VYSOKÁ ŠKOLA BÁŇSKÁ – TECHNICKÁ UNIVERZITA OSTRAVA

EKONOMICKÁ FAKULTA

KATEDRA SYSTÉMOVÉHO INŽENÝRSTVÍ

Racionalizace procesu řízení incidentů na IBM Mainframe v IT firmě
Incident Management Process Rationalization for the IBM Mainframe at the IT Company

Student: Bc. Marián Brtko

Vedoucí diplomové práce: Ing. Jan Ministr Ph.D.

Ostrava 2015

VŠB - Technická univerzita Ostrava
Ekonomická fakulta
Katedra aplikované informatiky

Zadání diplomové práce

Student: **Bc. Marián Brtko**

Studijní program: N6209 Systémové inženýrství a informatika

Studijní obor: 6209T025 Systémové inženýrství a informatika

Téma: Racionalizace procesu řízení incidentů na IBM Mainframe v IT firmě

Incident Management Process Rationalization for the IBM Mainframe at the IT Company

Zásady pro vypracování:

1. Úvod
2. Teoretická východiska servisní podpory při řešení incidentů na IBM Mainframe
3. Analýza současného stavu
4. Návrh racionalizace procesu řízení incidentů na IBM Mainframe
5. Závěr

Seznam použité literatury
Seznam zkratek
Prohlášení o využití výsledků diplomové práce
Seznam příloh
Přílohy

Seznam doporučené odborné literatury:

EBBERS, Mike, et al. *Introduction to the New Mainframe: z/OS Basics*. 3rd ed. Thousand Oaks: Vervante, 2011. 764 p. ISBN 978-07-384-3534-1.
EBEL, Nadin, et al. *ITIL 2011*. Brno: Computer Press, 2012. 216 s. ISBN 978-80-251-3732-1.
ROGERS, Paul and Alvalo SALLA. *ABCs of z/OS System Programming Volume 12*. Thousand Oaks: Vervante, 2010. 184 p. ISBN 978-07-384-3383-7.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Jan Ministr, Ph.D.**

Datum zadání: 21.11.2014
Datum odevzdání: 25.04.2015

Ing. Petr Rozehnal, Ph.D.
vedoucí katedry

prof. Dr. Ing. Dana Dluhošová
děkanka fakulty

„Místopřísežně prohlašuji, že jsem celou práci včetně všech příloh vypracoval samostatně“

.....

Obsah

1 Úvod

V súčasnosti využívanie bankomatov je základnou požiadavkou držiteľov platobných kariet. Bankomaty vydávajú držiteľom platobných kariet peňažnú hotovosť z bežných alebo úverových účtov, prípadne poskytuje ďalšie služby. Platí to, či už ide o platenie platobnými kartami za služby alebo tovar v nákupných centrách, objednávky letenky cez internet a podobne. Toto je len malá časť vecí, ktoré používame, ale len málokto vie ako to celé funguje. Aby sme mohli využívať spomínané služby, musí to bežať na nejakom informačnom systéme. Väčšina systémov, ktoré zabezpečujú spomínané služby sa nazýva Mainframe a je vyvíjaný americkou spoločnosťou IBM.

Cieľom tejto diplomovej práce je popísať základné pojmy z oblasti IBM Mainframe a ITIL-u. Následne analyzovať proces vytvárania incidentov na systémoch, identifikovať kritický incident a navrhnúť jeho racionalizáciu, ktorá bude následne štatisticky vyhodnotená.

V teoretickej časti práce sú uvedené teoretické východiská servisnej podpory pri riešení incidentov na IBM Mainframe. Táto teoretická časť diplomovej práce je rozdelená na dve časti. Prvá časť teoretických východísk sa zameriava na pojmy z oblasti ITIL-u. Druhá časť teoretických východísk diplomovej práce sa zameriava práve na Mainframe od spoločnosti IBM, kde čitateľ diplomovej práce bude oboznámený s bohatou históriou, vlastnosťami, hardwarovými komponentami, operačným systémom a niekoľkými najdôležitejšími produktmi, ktoré ponúkajú mainframy.

V praktickej časti diplomovej práce bude zmapovaný súčasný stav procesu tvorby a riešenia incidentov na systémoch IBM Mainframe. Prvá časť kapitoly približuje graficky a textovo proces tvorby alert procesu na systéme. Druhá časť kapitoly je zameraná na štatistické vyjadrenie vytvorených incidentov na kritických systémoch IBM Mainframe.

Diplomovú prácu uzatvára štvrtá kapitola, ktorá sa venuje identifikácii incidentov na kritických systémoch a následná selektizácia incidentu, pre ktorý bude navrhnutá racionalizácia procesu riešenia incidentov.

V závere diplomovej práce je štatisticky vyhodnotená racionalizácia daného procesu a celkové zhodnotenie práce.

Vzhľadom k tomu, že si firma, v ktorej bola diplomová práca vypracovávaná, nepraje uviesť svoje originálne meno, bude ďalej v práci uvádzaná pod fiktívnym názvom TC. Taktiež aj informácie o kritických systémoch, na ktorých boli vykonávané štatistické

vyhodnotenia a racionalizácia, sú internou záležitosťou firmy, ktorá zaručuje diskrétnosť zákazníkov, budú tieto systémy označené fiktívnymi kombináciami znakov a číslic.

2 Teoretické východiska servisnej podpory pri riešení incidentov na IBM Mainframe

2.1 ITIL – IT Infrastructure Library

ITIL predstavuje vo forme zbierky kníh rozsiahly a všeobecne dostupný návod pre správu služieb IT. Tu uvedené skúsenosti a odporúčania sa v medzičase stali najlepšími praktikami, defacto štandardom.

2.1.1 Služba, Životný cyklus služby

Služba, Služba IT

ITIL prichádza s poskytovaním a správou služieb IT pre podnikanie s jasným cieľom podporovať obchodný úspech zákazníka. Tomu poskytuje ITIL základ a úzko súvisí s pojatím správy služieb IT.

Správa služieb je definovaná ako systém špecifických organizačných schopností (kľúčových kompetencií), ktoré zákazníkovi prinášajú pridanú hodnotu vo forme služieb. Zdroje sú pomocou nasadených funkcií a procesov transformované na kvalitné služby. Výhody sú tak prevedené či už smerom k vlastnému podnikaniu, tak aj na stranu zákazníka. Základným bodom kľúčových publikácií ITIL je myšlienka životného cyklu a zameriava pozornosť na spojenie obchodných cieľov zákazníka s IT neustálym zdôrazňovaním faktu, že IT musí napomáhať tvorbe hodnôt (Ebel, 2012).

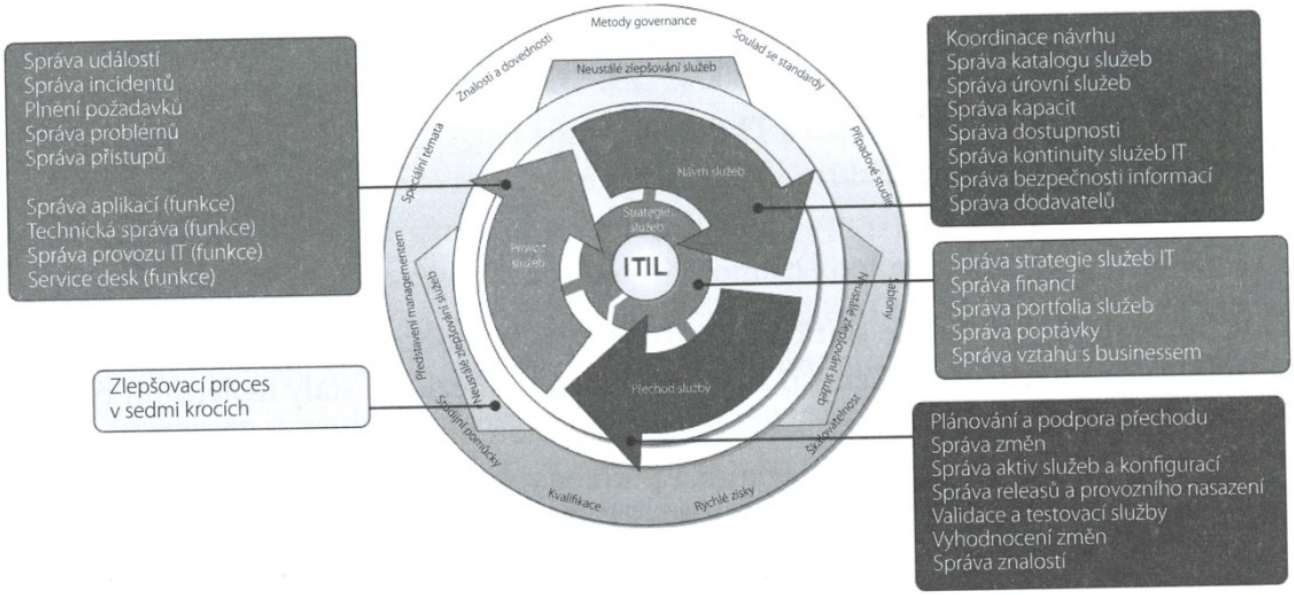
Na správu služieb (ITSM) je potreba nahliadať ako na takú implementáciu a správu služieb IT, ktorá odpovedá obchodným požiadavkám. Poskytovateľom služieb IT je správa služieb IT tvorená kombináciou personálu, procesov a technológií.

Služba je vlastne prostriedok tvorby hodnôt pre zákazníkov. Poskytuje zákazníkom zjednané výsledky, aniž by títo museli niesť zodpovednosť za špecifické náklady a riziká spojené so službami.

Služba IT je poskytovateľom služieb IT poskytovaná internému alebo externému zákazníkovi. Pozostáva z kombinácie informačných technológií, personálu a procesov. Služba IT orientovaná na zákazníka priamo podporuje podnikové procesy jedného či viac zákazníkov a v zmluve o úrovni služieb (Service Level Agreement, SLA) by mali byť definované ciele úrovní služieb. Ďalšie služby IT, nazývané podporné služby, nie sú podnikaním priamo využívané, ale pre prevádzku služieb orientovaných na zákazníka sú nevyhnutné (Ebel, 2012).

Životný cyklus služby

Za kľúčové publikácie ITIL je označované päť kníh. Každá z nich predstavuje jednu fázu životného cyklu a popisuje príslušné princípy, procesy, funkcie, organizačné a technologické aspekty a ďalšie príslušné témy. Štruktúru životného cyklu môžeme vidieť na nasledujúcom obrázku (Ebel, 2012).



