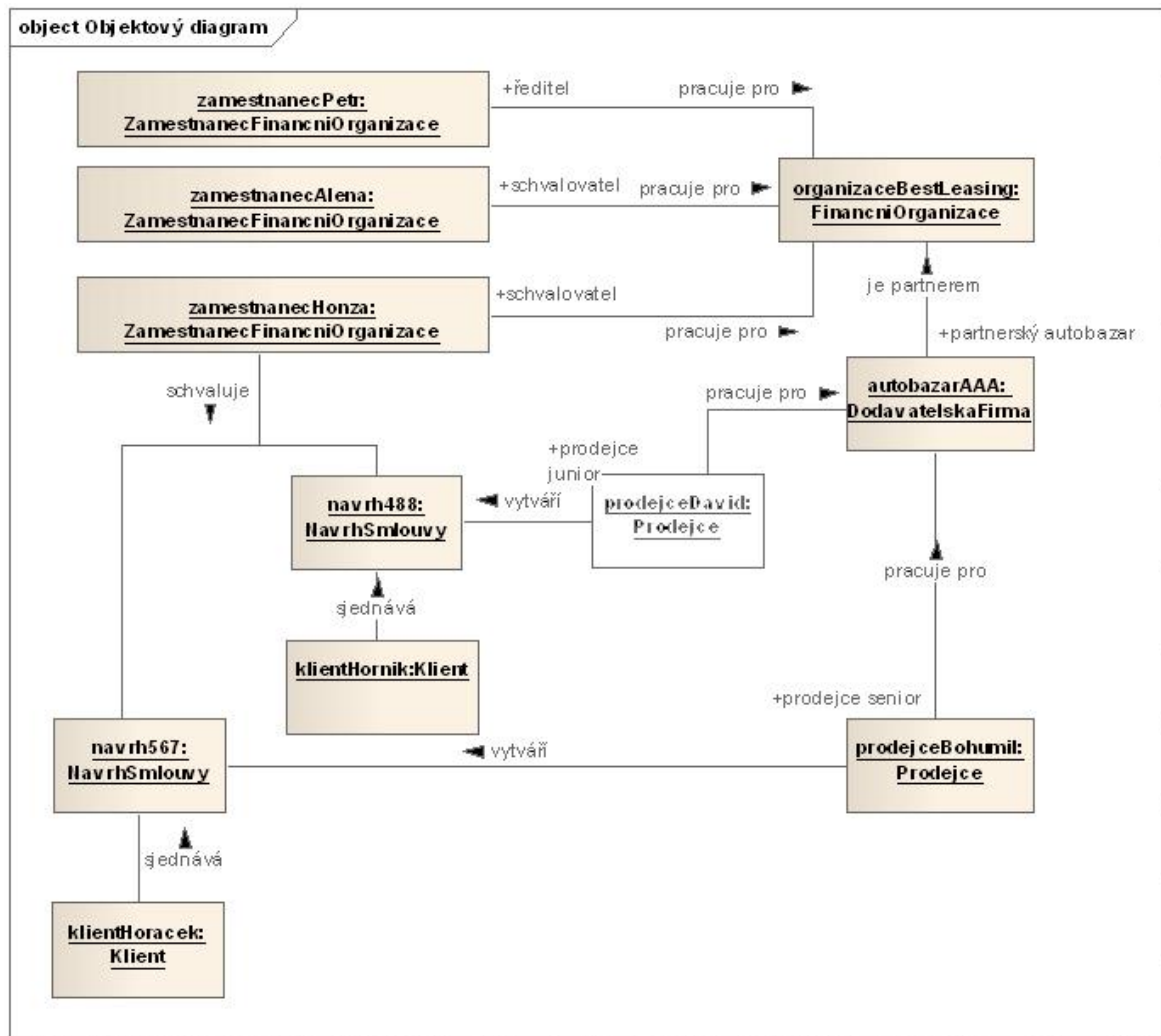
Abstraktnější popis systému pomocí modulárních částí – komponent, včetně vzájemného napojení.



Obr. 3: Příklad diagramu komponent

2.3.1.3 Objektový diagram

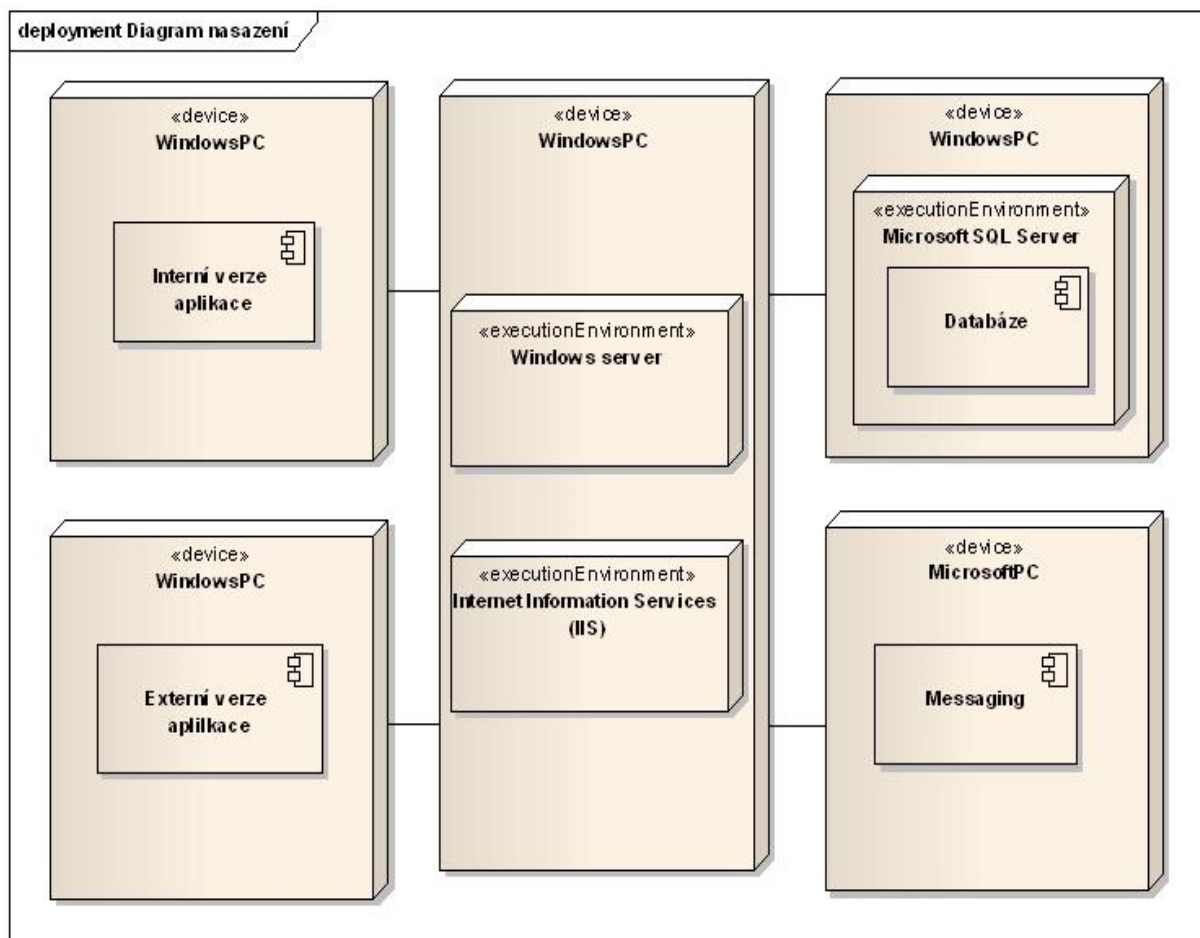
Je velice podobný diagramu tříd, ale popisuje navíc vztahy jednotlivých instancí objektů v různých stavech systému.



Obr. 4: Příklad objektového diagramu [16]

2.3.1.4 Diagram nasazení

Používá se v případě zavedení systému a hlavním specifíkem je výskyt tzv. uzlů.



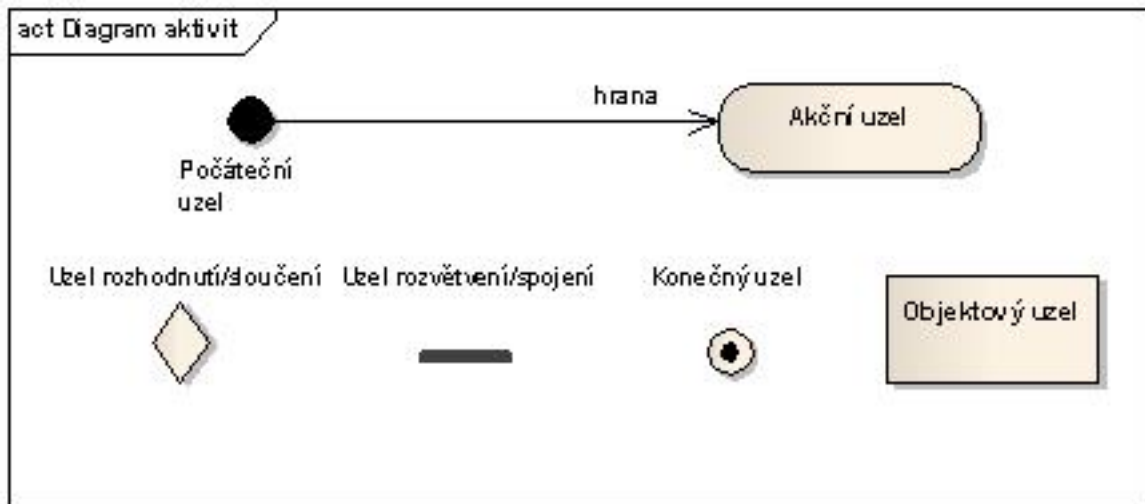
Obr. 5: Příklad diagramu nasazení [16]

Diagramy struktur jsou používány ve více technickém popisu a nemají pro tuto práci další využití, proto jsou zmíněny velice stroze a níže jsou uvedeny důležitější diagramy chování.

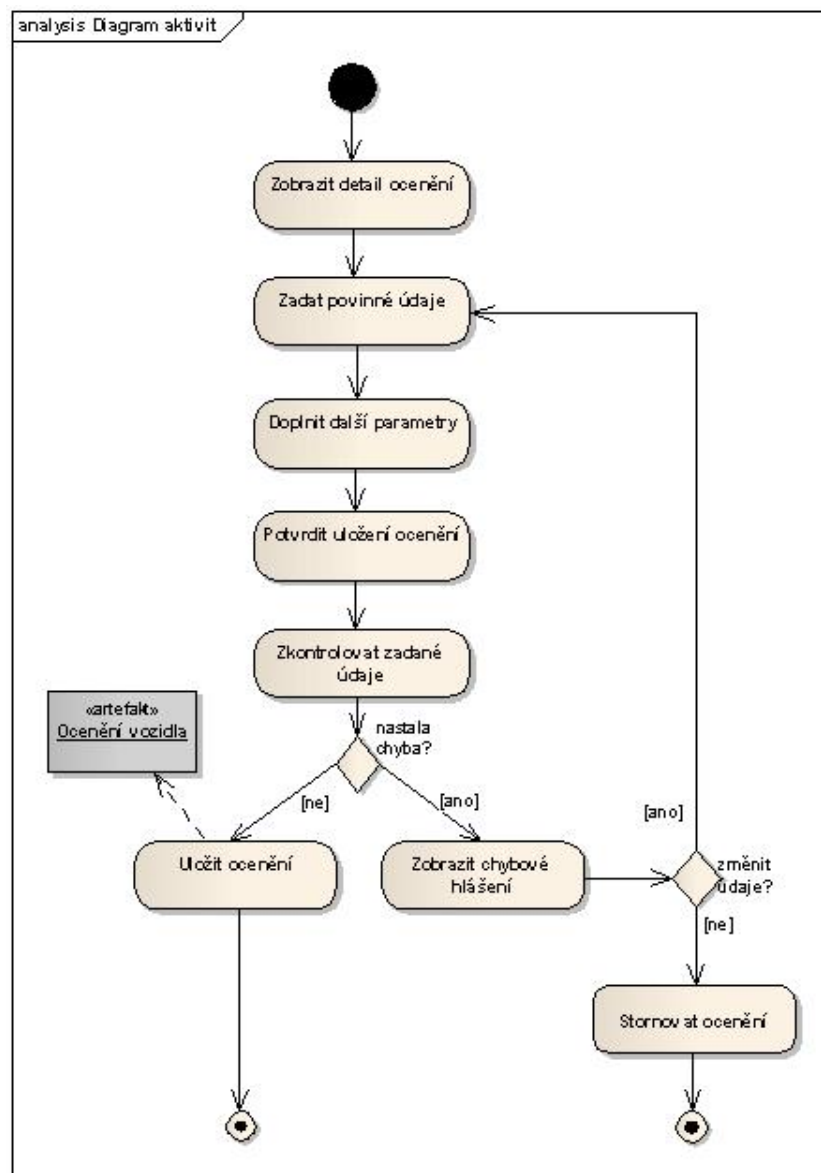
2.3.2 Diagramy chování

2.3.2.1 Diagram aktivit

Popisuje všechny interakce aktivit a pracovní postupy v systému za pomoci různých prvků. Modeluje procesy jako aktivity složené z uzlů propojených hranami, značící jednotlivé přechody mezi nimi. Pro lepší pochopení je přiložen Obr. 6 obsahující možné prvky. Aktivity mohou obsahovat další aktivity, či akce. Na Obr. 7 je následně zobrazen příklad diagramu aktivit.



Obr. 6: Popis prvků použitých v diagramu aktivit [16]

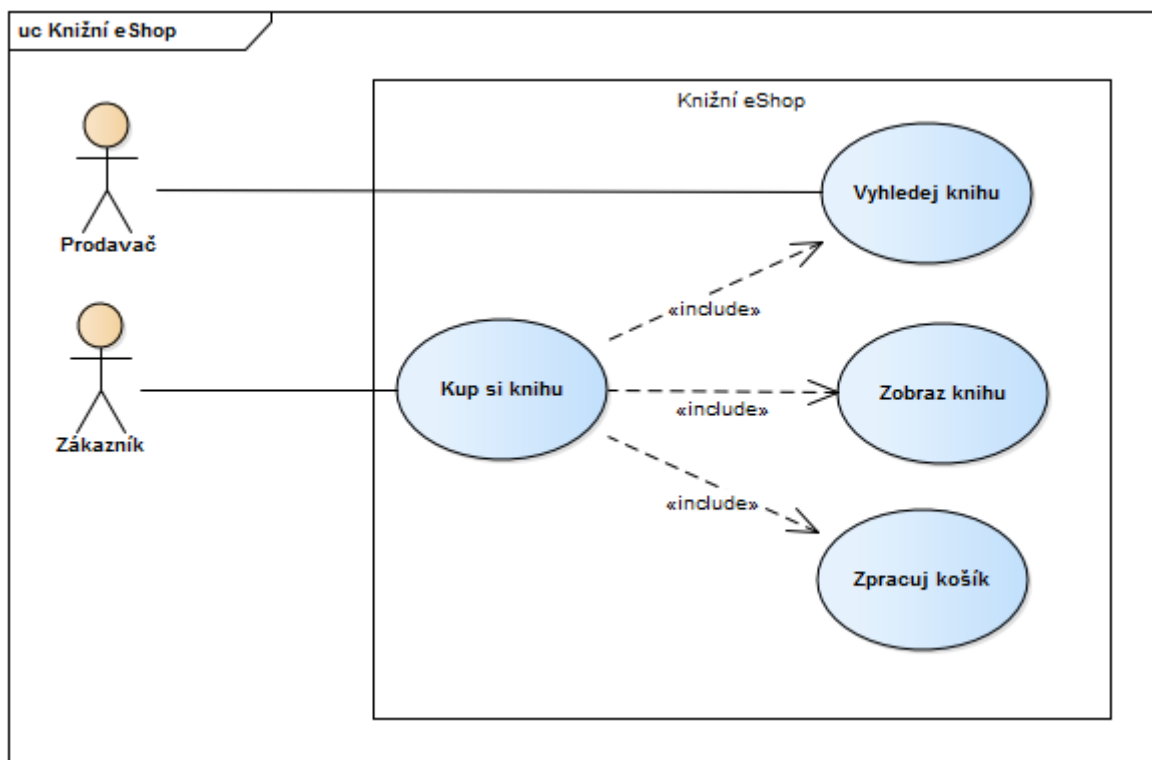


Obr. 7: Příklad diagramu aktivit [16]

Mohlo by se zdát, že je možné diagram aktivit použít na popis jednotlivých business procesů. Některé jednotlivé aktivity a jejich postup je opravdu možné za pomoci tohoto diagramu popsat, avšak rozšířenou zapisovací množinou je BPMN (Business Process Model Notation). Jedná se přímo navržený soubor určený k modelování podnikových procesů blíže popsany v kapitole 2.4.

2.3.2.2 Diagram případů užití

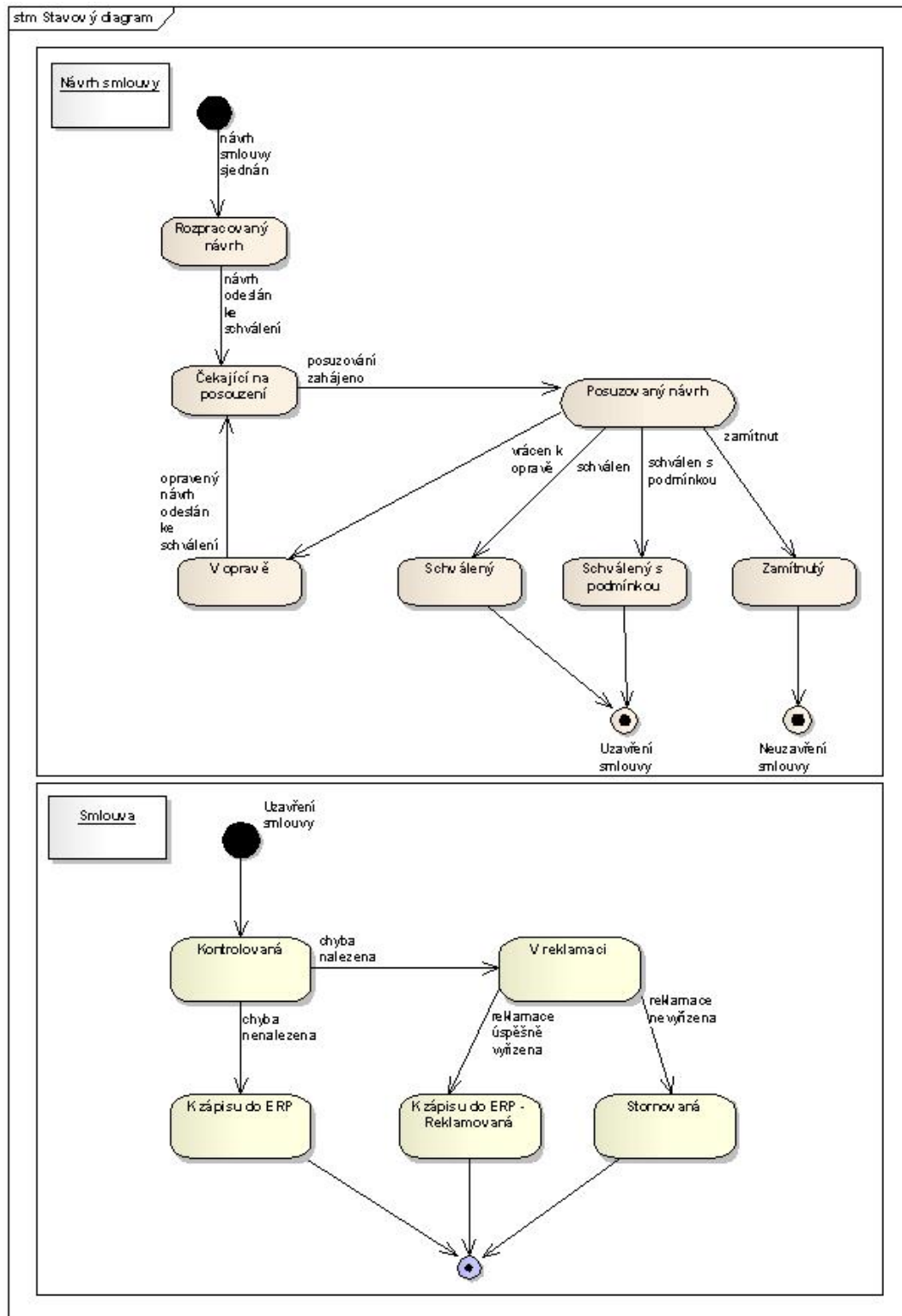
Hlavní vlastností tohoto diagramu je zachycení popisu chování systému z hlediska uživatele. Jsou zde zobrazeni konkrétní aktéři procesů a zároveň činnosti, které mohou v daném systému vykonávat. V tomto diagramu jsou také znázorněny funkční požadavky na systém. Jednotlivé případy užití mohou být přiřazeny více aktérům a je možné jejich propojení. Veškeré případy jsou zakresleny dovnitř hranic systému. Příklad takového diagramu je zobrazen na Obr. 8.



Obr. 8: Příklad diagramu případů užití

2.3.2.3 Stavový diagram

Ve formě konečného stavového automatu graficky popisuje systém včetně všech vzájemných přechodů. Jeho použití je především pro popis chování objektu pro různé případy užití. Může vypadat podobně jako na Obr. 9.

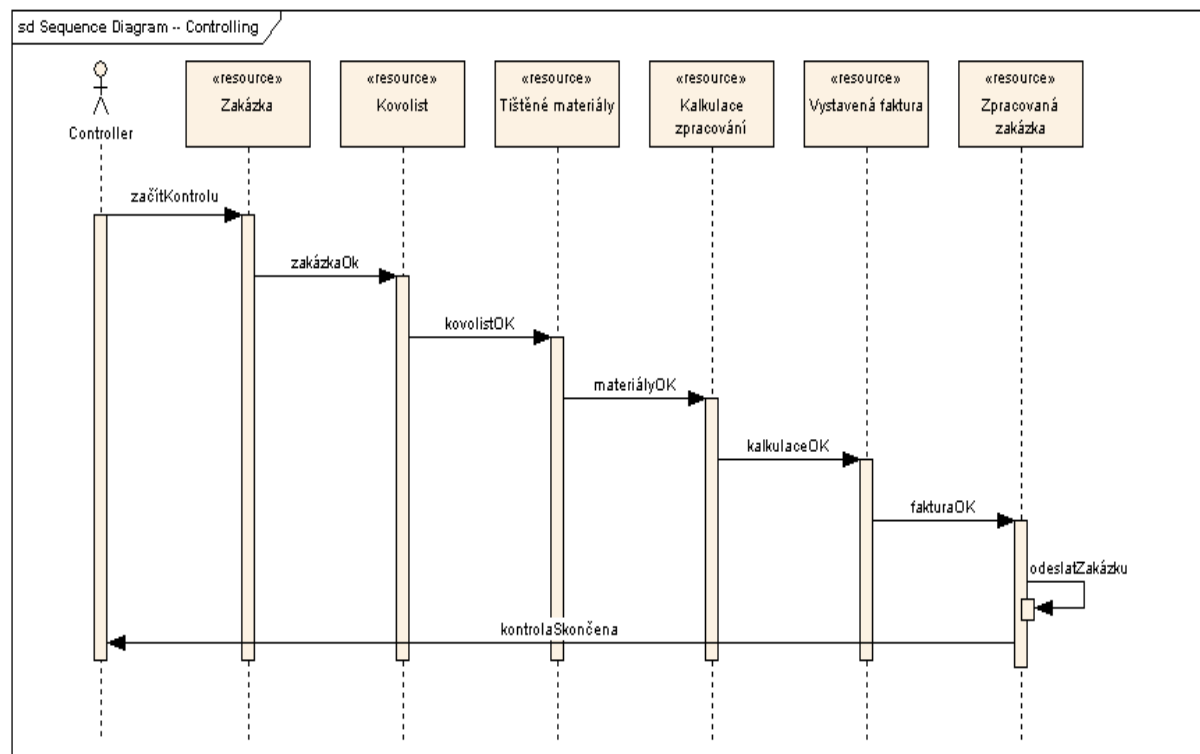


Obr. 9: Příklad stavového diagramu [16]

2.3.3 Diagramy interakce

2.3.3.1 Sekvenční diagram

Velmi často používaný diagram, který slouží ke grafickému zachycení průběhu zpracování aktivity v systému. Je z něj patrná časová souslednost aktivit jednotlivých objektů. Pro lepší představu je přiložen Obr. 10.



Obr. 10: Příklad sekvenčního diagramu [17]

2.4 BPMN

Business Process Model and Notation je soubor grafických objektů a pravidel, které slouží k modelování podnikových procesů za účelem porozumění technickou i netechnickou skupinou uživatelů. Mezi výhody BPMN patří vytvoření za účelem standardizace a nikoliv zisku, díky čemuž je standard volně použitelný. Umožňuje definovat a porozumět jednotlivým podnikovým procesům skrz zobrazení v grafických diagramech. Překonává rozdíly v porozumění problematice z různých typů pohledů, ačkoliv je relativně snadný na naučení, a zároveň dostatečně mocný k zobrazení všech možných podnikových procesů. Posledním specifickým je nezávislost na pouze jednom konkrétním výrobcí modelovacího software. [18]

Vývoj standardu je výsledkem činnosti Business Process Management Initiative (BPMI), která po více než dvouletém vývoji v květnu roku 2004 vydala BPMN ve verzi 1.0. Cílem vývoje bylo vytvoření konvence, která by byla snadno pochopitelná softwarovými

konzultanty, kteří provádí prvotní návrh procesů. Zároveň bylo nutné konvenci vytvořit tak, aby byla pochopitelná i softwarovými vývojáři, kteří business procesy musí implementovat v daném programovacím jazyku. Poslední skupinou osob, kteří museli také jazyku rozumět, byli manažeři za účelem monitorování konkrétních procesů. [19]

Vzhledem k podobným záměrům se BPMI sloučila s organizací OMG a společně v roce 2007 vydali standard BPMN 1.1 a o rok později BPMN 1.2. Vývoj pokračoval až ke standardu 2.0, který byl vydán v roce 2011. [20]

Postupným vývojem se funkcionalita BPMN rozšiřovala a umožnila transformaci do jednotného univerzálního jazyku, použitelného pro modelování podnikových procesů. Přispěla k tomu i adaptace Business Process Definition Metamodel (BPDM), který určuje sémantiku modelování jednotlivých procesů a jejich vzájemnou kompatibilitu mezi různými modelovacími nástroji. V praxi toto rozšíření obsahuje kromě grafické reprezentace i pseudokód, který umožňuje automatizovaně danému modelu porozumět. [21]

BPMN notace se skládá z různých elementů a pravidel použitých při modelování. Konkrétně se jedná o 4 typy objektů:

- Tokové objekty (flow objects)
- Spojovací objekty (connection objects)
- Plavecké dráhy (swim lanes)
- Artefakty (artefacts)

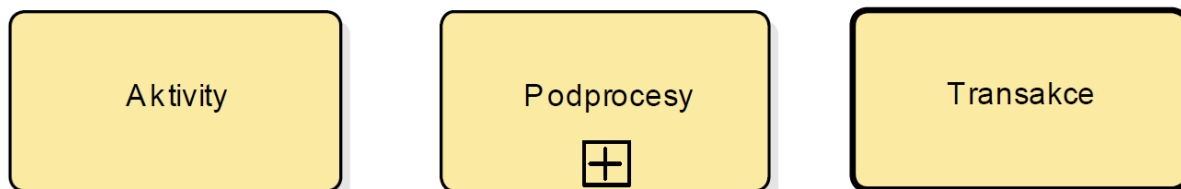
Tokové objekty jsou vyplněné zbylými třemi typy a obsahují konkrétní popisy jednotlivých procesů, včetně možných rozhodnutí (bran = gateways), očekávaných událostí (events) i vykonaných aktivit (aktivity). Níže jsou všechny objekty detailněji popsány.

2.4.1 Aktivita

Nejdůležitějším typem tokových objektů jsou aktivity. Popisují činnosti vykonávané v průběhu procesu. Jedná se například o vytvoření záznamu v seznamu, přijetí žádosti, zalití květin, či libovolný jinak definovaný úkon, který je dále nedělitelný a svým popisem jasně identifikuje veškeré činnosti, které se v rámci ní vykonávají.

Aktivity mohou obsahovat souhrn dalších činností, a v takovém případě se jedná o podprocesy. Ke grafickému značení je použito matematického znaku plus. V případě existence podprocesů v diagramu jejich obsah modelujeme v rámci vlastního diagramu aktivit.

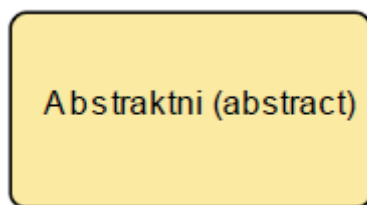
Speciálním typem podprocesu je transakce. Jedná se o soubor událostí, který musí být vždy vykonán celý, nebo je třeba zajistit, aby se nevykonala žádná určená akce. Je tedy definován jasný postup kroků, které je třeba vykonat, aby se proces dostal do výchozího stavu v případě, že se objeví nějaká nečekaná událost. Grafická reprezentace jednotlivých typů je na Obr. 11.



Obr. 11: Vzory grafického znázornění jednotlivých aktivit

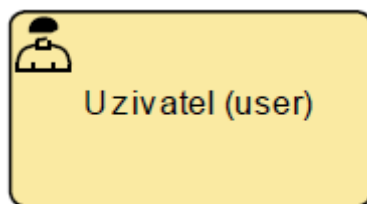
Pro přesnější popis existují různé druhy aktivit, které se využívají v rozličných případech. Toto upřesňující rozlišení detailněji popisuje daný firemní proces, jelikož grafickým znázorněním jasně definuje konkrétní typ úkolu. Jedná se o následující množinu:

- Abstraktní (abstract, Obr. 12) – Zobrazení interakce mezi aktivitou procesu a jiným procesem, či jiným účastníkem procesu. Jsou to jediné aktivity, které mohou interagovat mimo daný proces s okolním prostředím.



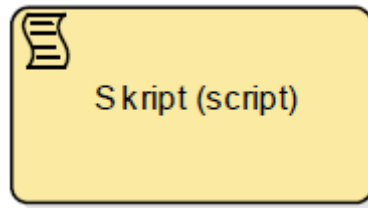
Obr. 12: Grafické znázornění abstraktní aktivity

- Uživatel (user, Obr. 13) – Akce, kterou vykonává daný uživatel, či jiná žijící osoba.



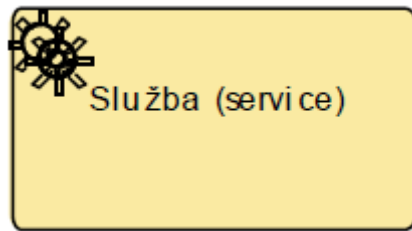
Obr. 13: Grafické znázornění uživatelské aktivity

- Skript (script, Obr. 14) – Část procesu vykonávaná naprogramovaným skriptem.



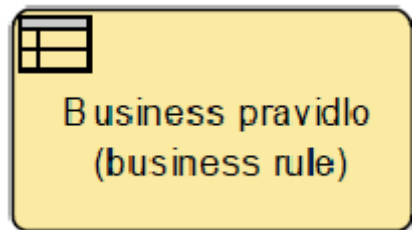
Obr. 14: Grafické znázornění skriptu

- Služba (service, Obr. 15) – Aktivita je stejně jako skript naprogramována, ale provádí se automaticky.



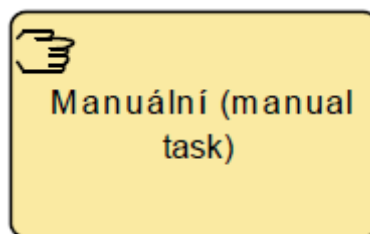
Obr. 15: Grafické znázornění služby

- Business pravidlo (business rule, Obr. 16) – Poskytuje mechanismus pro získání vstupu, který bude nadále využíván v rámci daného procesu. Jedná se například o výpočet hodnot.



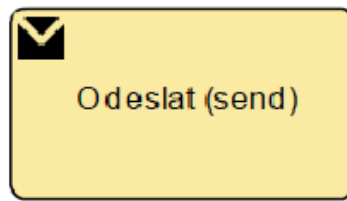
Obr. 16: Grafické znázornění business pravidla

- Manuální (manual, Obr. 17) – Aktivita je vykonávána bez jakéhokoliv podnikového procesu, nástroje, či programu.



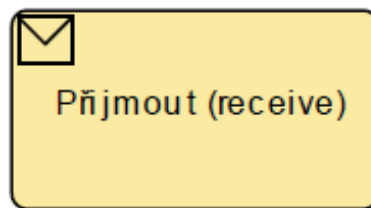
Obr. 17: Grafické znázornění manuální aktivity

- Odeslat (send, Obr. 18) – Aktivita odesílá nějakou zprávu, například dopis, v rámci procesu.



Obr. 18: Grafické znázornění aktivity odeslat

- Přijmout (receive, Obr. 19) – Opak aktivity odeslat, ve které se očekává příjem zprávy. Například zaslání požadovaných informací.



Obr. 19: Grafické znázornění aktivity přijmout

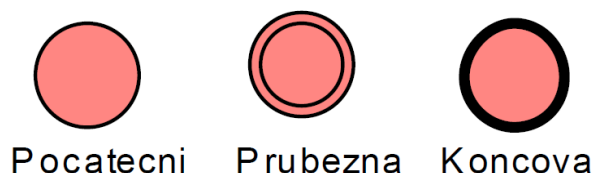
Kromě rozdělení aktivit na různé typy, je možné jednotlivé úkoly opakovat. Opakování lze provádět paralelně, nebo sériově. V případě paralelního procesu se vykonávají všechny instance v ten stejný čas. Pro sériové úlohy se čeká na dokončení jedné instance a až následně dochází ke spuštění další instance. Pro opakování instancí je nutno stanovit ohraničující podmínku, která zamezí nekonečnému opakování. Grafické zobrazení všech třech typů aktivit je na Obr. 20.



Obr. 20: Grafické znázornění většího počtu instancí

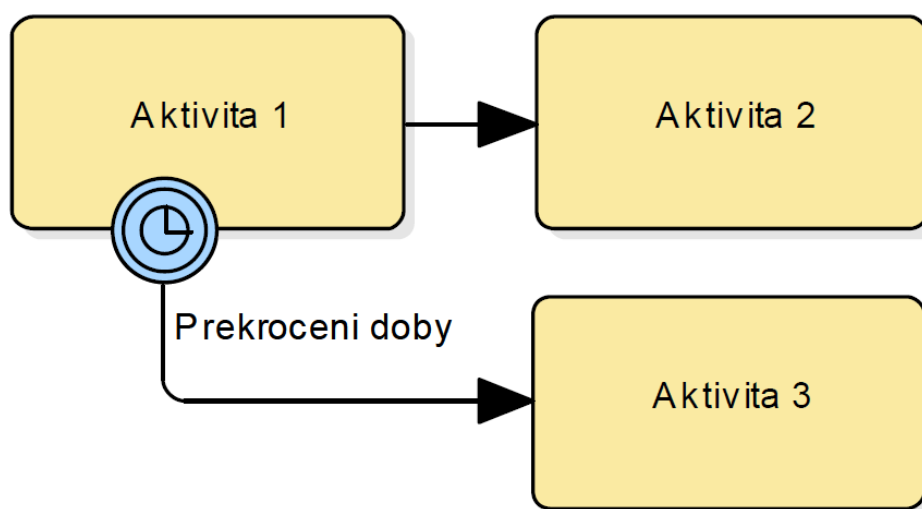
2.4.2 Události

Druhou množinou jsou události. Mohou být počáteční, průběžné, či koncové v závislosti na tom v jaké části procesu se vyskytují. Grafická reprezentace je ve formě kruhu s různým ohraničením, jak je uvedeno na Obr. 21.



Obr. 21: Grafické příklady různých událostí

Počáteční, ani koncové aktivity nepotřebují zvláštní popis, ale u průběžných je nutné zmínit dělení na dva druhy. Události typu catch a throw. Český ekvivalent se u zmíněných termínů neuvádí. Catch očekávají přijetí nějaké jiné události, která způsobí spuštění dané aktivity. Throw události naopak tyto spouštěcí události vytváří. Spouštěcí událostí může být například obdržení požadavku. Pokud je v grafické interpretaci průběžná událost připojena k aktivitě, označuje podmínku, při jejímž splnění se vykonávání dané činnosti ukončí. Příklad takového předběžného ukončení je zobrazen na Obr. 22.



Obr. 22: Příklad události připojené na aktivitu

Všechny tři události mohou být různých typů a níže je uveden jejich přehled:

- Nespecifikovaný (none, Obr. 23) – Blíže neurčená událost, která obsahuje doprovodný text k lepšímu určení typu.



Nespecifikovaný (none)

Obr. 23: Grafické zobrazení nespecifikované události

- Zrušení (cancel, Obr. 24) – Typ události, kterou se ukončuje proces.



Zruseni (cancel)

Obr. 24: Grafické zobrazení zrušení události

- Kompenzace (compensation, Obr. 25) – Využívá se v transakcích, pokud je třeba některé provedené aktivity vrátit zpět.



Kompenzace (compensation)

Obr. 25: Grafické zobrazení kompenzační události

- Podmínka (condition, Obr. 26) – Událost nastane pouze v případě splnění konkrétní podmínky.



Podmínka (condition)

Obr. 26: Grafické zobrazení události podmínky

- Eskalace (escalation, Obr. 27) – Je určena pro použití v podprocesu, pokud je potřeba předání toku do jiné dráhy.



Eskalace (escalation)

Obr. 27: Grafické zobrazení události eskalace

- Chyba (error, Obr. 28) – Jak je z názvu patrné, událost nastává v případě chyby.



Chyba (error)

Obr. 28: Grafické zobrazení chyby

- Link (link, Obr. 29) – Událost využívaná v případě propojování procesů.



Link (link)

Obr. 29: Grafické zobrazení linku

- Zpráva (message, Obr. 30) – Jedná se o příchozí či odchozí zprávu.



Zpráva (message)

Obr. 30: Grafické zobrazení události zprávy

- Násobná (multiple, Obr. 31) – Spojení více druhů událostí do jednoho symbolu.



Násobná (multiple)

Obr. 31: Grafické zobrazení násobné události

- Čas (timer, Obr. 32) – Událost nastává v určitém čase, nebo po uplynutí stanoveného času.



Čas (timer)

Obr. 32: Grafické zobrazení události času

- Signál (signal, Obr. 33) – Význam v použití při předávání informací v rámci procesů, nebo jejich částí.



Signál (signal)

Obr. 33: Grafické zobrazení události signálu

- Ukončení (terminate, Obr. 34) – Událost, která okamžitě ukončí všechny části procesu.



Ukončení (terminate)

Obr. 34: Grafické zobrazení události ukončení

2.4.3 Brány

Dalším prvkem, který určuje tok aktivit, jsou brány. Značí rozhodovací logiku a určují směr procesu, který jimi může být dělen, ale i spojován. V případě spojení je nutné splnit všechny vstupující kroky pro pokračování procesu. Existuje pět typů bran:

- Exkluzivní (exklusive, Obr. 35) – Ekvivalent logické spojky XOR v rozhodovacím procesu. V případě splnění podmínky jde tok alternativní cestou. Pokud není podmínka splněna, jde tok výchozím směrem.



Exkluzivní (exclusive)

Obr. 35: Grafické zobrazení exkluzivní brány

- Komplexní (complex, Obr. 36) – Určuje složitější podmínku a jednotlivé cesty jsou popsány.



Komplexní (complex)

Obr. 36: Grafické zobrazení komplexní brány

- Inkluzivní (inkluzive, Obr. 37) – Brána reprezentuje logickou spojku OR.



Inkluzivní (inkluzive)

Obr. 37: Grafické zobrazení inkluzivní brány

- Paralelní (paralel, Obr. 38) – Značí logickou spojku AND a tok je nadále procesován všemi výstupními větvemi.



Paralelní (paralel)

Obr. 38: Grafické zobrazení paralelní brány

- Podmíněné události (Event-based, Obr. 39) – Proces pokračuje tokem na základě průběžné události, která nastala.



Podmíněné události (Event-exclusive)

Obr. 39: Grafické zobrazení brány podmíněné události

2.4.4 Spojovací objekty

K propojování jednotlivých objektů slouží spojovací objekty (connecting objects), které jsou tři následující:

- Sekvenční tok (sequence flow) – Plná černá čára zakončená plnou šipkou značící spojení jednotlivých tokových objektů. Směr odpovídá směru toku procesu. V případě

výchozího toku se čára označuje lomítkem. Slouží ke spojování tokových objektů v rámci jednoho kontextu a díky tomu nesmí tok opustit vymezené hranice procesu. Při komunikaci s jiným kontextem se využívá toku zpráv.

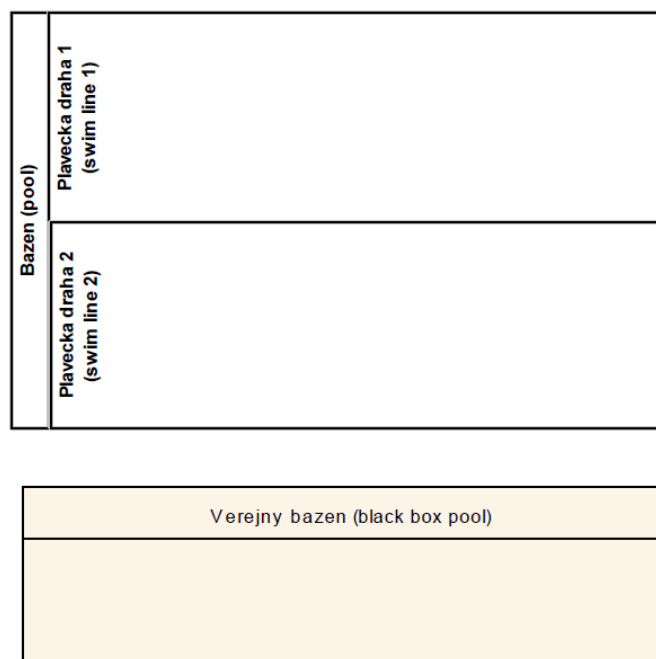
- Tok zpráv (message flow) – Použitelný v případě nutnosti komunikace mezi kontexty definovanými jednotlivými bazény (pools). Ke grafickému značení se využívá přerušovaná čára začínající kruhem ukončená nevyplněnou šipkou.
- Asociace (association) – Zobrazeno za pomoci přerušované čáry bez jakýchkoliv zakončení, slouží k propojování objektů za účelem poskytnutí dodatečných informací.

2.4.5 Bazény s plavecké dráhy

Pro rozdělení jednotlivých aktivit do logických celků (kontextů) se využívá bazénů (pools), které mohou být ještě dále děleny na plavecké dráhy (swim lines). Komunikace mezi dráhami není nijak omezena, ale mezi bazény se mohou pouze zasílat zprávy.

Plavecké dráhy, kterých může být více v jednom bazénu, blíže reflektují vnitřní strukturu daného kontextu. Každá dráha může být například oddělení v podniku, nebo se může jednat o přiřazení drah jednotlivým účastníkům procesu.

Zvláštním typem bazénu je veřejný bazén (black box), jehož aktivity, které jsou prováděny v rámci vykonávání, nejsou známy. Je používán v případě vykonávání procesu, třetí stranou. Zobrazení všech zmíněných termínů je na Obr. 40.



Obr. 40: Bazény a plavecke drahy

2.4.6 Artefakty

Poslední částí důležitou v modelování se v rámci BPMN využívají artefakty, které mohou být ve formě textových anotací, vstupně výstupních dat či textových zpráv.

Závěrem je vhodné zmínit, že pomocí BPMN je možné modelovat podnikové procesy na různých úrovních dle aktuální potřeby detailnosti. Je možné zobrazovat interakce a komunikace mezi jednotlivými odděleními, nebo například mezi konkrétními zaměstnanci, či zaměstnancem a daným systémem. V rámci celé práce byl pro zobrazení procesů použit asi nejrozšířenější modelovací program Enterprise Architect [22].

2.5 Hodnocení investic

Pro stanovení rozhodnutí, zda danou optimalizaci uskutečnit, je nutné provedení ekonomického zhodnocení. Díky tomuto rozhodnutí lze optimalizaci uchopit jako investici a za pomoci níže uvedených třech ekonomických ukazatelů určit, zda bude výhodné danou investici uskutečnit, či nikoliv.

Optimalizace bude probíhat s vynaložením zdrojů na začátku investice za účelem získání vyšších zisků v následujících letech. Důležitou roli tedy bude hrát v investici čas. Z ekonomického pohledu je možné použít metody statické, či dynamické.

2.5.1 Statické metody

Statické metody nezohledňují riziko, ale jejich výhodou je okamžité a rychlé určení celkových čistých příjmů při zavádění investice. Celkový čistý příjem u statických metod lze spočítat jako součet čistých příjmů jednotlivých let s odečtením prvotní investice. Vzorec pro výpočet je následující:

$$CP = -IN + CF_1 + CF_2 + \dots + CF_n = \sum_{i=1}^n CF_n$$

CP ... čistý příjem

IN ... investice

CF_n ... cash flow v daném roce

Pokud je čistý příjem větší než nula, tak se doporučuje investici přijmout. V případě záporné hodnoty se přijetí investice nedoporučuje. Pro získání reálného pohledu je však nutné započítat i riziko investice a zohlednit časovou hodnotu peněz, k čemuž se používají dynamické metody.

2.5.2 Dynamické metody

2.5.2.1 Čistá současná hodnota

Jedná se o metodu podobnou uvedené statické, avšak zohledňuje časovou hodnotu peněz. Čistá současná hodnota (NPV³) počítá s faktorem času a diskontuje čisté zisky v jednotlivých letech životnosti o diskontní míru. Výpočet je opět zobrazen níže:

$$NPV = -IN + \frac{CF_1}{(1+r)^1} + \frac{CF_2}{(1+r)^2} + \dots + \frac{CF_i}{(1+r)^i} =$$

$$NPV = -IN + \sum_{i=1}^n \frac{CF_i}{(1+r)^i}$$

IN ... investice

CF_n ... cash flow v daném roce

r ... diskontní míra

Diskontní míra vyjadřuje riziko investice či žádanou výnosnost. Může být určena pomocí vážených nákladů na kapitál. V rámci této práce však nebude možné z důvodu nevyužívání cizího kapitálu tuto metodu použít, a bude stanovení diskontu provedeno čistě subjektivně.