Obrázok 1: Štruktúra životného cyklu (Zdroj: Ebel, 2012)

- **Service Strategy** (stratégia služby) reprezentuje základné smerovania a ciele.
- **Service Design** (návrh služby) ponúka návody pre návrh a vývoj služieb a procesov. Predstavuje metódy a princípy, pomocou ktorých je možné previesť strategické ciele do portfólia služieb a aktív služieb.
- **Service Transition** (prechod služby) sa stará o zavedenie nových alebo pozmenených služieb do výrobného prostredia.
- **Service Operation** (prevádzka služieb) sa zaoberá činnosťami s ohľadom na účinnosť a efektivitu dodávky a prevádzky služieb.
- **Continual Service Improvement** (CSI) poskytuje nástroje a návody pre nepretržité zlepšovanie služieb a všetkých zmienených aspektov ako je návrh, zavedenie a prevádzka služieb IT.

Životný cyklus služby v podstate pokrýva celú dobu životnosti služby IT a teda od jej vzniku až po ukončenie, tj. až po okamih, kedy sa prestane prevádzkovať. U poskytovateľa služieb sa v rámci jeho stratégie služieb objavujú nové služby.

Správa úrovne služieb

V správe úrovne služieb (Service Level Management, SLM) sa menia a zmluvne ošetrujú požiadavky zákazníkov na produkty poskytovania služieb organizácie IT (služby).

SLM ako proces taktiež zaisťuje priebežnú kontrolu prisľúbenej úrovne služieb a vykazovania služieb (service reporting). Požiadavky zákazníkov na kvalitu a kvantitu služby môžu byť splnené len vtedy, keď ciele úrovne služieb sledujú obchodné požiadavky (Ebel, 2012).

Dohoda o úrovní služieb (Service Level Agreement, SLA) popisuje služby IT a kvalitatívne a kvantitatívne dohody o poskytovaných službách IT medzi zákazníkom a organizáciou IT pomocou netechnických výrazov. Po dobou trvania dohody slúži SLA ako zmluva o poskytovaní a riadení. SLA je možné koncipovať podľa rôznych kritérií:

a) **Definícia SLA z perspektívy služieb:** pri nej je pre konkrétnu službu vytvorená jedna SLA. Táto SLA potom platí pre všetkých zákazníkov príslušnej služby,
b) **Definícia SLA na základe zákazníkov:** pre všetky služby jedného zákazníka platí jedna SLA,
c) **Kombinácia SLA založených na službách a zákazníkoch**, tzv. viacúrovňová štruktúra, v ktorej sú napríklad zhrnuté všetky obecné aspekty do obecne platnej časti, zhodnej pre všetkých zákazníkov (Corporate Level) a špecifické detaily sú doplnené v dohodách založených na službách resp. zákazníkoch.

Jedna SLA rieši netechnickými termínmi nasledujúce aspekty:

a) **Popis (prehľad dohodnutých plnení) a definícia služby:** popis výsledkov služieb alebo dielčich plnení (rozsahu), prípadne povinností zákazníka ku spolupráci a dodaní potrebného k poskytovaniu služieb,
b) **Úrovne služieb:** vyjadrenie kvality služieb alebo dielčich plnení popísaných v definícii služby,

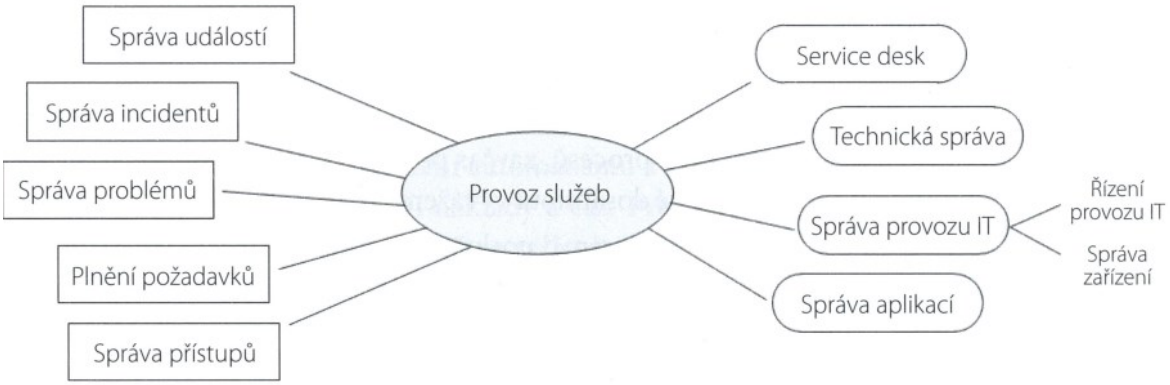
c) **Spracovanie zmien** (postupy zmien), schválenie zmien a s nimi súvisiacich nákladov, štandardov služieb a objemu služieb,

d) **Početnosť komunikácie a podávanie hlásení** a takisto aj vyúčtovanie služby, štruktúra cien a platobné podmienky,

e) **Predpisy týkajúce sa utajenia a zverejňovania informácií**, autorských práv, obmedzení zodpovednosti, právo odstúpenia od zmluvy (pre obe strany), trvanie, výpovedná lehota a povinnosti vyplývajúce zo zmluvy.

2.1.2 Prevádzka služieb

Služby sa spravujú a sú poskytované pomocou vykonávania prevádzkových riadiacich činností. Pritom nie je len dôležité, aby boli služby poskytované efektívne a za rozumnej ceny, ale aby sa taktiež zaistila spokojnosť zákazníka, že sa tak deje v rámci dohodnutých úrovní služieb. Mimo koordinácie odpovedajúcich funkcií, aktivít a procesov je prevádzka služieb zodpovedná rovnako za priebežnú správu technológií, ktoré sú potrebné pre poskytovanie a podporu služieb. Štruktúru prevádzky služieb znázorňuje nasledujúci obrázok (Ebel, 2012).



Obrázok 2: Prevádzka služieb (Zdroj: Ebel, 2012)

Prevádzka služieb ako súčasť správy služieb má zaisťovať, aby zákazník dostával odpovedajúcu hodnotu. Prevádzka služieb je zodpovedná za vykonávanie služieb, ktoré boli v predchádzajúcich fázach životného cyklu služby naplánované, navrhnuté a zostavené. Jedná sa tu napríklad o sledovanie výšky nákladov na službu alebo presadzovanie financovania nástrojov alebo činností ako sú školenia, ďalší rozvoj a zlepšenie služieb.

Procesy v rámci prevádzky služieb

Procesy fáze životného cyklu prevádzky služieb podporujú správu prevádzky služieb. Obsahujú návody pre účinné a efektívne poskytovanie služieb IT a ich podporu a údržbu, aby bolo zákazníkovi možné poskytovať definovanú hodnotu. Cieľom je stabilné, pokiaľ možno bezchybná prevádzka služieb, vyznačujúci sa dodržovaním dohôd o úrovni služieb (SLA). Na tieto ciele úrovní služieb sa zameriavajú procesy:

a) **Správa udalostí** – ponúka možnosť včasného odhalenia incidentov alebo dokonca aj problémov, a teda ich príčiny. Identifikuje možné nasadenia automatických nástrojov a tým znižuje časy výpadkov a vďaka cieleným upozorneniam na odchýlky šetrí náklady a čas,
b) **Správa incidentov** – zaznamenáva, skúma, kategorizuje, priradzuje prioritu a sleduje všetky poruchy služieb, aby ich bolo možné odstrániť čo najrýchlejšie a s minimálnymi dopadmi pre užívateľa. Poskytuje prvú pomoc a prípadne koordinuje ďalšie spracovanie následnými úrovňami podpory,
c) **Plnenie požiadaviek** – vznikol pre účinné a efektívne spracovanie všetkých požiadaviek kladených organizáciou IT. Jedná sa o samostatný proces pre požiadavky užívateľov, ktoré nesúvisia s poruchou,
d) **Správa problémov** – úlohou správy problémov je sledovanie problému od začiatku až do konca, teda od jeho identifikácie cez ďalšie analýzy, dokumentáciu až po odstránenie príčiny problému. Problémy je potreba riešiť a je možné im predchádzať,
e) **Správa prístupov** – je to proces, udeľujúci autorizovaným užívateľom práva k využívaniu služieb, zatiaľ čo neautorizovaným užívateľom je prístup zamietnutý. V niektorých organizáciách sa tieto úlohy označujú ako správa práv alebo správa identít.

Keďže sa moja diplomová práca venuje priamo správe incidentov, tento proces si priblížime dopodrobna.

Správa incidentov

Hlavným cieľom tohto procesu je čo najrýchlejšia obnova postihnutej služby pre návrat k „normálnemu prevádzkovému stavu“ s minimálnymi negatívnymi dopadmi na prevádzku podniku. Tento stav je popísaný v rámci SLA. Ďalšími cieľmi sú (Ebel, 2012):

a) Zaistiť aby všetci zúčastnení v rámci aktivít procesu používali štandardizované metódy a postupy,
b) Proaktívne zdieľanie incidentov podpore IT a zákazníkovi, napr. pri veľkých incidentoch pomocou intranetu alebo e-mailov, s cieľom zlepšiť tok informácií,
c) Udržanie, resp. zvýšenie spokojnosti zákazníka pomocou kvality služieb IT.