2.5.2.2 Vnitřní výnosové procento

Poslední metodou, která bude použita ke zhodnocení investice, je vnitřní výnosové procento (IRR⁴). Ekonomický ukazatel má základ v NPV, ale dohledává se taková diskontní míra, při které je současná hodnota investice rovna nule.

$$-IN + \sum_{i=1}^n \frac{CF_i}{(1+IRR)^i} = 0$$

Investice je přijata v případě, že IRR bude větší než stanovená diskontní míra. Větší IRR značí větší výnos z investice a tím pádem je investice žádanější. Tuto metodu lze použít pouze v případě, že jsou všechny peněžní toky konvenční, což znamená, že v průběhu celé investice se znaménko výnosů změní právě jednou. [23]

³ Net Present Value

⁴ Internal Rate of Return

2.6 Technický popis frameworku

V rámci implementace budou použity programovací jazyky představené v této kapitole. Jedná se o jazyk PHP (Personal Home Page), který společně s HTML (HyperText Markup Language) zajistí grafické rozhraní aplikace a pro implementaci veškerých automatizovaných procesů a skriptovací logiky bude použit PythonTM. Veškerá vyčítaná data budou uložena v databázi MySQL.

2.6.1 PHP

PHP je následníkem produktu PHP/FI a vytvořil jej v roce 1994 Rasmus Lerdorf. Prvotní účel byl pro sledování návštěvnosti přístupů jeho životopisu. Za tím účelem vzniklo označení „Personal Home Page tools“, který přetrvalo ve zkratce do dnes avšak s významem Hypertext Preprocessor. Postupem času byly skripty přepisovány a vývojem se v roce 2004 dosáhlo verze PHP 5, které je celosvětově používáno. [24]

Jedná se o skriptovací programovací jazyk, který na straně serveru vytváří stránku poté, co o ni uživatel požádal. Dokáže dynamicky generovat obsah stránky a není tedy nutné se omezovat jen na statický obsah a je možné informace měnit dle aktuální situace. Syntaxe je inspirována jazyky Java, Perl, C a Pascal. [25] Výhodou PHP je platformní nezávislost a v současné době je použit na 83% [26] všech internetových stránek.

2.6.2 HTML

HyperText Markup Language je značkovací jazyk, který slouží k tvorbě webových stránek. Vznikl rozšířením dříve používaného značkovacího jazyka Standard Generalized Markup Language (SGML). Na vývoji spolupracovali Tim Berners-Lee a Robert Cailliau v roce 1989. [27]

V aktuální verzi 5.0 se jazyk používá od roku 2014 a umožňuje nikoliv pouhé změny vlastností textu, ale přidává zásadní podporu multimédií a pro aplikace, které fungují i bez připojení k internetu.

2.6.3 Python

Python je poslední použitý vysokoúrovňový skriptovací programovací jazyk v rámci práce. Historie začíná v roce 1991, kdy Guido van Rossum vytvořil první verzi. Jazyk umožňuje používání objektů, procedur i funkcí. O jeho robustnosti svědčí použití například v nastavovacích skriptech linuxových operačních systémů. [28] V rámci frameworku zabezpečuje kromě rozhodovací logiky i napojení na virtuální farmu pomocí API. [29]

3 Praktická část

3.1 O firmě

Firma Gemalto N. V. vznikla v roce 2006 spojením firem Gemplus a Axalto. Hlavní sídlo je v Amsterdamu v čele s francouzským ředitelem společnosti Philippe Vallée. V roce 2017 se počet zaměstnanců společnosti pohyboval okolo 15 000 a příjmy v roce 2016 činily 3.1 miliard euro [30]. Společnost se je lídrem v oblasti digitální bezpečnosti, což dokládá i její slogan „security to be free“ viz Obr. 41. Postavení předního dodavatele v bezpečnostních řešeních utvrdilo i oznámení o spojení s firmou Thalles na konci roku 2017 [31]. Obě zmíněné korporace nabízí služby v různých oblastech digitální komunikace a jsou dlouholetými partnery.



Obr. 41: Logo firmy Gemalto [32]

Vyráběnými produkty jsou čipy do kreditních karet, sim karet, pasů, občanských průkazů, ale také programy, které s těmito bezpečnostními prvky komunikují. Společnost mimo jiné vyvíjí software pro komunikaci se svými produkty a je tedy možno například navázat zašifrovanou komunikaci se všemi sim kartami, které operátor vydal svým zákazníkům a oni je používají po celém světě bez nutné interakce člověka.

Dále je možné číst data z občanských průkazů vybavených biometrickým čipem, a zároveň tato data i okamžitě ověřovat s osobou, která se daným průkazem prokazuje. V neposlední řadě společnost v roce 2016 úspěšně spustila pilotní program, díky němuž je možné pomocí bezdrátové technologie NFC⁵, kterou je vybavena většina nových mobilních telefonů, si „nahrát klíče“ od svého osobního automobilu do mobilního zařízení a odemknout i nastartovat svůj vůz pouhým přiložením telefonu. Veškerá komunikace je samozřejmě šifrována a zabezpečena pomocí hesel uložených v sim kartě mobilního telefonu.

Gemalto má pobočky po celém světě a v rámci této práce budou zkoumány procesy probíhající převážně v pražské pobočce, která zaměstnává okolo 300 pracovníků. Konkrétně se jedná o procesy spojené s provozem virtuální infrastruktury.

⁵ NFC = Near Field Communication = komunikace mezi dvěma zařízeními na krátkou vzdálenost (do 4 cm)

3.2 Cílová skupina

Jak bylo zmíněno výše, firma Gemalto kromě vývoje hardware provádí především vývoj software. Jelikož každá instalace softwaru je zcela unikátní, je důležité si funkčnost všech komponent a jejich vzájemné propojení vyzkoušet. V případě, že by každý zaměstnanec potřeboval nový server pro každý testovací systém, byl by nejen provoz ale i pořízení takového množství zařízení velmi nákladné.

Za tím účelem, byla postupným nákupem výkonných serverů vytvořena virtualizační platforma, ve které si každý zaměstnanec může zažádat o virtuální stroj, na kterém následně testuje danou instalaci. Na základě principu fungování se dá říci, že se jedná se o privátní cloudové datacentrum založené na virtualizační platformě VMware [33].

Pozn.: Problematika fungování virtualizace je mimo rozsah této diplomové práce, a pro následující text postačí vysvětlení, že se jedná o velmi výkonné fyzické servery, na kterých je nainstalováno a spuštěno současně více operačních systémů (VM⁶) tak, jak je zobrazeno na Obr. 42. Pro lepší pochopení problematiky virtualizace je vhodné navštívit například webové stránky operačního systému Red Hat® [34].



Obr. 42: Grafické zobrazení virtualizace (tři fyzické servery s dvanácti virtuálními stroji)

Pro všechny pražské týmy je tato platforma spravována centrálně a díky velkému počtu virtuálních strojů, na začátku této práce se jednalo přibližně o 40 strojů⁷, byla správa značně neefektivní. Vzhledem k rostoucímu počtu požadavků na vytváření stále nových virtuálních strojů bylo rozhodnuto zanalyzovat a optimalizovat nejzásadnější procesy spojené s administrací této virtuální infrastruktury.

⁶ VM = Virtual machine = Virtuální stroj

⁷ Informace je k 31.12.2016

3.3 Původní procesy ve virtuální infrastruktuře

3.3.1 Vytvoření nového stroje

Proces vytvoření nového stroje se skládal ze tří podprocesů:

- Vytvoření názvu
- Vytvoření hardwarové instance stroje
- Instalace operačního systému

3.3.1.1 Vytvoření názvu

Název stroje slouží k jasné identifikaci systému a je složen ze tří částí. První z nich je unikátní šestimístný kód, kterým se ve firmě jednoznačně identifikuje daný projekt a s ním veškeré akce spojené. Tento kód je také používán při inventarizaci virtuálního stroje v rámci infrastruktury a rozhodování, na jaký projekt je třeba účtovat provoz tohoto stroje a úkony spojené s administrací.

Dále je využíváno dvou až šesti slovného názvu popisujícího obsah virtuálního stroje. Tato část slouží ke snazší identifikaci platformy a jednoznačnému určení toho, jaké produkty jsou uvnitř systému nainstalovány

Posledním potřebným údajem pro inventarizaci je vlastník, který má jako jediný právo rozhodovat o zásadních akcích spojených se strojem. Jméno dané osoby se nepíše celé, ale uvádí se ve speciálním LDAP⁸ tvaru konkrétní uživatelské jméno, které má každý zaměstnanec zcela unikátně přiděleno skrze celou organizaci.

Zmíněné tři údaje následně tvoří celý řetězec názvu virtuálního stroje ve farmě. Při vytváření názvu bylo na administrátorovi, aby iteračním dotazováním zjistil veškeré údaje, vytvořil unikátní název tak, aby splňoval veškerá pravidla a následně jej použil v další části procesu. Na Obr. 43 je vidět instance stroje Michala Proška, který na projektu DFE472 má stroj s SMS programem, databází typu MySQL⁹ a zároveň je zde nainstalována i monitorovací část produktu.

DFE472-SMSPROGRAM_MYSQLDATABASE_MONITORING-mprosek

Obr. 43: Příklad názvu virtuálního stroje

⁸ LDAP = *Lightweight Directory Access Protocol* = Centralizovaný adresář, který uchovává veškeré informace o uživatelských účtech

⁹ MySQL = Relační databázový model

Proces začínal zasláním žádosti o nový stroj administrátorovi s určením nového jména. V případě, že název neodpovídal specifikovanému formátu, administrátor se dotázal uživatele na nové jméno, případně mu navrhl alternativu. Po správném vytvoření jména byl proveden záznam do excelového seznamu obsahujícího všechny nainstalované stroje v infrastruktuře.

3.3.1.2 Vytvoření hardwarové instance stroje

V rámci této části procesu bylo potřeba nastavit hardwarové požadavky pro daný virtuální stroj. Zpravidla se jednalo o počet jader virtuálních procesorů (vCPU¹⁰), velikost RAM¹¹ paměti a velikost pevných disků (HDD¹²) přidělených dané instanci.

Vlastník zaslal sumarizované požadavky ve standardizované podobě emailem, administrátor je zkontroloval a v případě nejasností, či chybějících informací zahájil opět komunikaci s konkrétním uživatelem a objasnil nesrovnalosti. V rámci nápravy bylo třeba kontaktovat uživatele (emailem, na firemním komunikátoru Skype for Business®, či osobně). Po vyjasnění všech tří hodnot došlo ke kontrole, zda se v infrastruktuře nachází potřebné množství místa pro vytvoření daného virtuálního stroje. Pokud byly výpočetní prostředky dostatečné, bylo možné se přesunout k poslední části nasazení. V opačném případě se odstranil záznam z excelového seznamu a nebylo možné nový stroj vytvořit.

3.3.1.3 Instalace stroje

Instalace stroje navázala na předchozí nadefinování hardwarových požadavků. Přidala do něj další informaci, kterou byl výběr operačního systému a jeho nastavení.

V rámci virtuální infrastruktury si mohou uživatelé vybrat mezi různými verzemi operačního systému Red Hat, nebo CentOS. Jedná se o linuxové distribuce, které mají podobnou architekturu a liší se pouze formou poskytované podpory. Red Hat® je určen pro profesionální použití v systémech s vysokou dostupností, a proto se platí vysoké roční poplatky za možnost využití garantované technické podpory. Pro CentOS distribuce se jedná o podporu pouze pomocí uživatelského fóra a není garantovaná odpověď, ani čas k vyřešení problému.

Uvnitř pražské virtuální infrastruktury mají programátoři k dispozici systémy verzí 6 i nejnovější 7, včetně všech jejich subverzí o architekturách 32 i 64 bit. Pro všechny dostupné

¹⁰ Virtual Central Processor Unit

¹¹ Random Accessed Memory

¹² Hard Disk Drive

verze se jedná o přibližně 25 různých typů operačních systémů, které díky možnosti různých velikostí disků tvoří geometrické množství kombinací, jimiž je možné systém nastavit.

Po vytvoření instalačních souborů a dokončení instalace administrátor ověřil, že je systém nainstalován dle požadavků vlastníka a v případě chybného nastavení ručně dohledával zdroj chyb, který opravil a následně spustil instalaci znovu.

Při správném nastavení byl administrátorem odeslán email obsahující IP adresu¹³ a název nově vytvořeného stroje, čímž proces „vytvoření nového stroje“ skončil. Všeobecně známé heslo nebylo v této zprávě zasíláno. Pro lepší pochopení je proces zobrazen na Obr. 45.

3.3.2 Přejmenování stroje, změna vlastníka

V rámci životnosti virtuálního stroje je občas za potřeby změnit popis obsahu, projektový kód, či vlastníka daného VM. Proces samotný znamená „pouze“ přejmenování virtuálního stroje, avšak nejdříve bylo třeba administrátorem ověřit v seznamu všech doposud nasazených virtuálních strojů, zda nový název již v minulosti nebyl použit. Užití již dříve použitého jména by znamenalo chyby ve virtuální infrastruktuře a generování velkého množství chybových záznamů v logovacích souborech. Díky tomu administrátor při přejmenování vytvářel vždy nové unikátní jméno.

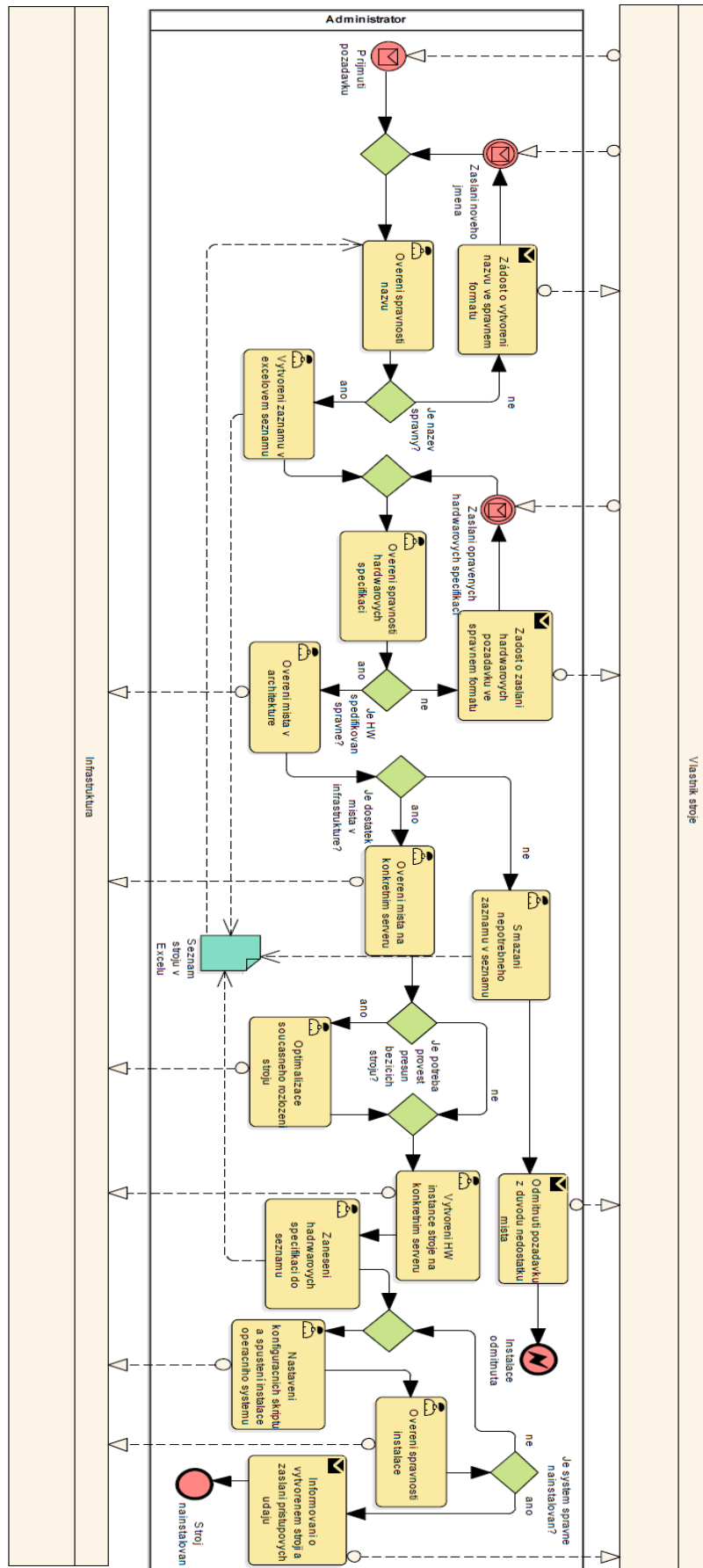
V procesu vlastník konkrétního stroje určil, jaký konkrétní stroj je třeba přejmenovat a poskytl požadované nové jméno. V případě nalezení administrátor, stejným způsobem jako při vytváření zkontroloval veškeré náležitosti nového jména, a po vyhledání konkrétního záznamu v seznamu i v infrastruktuře, provedl přejmenování.

Jestliže požadované nové jméno již bylo v minulosti použito pro jiný virtuální stroj, administrátor vyřešil tento problém návrhem nového jména, tedy přidáním vyšší číslice na konec názvu virtuálního stroje. Příklad takového jména, které muselo být uměle změněno na unikátní, je na Obr. 44. a celý proces následně na Obr. 46.

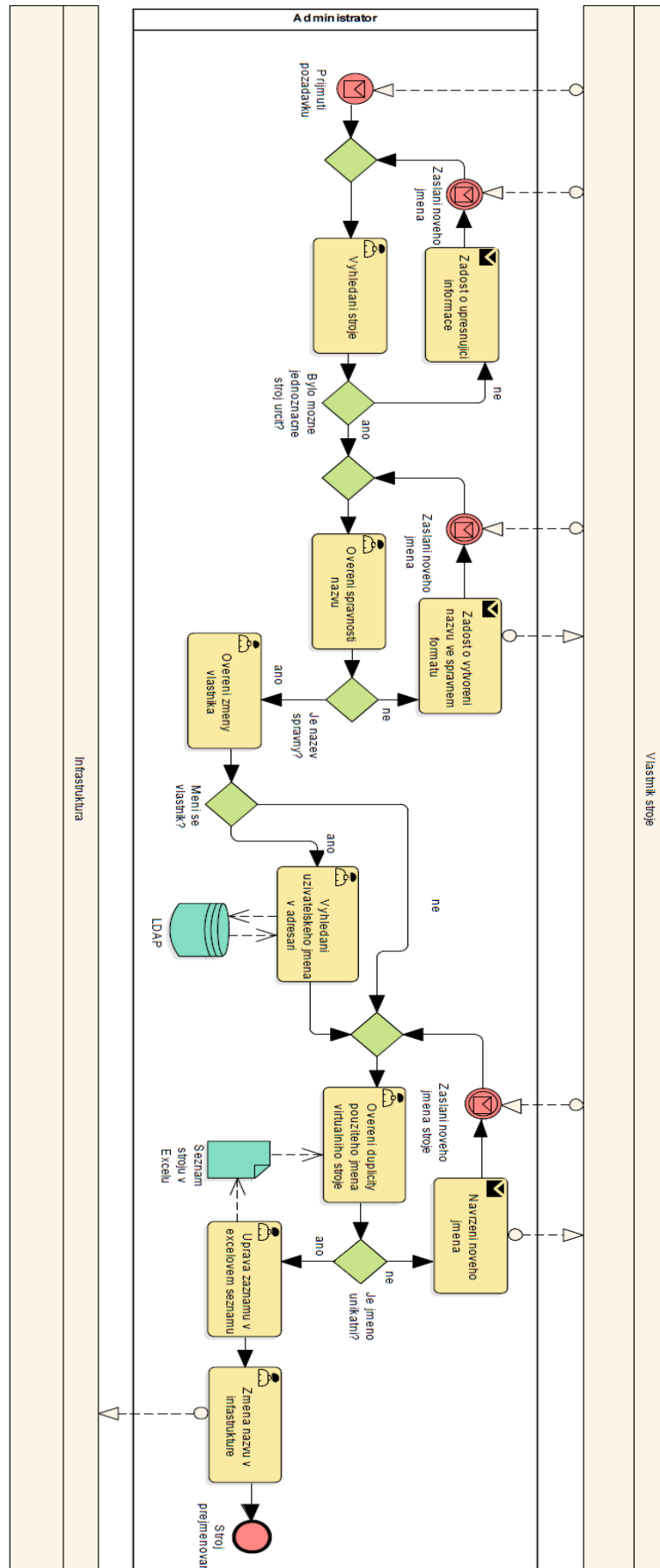
DFE472-SMSPROGRAM_MYSQLDATABASE_MONITORING_1-mprosek

Obr. 44: Příklad názvu v případě duplicity jmen

¹³ IP adresa = unikátní adresa každého počítače v počítačové síti [41]



Obr. 45: Proces vytvoření stroje

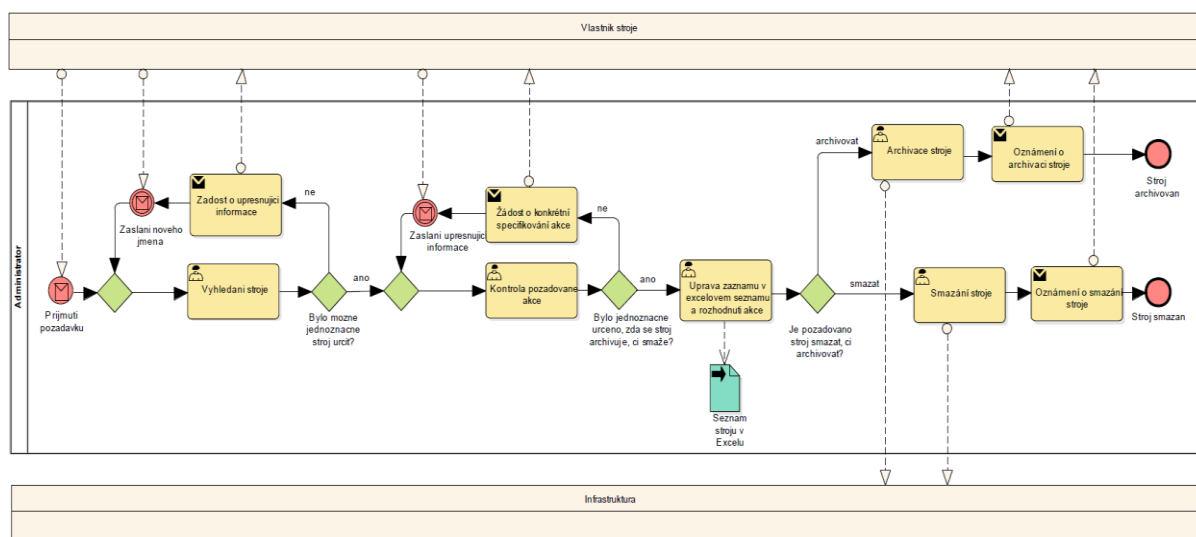


Obr. 46: Proces přejmenování stroje

3.3.3 Odstavení stroje

Odstavení stroje je relativně jednoduchý úkon, který spočívá v identifikaci a vyhledání stroje, který se může odstavit a potvrzení tohoto úkonu vlastníkem. Po zaslání žádosti bylo třeba administrátorem řádně ověřit, že byla jednoznačně specifikována akce, která se měla se strojem vykonat.

Pokud nebylo možné jednoznačně určit VM, které mělo být smazáno, docházelo opět k dodatečné komunikaci s vlastníkem. V procesu archivace nedocházelo k okamžitému smazání stroje, ale vypnutá instance se přesunula do archivního úložiště, ve kterém stroj zůstal na vždy pro případ, že by bylo třeba jej v budoucnu obnovit a pracovat s ním. Proces, který se dostal do fáze smazání, či archivace VM pokračoval zasláním notifikace vlastníkovi, jak je zobrazeno na Obr. 47.



Obr. 47: Proces odstavení stroje

3.3.4 Klonování stroje

Velmi často vznášeným požadavkem, ke kterému uživatelé potřebují součinnost s administrátorem virtuální infrastruktury, je klonování stroje. Klonování spočívá ve vytvoření zcela identické kopie zdrojového systému, tedy klonu.

Existují dva typy klonování:

- Vytvoření nového stroje
- Přepis existujícího stroje

V prvním případě na konci procesu existovala nová instance virtuálního serveru s novou, doposud nepoužitou IP adresou. V druhém případě byl současný běžící stroj smazán

a na novém stroji se nastavila IP adresa původního stroje. V obou případech bylo však nutné určit nové unikátní jméno virtuálního stroje, jelikož staré nemohlo být znovu použito z důvodu vytváření chybových záznamů, jak bylo popsáno výše.

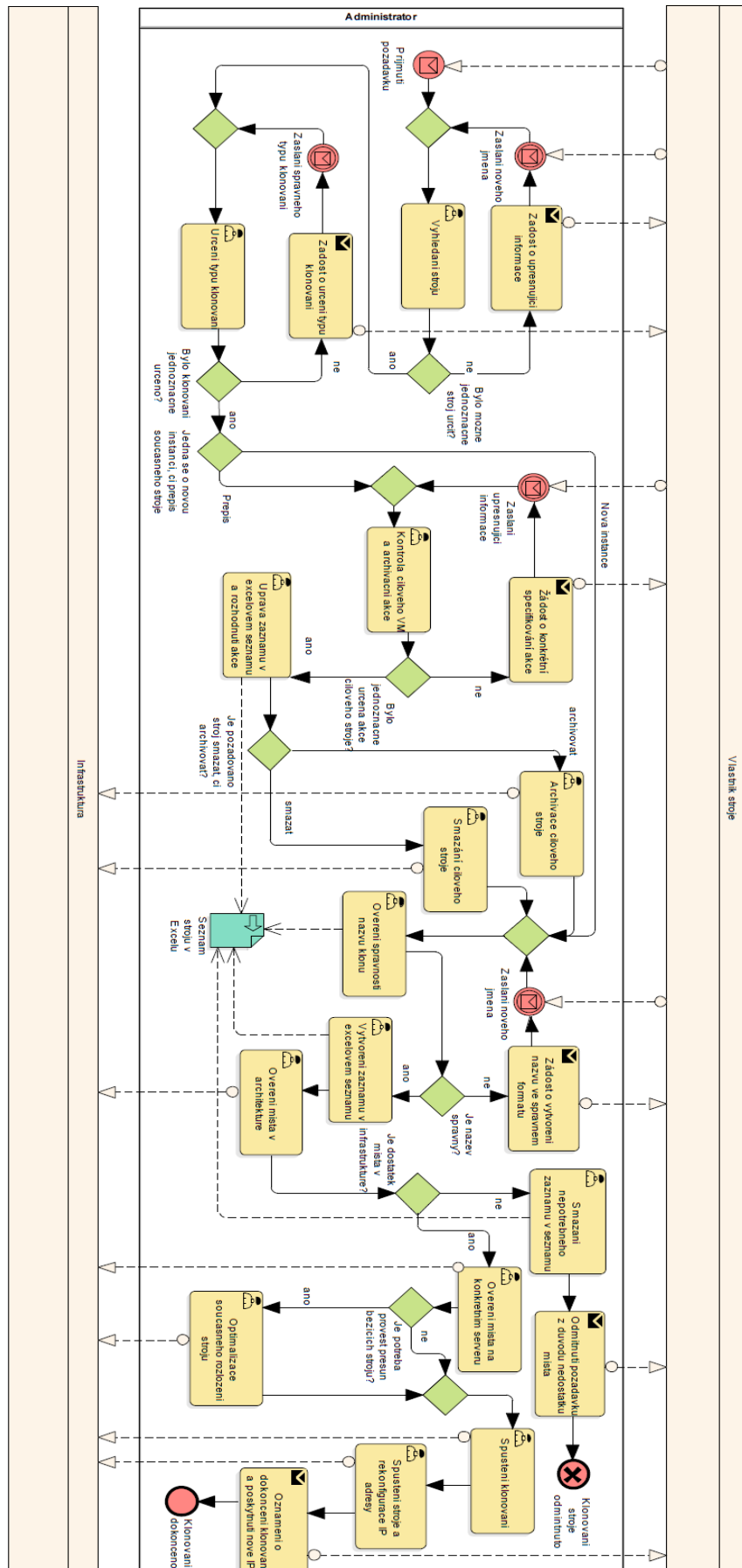
Proces zahajoval uživatel vznesením požadavku obsahující označení zdrojového VM a cílového stroje. Současně s tím bylo na uživateli, aby poskytl jméno nového virtuálního stroje. Administrátor po obdržení požadavku zkontroloval validitu vstupů a případně kontaktoval vlastníka pro upřesnění. Stejně jako v případě přejmenování bylo třeba v tomto procesu správně určit zdrojový stroj a pokud nebyla identifikace jasná, dané VM se dohledávalo. Zároveň docházelo k validování potřebných výpočetních prostředků.

Následně byl spuštěn proces klonování a čekalo se, dokud se data nezkopírují. Po dokončení bylo, pouze v první verzi klonování, dalším krokem změna IP adresy a vždy zaslání oznámení vlastníkovi o dokončení provedených změn. Proces je zobrazen na Obr. 48.

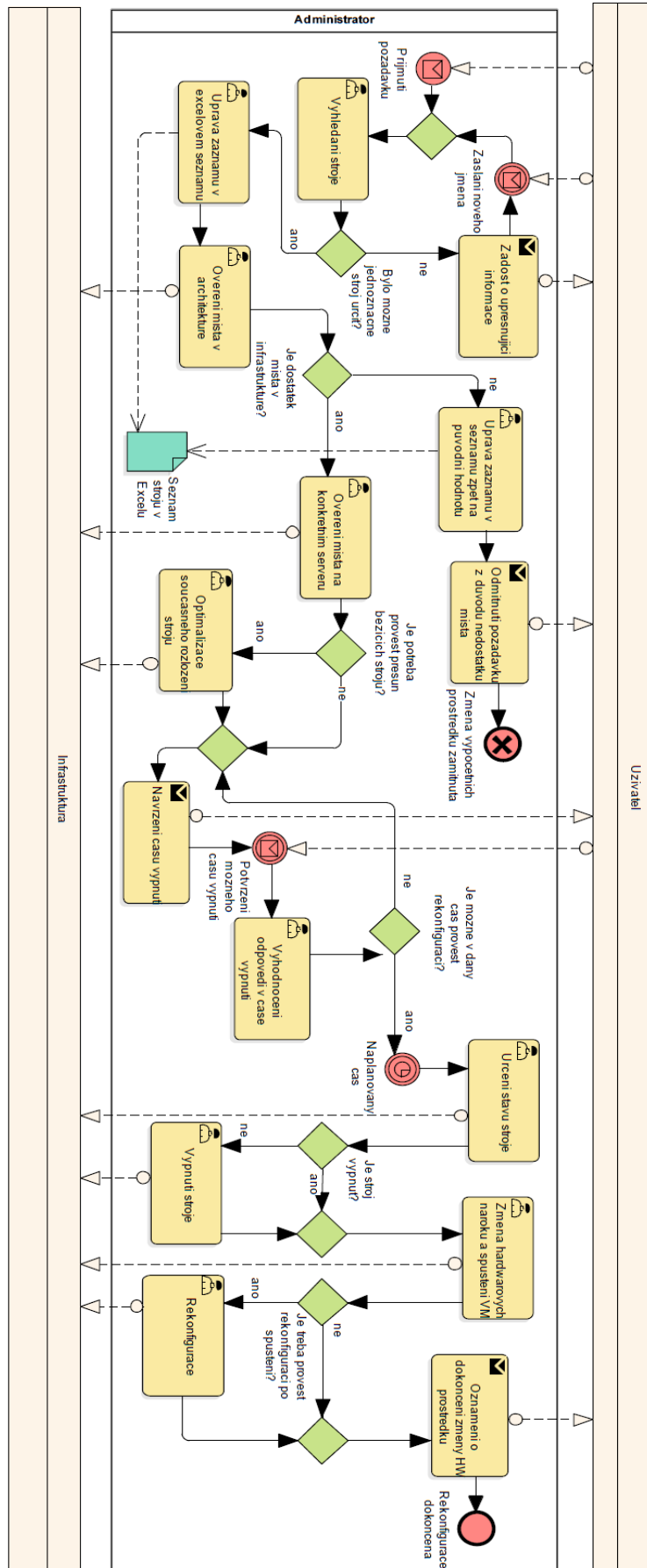
3.3.5 Změna výpočetních prostředků

V průběhu životnosti virtuálního stroje dochází ke změnám nároků na hardwarové požadavky. Množství procesorů, paměti i velikosti pevných disků může být variabilně zvyšováno či snižováno dle aktuální potřeby. Tento proces však vyžadoval součinnost administrátora platformy s vlastníkem, jelikož je třeba operační systém pro aplikování HW změny vypnout.

Průběh procesu na Obr. 49 byl následující. Uživatel zažádal o změnu prostředků, na což reagoval administrátor platformy. V případě, že byl stroj jasně definován a byly k dispozici výpočetní prostředky, tak žádost schválil. Následně začala komunikace o čase vypnutí. Po domluvě a vypnutí v daném čase proběhla požadovaná změna hardwarových prostředků. Administrátor po skončení rekonfiguračních úkonů uvedl stroj opět do provozu. V případě manipulace s velikostmi jednotlivých disků bylo zapotřebí provést i administrativní úkony po spuštění systému. Na konci procesu byl uživatel informován o změně a mohl systém opět využívat.



Obr. 48: Proces klonování stroje

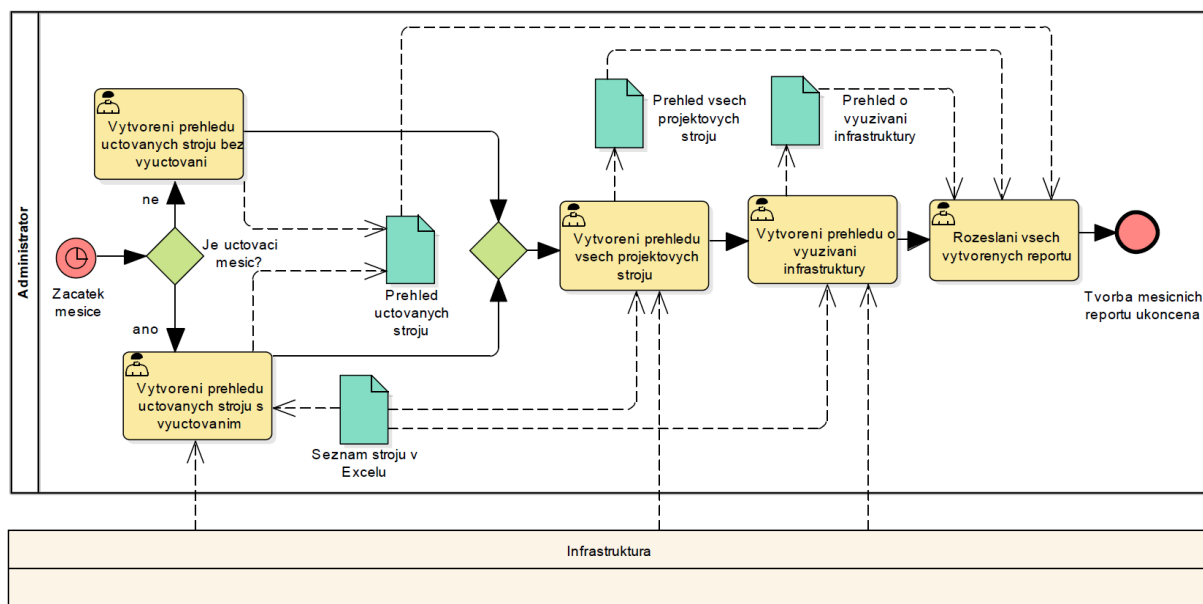


Obr. 49: Proces změny výpočetních prostředků

3.3.6 Generování informací o stavu virtuální farmy a účtování

Na začátku každého měsíce probíhalo ruční rozesílání informačních emailů společně s tvorbou interního vyúčtování pro konkrétní projekty. Výsledkem procesu byly tři souhrnné emaily obsahující seznam všech virtuálních strojů, seznam virtuálních strojů, které se budou po tři měsíčním období souhrnně účtovat a výkonový přehled o využití infrastruktury.

Manuální tvorba těchto přehledných seznamů spadala do zodpovědnosti administrátora. Probíhala pomocí kopírování dat z excelového přehledu ale i z infrastruktury do emailů a ruční tvorbou veškerých přehledů výkonnosti, jak je vidět v procesu na Obr. 50.



Obr. 50: Generování informací o stavu virtuální infrastruktury a účtování

3.4 Nalezení neoptimálních procesů

Veškeré zmíněné procesy byly postupně prozkoumány a v této kapitole jsou popsány neoptimální činnosti, které se v nich prováděly.

3.4.1 Vytvoření nového stroje

3.4.1.1 Vytvoření názvu

V průběhu tohoto procesu nastával velký problém v opakovaném dotazování se na stále stejná data. Dodávání neúplných informací a neznalost požadavků stran uživatelů, stejně jako neochota detailně vyplňovat standardizované emailové šablony, byly důvodem pro zbytečné prodlužování procesu. Díky těmto nedostatkům byl administrátor nucen se iteračně dotazovat na chybějící údaje uživatelů a stále vysvětlovat ty stejné podmínky pro vyváření názvů strojů, což je patrné z Obr. 45.

Problém se skrýval i v manuálním dohledávání uživatelského jména vlastníka, které v optimistickém případě znamenalo práci na 1 minutu a 10 kliknutí kurzorem myši se současným zkopírováním uživatelského jména do názvu. Toto kopírování však na základě zkušeností často způsobovalo chybu ve formě chybějícího písmena na začátku či na konci uživatelského jména, díky chybnému označení slova při kopírování.

3.4.1.2 Vytvoření hardwarové instance

Uvnitř těchto dvou podprocesů bylo časově náročné především manuální nastavování veškerých parametrů. Vytvoření stroje spočívalo v otevření webové stránky s konzolou, do které se vyplnily hardwarové požadavky na stroj, viz Obr. 51. Jednalo se o pouhou transformaci údajů uvedených v emailu, či komunikátoru, od zaměstnance do jiného nástroje. V rámci přepisu většinou nevnikaly chyby, avšak současně se jednalo o časově zatěžující práci pro administrátora, která nepřinášela žádný užitek a byla pouhou repeticí stále stejného postupu.

Dalším problémem byl fakt, že virtuální infrastrukturu zajišťovalo 7 velmi málo výkonných serverů, které měly různou kapacitu v rozmezí 16 – 64 GB RAM. Díky tomu bylo potřeba při vytváření každého nového stroje pečlivě dohledat, na jakém fyzickém stroji lze nový virtuální stroj spustit, aby nedošlo k zahlcení serveru, případně k učinit rozhodnutí, jaké virtuální stroje je nutné přesunout na jiné fyzické servery. Dohledávání místa v případě více zaplněné infrastruktury¹⁴ byl velice časově náročný úkol, který zdaleka nepřinášel užitek.

¹⁴ Zaplněná infrastruktura znamená využití veškeré RAM paměti alespoň na 80%

New Virtual Machine

✓ 1 Select a creation type
 ✓ 2 Select a name and folder
 ✓ 3 Select a compute resource
 ✓ 4 Select storage
 ✓ 5 Select compatibility
 ✓ 6 Select a guest OS
7 Customize hardware
 8 Ready to complete

Customize hardware
 Configure the virtual machine hardware

Virtual Hardware VM Options

ADD NEW DEVICE

> CPU *	1		
> Memory	2	GB	
> New Hard disk *	16	GB	
> New SCSI controller *	VMware Paravirtual		
> New Network *	TEST		<input checked="" type="checkbox"/> Connect...
> New CD/DVD Drive *	Client Device		<input type="checkbox"/> Connect...
> Video card	-		
VMCI device	Device on the virtual machine PCI bus that provides support for the virtual machine communication interface		
New SATA Controller			

CANCEL BACK NEXT

Obr. 51: Rozhraní pro vytvoření nového virtuálního stroje

Po vytvoření stroje bylo nutné přenesení MAC adresy z administrace virtualizační části do konfiguračních souborů, společně s jejich následnou modifikací.

3.4.1.3 Instalace stroje

Bylo třeba provést výběr správné distribuce systému, následně nastavit konfigurační soubory systému tak, aby vše odpovídalo potřebám programátora, a zároveň instalátor byl schopen dané konfiguraci porozumět. Změna konfigurace zahrnovala manuální vytvoření pěti konfiguračních souborů, v žádném z nich se nesměla udělat chyba a veškeré informace na sebe musely odkazovat přesně tak, jak byly zadány v jiném souboru.

Všechny soubory byly vytvářeny manuálně a v případě nekorespondování jmenných odkazů, způsobovaly neúspěšnou instalaci celého systému, či chybné nastavení stroje, které nesplňovalo požadavky vlastníka. Následná náprava, která byla nutná průměrně u každé třetí

instalace¹⁵, poté znamenala manuální dohledávání chyby a tím pádem opět zbytečně vynaložený čas administrátora platformy.

3.4.2 Přidělení přístupu

Neexistence centralizované databáze, která by obsahovala seznamy všech uživatelů, kteří měli mít k daným strojům přístup, znamenala nemožnost jakkoliv efektivně spravovat uživatelská práva. Díky tomu se na všech systémech používalo stejné administrátorské heslo, a proto tento podproces nebyl nijak uveden ve výčtu stávajících procesů.

Tento přístup neumožňoval, aby k danému virtuálnímu stroji měla přístup pouze konkrétní skupina uživatelů. Naopak každý uživatel mohl pracovat na libovolném stroji ve farmě dle jeho uvážení, případně mohl omylem začít pracovat na jiném stroji, jelikož znal přihlašovací údaje, ale zadal nechtěně špatnou adresu serveru, čímž se úspěšně připojil na jiný stroj.

V případě nutnosti byly pro jednotlivé stroje ručním způsobem uživatelské účty vytvářeny, ale nikoliv pod centrálním dohledem administrátora virtualizační infrastruktury, nýbrž vždy pod zodpovědností vlastníka stroje.

3.4.3 Přejmenování stroje, změna vlastníka

Při přejmenování se kromě nutnosti správné syntaxe názvu trávil enormní čas dohledáváním jmen, které již byly použity a porovnáváním s nově vytvořeným označením. Druhou částí bodem procesu, kterou bylo vhodné optimalizovat, se ukázalo manuální kopírování a dohledávání uživatelských jmen z LDAP adresáře, při kterém vznikalo množství chyb díky nesprávnému označování kopírovaného textu. Zároveň nalezení konkrétního uživatelského jména nebyl snadný úkon v rámci jednotek vteřin, ale spíše v rámci minut.

3.4.4 Odstavení stroje

Jak bylo zmíněno v popisu současného procesu, je odstavení stroje relativně snadným úkonem. Úskalí tohoto procesu však bylo opět v komunikaci s vlastníky virtuálních strojů při zaslání požadavku na odstranění stroje. Jelikož nebyli vlastníci jasně obeznámeni s možnostmi platformy, pravidelně nespecifikovali, zda chtějí stroj zcela smazat, či přesunout do archivu a zde ponechat na vždy. Druhým problémem bylo nepřesné a neúplné označování strojů, které se nesnadno dohledávaly, a hrozilo smazání nesprávného stroje.

¹⁵ Subjektivní hodnocení administrátora platformy

V případě, že by se při mazání administrátor spletl, či by vlastník určil nepřesně zdrojový stroj, mohl daný přečin znamenat ztrátu desítek až stovek MD¹⁶. Původní proces neměl proti této chybě naprosto žádnou ochranu a po zaslání požadavku na smazání docházelo okamžitě k trvalému odstranění stroje.

3.4.5 Klonování stroje

Vytváření klonu je samo o sobě činností náročnou na čas, jelikož kopírování jednoho průměrně velkého stroje trvá přibližně 4 hodiny. Před započítím klonování nastávaly stejné problémy, již zmíněné v rámci přejmenovávání. Po dokončení klonování bylo třeba navíc přenastavit cílovou instanci na jinou IP adresu, než která se vyskytovala na zdrojovém stroji. Toto znamenalo, že administrátor musel sledovat proces klonování, a ve chvíli kdy se úloha dokončila, přistoupil k manuální změně parametrů. V případě varianty přepisu již běžící instance bylo ještě zapotřebí vypnout a smazat, či přesunout do archivního úložiště současný stroj. Pokud došlo k nesprávné výměně informací a byl smazáno chybné VM, mohla tato chyba znamenat, stejně jako v procesu „Odstavení stroje“ ztrátu desítek MD.

Prodlužování procesu navíc opět působili vlastníci, jelikož neuváděli akci, která měla být se současným strojem provedena. Proces klonování v sobě v podstatě zahrnuje podproces vytvoření nového názvu a proces „Odstavení stroje“.

3.4.6 Změna výpočetních prostředků

Nevýhodou tohoto procesu byla nutnost zapojení alespoň dvou osob, které spolu musely interagovat. Nalezení vhodného času, kdy se s danou platformou nepracovalo a zároveň zajištění toho aby ve stejné době mohl administrátor přenastavit prostředky, bylo často velice náročné a požadovalo velké množství času pouze na podproces domlouvání se. Samotná změna prostředků byla neefektivní pouze pro změny v diskových velikostech, jelikož vyžadovala následnou manuální rekonfiguraci uvnitř operačního systému, která byla časově náročná a znamenala jen opakování stále stejných úkonů.

Poslední nevýhodou procesu byla skutečnost, že nedocházelo ke schvalování změny výpočetních prostředků vlastníkem a díky tomu vyvstávaly dohady v případě účtování vyšších částek, odpovídajících vyšším hw požadavkům, než které vlastník v době nasazení stroje schválil.

¹⁶ MD = Man Day = 1 den práce pro 1 pracovníka

3.4.7 Generování informací o stavu virtuální farmy a účtování

V rámci všech výše uvedených procesů bylo nutné po každém úkonu, do excelového seznamu strojů, korektně zapsat všechny změněné údaje, které musely korespondovat s vlastnostmi virtuálních strojů ve farmě. Toto manuální přepisování bylo jednak časově náročné, ale převážně neefektivní z důvodu velké chybovosti, která vznikala opomenutím inventarizace jako součásti procesu, či zapsáním nesprávných hodnot do excelového souboru.

Nesprávné hodnoty byly občas objeveny při vytváření souhrnných reportů. Nebylo však výjimkou, že byla chyba nalezena až s odstupem několika měsíců, což způsobilo nesprávné naúčtování výpočetních prostředků ostatním týmům¹⁷.

V rámci tohoto procesu docházelo k velkému množství času stráveného na tvorbě reportů. Čas potřebný k vytvoření přehledů byl přibližně 3 hodiny měsíčně. V každém kroku procesu zároveň vznikalo riziko vytvoření chyby, jejímž důsledkem mohlo být nesprávné zaúčtování, které vedlo k nižšímu výdělku celého týmu. Dále bylo nutné, aby administrátor byl vždy na začátku měsíce k dispozici, jelikož v opačném případě nebyl report vytvořen a tím pádem nemohlo dojít ani k vyúčtování běžících strojů.