Princípy a pojmy zo správy incidentov

Proces sa skladá predovšetkým z reakčných úloh. Je dôležité, aby sa s poruchami nejednalo len ako so vzniknutými, ale aj s potenciálnymi narušeniami (Ebel, 2012).

a) **Incident** – je neplánované prerušenie alebo zníženie kvality služby IT. Jedná sa o udalosť, ktorá nepatrí k bežnej prevádzke a predstavuje skutočné alebo potenciálne narušenie alebo zníženie kvality služby. Príčinu jedného ale aj viac incidentov je spravidla možné vystopovať späť ku konkrétnemu problému,
b) **Známe chyby a náhradné riešenie** – pokiaľ je podstata poruchy už známa a zdokumentovaná, ale doposiaľ nie je odstránená, hovorí sa o známej chybe (Known error). Dopady incidentu je možné zmierniť alebo odstrániť pomocou náhradného riešenia (Workaround). Síce to nerieši skutočný problém, ale pomáha to užívateľom pracovať ďalej bežným spôsobom do doby, keď bude existovať konečné riešenie,
c) **Záznam incidentu** – obsahuje detaily konkrétneho incidentu. Tento záznam môže naberať rôzne stavy, napr. otvorený, spracováva sa, vyriešený, uzavretý,
d) **Veľký incident** – najvyššia kategória incidentov, čo sa týka dopadu. Vedú k významnému prerušeniu služieb pre podnik a preto sa riešia oddelenými postupmi a neodkladne,
e) **Eskalácia** – pri eskalácii sa prideľujú dodatočné prostriedky a zdroje pre splnenie dohodnutých cieľov úrovne služieb, resp. požiadaviek zákazníka,
f) **Model incidentu** – sú založené na kategorizácii a vyhodnotení opakujúcich sa porúch a popisujú príslušné kroky pre ich spracovanie. Tento prístup podporujú nástroje a odpovedajúce dokumenty.

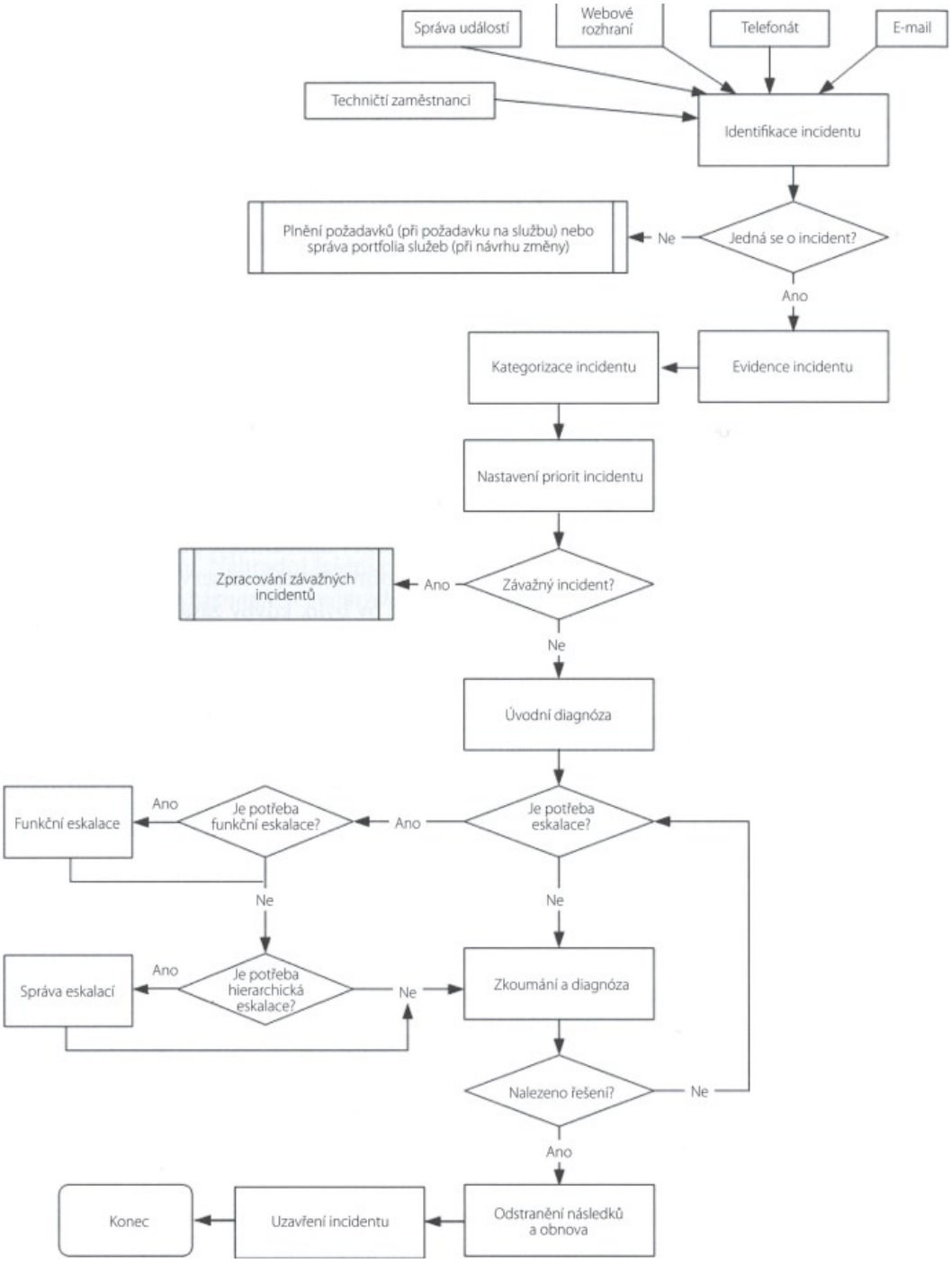
Vstup a spúšťač procesu

Incidenty sa ako vstup procesu dostávajú ku správe incidentov buď od užívateľa priamo cez webové záznamy cez samoobslužný portál, alebo zavolaním na service desk (alebo e-mail, fax atď.), alebo priamo a automaticky cez rozhranie správy incidentov. Poruchy môžu hlásiť aj ďalší technickí pracovníci a dodávatelia.

Aktivity v rámci správy incidentov

Behom procesu prebiehajú nasledovné aktivity, ktoré sú znázornené na obrázku číslo 3:

a) **Identifikácia incidentu** – zavolaním užívateľa na service desk a následné nasmerovanie pozornosti na konkrétnu chybu alebo zásahom monitorovacích opatrení ku sledovaniu (potenciálnych) výpadkov a odštartovanie procesu správy incidentov ešte predtým, než to ovplyvní bežné prácu užívateľa (Ebel, 2012),
b) **Záznam incidentu** – prevzatie hlásenia poruchy, opatrenie dát časovým razítkom a to nezávisle na spôsobu prijatia hlásenia o incidente,
c) **Kategorizácia** – má zásadný vplyv na ďalšie kroky a taktiež na zmysluplnosť informácií správy a ďalšie vyhodnotenie,
d) **Pridelenie priority** – pridelená priorita incidentu rozhoduje o ďalšom spracovaní, napr. o rýchlosti riešenia. Vzorec je: Priorita = dopad + naliehavosť. Dopad znamená odhad, koľko môže byť zasiahnutých užívateľov a naliehavosť vyjadruje ako rýchlo potrebuje podnik riešenie. Priorita incidentu sa môže v priebehu času meniť,
e) **Úvodná diagnóza** – vo chvíli, keď sa incident dostane ku správe incidentov, zamestnanec service desku spravidla už behom prvého telefonického kontaktu s užívateľom overuje, čo sa stalo a skúša prvotné riešenie pre obnovu (napr. pomocou modelov incidentov). Pokiaľ sa daný incident nepodarí vyriešiť dochádza k eskalácii,
f) **Preskúmanie a diagnóza** – aby bolo možné vykonať obnovu, zisťujú zúčastnené skupiny pomocou relevantných akcií, čo sa stalo zle,
g) **Riešenie a obnova** – ako náhle je identifikované možné riešenie, malo by byť použité a otestované. Na základe možného riešenia sa vykonáva opatrenie k jeho uskutočneniu a obnoveniu služby,
h) **Ukončenie** – aby bolo možné uzatvoriť záznam incidentu, kontroluje control desk úspešnosť poruchy a úplnosť dát. Končené uzavretie by malo nasledovať po konzultácii s užívateľom.



Obrázok 3: Aktivity v rámci správy incidentov (Zdroj: Ebel, 2012)

Nezávisle na aktivitách procesu sú podľa dohody vykonávané obecné hlásenia procesu. Všetky aktivity sú zaznamenávané v zázname incidentu (Ebel, 2012).

Výstup správy incidentov

Medzi rôzne výsledky procesu patrí hlavne obnovená služba a uzavretý, vyriešený incident. Patrí k nim aj pozitívna spätná väzba po informovaní zákazníka o ukončení.

Možným výsledkom procesu je aj problém, a teda záznam problému, ktorý je otvorený za účelom zahájenia hľadania príčiny.

Role v rámci správy incidentu

Manažér incidentu je zodpovedný za kompletný priebeh správy incidentov na všetkých úrovniach podpory a za spracovanie veľkých incidentov. V rámci procesu je aktívny analytik prvej línie, analytik druhej línie ako člen podpornej skupiny a analytik tretej línie ako súčasť internej technickej skupiny alebo podpory poskytovanej tretou stranou. Dôležitú funkciu preberá Service desk (väčšinou analytik prvej línie) ktorý má neustále administratívnu kontrolu nad životným cyklom incidentu.

Service Desk

SD slúži užívateľom ako primárne kontaktné miesto pri hlásení porúch služieb (incidentov) a žiadostí o službu. Okrem toho slúži aj ako interné koordinačné miesto rôznych procesov a skupín IT. Predstavuje jediné kontaktné miesto (SPoC) a je prvou líniou podpory organizácie IT. Typický service desk má na starosti incidenty a požiadavky na službu. Táto funkcia teda predovšetkým plnenie aktivity procesov správy incidentov a plnenie požiadaviek. Ďalšie aktivity ako napr. dotazovanie zákazníkov / zamestnancov na spokojnosť alebo poskytovanie informácií užívateľom, sa pridávajú podľa konkrétneho podniku (Ebel, 2012).

Správa zmien (Change management)

Ťažiskom správy zmien je snaha o minimalizáciu výpadkov a incidentov spôsobených zmenami (changes). Vďaka štandardizovaným metódam a postupom by mali zmeny prebiehať rýchlo a kontrolovane. Účelom tohto procesu je preto riadenie životného cyklu všetkých zmien s cieľom získať z nich maximálny úžitok a zároveň minimalizovať nebezpečenstvo výpadku poskytovania služby.