¹⁷ Pozn. Administrátora: Převážně se jednalo o naúčtování nižších částek

3.5 Navržení nových procesů

Z popisu původních procesů bylo zřejmé, že všechny operační procesy spojené s provozem virtuální infrastruktury obsahovaly velké množství časově náročných operací. Zároveň se vyskytovalo mnoho méně časově náročných kroků, které však bylo nutné přesně dodržet a na žádný z nich se nesmělo nezapomenout. Současně byla většina úkonů prováděna za pomoci opakování stále stejných akcí, které nepřinášely žádný přidaný užitek, prodlužovaly dobu procesu, zvyšovaly chybovost a požadovaly součinnost více osob na jednotlivých činnostech, což zbytečně prodlužovalo dobu celého procesu.

V rámci práce byly navrženy procesy nové, které zmíněné neoptimální kroky za použití automatizace eliminovaly a snížily tak časovou náročnost i chybovost a zároveň rozšířily využitelnost platformy. Nedílným důsledkem optimalizace je i celkové zjednodušení procesů z pohledu jednotlivých účastníků.

Při vytváření procesů byl vyvinut automatizační framework¹⁸, který je zprovozněn na operačním systému Red Hat®. Informace o všech strojích, včetně zašifrovaných hesel, se uchovávají v databázi MySQL [35] a webové rozhraní pro interakci s vlastníky bylo naprogramováno v jazyku PHP¹⁹ [36]. Napojení na virtualizační platformu VMware bylo provedeno za pomoci API²⁰ rozhraní, konkrétně v implementaci „VMware vSphere Automation SDK for Python“ [29].

3.5.1 Vytvoření nového stroje

Proces, který díky automatizaci umožňuje snazší a rychlejší nasazení téměř bez nutnosti interakce administrátora. Uživatel zadá po přihlášení do formuláře na webovém rozhraní frameworku požadovaný název virtuálního stroje. V případě, že syntaxe neodpovídá zadaným pravidlům, nebo je některá položka vyplněna chybně, vnitřní logika frameworku oznámí uživateli, jaká část je třeba opravit. Současně je ověřována i duplicita názvů veškerých použitých strojů v infrastruktuře.

Po úspěšně vytvořeném unikátním názvu je zobrazena na Obr. 52 stránka, do které jsou vkládány požadavky na výpočetní prostředky a typ operačního systému. V případě chybného

¹⁸ Soubor knihoven, programů, skriptů a postupů, které umožní rutinní úkoly abstrahovat do generických modulů, které lze snadno opakovat. [40]

¹⁹ PHP = Skriptovací programovací jazyk [43]

²⁰ API = Application Programming Interface = Rozhraní, které umožňuje dvěma aplikacím spolu navzájem komunikovat na základě jednoznačné syntaxe předem definovaných příkazů [42]

zadání údajů framework vlastníka opět upozorní a požádá o nápravu i s vysvětlením, jaký údaj je chybně zadán.

VM creation request

Owner Mandatory. VM owner responsible for the further VM operations

Description Mandatory. VM description outlining the VM purpose separated by underscore.

Project code Mandatory. Project code the VM is assigned to

CPU (threads) Mandatory. Number of CPU threads that will be available to the OS in thread instance

RAM (GB) Mandatory. Ammount of memory available to the OS in GB

System disk size (GB) Mandatory. Disk size allocated for the OS in GB

Mountpoint 1 path and size (GB) Optional. Disk size allocated for additional mountpoint in GB

Mountpoint 2 path and size (GB) Optional. Disk size allocated for additional mountpoint in GB

Mountpoint 3 path and size (GB) Optional. Disk size allocated for additional mountpoint in GB

Guest/OS type Mandatory. Type of OS to be installed on the VM or Guest type. Select only one of those.

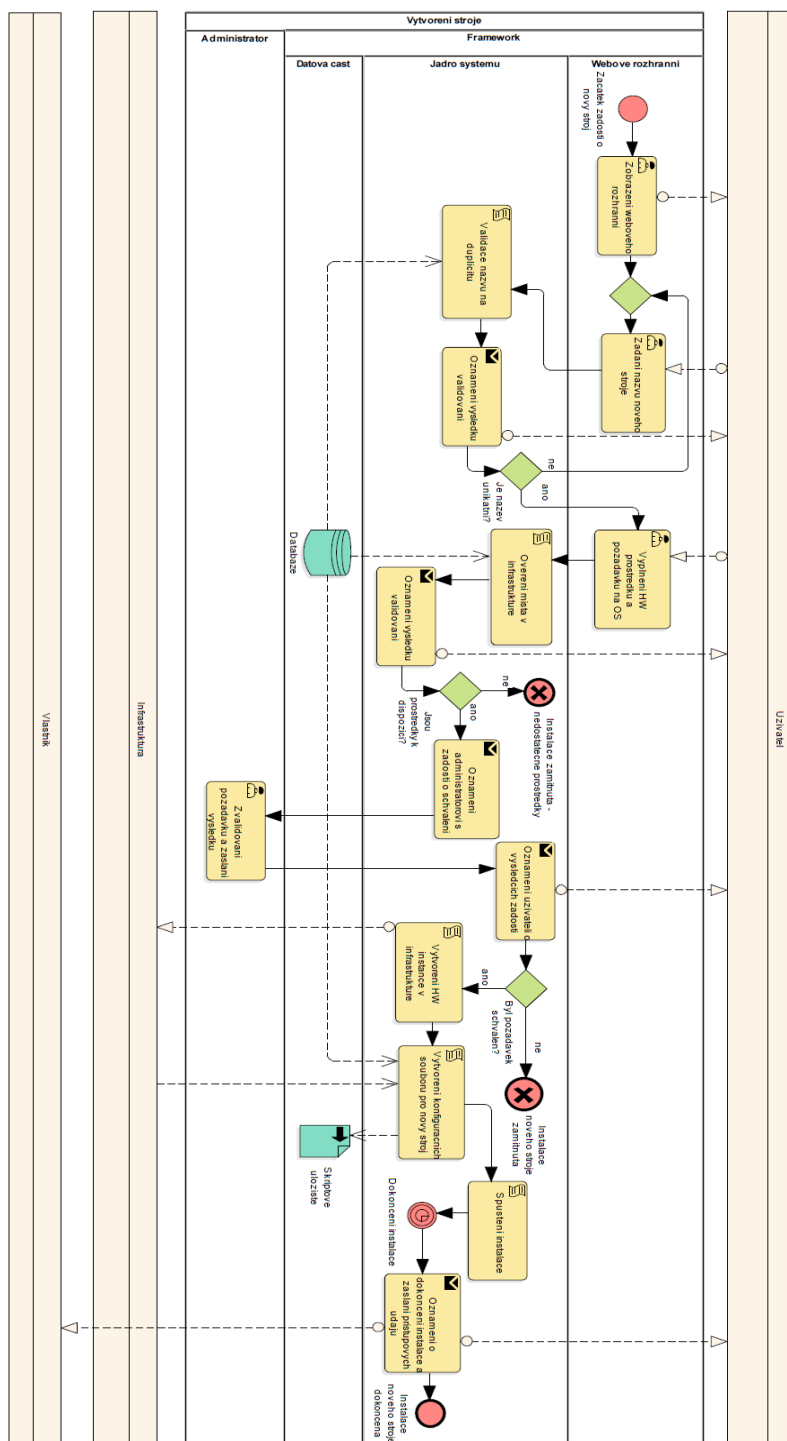
Hostname Optional. Fill in to override the default generated hostname.

Comment

[Review new VM request](#)

Obr. 52: Příklad vyplněného webového formuláře pro vytvoření nového stroje

Následně pokračuje aplikační logika automatickou kontrolou infrastruktury, kde ověřuje volné výpočetní prostředky. Po kladné automatické validaci všech vstupů se zašle email administrátorovi a čeká se na schválení/zamítnutí vytvoření stroje. V případě, že administrátor vytvoření stroje schválí, framework zajistí vytvoření hardwarové instance ve virtuální infrastruktuře, vygenerování potřebných souborů i interakci s virtualizační farmou, která zajistí spuštění instalace dle zadaných požadavků.



Obr. 53: Proces vytvoření stroje - Optimalizováno

Schvalování nově využitých hardwarových prostředků bylo ponecháno, jako obrana před uživateli, aby vytvářeli jen nutně potřebné množství strojů. Úspěšnému schválení následuje vytvoření hardwarové instance stroje, generování instalačních skriptů a spuštění automatizované instalace.

Již před optimalizací procesu byla instalace prováděna pomocí Kickstart [37] souborů, které umožňují linuxový operační systém nainstalovat tak, jak si libovolný vlastník přeje. Vytváření souborů však probíhalo pouze manuální cestou. Díky vytvoření automatizačních skriptů, které jsou napojeny přímo na informace zadané od programátora do webového rozhraní, jsou soubory vygenerovány automaticky a během několika vteřin i nakopírovány do správných umístění. Nutnost zapojení administrátora je opět zcela eliminována a zároveň je naprosto odstraněn prostor pro možnost vytvoření jakékoliv chyby při nastavování platformy.

Po dokončení instalace systému je vlastníkovi stroje zaslán zašifrovaný email obsahující přístupové údaje i s náhodně vygenerovaným administrátorským heslem, jež je na stroji nastaveno a zajišťuje tak možnou selekci přístupů. Vzhledem k diversifikaci hesel je vytvořen nový proces „Přidělení přístupu“.

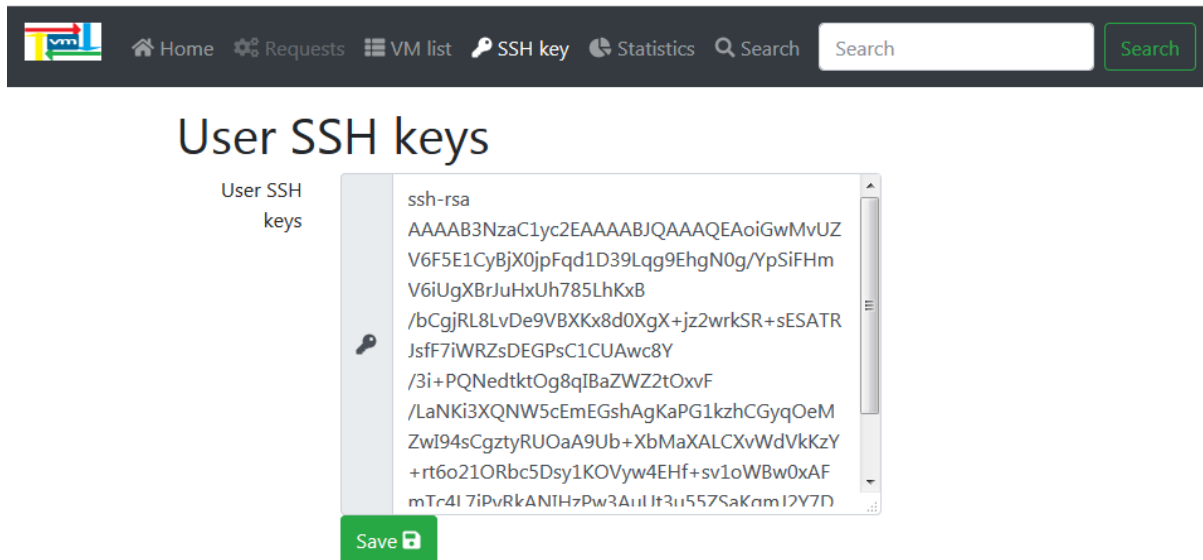
Všechny původní podprocesy nejsou nadále uváděny odděleně, jelikož se jedná pouze o vyplnění standardizovaného webového formuláře, který zajišťuje i validaci vstupů. Vzhledem k propojení formuláře, databáze i virtualizační farmy, probíhá přenos všech údajů zcela automatizovaně, čímž se šetří čas, který je třeba při instalaci strávit jak na straně administrátora, tak i uživatele.

3.5.2 Přidělení přístupu

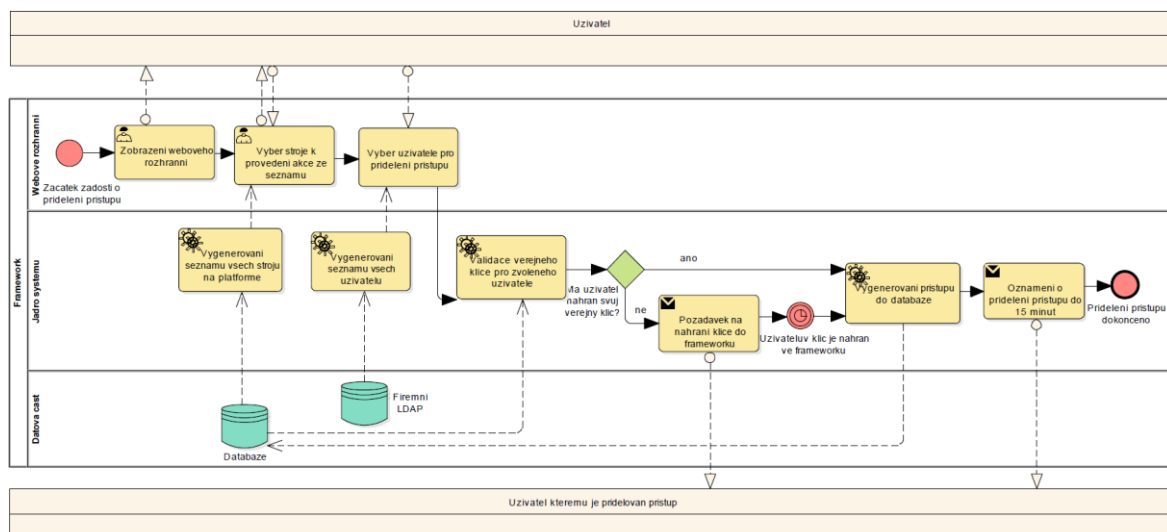
Vzhledem k vytváření diversifikovaných hesel je možné přidělovat práva přístupu pouze konkrétním uživatelům. Administrátorské heslo je určeno pouze pro vlastníka v případě nouze. Veškeré přístupy ostatních uživatelů, kteří mají mít možnost se strojem pracovat, jsou dále řešeny pomocí systémů RSA veřejných klíčů [38].

Všichni známí uživatelé mají své veřejné klíče uloženy v databázi frameworku a vlastník pouze vybírá z firemního adresáře, kterému člověku má být přístup přidělen nebo odebrán. V případě, že uživatel nemá svůj klíč uložen v databázi, je mu zaslán informační email s pokyny. Po nahrání klíče do systému dle Obr. 54, je mu automaticky přidělen přístup na všechny stroje, ke kterým má mít přístup. Přidělování přístupů není schvalováno vlastníkem ani administrátorem systému, protože se nepředpokládá, že by někdo cíleně chtěl škodit

na strojích, na kterých nemá mít přístup a je zvolena varianta volného přiřazení přístupu za účelem zjednodušení procesu na Obr. 55.



Obr. 54: Formulář pro nahrání veřejného klíče do farmy



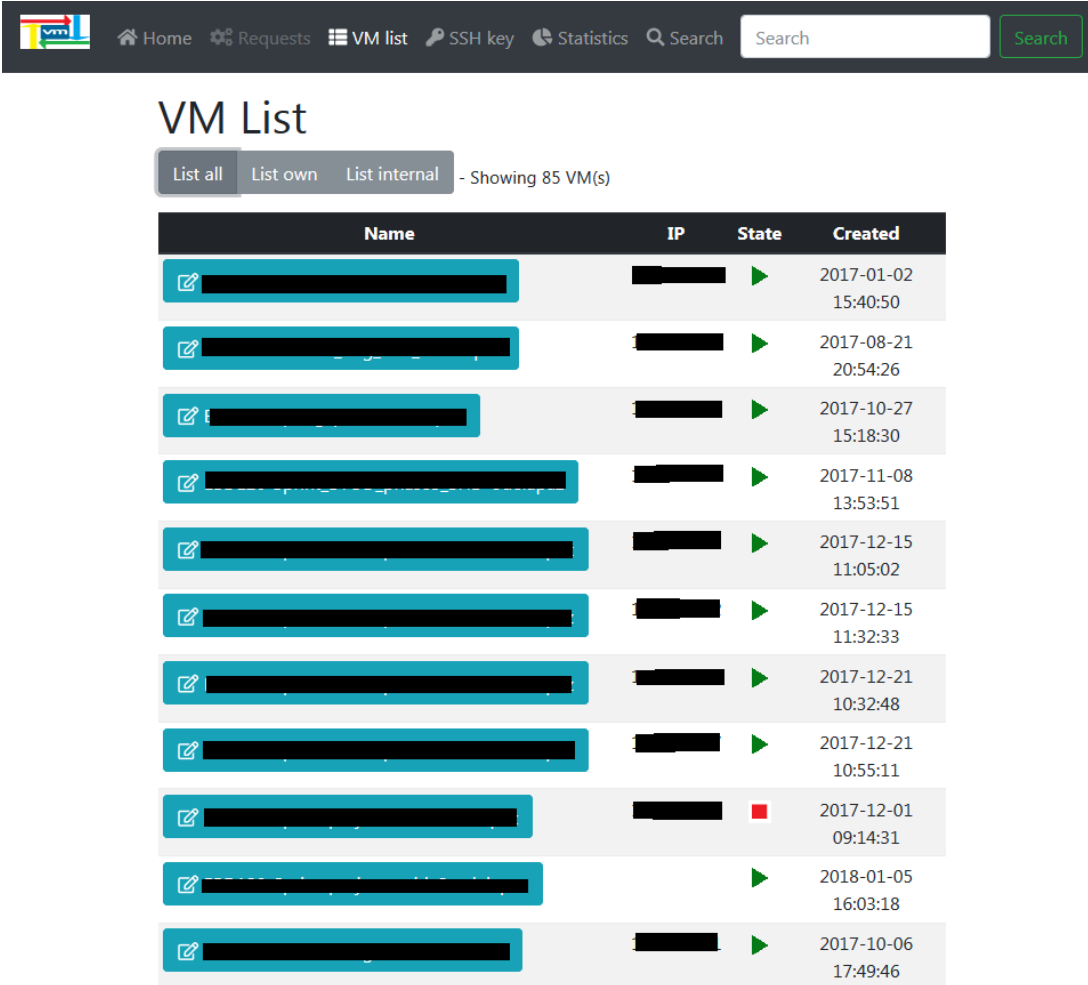
Obr. 55: Přidělení přístupu - Optimalizováno

Framework v pravidelných intervalech kontroluje seznam povolených uživatelů a porovnává s údaji nastavenými v databázi. V případě disproporce v konfiguraci je na stroj zapsána validní konfigurace z databáze a vlastník je informován o změně. Tato naplánovaná úloha se spouští každých 15 minut a není součástí procesu přidělení přístupu.

Zcela automatizovaný proces, který zásadně zvyšuje bezpečnost všech strojů, by bez automatizačního frameworku a centralizované databáze vůbec nemohl vzniknout.

3.5.3 Přejmenování stroje, změna vlastníka

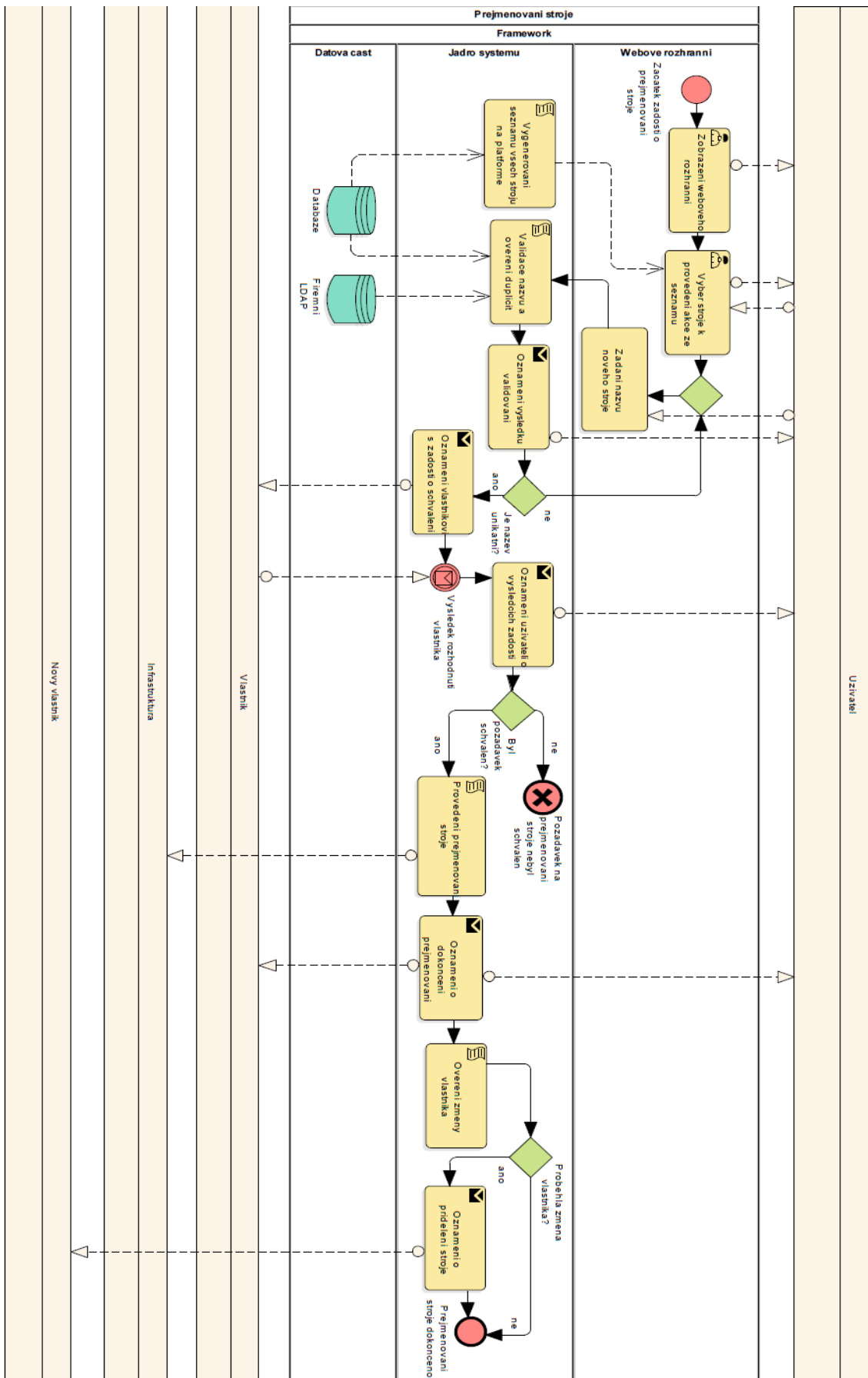
Plně automatizovaný proces přejmenování/změna vlastníka ve webovém formuláři, jenž po výběru stroje Obr. 56 okamžitě ověří nové jméno a v případě duplicity s již použitým jménem v minulosti oznámí tuto skutečnost vlastníkovi a požádá o změnu jména. V případě korektně zadaného nového jména dojde k zaslání oznámení současněmu vlastníkovi s dotazem na schválení.



Name	IP	State	Created
[REDACTED]	[REDACTED]	▶	2017-01-02 15:40:50
[REDACTED]	[REDACTED]	▶	2017-08-21 20:54:26
[REDACTED]	[REDACTED]	▶	2017-10-27 15:18:30
[REDACTED]	[REDACTED]	▶	2017-11-08 13:53:51
[REDACTED]	[REDACTED]	▶	2017-12-15 11:05:02
[REDACTED]	[REDACTED]	▶	2017-12-15 11:32:33
[REDACTED]	[REDACTED]	▶	2017-12-21 10:32:48
[REDACTED]	[REDACTED]	▶	2017-12-21 10:55:11
[REDACTED]	[REDACTED]	■	2017-12-01 09:14:31
[REDACTED]	[REDACTED]	▶	2018-01-05 16:03:18
[REDACTED]	[REDACTED]	▶	2017-10-06 17:49:46

Obr. 56: Přehledné zobrazení všech virtuálních strojů pro výběr akce

Kladná odpověď znamená okamžité přejmenování stroje. V případě odmítnutí je proces ukončen a přejmenování není provedeno. V obou částech procesu se zasílá oznámení žadateli, a pokud se v rámci přejmenování mění i vlastník, zašle se i informační email o přidělení stroje novému vlastníkovi. Administrátor infrastruktury touto akcí není nijak zatěžován, jak je vidět na Obr. 57, čímž je jednak naprosto vyloučena chyba lidského faktoru, ale také není třeba k tomuto procesu využívat časových prostředků administrátora a díky tomu lze celý proces provádět levněji. O ekonomické stránce veškeré implementace a následné správy infrastruktury se píše v kapitole 3.7.



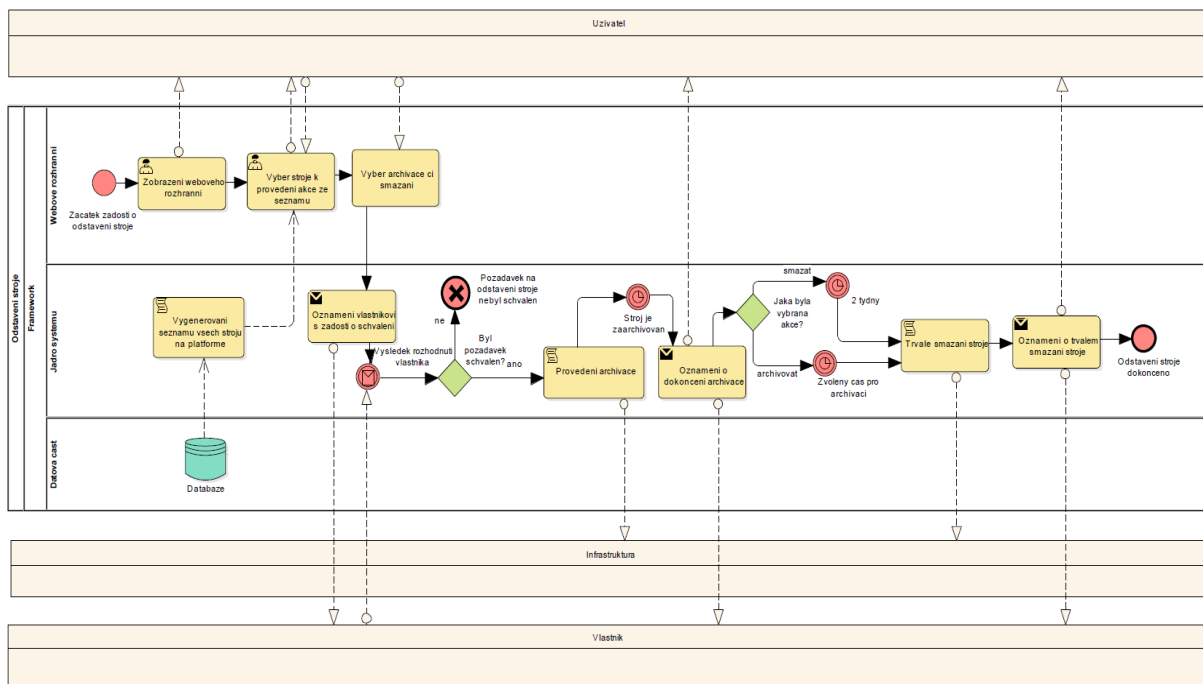
Obr. 57: Proces přejmenování / změna vlastníka - Optimalizováno

3.5.4 Odstavení stroje

Automatizace umožnila implementaci tzv. „bezpečnostního mechanismu“, který zabezpečuje, že v případě jakékoliv žádosti o smazání stroje dojde nejdříve k archivaci daného serveru a nejdříve po uplynutí doby dvou týdnů se zvolený virtuální stroj smaže bez možnosti obnovy. V případě archivace framework automaticky stroj vypne a přesune jej do archivního úložiště, ve kterém je uložen po dobu, kterou zvolil vlastník daného VM.

Nový proces na Obr. 58 zcela vypouští nutnost zapojit administrátora do jakékoliv části odstavení. Uživatel si sám vybere konkrétní VM, žádost je opět schválena vlastníkem a v případě souhlasu se framework sám postará o veškeré přesuny a emailová oznámení.

V době, kdy je stroj přesunut do archivního úložiště jej může administrátor ručně obnovit. Zvolená funkcionality nebyla záměrně zanesena jako součást procesu odstavení, jelikož tento požadavek se vyskytuje méně než jednou ročně a jedná se o identické akce, jako v procesu „Klonování stroje“ s vytvořením nové instance.



Obr. 58: Odstavení stroje - Optimalizováno

3.5.5 Klonování stroje

Druhý nejčastější proces, který v optimalizované podobě nepředstavuje téměř žádný strávený čas pro administrátora platformy. Uživatel ve webovém rozhraní vybere zdrojový server a typ klonování (vytvoření nové instance, či přepis). V případě přepisu se framework

dotáže uživatele, zda chce současné VM archivovat, či jej trvale smazat, čímž je akce jednoznačně definována.

Pokud se jedná o přepis stroje, je automaticky zaslán email s dotazem vlastníkovi cílového stroje, zda souhlasí s přepisem. V případě vytvoření nové instance stroje se žádá o schválení využití nových výpočetních prostředků administrátora infrastruktury.

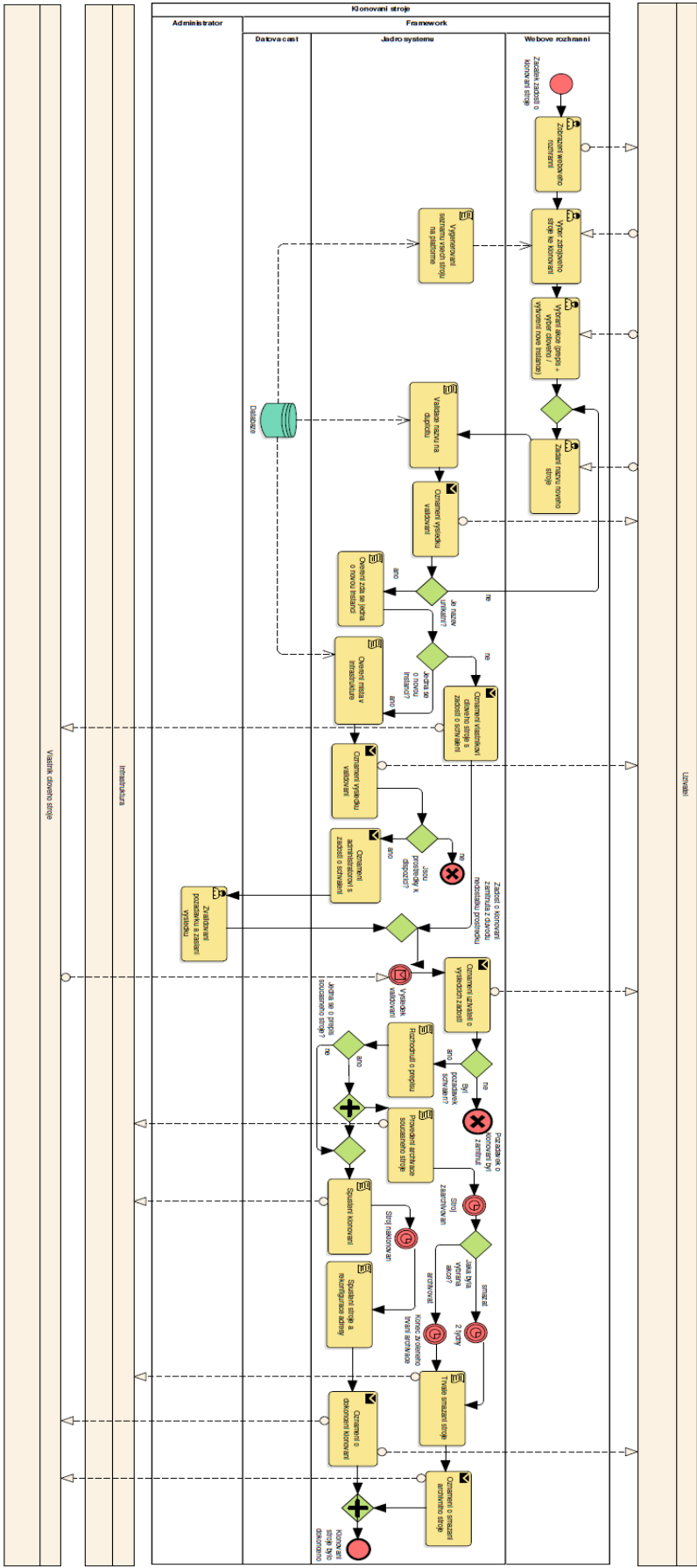
Při vytvoření nové instance je zaslán email administrátorovi, aby odsouhlasil vytvoření nového stroje. Po schválení systém automaticky naklonuje dané VM, včetně následné rekonfigurace IP adresy. Dokončení je automaticky oznámeno emailem vlastníkovi a uživateli. Pro lepší pochopení je celý proces je zobrazen na Obr. 59.

3.5.6 Změna výpočetních prostředků

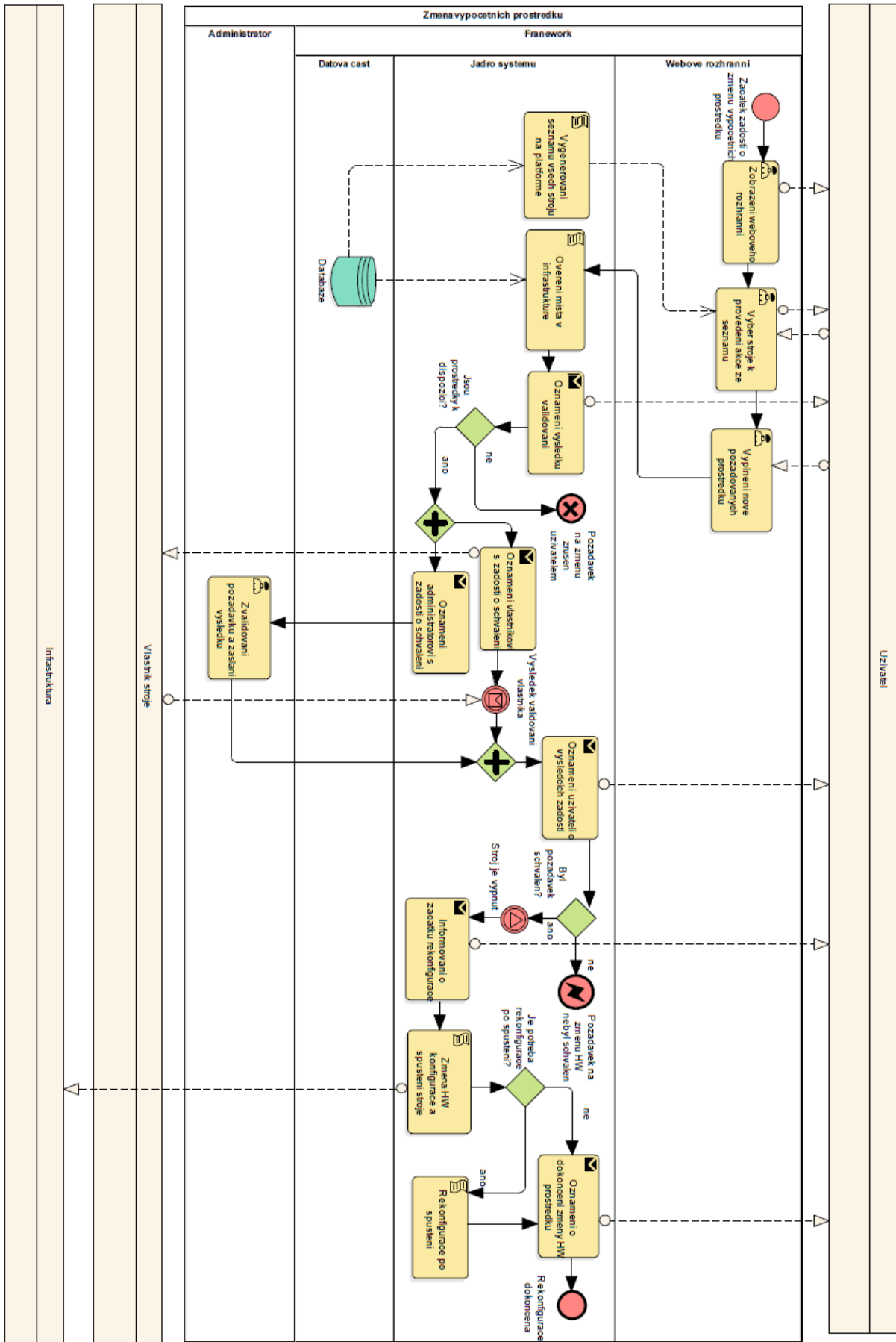
Díky využití automatizačního frameworku napojeného na API virtualizační farmy je možnost veškeré změny v přiděleném hardwaru serverů provádět za pomoci skriptů. Uživatel platformy zadá nově požadované velikosti výpočetních prostředků pro konkrétně vybraný stroj a framework, pokud jsou prostředky k dispozici, zašle email administrátorovi i vlastníkovi, s dotazy na schválení nových prostředků.

Jakmile jsou obě části paralelního procesu potvrzeny, vlastník obdrží email s výsledkem rozhodnutí a systém čeká, až bude dané VM vypnuté. Po detekování vypnutého vypnutý stroje, oznámí framework začátek práce se strojem a dojde k automatické rekonfiguraci i následnému zapnutí stroje.

Po spuštění jsou automatizovány všechny akce, které je třeba provést na spuštěném stroji a dokončení rekonfigurace je opět notifikováno emailovým oznámením uživateli, který o změnu požádal. Tato radikální změna v administraci platformy opět výrazným způsobem eliminuje zapojení administrátora, jak je vidět na Obr. 60, do běžných úkonů spojených se správou infrastruktury.



Obr. 59: Klonování stroje – Optimalizováno



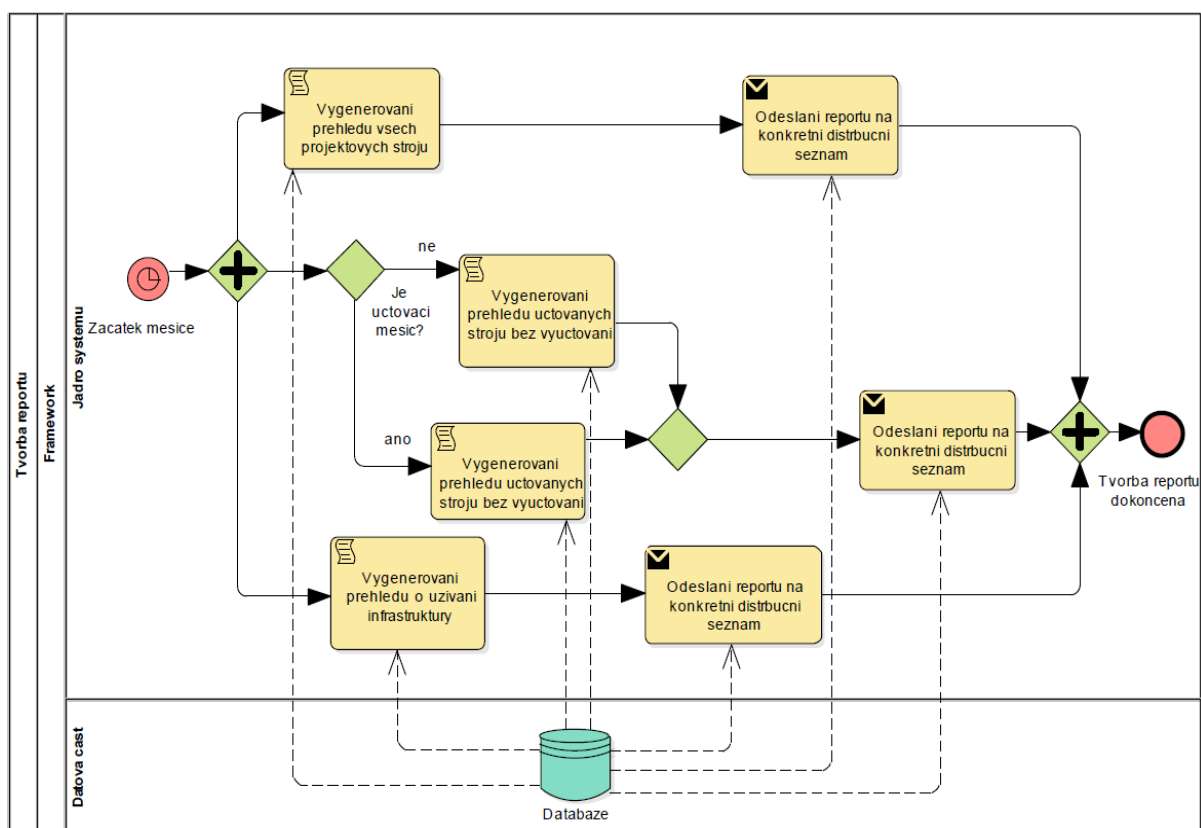
Obr. 60: Změna výpočetních prostředků - Optimalizováno

3.5.7 Generování informací o stavu virtuální farmy a účtování

Vyčítání všech informací o strojích a jejich uložení do jednotné databáze, které probíhá pomocí naplánované úlohy zcela automatizovanou formou v pravidelných intervalech, umožňuje zcela bezchybnou inventarizaci bez nutnosti zapojení administrátora. Automatizace zcela odstranila nutnost jakéhokoliv manuálního vpisování informací do databáze po provedení jakékoliv změny v infrastruktuře. Zároveň také přinesla způsob, jakým lze reporty generovat bez zásahu administrátora.

Pomocí skriptu jsou v měsíčních intervalech vygenerovány všechny tři emaily, obsahující seznam spuštěných strojů, stejně jako přehled s vyúčtováním, či report s využitím virtualizační farmy. Reporty jsou vytvořeny bez jakéhokoliv zásahu administrátora a zároveň bez možnosti zanesení jakékoliv chyby, či nutnosti kontaktovat infrastrukturu, jak je zobrazeno na Obr. 61.

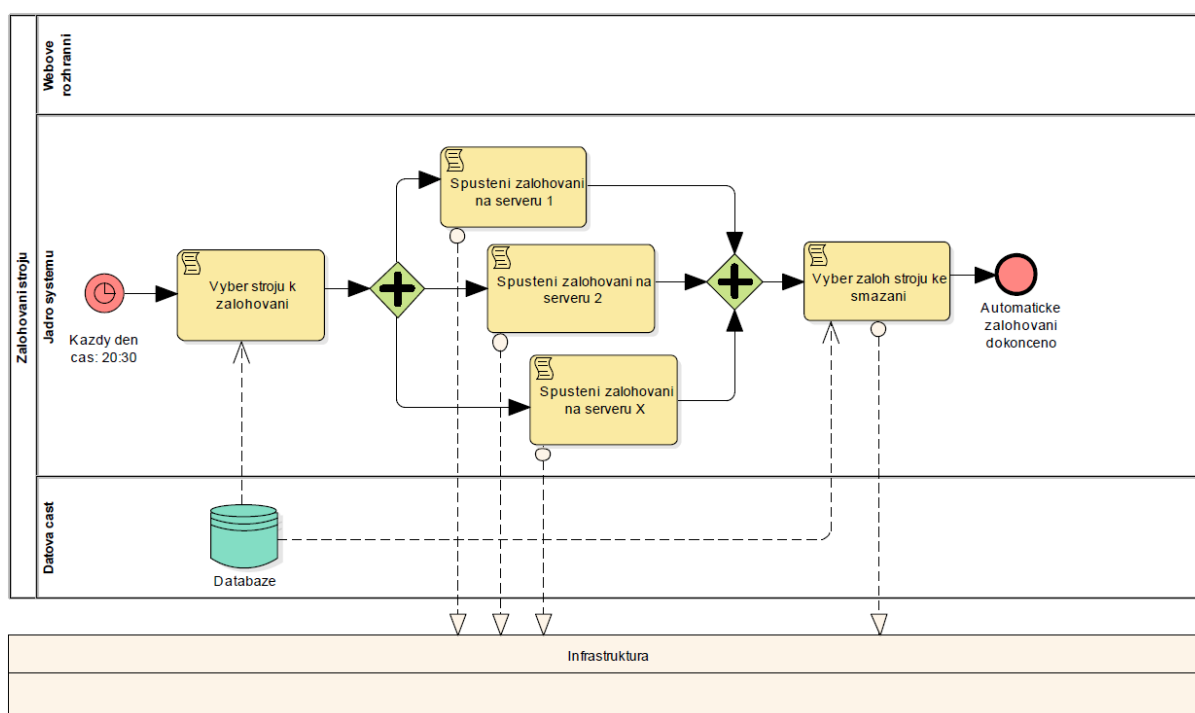
Proces samotný pouze vybírá zvolené informace z databáze za pomoci SQL dotazů, jejichž výsledky jsou za použití správného formátování vloženy do těla emailu a odeslány na předem zadané adresy.



Obr. 61: Generování informací o stavu virtuální infrastruktury a účtování - Optimalizováno

3.5.8 Automatické zálohování

Druhým nově vytvořeným procesem na Obr. 62, který nebylo možné snadno bez frameworku nasadit, je automatické zálohování. Proces probíhá zcela autonomně a jeho cílem je pro každý virtuální stroj vytvořit zálohu. Framework zajišťuje, aby se každé dva týdny vytvořil klon pro každý konkrétní stroj, který se následně uchovává po dobu jednoho měsíce. Díky automatizované komunikaci s farmou i centralizovanému seznamu virtuálních strojů je možné snadno určit jednoduchým SQL dotazem, které servery je třeba zálohovat, i jaké dříve vytvořené zálohy je možno smazat.



Obr. 62: Automatické zálohování - Optimalizováno

Jelikož se při tomto „automatickém klonování“ vytváří požadavky pouze na diskové úložiště, které je speciálně pro zálohy vytvořeno v dostatečné výši, není třeba žádného dalšího schvalování a administrátor v rámci monitorovacího frameworku pouze kontroluje dostupnou kapacitu záložního úložiště.

Obnovy z těchto záloh probíhají manuálně, velmi zřídka a vždy v součinnosti s administrátorem platformy, převážně na úrovni jednotlivých souborů. Z toho důvodu funkcionalita obnovy není, nijak procesně automatizována, a ve webovém rozhraní na Obr. 63 je pouze zobrazen přehled záloh daného stroje.