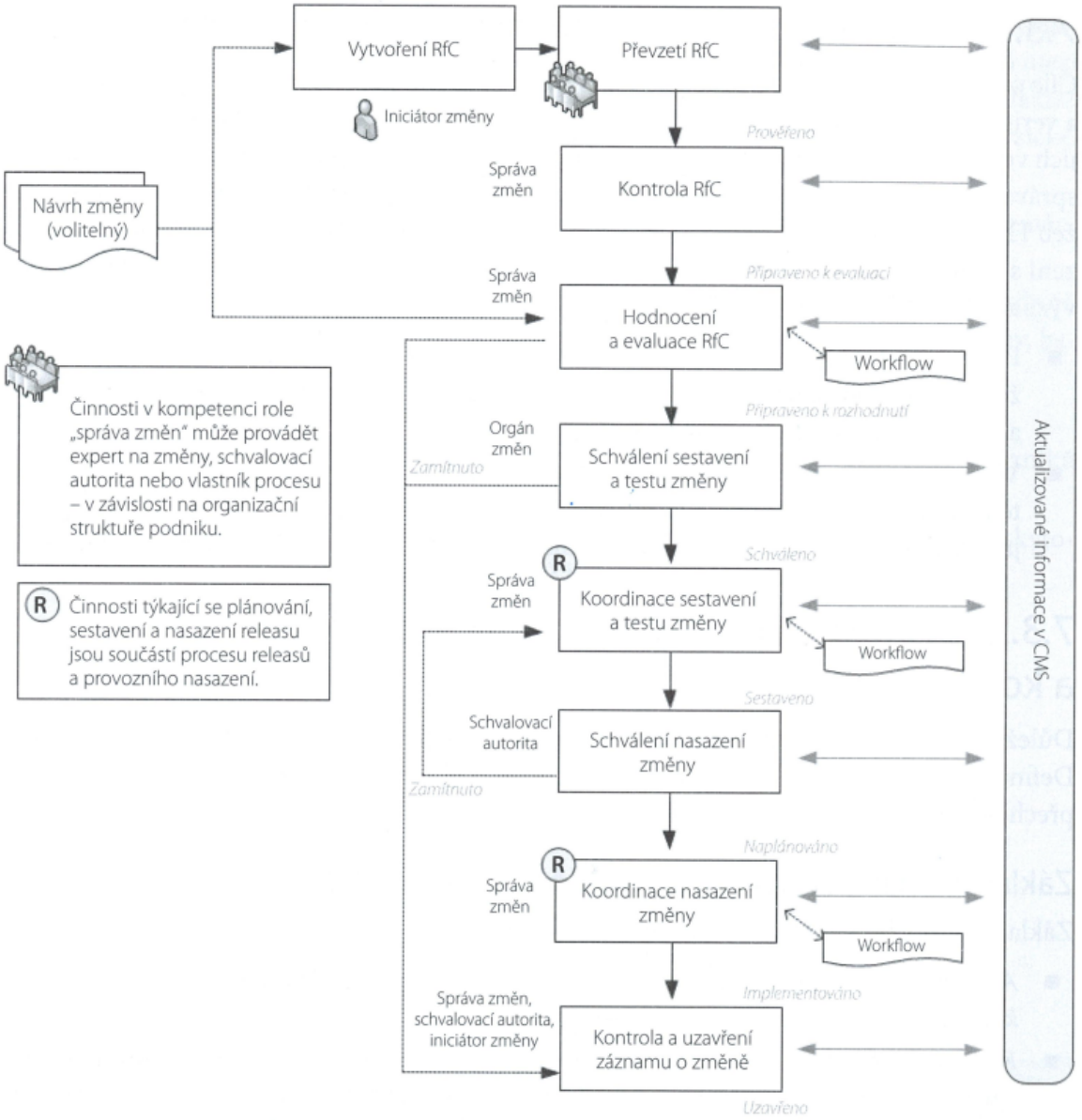
Veľké počty zmien a ich relatívne ďalekosiahlejšie následky spôsobujú nutnosť ich systematického a kontrolovaného plánovania a riadenia v súlade s celkovou optimalizáciou obchodných rizík. Cieľom správy zmien je reakcia na meniace sa obchodné požiadavky pri súčasnej maximalizácii úžitku a redukcii počtu incidentov a výpadkov. Malo by tak byť

zaistené, že zmeny budú zaznamenané, posúdené, schválené, opatrené prioritou, naplánované, otestované, implementované a dokumentované kontrolovaným spôsobom.

Zmena (change) je termín pre pridanie, úpravu alebo odstránenie konkrétnej CI (konfiguračnej položky, configuration item). Zmena je iniciovaná prostredníctvom žiadosti o zmenu (RfC). Tá predstavuje žiadosť o vykonanie zmeny jednej alebo viac CI. Záznam o zmene (change record) obsahuje všetky údaje o každej vykonávanej zmene a zároveň dokumentuje jej životný cyklus. Patrí sem jednoznačné ID pre RfC, meno a ďalšie informácie o žiadateľovi, označenie, dôvod a popis zmeny ako aj závislosti, dopady a odhady rizík. Je potrebné zaznamenať väzbu na CI, ktorých sa to týka. Priebeh zmien je zobrazený na obrázku číslo 4 (Ebel, 2012).

Rôzne typy zmien vyžadujú rôzne prístupy k vykonaniu:

a) **Normálna zmena** – týka sa zmeny jednej alebo viac konfiguračných položiek alebo služby, ktorú je potrebné formálne vyžiadať a schváliť a prechádza správou zmien,
b) **Štandardná zmena** – sú úpravy s nízkym rizikom a štandardizovanými postupmi spracovania, ich schvaľovací proces prebehol už predtým a považujú sa tak za „predschválené“,
c) **Naliehavá zmena** – tzv. emergency change, ktorú je potrebné vykonať tak rýchlo, ako to je možné. Dokumentácia a nutná kontrola prebieha najčastejšie následne, tj. po dokončení zmeny.



Obrázok 4: Správa zmien (Zdroj: Ebel, 2012)

2.1.3 Vzťah ITIL a CobiT

Cobit od verzie 4 (súčasná verzia je 5) je plne kompatibilný s ITIL, tzn. základné princípy riadenia IT prostredia sú buďto úplne zhodné (napr. riadenie incidentov, požiadaviek na službu, problémov) alebo aspoň nie sú v principiálnom rozpore, aj keď sú z pohľadu organizačne-procesného pojaté mierne odlišným spôsobom (napr. informačná bezpečnosť, životný cyklus služby IT) (Bestpractice, 2015).

CobiT má oproti ITIL dve základné výhody a jednu veľkú nevýhodu (Bestpractice, 2015).:

a) CobiT predstavuje jednoducho uchopiteľný auditný a riadiaci rámec, ktorý umožňuje priamo a konkrétnym spôsobom prepojiť IT stratégie s celopodnikovou stratégiou, resp. nastaviť strategické riadenie podnikovej informatiky v zhode so strategickými požiadavkami businessu. Prestože ITIL obsahuje samostatnú publikáciu Service Strategy, je aplikácia týchto „best practice“ veľmi obtiažné uchopiteľná, a preto je vhodnejšia pre nastavenie IT stratégie použiť CobiT,
b) CobiT zahrňuje všetky aspekty riadenia podnikovej informatiky, kdežto v ITIL chýbajú celé veľké oblasti, hlavne riadenie IT projektov, riadenie ľudských IT zdrojov a oblasť IT developmentu. Pokiaľ chce mať CIO (chief information officer = riaditeľ útvaru podnikovej informatiky) istotu, že organizačne a manžérsky pokrýva všetky oblasti, ktorým by sa mal z titulu svojej zodpovednosti venovať, tak s ITIL nemôže vystačiť a musí siahať po CobiT, ktorý mu pomôže sa rýchlo zorientovať a v akejkoľvek situácii vyhodnotiť, ktorý aspekt IT managemenetu nemá pod kontrolou,
c) Základná nevýhoda CobiT je v tom, že je určený predovšetkým ku strategickému riadeniu IT prostredia a k vykonaniu jeho auditu, resp. k rýchlemu odhaleniu chýb pri jeho riadení. CobiT už nehovorí takmer nič o tom, ako designovať a implementovať procesy, aktivity, funkcie a role, ktoré by zaistili splnenie princípov riadenia, ktoré CobiT tak jednoducho a prehľadne popisuje z pohľadu audítora. CobiT neobsahuje niekedy ani základné definície, v oblasti popisu procesov prichádza len s výčtom vstupov, výstupov, rolí a aktivít, avšak jedná sa len o ich pomenovania a podobnejší popis toho, čo je tým myslené, väčšinou chýba. ITIL oproti tomu obsahuje podrobné popisy, definície, ukážky šablón, diagramy, modely atď. Implementácia systémov riadenia služieb IT čisto podľa špecifikácie CobiT je teda veľmi komplikovaná, ne-li nemožná.

Spoločné využitie ITIL a CobiT.

a) CobiT pre vykonanie auditu a identifikáciu prvkov riadenia IT prostredia, ktoré nemáme pod kontrolou, a pre tvorbu systémov prepojenia IT stratégie s celopodnikovou stratégiou,

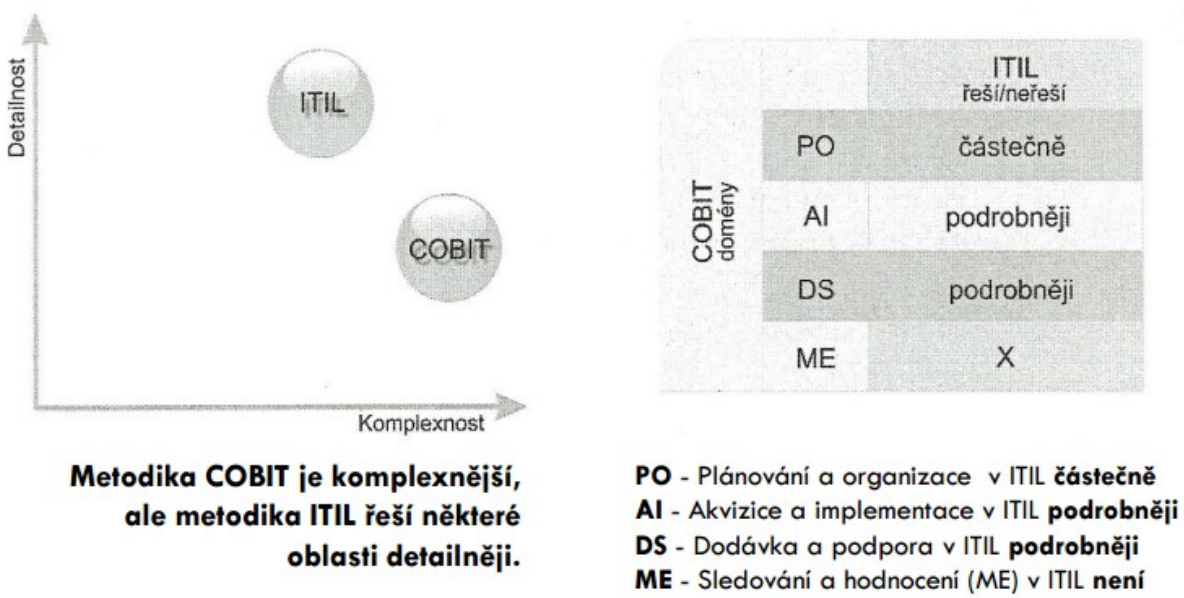
b) ITIL pre vytvorenie detailného designu a následnú implementáciu systému operatívne-taktického riadenia služieb v súlade s IT stratégiou vytvorenou podľa CobiT.

2.1.4 Porovnanie metodik ITIL a CobiT

Tabuľka 1: Porovnanie metodík ITIL a CobiT (Zdroj: Bestpractice, 2015)

IT management versus IT Governance	
IT management sa sústreduje na efektívne poskytovanie služieb a produktov IT a na účinné riadenie rozvoja a prevádzky IT	IT Governance je odpovednosťou najvyššieho vedenia, ktoré svojím príkladom, organizačným usporiadaním a vnútornými procesmi zaisťuje, že IT prispieva a prehlbuje stratégiu a dlhodobé ciele organizácie
Snahou IT managementu je efektívne a účinne realizovať stanovené ciele, a preto sa orientuje na taktické a operatívne riadenie. IT management je realizovaný hlavne v rámci útvarov IT	IT governance má širšie poňatie a jeho snahou je definovať strategické ciele IT v súlade s potrebami a záujmami celej organizácie
Medzinárodne uznávaným rámcom IT managementu je ITIL	Medzinárodne uznávaným rámcom IT governance je CobiT

Na nasledujúcom obrázku môžeme vidieť porovnanie metodík ITIL a CobiT.



Obrázok 5: Porovnanie metodík ITIL a CobiT (Zdroj: Bestpractice, 2015)

2.2 IBM Mainframe

Všetci poznáme počítače, pravdepodobne sme počuli o superpočítačoch. Mainframy nie sú tak známe. Historicky, mainframy sú spojené s centralizovanými výpočtami na rozdiel od distribuovaných výpočtov. Napriek prevahe mainframov v obchodnom svete, tieto stroje nie sú veľmi viditeľné pre verejnosť, akademickú komunitu a dokonca aj pre mnoho skúsených IT profesionálov. Koniec koncov, kto z nás potrebuje priamy prístup k mainframom? Po pravde povedané, všetci z nás sme mainframe používatelia, či už si to uvedomujeme alebo nie (FEEDIT, 2014).

Mainframy, označované aj ako sálové počítače alebo Big Iron, sú počítače používané hlavne veľkými organizáciami pre kritické aplikácie, typicky hromadné spracovanie dát, priemyslu a spotrebiteľových štatistík, ERP a spracovanie finančných transakcií. Termín vznikol pravdepodobne z prvých mainframov, ktoré boli uložene v enormne veľkých kovových boxoch. V súčasnej dobe sa v praxi tento termín používa pre počítače kompatibilné s IBM System / 360, ktorý bol predstavený na trh v roku 1964. Najnovší model je IBM System z196 od spoločnosti IBM (ComputerWorld, 2014).

Užitočnosť mainframov dokazuje aj fakt, že s nimi vyše 50 rokov od ich uvedenia na trh stále pracujeme. V roku 1996 Stewart Alsop Jr., vice-prezident spoločnosti InfoWorld, ktorá sa zaoberá informačnými technológiami sa predovšetkým dostal do povedomia výrokom, že: "Predpovedám, že posledný mainframe bude odpojený 15. Marca 1996.". Vo februári roku 2002 svoj výrok zmenil na: "Je jasné, že firemní zákazníci ešte chcú mať centrálne riadené, veľmi predvídateľné, spoľahlivé výpočtové systémy, presne ten druh systémov, ktorými sa zaoberá firma IBM."(Elliott, 2014)

Možno si ani neuvedomujeme, ale s mainframom sa stretávame dennodenne. Či už pri vyberaní hotovosti z bankomatov, nakupovaní v obchodný centrách, rezerváciách leteniek, prechádzaní cez križovatku so svetelnou signalizáciou, pri kontaktovaní polície, hasičov alebo záchrannej služby, pri termináloch v bankách, poisťovniach a v iných oblastiach služieb.

V dnešnej dobe je za pomoci mainframov spracovávaných viac než 70% podnikových dát a 71% najvýznamnejších globálnych spoločnosti zo zoznamu Fortune 500 prevádzkuje hlavnú časť svojho businessu za pomoci mainframov. Mainframe navyše práve teraz prechádza vývojom, ktorý sa môže ukázať rovnako revolučný ako jeho predstavenie pred päťdesiatimi rokmi (FEEDIT, 2014).

Päťdesiat rokov mainframu zmenilo každodenný život na našej planéte a učinilo ju bezpochyby o niečo chytrejšiu. Predstava o tom, kam nás mainframe posunie o ďalších 50 rokov, vyvoláva nepatrné vzrušenie a vysoké očakávania spoločnosti.

2.2.1 Mainframe verzus superpočítač

Rozdiel medzi mainframom a superpočítačom je v tom, že superpočítače sa vo všeobecnosti zameriavajú na problémy, ktoré sú limitované výpočetnou rýchlosťou, zatiaľ čo mainframy sa zameriavajú na problémy limitované na vstupe / výstupe a ktoré vyžadujú extrémne vysokú spoľahlivosť a sú schopné riešiť súčasne násobky podnikových problémov (Aspg, 2014).

Superpočítače sú optimalizované pre komplikované výpočty, ktoré spotrebovávajú veľké množstvo pamäte, pokiaľ mainframy sú optimalizované pre relatívne jednoduché výpočty , ktoré zahŕňajú veľké množstvo externých dát. Napríklad, predpovede počasia sú optimalizované pre superpočítače, kdež to spracovania miezd, poistení, bankomatové služby sú viac vhodné pre mainframe.

Superpočítač sú často využívané pre jednu alebo niekoľko málo konkrétnych inštitucionýlnych úloh, ako napríklad simulácia a modelovanie. Mainframy dokážu zvládnut širšiu škálu úloh ako napríklad spracovanie dát alebo skladovanie dát.

Tabuľka 2: Zosumarizovanie vlastností Mainframe a Superpočítač (Zdroj: Aspg, 2014)

Mainframe	Superpočítač
Dokáže spustiť viac programov súčasne	Zameranie na rýchlosť a rýchlejší výkon
Podporuje nový aj starý software (spätná kompatibilita)	Výpočetný výkon zameriava na vykonávanie niekoľkých programov najrýchlejsie ako je možné
Podporuje mnoho súbežných užívateľov	Zvyčajne beží na maximálny výkon, svoje zdroje spracovania sústreduje k riešeniu konkrétneho problému
Nepretržitá prevádzka	Výkon je meraný v tzv. FLOPS (Floating Point Operations per Second)
Výkon v miliónoch inštrukcií za sekundu (MIPS)	Vykonáva zložité výpočty pomocou veľkej internej pamäte
Vykonanie úloh z veľkého množstva externých dát	Vyhradené účely pre modely technických alebo vedeckých modelov

2.2.2 Vlastnosti IBM Mainframe

Skratka RAS, ktorý je v originále využívaný hlavne firmou IBM, dokáže najlepšie vyjadriť základné vlastnosti mainframov. Jedná sa o skratku 3 definícií mainframov a to (Kettner, 2011):

a) **Spoľahlivosť** (Reliability),
b) **Dostupnosť** (Availability),
c) **Prevádzkyschopnosť** (Serviceability).

Atributy RAS by mal mať určite každý počítač, ale u mainframov sa naň kladie najvyšší dôraz. A to z dôvodu, že mainframy z 99% pracujú s citlivými údajmi alebo transakciami, ktoré majú kritický dopad na spoločnosť. Preto má mnoho súčasti systému schopnosť kontrolovať samého seba a prípadne sa aj opraviť (spoľahlivosť), v prípade poruchy sa nezničí zvyšok systému a zničený hardware sa nahradí paralelnou súčasťou (dostupnosť) a bez toho aby bolo nutné beh operačného systému meniť, môže byť zničený hardware vymenený za nový (prevádzkyschopnosť). Takýto systém väčšinou nemusí byť vôbec pozastavený z prevádzky kvôli vylepšeniam alebo opravám a keď, tak veľmi na krátky čas. Tieto charakteristiky platia pre hardware a takisto aj pre software (Kettner, 2011).

Ďalšími dôležitými vlastnosťami je veľmi vysoké zabezpečenie dát, prispôsobivosť systému novým vlastnostiam (nové procesory, pamäť atď.) a spätná kompatibilita. To nám umožňuje spúšťať programy, ktoré boli vývojarmi naprogramované aj pred 40 rokmi.

2.2.3 Milníky IBM Mainframe

Nasledujúca časová osa zobrazuje najdôležitejšie udalosti týkajúce sa vývoja IBM Mainframe (Kettner, 2011).

7. Apríl 1964 – IBM ohlasuje systém System/360. Rodinu piatich výkonných počítačov, na ktorých beží rovnaký OS a dokážu používať 44 periférnych zariadení

1968 – zavedenie Customer Information Control System – CICS,

1972 – Spoločnosť IBM ohlásila virtualizáciu VM s operačným systémom VM/370,

1972 – SAP vyvíja revolučný ERP systém pre S/370. Prvýkrát môžu firmy umiestniť objednávky a sledovať zásoby v reálnom čase,

1976 – SAS software vytvára novú konkurenčnú výhodu: Business Intelligence,

1988 – IBM predstavuje PR/SM. To umožňuje vytvorenie logických partícií – LPAR,

1988 – IBM zavádza systém pre správu relačnýc databáz,

1988 – MVS/ESA a VM/XA operačné systémy boli predstavené, odľahčujú pamäťové obmedzenia, ktoré limitovali veľkosť aplikácií pomocou nových architektonických konštrukcií,

1994 – Paralelný sysplex (parallel sysplex) bol oznámený,

1994 – oznámenie UNIX-u na mainframoch,

1995 – procesory založené na CMOS boli zavedené do prostredia mainframov,

2000 – predstavenie IBM zSeries na trh a 64-bitový operačný systém z/OS,

2003 – špeciálny procesor Linux-u zIFL je predstavený na zSeries mainframoch,

2004 – špecializovaný prcoesor JAVA zAAP je predstavený na zSeries mainframoch,

2010 – uvedenie z196 rodiny mainframov s zBX rozšírením.