The screenshot displays a web interface for VM management. At the top, there is a navigation bar with icons for Home, Requests, VM list, SSH key, Statistics, and Search. Below this, the VM's state is shown as "VM is in running state". Resource allocation is detailed as follows:

- Allocated vCPU: 2 threads
- Allocated RAM: 8192 MB
- Allocated storage: 111 GB

The "Connectivity" section includes:

- Console: [https://\[redacted\]/vsphere-client](https://[redacted]/vsphere-client)
- ESXi host: [redacted]
- SSH access status: OK
- Root password: Send to email @
- root SSH keys: Additional SSH keys for the root account (Save button)
- Monitoring data: Nagios page

The "Backups" section shows two backup entries:

- Backup [redacted] created 2018-01-02 21:33:14
- Backup [redacted] created 2017-12-18 21:26:56

Obr. 63: Webové rozhraní frameworku obsahující informace o daném stroji

3.6 Požadavky na framework

Na základě analýzy stávajících procesů a návrhu nových bylo identifikováno, že některé části činností jsou velmi podobné. Dále bylo zjištěno, že nejvíce času bylo spotřebováváno na správnou identifikaci požadavku a poskytnutí korektních vstupů. Poslední společnou doménou všech procesů je opakování stále stejných činností, které jen zabírají čas, a v případě manuálního vykonávání prodlužují dobu nutnou k nasazení a zároveň zvyšují riziko zanesení chyby.

Nově vytvořený framework obsahuje následující případy užití jednotlivých aktérů. Libovolný uživatel může požádat o všechny procesy výše zkoumané v rámci této práce. Vlastník stroje potvrzuje přejmenování v případě, že se mění vlastník, dále odstavení systému a změnu výpočetních prostředků. Administrátorova funkce je schvalování nově využívaných hardwarových prostředků při změně a vytváření nového stroje. Naopak administrátor po nasazení automatizačního frameworku ztratil zcela kontrolu nad přejmenováváním, odstavením i generováním reportů.

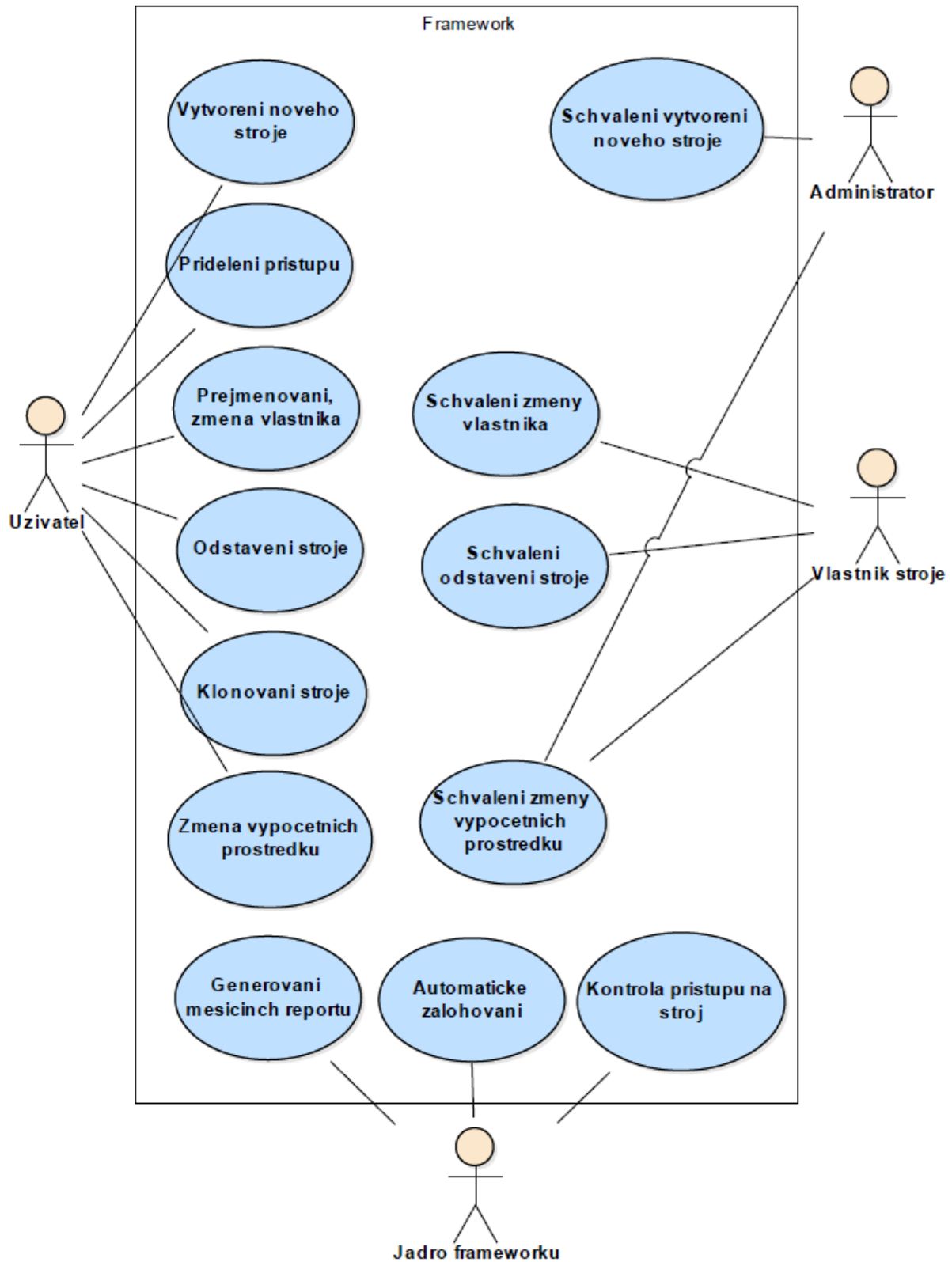
V případech užití záměrně není zaneseno schválení klonování, jelikož tento případ je zahrnut pro administrátora v rámci schválení vytvoření nového stroje a pro vlastníka stroje v rámci schválení odstavení stroje.

Posledním účastníkem je framework samotný, který kromě participace na všech procesech i navíc pravidelně provádí inventarizaci farmy do databáze, kontrolu přístupů na stroje a tvorbu měsíčních reportů. Pro ucelený přehled jsou na Obr. 64 zobrazeny případy užití v rámci vytvořeného automatizačního frameworku.

Další případ užití, který není v práci zpracován, ale díky centralizované databázi i unifikovanému přístupu ke správě virtuální infrastruktury je již implementován, je napojení dalšího automatizační nástroje.

V současné době již existuje napojení na automatizační nástroj Ansible [39], který slouží k instalaci softwarových produktů vyvinutých firmou Gemalto bez nutnosti jakékoliv interakce uživatele. Napojení již bylo úspěšně ověřeno a v druhé polovině roku 2017 vzniklo okolo dvaceti zcela automatizovaně nainstalovaných strojů, díky čemuž byl zkrácen čas nasazení z 5 MD na 1 hodinu pro každou instalaci stroje.

Zdrojové kódy implementovaného frameworku jsou součástí této práce na příloženém DVD médiu.



Obr. 64: Případy užití v automatizačním frameworku

3.7 Analýza nákladů pro vytvoření frameworku

V následující kapitole je popsána analýza nákladů pro všechny původní a zároveň i optimalizované procesy, která byla provedena před nasazením frameworku a která sloužila jako podklad pro rozhodnutí managementu, zda bude výhodné danou optimalizaci implementovat, či zda ponechat procesy v původní podobě. V prvním kroku byly identifikovány časové náročnosti jednotlivých procesů.

3.7.1 Časová náročnost původních procesů

Není možné zcela přesně určit časové náročnosti jednotlivých úkonů, jelikož ve většině případů zahrnuje určení časové náročnosti i schopnost uživatele daného stroje konkrétně specifikovat požadavky. Pro určení časových náročností byl proto vybrán časový modus daných procesů, určený na základě subjektivních zkušeností administrátorů platformy. Níže uvedené časy odpovídají času, který musel administrátor strávit na jednotlivém požadavku a nezapočítává se tak čas patřící uživateli či vlastníkovi.

3.7.1.1 Klonování stroje

Nejkomplexnější akcí, ze které bylo možno odvodit časy i pro další procesy bylo klonování, jelikož zahrnuje částečně proces vytvoření nového stroje a zcela i proces odstavení. Prvním krokem je vyhledání stroje. Tento úkon, vzhledem k nutnosti manuálního výběru informací z excelového seznamu a porovnání s reálnými údaji na platformě trval tři minuty v případě jedné žádosti o upřesňující informace. Druh klonování, stejně jako výběr typu archivace, nebyl určen téměř v žádné žádosti, a proto se vždy administrátor dodatečně dotazoval, což mu zabralo jednu minutu pro každou akci.

Rozdíl mezi archivací a mazáním byl z pohledu času rozdílný. Na smazání stroje se spotřebovaly tři minuty, oproti patnácti minutám potřebným na archivaci. V druhém případě se navíc jednalo o akce, které mezi sebou měly odstup přibližně čtyř hodin a administrátor musel průběžně kontrolovat stav archivace a stále se připojovat do infrastruktury. Z množství všech požadavků bylo identifikováno zastoupení obou dvou procesů ve stejné četnosti, a tak bylo v rámci analýzy počítáno s průměrem těchto dvou hodnot, tedy devět minut. Následující úprava záznamu v Excelovém seznamu byla ohodnocena standardizovanými dvěma minutami pro většinu všech změn. Oproti tomu dotazování se na jméno ve správném formátu průměrně zabíralo okolo tří minut.

Časově velmi nevyzpytatelným úkonem bylo ověření místa a optimalizace rozložení jednotlivých strojů. Na základě obsazenosti infrastruktury bylo možné na těchto aktivitách

strávit i patnáct minut, ale zároveň se vyskytovaly i případy, kdy byl krok zcela přeskokován. Vzhledem ke zkušenostem administrátor ohodnotil náročnost na jedenáct minut.

Spuštění klonování včetně vyplnění nového jména a výběru konkrétních hardwarových prostředků v infrastruktuře trvalo pět minut. Rekonfigurace systému po spuštění ve většině případů trvala dvanáct minut. Zaslání oznámení s informacemi o stroji bylo obdobně náročné pro všechny procesy a znamenalo tři minuty administrátorova času. Celkový čas běžného procesu následně činil padesát minut.

3.7.1.2 Vytvoření nového stroje

Většina jednotlivých akcí byla popsána ve výše uvedené kapitole klonování stroje. Chybějícími úkony se stalo ověření hardwarové specifikace, což byl snadný úkol na jednu minutu. Oproti tomu vytvoření hardwarové instance v infrastruktuře trvalo sedm minut administrátorova času. Navazujícím, doposud nezmapovaným krokem, bylo natavení konfiguračních skriptů pro instalaci operačního systému. Tento manuální proces i v případě předpřipravených šablon trval administrátorovi průměrně osmnáct minut. Ověření instalace i s připočtením času na nápravu trvalo okolo osmi minut. Po připočtení všech již zmapovaných kroků se výsledný čas procesu vyšplhal na padesát tři minut.

3.7.1.3 Přejmenování stroje

V rámci tohoto procesu se zanedbávala aktivita pro rozlišení, zda se mění vlastník, jelikož se jednalo o úkon v rámci vteřin. Vyhledání jména v adresáři bylo již zahrnuto do aktivity vytvoření nového jména. Na samotný úkon přejmenování stroje v infrastruktuře byly vyčleněny dvě minuty. V součtu proces přejmenování trval deset minut.

3.7.1.4 Odstavení stroje

Všechny části celého odstavení byly popsány v rámci klonování. Poslední krok - zaslání oznámení - bylo možno díky časové nenáročnosti ignorovat. Celkem odstavení zabralo patnáct minut.

3.7.1.5 Změna výpočetních prostředků

Po projití již známými částmi procesu bylo potřeba vyjednat čas vypnutí. Určení času, ve kterém se s platformou nepracuje a zároveň je k dispozici administrátor, bylo odhadnuto na průměrných deset minut. Samotná změna zabrala pouhé dvě minuty, ale v případě následné rekonfigurace disků na běžícím systému bylo třeba připočítat dalších devět minut. Celkem tedy třicet sedm minut.

3.7.1.6 Generování informací o stavu virtuální infrastruktury a účtování

Specifickým úkonem se stala tvorba měsíčních přehledů. Vzhledem k dohledávání veškerých informací a manuální tvorbě daných výstupů byla na základě zkušeností odhadnuta pravidelná náročnost na tři hodiny. Všechny časy původních procesů jsou zobrazeny v Tab. 1.

Název aktivity	Čas (min)	Klonování	Vytvoření nového stroje	Přejmenování stroje	Odstavení stroje	Změna výpočetních prostředků	Generování reportů a účtování
Vyhledání stroje s jednou žádostí o upřesnění	3	x		x	x	x	
Dotázání se na typ klonování	1	x					
Dotázání se na typ odstavení	1	x			x		
Smazání stroje	3						
Archivace stroje	15						
Smazání / archivace	9	x			x		
Úprava excelového seznamu	2	x	x	x	x	x	
Vytvoření správného jména	3	x	x	x			
Ověření místa a optimalizace rozložení	11	x	x			x	
Spuštění klonování	5	x					
Rekonfigurace stroje	12	x					
Zaslání informací o stroji	3	x	x				
Ověření hardwarové specifikace	1		x				
Vytvoření hardwarové konfigurace ve farmě	7		x				
Vytvoření konfiguračních souborů pro instalaci	18		x				
Ověření správnosti instalace	8		x				
Přejmenování stroje v infrastruktuře	2			x			
Vyjednávání o čase vypnutí	10					x	
Změna hardwarových prostředků	2					x	
Rekonfigurace běžícího systému	9					x	
Tvorba reportů	180						x
Celkové trvání		50	53	10	15	37	180

Tab. 1: Přehled časových náročností původních procesů (min)

3.7.2 Časová náročnost optimalizovaných procesů

Většina nově navržených procesů eliminuje nutnost zapojení administrátora na nezbytné minimum. Díky automatizaci se většina úkonů provádí zcela automatizovaně a administrátor pouze rozhoduje o schválení, či zamítnutí nového požadavku, což je úkol na jednu minutu. Schvalování probíhá při vytvoření nového stroje a změně výpočetních prostředků.

V případě klonování administrátor schvaluje jen v případě vytvoření nové instance stroje. Pro účely analýzy však bylo počítáno s rozhodováním při každém klonování. V Tab. 2 je zobrazeno srovnání časové náročnosti administrátora původních procesů a procesů nově vytvořených.

	Starý proces (min)	Optimalizovaný proces (min)
Vytvoření nového stroje	53	1
Přidělení přístupu	N/A	0
Přejmenování stroje	10	0
Odstavení stroje	15	0
Klonování	50	1
Změna výpočetních prostředků	37	1
Generování reportů a účtování	180	0
Automatické zálohování	N/A	0

Tab. 2: Srovnání časů původních a optimalizovaných procesů (min)

3.7.3 Vlastní náklady na zkoumané procesy

Veškeré zmíněné aktivity byly vykonávány administrátorem, který je zaměstnancem firmy Gemalto s r.o. na plný úvazek. Jelikož jsou lidé v rámci korporátu rozděleni do určitých týmů, je jedna hodina každého pracovníka přesně vypočítána a ohodnocena. Díky této skutečnosti bylo snadné určit vlastní náklady na daný proces. Interně získaná hodnota se pouze vynásobila množstvím času, které bylo nutné na daném úkolu strávit.

Při analýze nákladů na jednotlivé procesy se počítalo se skutečnou částkou, kterou je ohodnocena jedna hodina administrátora, avšak z důvodu přísné interní politiky byla pro účely této práce zvolena zcela náhodná hodnota 1200 Kč. Po přepočtu dostáváme tedy dvacet korun za jednu minutu času administrátora platformy.

Vzhledem k rozdělení firmy na týmy nijak do analýzy nevstupují časy uživatelů a vlastníků jednotlivých strojů. Na základě odhadu se však jedná o podobné, či menší množství

času, které musí ostatní aktéři na jednotlivých procesech strávit a proto je tato veličina z procesu zcela vypuštěna. Žádné další přímé náklady pro procesy nebyly známy, a proto je souhrnný přehled nákladů uveden v Tab. 3.

	Starý proces (Kč)	Optimalizovaný proces (Kč)
Vytvoření nového stroje	1060	20
Přidělení přístupu	N/A	0
Přejmenování stroje	200	0
Odstavení stroje	300	0
Klonování	1000	20
Změna výpočetních prostředků	740	20
Generování reportů a účtování	3600	0
Automatické zálohování	N/A	0

Tab. 3: Porovnání vlastních nákladů původních a optimalizovaných procesů (Kč)

Uvedené náklady na staré procesy byly vypočítány pro průměrně náročné požadavky na základě četnosti jejich výskytů v roce 2016. V případě optimalizovaných procesů se zmíněné náklady mohou pouze snížit, ale nikoliv zvýšit.

3.7.4 Odhad nákladů na implementaci jednotlivých procesů v rámci frameworku

Pro umožnění přechodu na optimalizované procesy bylo za potřebí implementovat následující funkcionality, ze kterých je automatizační framework složen:

- Databáze (Navržení schématu, implementace tabulek)
- Webové rozhraní (Grafická část všech případů užití, a příslušné generovací skripty)
- Rozhodovací logika všech případů užití a nastavovací skripty
- Skripty pro naplánované úlohy
- Napojení na mailový server včetně šifrování zpráv
- Zabezpečené napojení na firemní adresář uživatelů LDAP
- Plná implementace komunikace s virtuální farmou přes VMware API

Veškerá implementace frameworku byla cílena svými parametry na schopnosti administrátora infrastruktury, který naprogramoval veškeré potřebné komponenty. Díky

tomuto předpokladu stačilo určit časovou náročnost implementace jednotlivých akcí a následně čas vyjádřit stejným způsobem, jako tomu bylo při zkoumání vlastních nákladů jednotlivých procesů. Detailní rozpis odhadu náročnosti jednotlivých akcí je zobrazen v Tab. 4.

Funkcionalita	Náročnost (MD)
Návrh databáze a implementace schématu	2
Výběr konkrétního stroje	1
Validace názvu stroje včetně zobrazení výsledku	2
Vložení HW prostředků	1
Ověření místa v infrastruktuře a přesun strojů včetně zobrazení výsledku	3
Oznámení výsledku dané akce na email	1
Zaslání žádosti na schválení včetně linku pro odpověď	1
Vytvoření hardwarové instance v infrastruktuře	1
Automatizované vygenerování instalačních souborů	5
Spuštění instalace	1
Výběr akce, která se má se strojem provést	1
Napojení na firemní LDAP a formulář pro výběr uživatelů	2
Ověření veřejného klíče a formulář pro nahrání	1
Automatizované ověřování přístupu na strojích	3
Formulář na odstavení stroje	1
Provedení archivace	1
Smazání stroje	1
Rekonfigurace stroje po klonování	3
Změna hardwarové konfigurace vypnutého stroje	2
Změna hardwarové konfigurace po spuštění	3
Oznámení výsledku na zašifrovaný email	1
Generování informací o stavu a účtování	2
Automatické zálohování	4
Propojení všech jednotlivých částí a vytvoření komplexní logiky	5
Suma	48

Tab. 4: Odhad náročnosti implementace jednotlivých aktivit (MD)

Jak bylo zmíněno, implementace byla prováděna administrátorem infrastruktury, jehož denní sazba byla interně stanovena na devět tisíc šest set korun za člověkodenní. Celkové náklady na vytvoření frameworku se vyšplhaly na 460 800 Kč. Zároveň byl vytvořen pesimistický odhad, která počítá s možnými problémy a přidává rezervních deset procent z vypočítaných dní, tedy přibližně deset člověkodenní, díky čemuž byla výše nákladu odhadnuta na 556 800 Kč.

3.8 Zhodnocení investice

V následující kapitole je popsáno rozhodování, které předcházelo započetí implementace na začátku roku 2017. Bylo použito ukazatelů uvedených v teoretické části práce 2.5, konkrétně čisté současné hodnoty, pro jejíž výpočet se určila diskontní míra a odhad peněžních toků v rámci životnosti pěti leté investice.

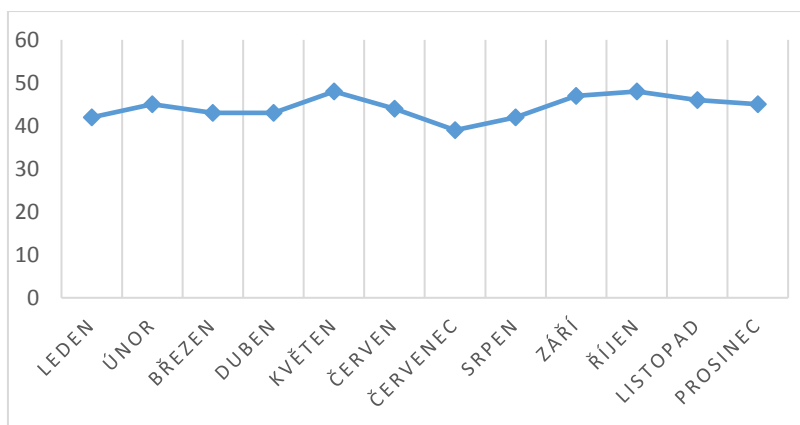
3.8.1 Diskontní míra

Určení této hodnoty je stanoveno vnitřní ekonomickou strukturou a jedná se údaj jasně definovaný pro každý tým v rámci firmy. Jelikož není povoleno vedením firmy použít přesnou hodnotu, se kterou se počítalo při rozhodování, v rámci práce byl tento ukazatel stanoven na 8 %, což je požadovaný výnos investice. Inflace byla při rozhodování na relativně nízké úrovni, stejně jako diskontní sazby určené bankami.

Z důvodu rozdělení podniku na oddělené jednotky nebylo možné použít ROE, jelikož tým neměl v minulosti žádné investice. Cizí kapitál také není nevyužíván, a proto se ke stanovení nevyužilo hodnoty WACC ale definovaného ukazatele.

3.8.2 Cash flow

Veškeré procesy, které jsou navázány na virtuální infrastrukturu, mají v pražské pobočce vlastní rozpočet a je díky tomu relativně snadné zjistit libovolný ekonomický ukazatel. Na základě pravidelných měsíčních reportů bylo možné dohledat přesné množství všech virtuálních strojů, které byly v rámci roku ve farmě průběžně spuštěny a společně s daty z excelového seznamu odhadnout množství akcí, které se v rámci jednotlivých měsíců vykonalo. Přehledné zobrazení celkového množství virtuálních strojů je na Obr. 65. Průměrné obsazení farmy bylo okolo 45 strojů, což přibližně odpovídá devadesáti procentnímu využití serverů.



Obr. 65: Množství virtuálních strojů v roce 2016

Z výše uvedených údajů bylo možné na základě množství všech strojů ve farmě za rok, vynásobených částkou 800 Kč / stroj / měsíc²¹, spočítat provozní výnosy farmy. Tento údaj zcela korespondoval s interními účetními výkazy prováděnými mezi jednotlivými týmy.

$$Výnosy_{45} = 532 * 800 = 425\ 600\ Kč$$

V rámci optimalizace bylo uvažováno o výměně serverů, které jsou pro běh veškerých strojů používány. Pro původní servery se platí roční poplatek 50 000 Kč, který zahrnuje podporu v případě poruchy. Při nákupu serverů nových by se poplatek snížil na 5 000 Kč a prvotní investice by byla 200 000 Kč. Hodnoty pro nákup serverů byly převzaty z cenové nabídky vypracované dodavatelem hardwaru pro pražskou pobočku.