2.2.4 Pracovné úlohy mainframu

Mainframy plnia 2 typy pracovných úloh, je to:

a) Dávkové spracovanie (batch processing),
b) Online spracovanie transakcií (online transaction).

Dávkové spracovanie

Je to pracovná úloha, kde mainframe dostane na začiatku určitý vstup v podobe dát (napríklad niekoľko terabajtov) a vytvorí z neho použiteľný výstup. Dávková úloha môže trvať niekoľko sekúnd, ale aj niekoľko hodín. Napríklad v banke ma mainframe vstup databázu klientov a ako výstup to bude pravidelné mesačné vyúčtovanie pre každého z nich, štatistická správa pre vedenie podniku, správa pre spoločnosť spravujúcu kreditné karty a vytvorenie záložnej kópie dát. Tieto úlohy pozostávajú z väčšieho množstva pod úloh a vytvárajú informácie pre veľké množstvo užívateľov. Beh úloh na mainframe je bez zásahu užívateľa a všetky dávkové úlohy majú vopred presné nadefinovaný čas začiatku a konca (Kettner, 2011).

Online spracovanie transakcií

Jedná sa o pracovnú úlohu, kde zákazník komunikuje s mainframom za pomoci terminálu a ten mu poskytuje žiadané služby. Typickým príkladom je výber peňažných prostriedkov z bankomatu alebo webový rezervačný systém leteniek. Dôležitým kritériom pre rozhodnutie či pre danú činnosť nasadíme mainframe alebo obyčajný server, rozhodujú tri kritéria:

a) Počet klientov, ktorý pristupujú k systému v ľubovoľnom danom čase,
b) Počet transakcií za sekundu,
c) Dostupnosť služby (24/7).

Transakcie sú charakterizované malým objemom vstupných dát, veľmi krátkou dobou behu (menej ako 1 sekunda), vysokým počtom transakcií vykonávaných naraz mnohými užívateľmi, vysokou bezpečnosťou a časovou dostupnosťou.

2.2.5 Obsluha mainframov

Mainframe systémy sú navrhnuté tak, aby slúžili veľkému počtu ľudí. Vzhľadom k veľkému počtu užívateľov a aplikácii, ktoré bežia na systéme sú potrebné rôzne úlohy pre prevádzku a podporu mainframového systému. V oblasti informačných technológií sa jednotlivé úlohy označujú:

a) **Systémový programátor** – stará sa o operačný systém ako celok (inštalácia, zmeny v nastaveniach, plánovanie výkonu),
b) **Správca systému** – každodenná údržba systému,
c) **Aplikační dizajnéri a programátori** – príprava nových programov pre mainframy,
d) **Systémový operátor** – správa veľkých podsystémov, sledovanie správnej spolupráce medzi softwarom a hardwarom,
e) **Analytik riadenia výroby** – kontrola správnosti priebehu úloh.

2.2.6 Výhody a nevýhody mainframov

Hlavnou výhodou mainframov je ich dostupnosť, ktorá je obdivuhodná. Systémy bežia 99,999% času, čo si je možné predstaviť ako 5 minútový prestávku počas celoročnej prevádzky za jeden rok. U unixových serverov to je približne 23,6 hodiny za rok. Výpadky a odstávky mainframových systémov sú spôsobené plánovanými zmenami, neplánovanými chybami alebo živelnými pohromami. K prepnutiu na paralelný systém a udržovanie

aktuálnosti dát na oboch systémoch sa využíva technológia GDPS. Vďaka nej sa mainframe z havárie úplne spamätá za menej ako hodinu. GDPS je kombinácia technológie paralelného sysplexu (zdieanie dát a zdvojenie súčastí), synchronizovaného vzdialeného kopírovania a automatického opravovania chýb. Jedinou podmienkou pre fungovanie je, aby vzdialenosť primárneho a sekundárneho systému neprekročila 40 km (Rogers a Salla, 2010).

Nevýhodou mainframe je jeho cena. IBM svoje systéme prenajíma a suma za prenájom sa odvíja podľa produktov, ktoré si sami navolíme. To znamená, že náklady na prevádzku sú vysoké a nemôže si ich dovoliť každá firma. Takisto sa na trhu nachádza veľmi málo odborných pracovníkov, ktorí mainframom rozumejú.

2.2.7 Úvod do hardwarových systémov IBM Mainframe

Obsah mainframe boxu

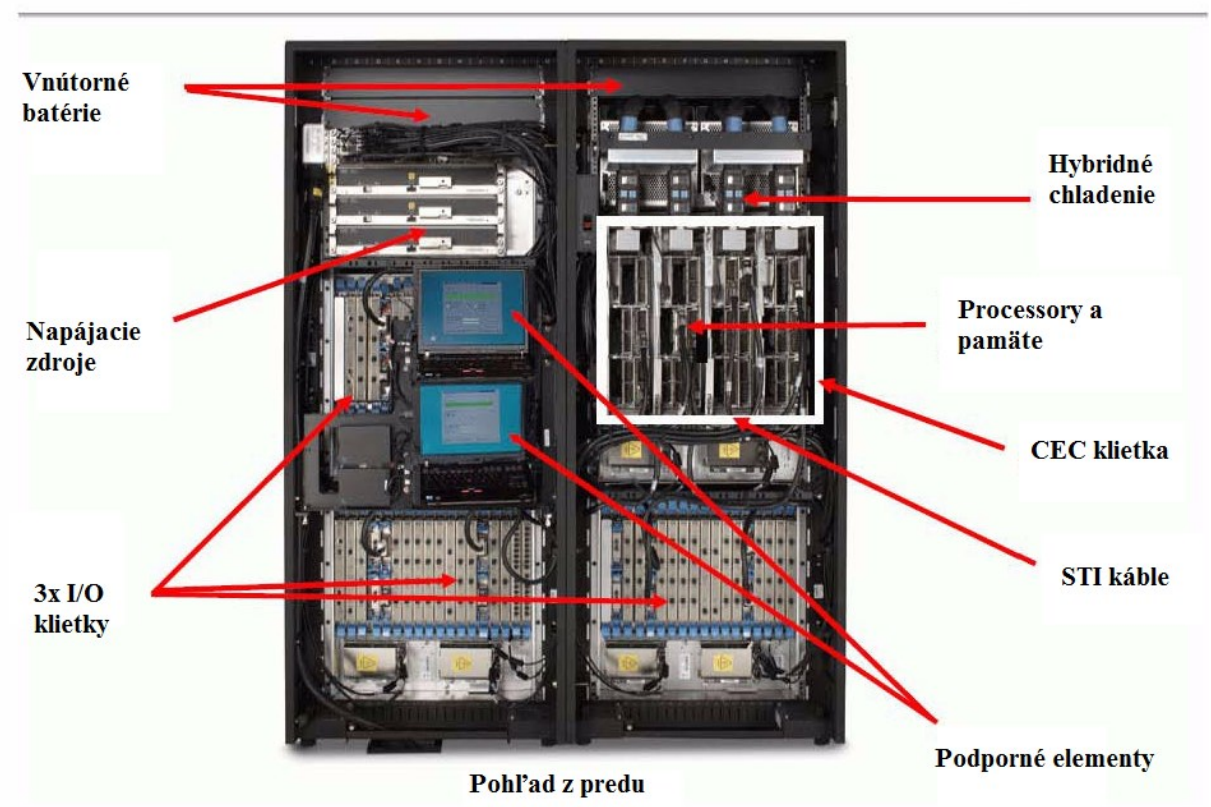
Central procesor complex, skratka CPC, je označenie pre mainframe „box“, teda fyzická kolekcia hardwaru, ktorá zahŕňa:

a) Centrálny sklad (central storage),
b) Jeden alebo viac centrálnych procesorov,
c) Kanály (channels, sieťová karta je súčasťou CSS).

Súčasťou boxu sú samozrejme aj iné veci, ako napríklad chladiace sústavy, ktoré môžu byť vzduchové alebo vodné, batérie pre záložné napájanie, podporne elementy (support element, SE) a iné. Nenachádzajú sa tu žiadne I/O zariadenia ako napríklad disky. Akékoľvek periférne zariadenie komunikuje s mainframom cez CSS (channel subsystem), sú umiestnené mimo box a musia byť objednané zvlášť. Všetky súčiastky, ktoré sa nachadzajú v boxe sú redundantné. To znamená, že sú minimálne dvakrát v boxe, pre prípad, že sa nejaká pokazí, aby sa automaticky nahradila (Rogers a Salla, 2010).

Podporný element (SE) je centrálnym bodom kontroly v každom mainframovom systéme. Od systémov **Mainframe z9** je realizovaná prostredníctvom ThinkPad laptopu, ktorý je natrvalo namontovaný do mainframe boxu a nemôže byť odstránený.

I/O kanále sú súčasťou CSS, ktoré poskytujú pripojenie pre výmenu dát medzi servermi a externými zariadeniami (tlačiarne, disky, pásky, robot pásky, terminále). Obsah mainframe boxu vyobrazuje nasledujúci obrázok.



Obrázok 6: IBM Mainframe Box (Zdroj: vlastný)

Logicke partície - LPAR

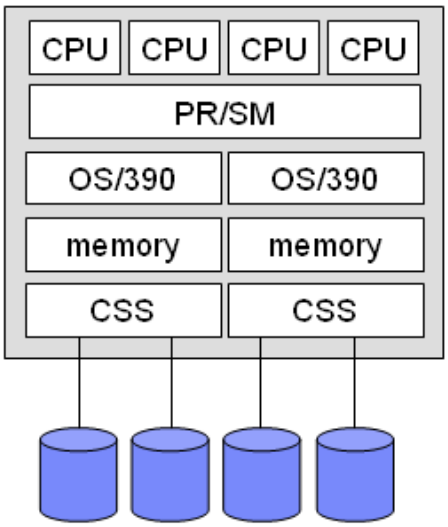
Takmer všetky mainframy majú schopnosť behu (alebo host-u) viac operačných systémov a tým pôsobiť nie ako jediný počítač, ale ako množstvo virtuálnych strojov. V tejto úlohe, jediný mainframe môže nahradiť desiatky alebo dokonca aj stovky menších serverov, znižuje tým náklady na správu a administratívne náklady a zároveň poskytuje výrazne lepšiu škálovateľnosť a spoľahlivosť (Rogers a Salla, 2010).