Tyto nové servery by byly více výkonné, a díky tomu by se mohl zvýšit průměrný počet nasazených strojů z aktuálních 45 na 75. Vzhledem k množství požadavků o vytvoření nových strojů za uplynulá období je naplnění kapacity reálné, proto je níže v ekonomickém hodnocení zahrnuta varianta jak se zvýšením množství strojů, tak i bez nich. Elektrická energie se nijak neúčtuje a je placena z rozpočtu jiného týmu firmy. Výnosy by v případě zvýšení počtu strojů byly 720 000 Kč za rok.

$$Výnosy_{75} = 900 * 800 = 720\ 000\ Kč$$

Pro výpočet variabilních nákladů bylo za potřebí zjistit množství vykonaných úkonů. Průměrné odhadnuté počty jednotlivých úkonů, vycházely z archivní emailové komunikace a zmíněného excelového seznamu, který obsahoval i historii. Každý měsíc se v roce 2016 jednalo přibližně o množství úkonů uvedené v Tab. 5.

²¹ Částka, kterou je běh každého stroje účtován. Z důvodu firemních předpisů je opět tato hodnota v práci uvedena jako fiktivní.

Název procesu	Množství úkonů / měsíc
Vytvoření nového stroje	5
Přidělení přístupu	0
Přejmenování stroje	1
Odstavení stroje	5
Klonování	6
Změna výpočetních prostředků	4
Generování reportů a účtování	1
Automatické zálohování	0

Tab. 5: Průměrné množství analyzovaných úkonů za měsíc

K uvedeným údajům bylo ještě potřeba přičíst náklady na používaný software VMware® a Red Hat® stojící ročně 75 000 Kč.

Náročnosti uvedené v Tab. 2 se vynásobily množstvím úkonů z Tab. 5 a definovanou minutovou sazbou administrátora. Následovalo násobení dvanácti měsíci a přičtení zmíněných fixních nákladů, čímž se v Tab. 6 vypočítaly celkové náklady na administraci platformy za zkoumaný rok pro původní i optimalizované procesy.

Přidělení přístupu i automatické zálohování bylo při výpočtu vlastních nákladů zcela vypuštěno, jelikož v původních procesech tyto případy užití nevystupovaly a v rámci nových procesů nevytváří žádnou časovou náročnost pro administrátora platformy. Náklady v případě zvýšení počtu strojů představují minimální rozdíl, a proto jsou v rámci zhodnocení zanedbány.

Název procesu	Úkonů / měsíc	Původně (Kč)	Optimalizovaně (Kč)	Optimalizovaně s nákupem HW (Kč)
Vytvoření nového stroje	5	5,300	100	100
Přidělení přístupu	0	0	0	0
Přejmenování stroje	1	200	0	0
Odstavení stroje	5	1,500	0	0
Klonování	6	6,000	120	120
Změna výpočetních prostředků	4	2,960	80	80
Generování reportů a účtování	1	3,600	0	0
Automatické zálohování	0	0	0	0
Měsíčně za administrátora		19,560	300	300
Softwarové licence	-	75,000	75,000	75,000
HW podpora	-	50,000	50,000	5,000
Celkové vlastní náklady / rok		359,720	128,600	83,600

Tab. 6: Vlastní náklady na původní i optimalizované procesy (Kč)

Na základě vypočítaných údajů již bylo možné spočítat provozní cash flow pro všechny varianty viz Tab. 7.

CF	Částka (Kč)
Původní procesy	65,880
CF optimalizovaného procesu bez nákupu (BZ) serverů	297,000
CF optimalizovaného procesu s nákupem a zvýšením (SZ) počtu serverů	636,400
CF optimalizovaného procesu s nákupem a bez zvýšení (BZ) počtu serverů	342,000

Tab. 7: CF jednotlivých variant

3.8.3 Čistá současná hodnota

Posledním krokem pro rozhodnutí, zda implementovat automatizační framework byl výpočet čisté současné hodnoty pro všechny varianty. Definice vztahu byla uvedena v 2.5.2.1. Níže v Tab. 8 jsou uvedeny přehledně vstupy potřebné pro výpočet, které společně s Tab. 7 tvoří veškeré potřebné vstupy.

Údaj	Hodnota
IN optimistická (OP) bez nákupu (BN) serverů	460,800
IN optimistická (OP) s nákupem (SN) serverů	660,800
IN pesimistická (PE) bez nákupu (BN) serverů	556,800
IN pesimistická (PE) s nákupem (SN) serverů	756,800
t	4 roky
r	0.08

Tab. 8: Souhrnné vstupy pro výpočet NPV investice

Následně jsou uvedeny výpočty variant optimalizace s nákupem serverů i bez nich společně se scénářem současného počtu strojů i zvýšeného v optimistické i pesimistické verzi.

$$NPV_{OP\ BN\ BZ} = -460\ 800 + \frac{-163\ 800}{(1+0,08)^1} + \frac{133\ 200}{(1+0,08)^2} + \frac{430\ 200}{(1+0,08)^3} + \frac{727\ 200}{(1+0,08)^4} = 377\ 751\ K\check{c}$$

$$NPV_{OP\ SN\ SZ} = -660\ 800 + \frac{-24\ 400}{(1+0,08)^1} + \frac{612\ 000}{(1+0,08)^2} + \frac{1\ 248\ 400}{(1+0,08)^3} + \frac{1\ 884\ 800}{(1+0,08)^4} = 2\ 217\ 703\ K\check{c}$$

$$NPV_{OP\ SN\ BZ} = -660\ 800 + \frac{-363\ 800}{(1+0,08)^1} + \frac{-66\ 800}{(1+0,08)^2} + \frac{230\ 200}{(1+0,08)^3} + \frac{527\ 200}{(1+0,08)^4} = -484\ 674\ K\check{c}$$

$$NPV_{PE\ BN\ BZ} = -556\ 800 + \frac{-259\ 800}{(1+0,08)^1} + \frac{37\ 200}{(1+0,08)^2} + \frac{334\ 200}{(1+0,08)^3} + \frac{631\ 200}{(1+0,08)^4} = -36\ 213\ K\check{c}$$

$$NPV_{PE\ SN\ SZ} = -756\ 800 + \frac{-120\ 400}{(1+0,08)^1} + \frac{516\ 000}{(1+0,08)^2} + \frac{1\ 152\ 400}{(1+0,08)^3} + \frac{1\ 788\ 800}{(1+0,08)^4} = 1\ 803\ 739\ K\check{c}$$

$$NPV_{PE\ SN\ BZ} = -756\ 800 + \frac{-414\ 800}{(1+0,08)^1} + \frac{-72\ 800}{(1+0,08)^2} + \frac{269\ 200}{(1+0,08)^3} + \frac{611\ 200}{(1+0,08)^4} = -540\ 338\ K\check{c}$$

Čistá současná hodnota vyšla kladně i záporně pro různé varianty. V případě, optimistické varianty vyšla nevýhodně pouze v případě nákupu nových serverů, bez zvýšení nárůstu strojů. V pesimistické verzi byla těsně pod hranicí přijatelnosti pro verzi bez nákupu

serverů a opět velice nevhodně pro variantu s nákupem bez zvýšení počtu strojů. Nebylo tedy jasné, zda se pro investici rozhodnout či nikoliv.

Druhým použitým ukazatelem bylo vnitřní výnosové procento. Bylo počítáno pouze pro varianty s vhodným výsledkem NPV

$$IRR_{OP\ BN\ BZ} = 26\ \%$$

$$IRR_{OP\ SN\ SZ} = 72\ \%$$

$$IRR_{PE\ BN\ BZ} = -3\ \%$$

$$IRR_{PE\ SN\ SZ} = 57\ \%$$

Stejně jako čistá současná hodnota ukázalo vnitřní výnosové procento nevhodnost investice v pesimistické variantě bez nákupu serverů.

Rozhodnutí managementu bylo ovlivněno vírou v to, že bude v případě navýšení kapacity i úměrně navýšeno množství spuštěných virtuálních strojů v infrastruktuře. Zároveň byl důvodem k implementaci fakt, že v případě automatizace budou k dispozici automatické zálohy, které zásadně zvyšují bezpečnost jednotlivých strojů ve farmě a snižují možnou finanční ztrátu v případě nechtěného smazání dat, či stroje. Díky tomu je přijatelný i pesimistický případ, při kterém je IRR mínus tři procenta.

Implementace frameworku proběhla na počátku roku 2017 a náklady na vytvoření byly nižší, než uváděla optimistická varianta. Důvodem pro nižší implementační náklady byla implementace rozhraní ve volném čase administrátora a v rámci tvorby této práce, díky čemuž se nejednalo o práci v rámci standardní pracovní doby administrátora a proto byla část implementace provedena zadarmo.

4 Závěr

Cílem práce bylo provést analýzu a optimalizaci business procesů ve firmě Gemalto s.r.o. V první části práce byly vysvětleny teoretické pojmy, které je nutné znát pro orientaci v problematice. Bylo definováno, co znamená podnikový proces, jaká je návaznost na UML a dále bylo provedeno propojení znalostí s BPMN.

Zároveň se diskutovala problematika hodnocení investic včetně letmého popisu technických částí použitých uvnitř frameworku. Na základě teoretických znalostí byly stanoveny všechny základy potřebné pro analýzu, vyhodnocení a optimalizaci procesů.

V druhé části práce byl proveden popis stávajících procesů, včetně identifikování míst, ve kterých je možné provést zlepšení. Následoval návrh nových procesů s použitím automatizačního frameworku včetně vypočtení ekonomických ukazatelů, které hrály klíčovou roli při rozhodování managementu o implementaci dané optimalizace.

Na základě akceptování výsledků těchto ukazatelů byla provedena implementace automatizačního frameworku včetně nákupu nových serverů umožňujících rozšíření kapacity celé infrastruktury a optimalizaci všech zmíněných procesů. Automatizace umožnila kromě snížení času, který musí trávit administrátor na jednotlivých úkolech také snížení fixních nákladů na provoz serverové farmy.

Investice byla naplánovaná na dobu čtyř let a měla by mít při subjektivně stanovené diskontní míře 8% výnosnost přes dva miliony korun. Vzhledem k tomu, že optimalizace i investice byla provedena na začátku roku 2017, bylo možné za uplynulý rok potvrdit pravdivost vypočítaných odhadů, které přibližně korespondují s vypočítanými údaji pro optimistický odhad včetně zvýšení naplnění infrastruktury.

Disproporce mezi odhadovanými náklady a reálnou cenou za implementaci vznikla kvůli zmíněnému provedení části integrace frameworku ve volném čase administrátorů infrastruktury, s čímž se při vytváření odhadů nepočítalo. Dobrovolná iniciativa pracovníků však umožnila snížení nákladů a díky tomu urychlení návratnosti investice.

Implementovaný framework značně rozšířil možnosti využívání farmy a kromě implementace procesu automatického zálohování je rozšiřován o další funkčnosti, jako je například automatická instalace programů na jednotlivé servery, nebo automatická kontrola funkčnosti severů včetně implementace automatizovaných aktualizací pro všechny běžící stroje v infrastruktuře.

Pro rok 2018 se předpokládá kromě zmíněných napojení na automatizační framework dále kompletní automatizace monitorování funkčnosti nejen jednotlivých serverů, ale také programů na nich běžících.

Dále bude framework rozšířen o automatickou kontrolu využívání stroje a v případě nevyužívání po dobu delší než dva měsíce, bude automaticky vlastníkově zasláno oznámení, zda může být daný stroj archivován, jelikož na něm nebyla zaznamenána po jistou dobu žádná aktivita.

5 Citovaná literatura

- [1] D. Kirk, „How Much Is Intelligent Automation Saving You?“, [Online]. Available: <https://www.forbes.com/sites/kpmg/2017/09/21/how-much-is-intelligent-automation-saving-you/#73ceef7604c>. [Přístup získán 02 10 2017].
- [2] C. Prof. Ing. Václav Řepa, Procesně řízená organizace, Praha: Grada Publishing, a.s., 2012.
- [3] J. V. a. kol., Management - Základy, moderní manažerské přístupy, výkonnost a prosperita, Praha: Management press, 2009.
- [4] W. A. R. Robert C. Ford, „Cross-Functional Structures: A Review and Integration of Matrix Organization and Project Management“, 1992.
- [5] „NÁLEŽITOSTI PROJEKTOVÉHO PŘÍSTUPU“, [Online]. Available: https://is.muni.cz/do/econ/soubory/studium/topsec/NALEZITOSTI_PROJEKTOVEH_O_PRISTUPU.pdf. [Přístup získán 24 08 2017].
- [6] A. Hofer, „Procesní přístup k řízení organizací a jeho podpora“, Praha, 2006.
- [7] A. S. Peter O'Neill, „Business Process Reengineering A review of recent literature“, 1999.
- [8] „ITIL - Procesní řízení“, [Online]. Available: <http://www.ital.cz/index.php?id=914>. [Přístup získán 31 03 2017].
- [9] J. Mlejnek, „Procesní řízení - přednáška“, Praha, 2011.
- [10] D. J. Arlow, „OO Analysis and Design with UML 2 and UP“, Zurich, 2015.
- [11] „OOSE development“, [Online]. Available: <http://cgi.omg.org/news/pr97/umlprimer.html>. [Přístup získán 02 05 2017].
- [12] „History of UML“, [Online]. Available: http://www.cs.pomona.edu/classes/cs121/supp/UML%20tutorial/history_of_uml.htm. [Přístup získán 08 09 2017].

- [13] „History of UML: Methods and Notations,“ [Online]. Available: <https://sourcemaking.com/uml/basic-principles-and-background/history-of-uml-methods-and-notations>. [Přístup získán 22 11 2017].
- [14] „About the Unified Modeling Language Specification Version 2.5,“ [Online]. Available: <http://www.omg.org/spec/UML/About-UML/>. [Přístup získán 18 03 2017].
- [15] [Online]. Available: http://mpavus.wz.cz/uml/img/class_1.gif. [Přístup získán 09 02 2017].
- [16] „Příklady použití diagramů UML 2.0,“ [Online]. Available: <http://uml.czweb.org/>. [Přístup získán 6 10 2017].
- [17] [Online]. Available: <https://is.mendelu.cz/eknihovna/opory/download.pl?objekt=17502>. [Přístup získán 25 05 2017].
- [18] „Introduction to BPMN Part I,“ [Online]. Available: <https://www.visual-paradigm.com/tutorials/bpmn1.jsp>. [Přístup získán 14 04 2017].
- [19] „Introduction to BPMN - Stephen A. White,“ IBM Corporation, [Online]. Available: http://yoann.nogues.free.fr/IMG/pdf/07-04_WP_Intro_to_BPMN_-_White-2.pdf. [Přístup získán 12 04 2017].
- [20] A. T. Michele Chinosi, „Computer Standards & Interfaces,“ Italy, 2012.
- [21] J. R. Martin Owen, „BPMN and Business Process Management: Introduction to the New Business Process Modeling Standard,“ Popkin Software, New York, 2003.
- [22] „Enterprise Architect,“ [Online]. Available: <http://sparxsystems.com/>. [Přístup získán 28 02 2017].
- [23] S. Hana, Ekonomické a finanční řízení pro neekonomy, 2., aktualizované a rozšířené vydání, Praha: Grada, 2012.
- [24] „History of PHP,“ [Online]. Available: <http://php.net/manual/en/history.php.php>. [Přístup získán 15 02 2017].

- [25] L. T. Luke Welling, *Mistrovství - PHP a MySQL*, COMPUTER PRESS, 2017.
- [26] „Historical trends in the usage of server-side programming languages for websites,“ [Online]. Available: https://w3techs.com/technologies/history_overview/programming_language. [Přístup získán 2 08 2017].
- [27] *HTML SourceBook*, New York: John Wiley & Sons, Inc., 1995.
- [28] P. A. Zandbergen, *Python Scripting for ArcGIS*, 2013.
- [29] „VMware vSphere Automation SDK for Python,“ [Online]. Available: <https://code.vmware.com/web/sdk/65/vsphere-automation-python>. [Přístup získán 26 01 2017].
- [30] Gemalto, „Financial review,“ 22 11 2017. [Online]. Available: http://fr.sitestat.com/gemalto/gemalto/s?corp-ar-2016-fin-review&ns_type=pdf.
- [31] „Thales and Gemalto venture,“ 17 12 2017. [Online]. Available: <https://www.gemalto.com/press-site/Documents/2017/2017-12-17-Thales-Gemalto-acquisition-announcement-press-release.pdf>.
- [32] „Gemalto logo,“ [Online]. Available: https://www.gemalto.com/_catalogs/masterpage/Gemalto/assets/logo-gemalto-340x120.gif. [Přístup získán 03 08 2017].
- [33] „VMware datacenter,“ [Online]. Available: <https://www.vmware.com/products/datacenter-virtualization.html>. [Přístup získán 06 08 2017].
- [34] „Understanding virtualization,“ [Online]. Available: <https://www.redhat.com/en/topics/virtualization>. [Přístup získán 30 06 2017].
- [35] „MySQL,“ [Online]. Available: <https://www.mysql.com/>. [Přístup získán 04 10 2017].
- [36] „PHP,“ 23 10 2016. [Online]. Available: <http://php.net/>.
- [37] „26.2. How Do You Perform a Kickstart Installation?,“ [Online]. Available: <https://access.redhat.com/documentation/en->

- us/red_hat_enterprise_linux/7/html/installation_guide/sect-kickstart-howto. [Přístup získán 06 10 2017].
- [38] „Public Key Cryptography,“ [Online]. Available: https://www.tutorialspoint.com/cryptography/public_key_encryption.htm. [Přístup získán 15 05 2017].
- [39] „Ansible,“ [Online]. Available: <https://www.ansible.com/>. [Přístup získán 01 03 2017].
- [40] „What is a framework,“ [Online]. Available: <https://alistapart.com/article/frameworksfordesigners#section1>. [Přístup získán 26 02 2017].
- [41] „Cisco IP addressing,“ [Online]. Available: <https://www.cisco.com/c/en/us/support/docs/ip/routing-information-protocol-rip/13788-3.html>. [Přístup získán 16 08 2016].
- [42] „MuleSoft API description,“ [Online]. Available: <https://www.mulesoft.com/resources/api/what-is-an-api>. [Přístup získán 8 02 2017].
- [43] „w3schools - PHP,“ [Online]. Available: <https://www.w3schools.com/php/>. [Přístup získán 25 01 2017].
- [44] „Class diagram - diagram tříd,“ [Online]. Available: <http://mpavus.wz.cz/uml/uml-class-5.php>. [Přístup získán 16 09 2017].

6 Seznam obrázků

Obr. 1: Přehled UML diagramů	12
Obr. 2: Příklad diagramu tříd [15]	13
Obr. 3: Příklad diagramu komponent.....	13
Obr. 4: Příklad objektového diagramu [16]	14
Obr. 5: Příklad diagramu nasazení [16]	15
Obr. 6: Popis prvků použitých v diagramu aktivit [16]	16
Obr. 7: Příklad diagramu aktivit [16].....	16
Obr. 8: Příklad diagramu případů užití	17
Obr. 9: Příklad stavového diagramu [16].....	18
Obr. 10: Příklad sekvenčního diagramu [17].....	19
Obr. 11: Vzory grafického znázornění jednotlivých aktivit	21
Obr. 12: Grafické znázornění abstraktní aktivity.....	21
Obr. 13: Grafické znázornění uživatelské aktivity	21
Obr. 14: Grafické znázornění skriptu.....	22
Obr. 15: Grafické znázornění služby	22
Obr. 16: Grafické znázornění business pravidla	22
Obr. 17: Grafické znázornění manuální aktivity.....	22
Obr. 18: Grafické znázornění aktivity odeslat	23
Obr. 19: Grafické znázornění aktivity přijmout.....	23
Obr. 20: Grafické znázornění většího počtu instancí.....	23
Obr. 21: Grafické příklady různých událostí	24
Obr. 22: Příklad události připojené na aktivitě	24
Obr. 23: Grafické zobrazení nspecifikované události	24
Obr. 24: Grafické zobrazení zrušení události	25
Obr. 25: Grafické zobrazení kompenzační události.....	25
Obr. 26: Grafické zobrazení události podmínky.....	25
Obr. 27: Grafické zobrazení události eskalace	25
Obr. 28: Grafické zobrazení chyby	26
Obr. 29: Grafické zobrazení linku	26
Obr. 30: Grafické zobrazení události zprávy	26
Obr. 31: Grafické zobrazení násobné události	26
Obr. 32: Grafické zobrazení události času	26

Obr. 33: Grafické zobrazení události signálu	27
Obr. 34: Grafické zobrazení události ukončení	27
Obr. 35: Grafické zobrazení exkluzivní brány.....	27
Obr. 36: Grafické zobrazení komplexní brány	28
Obr. 37: Grafické zobrazení inkluzivní brány	28
Obr. 38: Grafické zobrazení paralelní brány.....	28
Obr. 39: Grafické zobrazení brány podmíněné událostí	28
Obr. 40: Bazény a plavecké drahy	29
Obr. 41: Logo firmy Gemalto [32]	33
Obr. 42: Grafické zobrazení virtualizace (tři fyzické servery s dvanácti virtuálními stroji) ...	34
Obr. 43: Příklad názvu virtuálního stroje.....	35
Obr. 44: Příklad názvu v případě duplicity jmen.....	37
Obr. 45: Proces vytvoření stroje	38
Obr. 46: Proces přejmenování stroje.....	39
Obr. 47: Proces odstavení stroje	40
Obr. 48: Proces klonování stroje.....	42
Obr. 49: Proces změny výpočetních prostředků	43
Obr. 50: Generování informací o stavu virtuální infrastruktury a účtování	44
Obr. 51: Rozhraní pro vytvoření nového virtuálního stroje.....	46
Obr. 52: Příklad vyplněného webového formuláře pro vytvoření nového stroje	51
Obr. 53: Proces vytvoření stroje - Optimalizováno	52
Obr. 54: Formulář pro nahrání veřejného klíče do farmy.....	54
Obr. 55: Přidělení přístupu - Optimalizováno.....	54
Obr. 56: Přehledné zobrazení všech virtuálních strojů pro výběr akce	55
Obr. 57: Proces přejmenování / změna vlastníka - Optimalizováno	56
Obr. 58: Odstavení stroje - Optimalizováno	57
Obr. 59: Klonování stroje – Optimalizováno.....	59
Obr. 60: Změna výpočetních prostředků - Optimalizováno	60
Obr. 61: Generování informací o stavu virtuální infrastruktury a účtování - Optimalizováno	61
Obr. 62: Automatické zálohování - Optimalizováno.....	62
Obr. 63: Webové rozhraní frameworku obsahující informace o daném stroji	63
Obr. 64: Případy užití v automatizačním frameworku.....	65
Obr. 65: Množství virtuálních strojů v roce 2016.....	73

7 Seznam tabulek

Tab. 1: Přehled časových náročností původních procesů (min)	68
Tab. 2: Srovnání časů původních a optimalizovaných procesů (min)	69
Tab. 3: Porovnání vlastních nákladů původních a optimalizovaných procesů (Kč).....	70
Tab. 4: Odhad náročnosti implementace jednotlivých aktivit (MD)	71
Tab. 5: Průměrné množství analyzovaných úkonů za měsíc	74
Tab. 6: Vlastní náklady na původní i optimalizované procesy (Kč).....	74
Tab. 7: CF jednotlivých variant	75
Tab. 8: Souhrnné vstupy pro výpočet NPV investice	75
Tab. 9: Seznam souborů na příloženém CD	86

8 Seznam souborů na přiloženém CD

Název	Popis
./Source.zip	Zdrojové kódy frameworku
./DP_2018_Sedlacek_Ondrej.pdf	Diplomová práce

Tab. 9: Seznam souborů na přiloženém CD