Moderné mainframy (IBM zSeries) ponúkajú tri úrovne virtualizácie:

a) Logické oddiely (LPAR),
b) Virtuálne stroje (za pomoci z/VM operačného systému),
c) Prostredníctvom svojich operačných systémov (hlavne z/OS, ale aj Linux a Java).

Procesorové zdroje (PR) a systém management (SM) umožňuje spustiť viac (2-60) logický partícií (LPAR). Procesory a kanály môžu byť zdielané medzi LPAR-mi alebo dedikované na konkrétny LPAR. Každý LPAR má svoju vlastnú kópiu operačné systému.

LPAR možno upravovať individuálne a nové môžu byť dynamicky pridávané. LPAR sa správa ako samostatný server.



Obrázok 7: Logická partícia LPAR (Zdroj: Rogers a Salla, 2010)

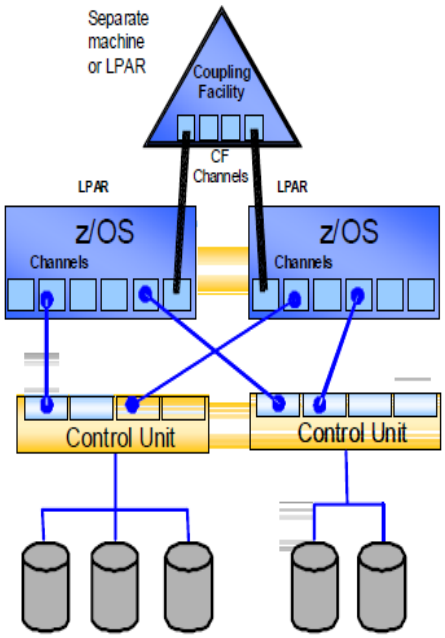
LPAR znamená logický oddiel a je výkonný hardware a firmware funkcia, ktorá je implementovaná vo všetkých mainframoch. Za pomoci tejto funkcie je možné vytvoriť partície medzi ktorými môže byť zdielané procesory a I/O subsystémy. Pamäť nie je zdielaná a musí byť dedikovaná pre každý LPAR zvlášť.

Paralelný sysplex (parallel sysplex)

Je to clusteringová technológia, ktorá umožňuje súbežný prístup pre čítanie a zápis k zdielaným dátam zo všetkých procesných systémov bez vplyvu na výkon alebo integritu dát v konfigurácii. Výhody paralelného sysplexu sú (Rogers a Salla, 2010):

a) Dynamická záťaž systému je rozložená cez všetky systémy,
b) Vylepšuje dostupnosť pre plánované a neplánované výpadky,
c) Zobrazuje multisystémové prostredie ako jeden logický zdroj.

Technológia paralelného sysplexu pomáha zabezpečiť neustálu dostupnosť. To umožňuje prepojiť 32 LPARov dohromady. Sysplex dokáže dynamicky pridávať a meniť systém bez jediného bodu zlyhania. Za pomoci state-of-art cluster technológie, viaceré z/OS image môžu pracovať efektívnejšie a dosiahnuť tak 99,999% dostupnosť.



Obrázok 8: Paralelný sysplex (Zdroj: Rogers a Salla, 2010)

2.2.8 Operačný systém z/OS

Najrozšírenejší operačný systém pre IBM Mainframe je z/OS. Disponuje 64-bitovou architektúrou a ideálne sa hodí pre spracovanie veľkých záťaží pre mnoho súčasných užívateľov. Je dizajnovaný pre (Rogers a Salla, 2010):

a) Obsluhovanie tisícok užívateľov súčasne,
b) I/O intenzívne výpočty,
c) Spracovanie veľmi veľkých pracovných záťaží,
d) Zabezpečené spustenie kritických aplikácií.

Vzhľadom k tomu, že ťaží z viac ako 40 rokov inovácií, jeho dizajn ponúka veľmi stabilné a bezpečné prostredie, s priemerným „up“ časom 99,999%. To znamená, že z/OS toleruje len 5,26 minúty „downtime“ za rok a tým je to najspoľahlivejší operačný systém. z/OS má v sebe priamo zabudovaný UNIX systém, ktorý sa nazýva OMVS.

Operačný systém z/OS ponúka viac ako 70 funkcií, ako napríklad:

a) z/OS XML,
b) komunikačný server, ktorý umožňuje šifrovanie siet ako IPSec,
c) podporuje viacero programovacích jazykov – Java, PHP, Perl, Cobol, Fortran, PL/1, Rexx, Assembler, správu úložísk, atď.

Operačný systém z/OS používa koncept virtuálnych pamätí a nazýva sa MVS (multiple virtual storage). Virtuálna pamäť sa sama o sebe nazýva adresné miesto (address space) a je ilúzia vytvorená z/architektúrou spoločne so z/OS a definuje, že programu sa zdá, že má viac centrálnych pamätí ako skutočne má.

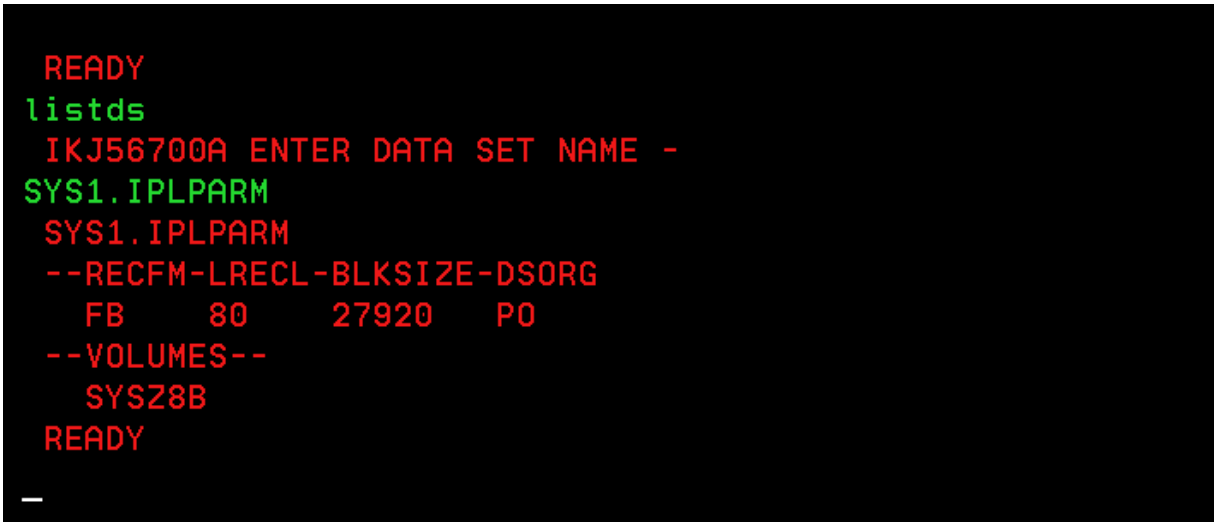
Charakteristiky adresného miesta:

a) Každá štartovacia úloha (STC – started task, napr. dlho trvajúca služba ako TCP/IP, DB2 atď.) spúšťa svoje vlastné adresné miesto,
b) Každý batch iniciátor spúšťa svoje vlastné adresné miesto (jeden batch iniciátor spúšťa len jednu batch úlohu)
c) Každý TSO užívateľ má svoje vlastné adresné miesto
d) Každé adresné pole má svoje vlastné unikátne číslo,
e) Nové adresné pole sa vytvára v momente, keď sa naštartuje nová úloha, alebo po nalogovaní užívateľa na systém.
f) Adresné polia sú navzájom izolované od seba, čiže chyba v jednom adresnom poli neovplyvňuje druhé adresné pole alebo operačný systém sám o sebe.

Interakcia s operačným systémom z/OS

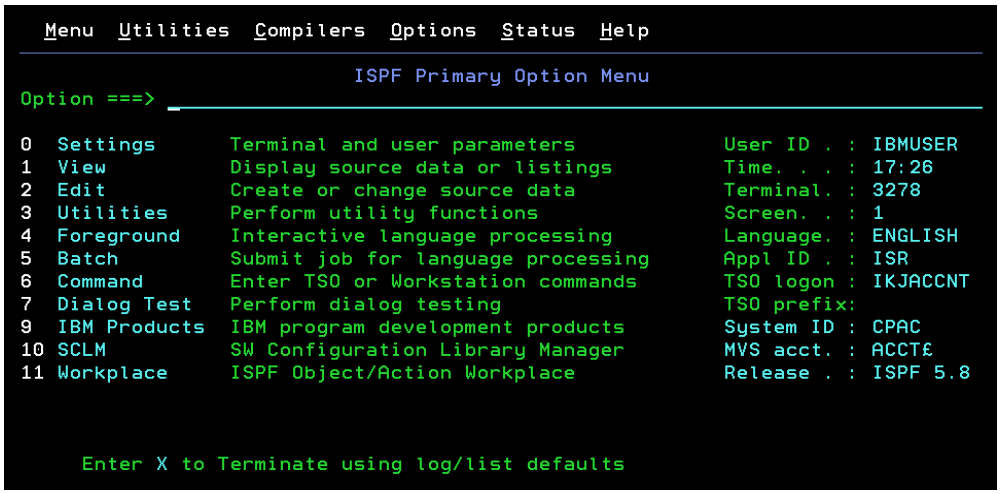
Slovo interakcia tu znamená, že užívatelia (niekedy desiatky tisíc z nich súbežne) môžu používať systém cez priamu interakciu ako sú príkazy v menu v štýle užívateľského rozhrania. Táto kapitola poskytuje prehľad o nich. Sú to (Kettner, 2011):

a) **Time Sharing Option/Extensions (TSO/E)** – umožňuje užívateľovi sa nalogovať do systému z/OS a používať limitované množstvo základných príkazov v natívnom móde



Obrázok 9: TSO/E (Zdroj: vlastný)

b) **Interactive System Productivity Facility (ISPF)** – ISPF je menu-ové rozhranie, ktoré slúži pre interakciu užívateľa so z/OS. Poskytuje nástroje, editor a ISPF aplikácie pre užívateľa v rozsahu povolenom podľa rôznych bezpečnostných kontrol. Užívateľ ISPF má prístup k väčšine z/OS systémových funkcií (to sa samozrejme odvíja na vyššie spomenutej bezpečnostnej kontrole). ISPF môžeme vnímať ako rozhranie pre správu systému a vývoj rozhrania pre tradičné z/OS programovanie.



Obrázok 10: ISPF menu (Zdroj: vlastný)

c) **z/OS UNIX interaktívne rozhranie** – poskytuje príkazové rozhranie pre prostredie z/OS UNIX. Je možné pristúpiť pomocou prihlásenia cez TSO/E alebo pomocou diaľkového prihlásenia TCP/IP (rlogin)

Súborový systém z/OS

V prostredí mainframov nie sú súborové systémy veľmi jednoduché. V skutočnosti nie sú založené na rovnakej technológii, ktorá sa používa v distribuovaných prostrediach systémov. Súborový systém z/OS je veľmi špecifický a komplexný. Na rozdieľ od tradičných systémov ako UNIX alebo Windows, ktorý používa unikátny hierarchický súborový systém, z/OS používa niekoľko typov dát, ktoré sa od seba líšia, či už štruktúrou alebo prístupovou metódou, aby mohli byť efektívnejšie v závislosti na ich využití. Namiesto toho používa katalógový systém, odkazujúci, že každý dátový súbor v systéme je k dispozícii. Názov datasetu musí dodržiavať určité pravidlá (Kettner, 2011):

a) Môže byť zložený až zo 44 znakov,
b) Pozostáva z častí nazvaných kvalifikátory, ktoré sa oddeľujú bodkou,
c) Každý kvalifikátor môže byť 8 znakov dlhý, napríklad: SYS1.ZOS113.PARMLIB.

Rozoznávame niekoľko typov datasetov, ale tieto tri sú najčastejšie:

a) **Sekvenčné datasety** – nazývané PS (physical sequential) – jedná sa o veľmi jednoduchý typ, ktorý môže byť videný ako súbor, tvorený sekvenciou jedného alebo viacerých záznamov,
b) **Delené datasety** – nazývane PDS (paritioned datasets) - môžeme ich vidieť ako zložky (folders), ktoré pozostávajú z kolekcie sekvenčných datasetov,
c) **VSAM – Virtual Storage Access Method** – tento termín sa používa pre špeciálne údaje ako aj pre súvisiace prístupové metódy. Vďaka ich štruktúre, VSAM súbory neuveriteľne zlepšujú výkon prístupu čítania. Tento typ datasetov používajú napr. DB2 a IMS databáze. Jedná sa o jednej z najzložitejších typov datasetu.

Keďže operačný systém z/OS nepoužíva hierarchický systém a jeho systém súborov nemá koncepciu „root-a“, musí využívať iný spôsob alokácie dát. Tento spôsob sa nazýva „katalóg. Katalóg opisuje atributy datasetov, vrátane ich umiestnenia. Rozoznávame 2 typy katalógov:

a) **Master katalóg** – (master catalog) – kde sú systémové datasety ako napr. kritické zaťaženie modulu,
b) **Užívateľský katalóg** – ktorý sa vzťahuje na ostatné špecifické datasety, ktoré sú nadefinované v podnikovej štruktúre a politike. Tieto katalógy sú katalogizované v hlavnom master katalógu.

2.3 Produkty z/OS

2.3.1 JCL – Job Control Language

JCL je skriptovací jazyk, ktorý inštruuje systém ako má spustiť daný program. JCL je väčšinou opis dávkového programu (batch program) a opisuje jeho parametre, zdroje vstupov a výstupov a i. V rámci každej úlohy sú príkazy usporiadané do krokov úlohy. Každý krok pozostáva z príkazov nutných pre beh jedného programu. Poznáme tri základné príkazy JCL (Kettner, 2011):

a) **Parameter JOB** – prvá JCL inštrukcia, ktorá poskytuje názov JCL a dôležité informácie bezpečnostného, administratívneho a identifikačného rázu. Slúži takisto k vyznačeniu začiatku úlohy.
b) **Parameter EXEC** – tento príkaz označuje začiatok každého kroku programu a v jeho argumentoch špecifikujeme, ktorý program sa má spustiť.. Úloha môže týchto krokov zahrňovať maximálne 255, vyšší počet krokov vyvolá chybu programu.
c) **Parameter DD** – definuje datasety, vstupné a výstupné zdroje, ktoré program potrebuje. Každa DD karta je spojená s konkrétnym EXEC parametrom a následne s jednotlivými krokmi JCL-kódu. Jedná sa o najzložitejší parameter JCL.

Nasledujúci príklad JCL nám skopíruje dataset „SYS1.IPLPARM“ zo „SYSZ8B“ úložiska (DASD) do nového datasetu „SYS1.IPLPARM“ na „TARG00“ úložisku, použitím IEBCOPY funkcie.

```
//JOBCOPY1   JOB 1,'IEBCOPY',CLASS=A,MSGCLASS=W,TIME=1440,
//            MSGLEVEL=(1,1),NOTIFY=&SYSUID,REGION=0M
//STEP1 EXEC PGM=IEBCOPY
//SYSPRINT   DD SYSOUT=*
//SYSUT1     DD DSNAME=SYS1.IPLPARM,UNIT=3390,VOL=SER=SYSZ8B,
//           DISP=SHR , SYSUT2 DD DSNAME=SYS1.IPLPARM,UNIT=3390,
//           VOL=SER=TARG00,  LIKE=SYS1.IPLPARM,DISP=(NEW,KEEP)
```

Obrázok 11: JCL parametre (Zdroj: vlastný)

2.3.2 Prostriedok pre zobrazovanie a prehľadávanie manipulačného priestoru - SDSF (Spool Display and Search Facility)

Jedná sa o veľmi mocný nástroj pri práci so z/OS, vďaka ktorému, môžu administrátori systému kontrolovať v reálnom čase I/O prostriedky a úlohy prechádzajúce

systémom. Pomocou SDSF je možné zadávať taktiež príkazy JES2 a celému systému (pokiaľ k tomu máme oprávnenia) (Kettner, 2011).

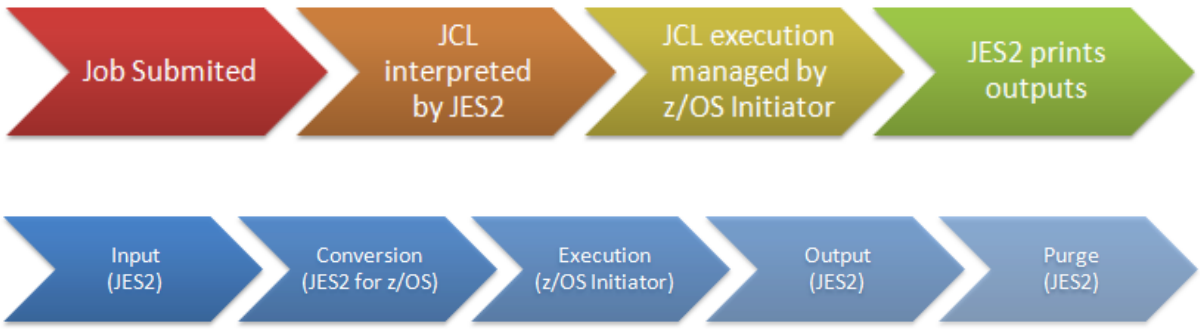
SDSF umožňuje správcom:

a) Zobraziť výstup úloh,
b) Ovládať systémové príkazy,
c) Spravovať úlohy,
d) Monitorovanie práve prebiehajúcich úloh,
e) Zobrazenie celého systémového logu a vyhľadávanie ľubovoľného reťazca v ňom,
f) Proces riadenia úloh (držanie, uvoľnenie, zrušenie, zmazanie úlohy).

2.3.3 JES2 (Job Entry Subsystem)

JES2 sa stará o úlohu (workload) pred a po priebehu programu. Jeho úlohou je (Kettner, 2011):

a) Prijať úlohu ku spracovaniu,
b) Prevedenie do strojovo čitatelnej podoby,
c) Postupne odosielať úlohu k vyhodnoteniu,
d) Kontrolovať a spracovať výstup,
e) Odstrániť výstup zo systému.



Obrázok 12: Popis cesty úloh systémom za pomoci JES2 (Zdroj: Kettner, 2011)

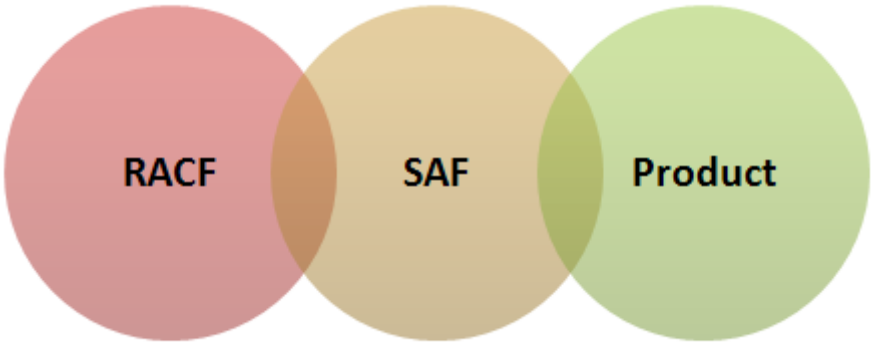
Obrázok nám popisuje cestu úlohy systémom, ktorý používa JES2. Úloha prejde šiestimi fázami – vstupom (input) -> prevodom (conversion) -> spracovaním (processing) -> výstupom (output) -> uložením (hard-copy) -> zmazaním(purge).

2.3.4 Bezpečnostný server na mainframe – RACF

Bezpečnosť je potrebné mať na kritických systémoch, keďže na mainframoch bežia systémy, ktoré sa zaoberajú citlivými a dôvernými dátami, ako sú bankové účty, poistenia alebo citlivé dáta o zákazníkoch. Preto je potrebné mať veľmi spoľahlivú aplikáciu, ktorá obmedzí všetky prístupy.

z/OS správa bezpečnosti pozostáva z 2 hlavných komponent (Kettner, 2011):

a) **Bezpečnostné autorizovacie zariadenia** (SAF, The Security Authorisation Facility) – je funkcia jadra, ktorá identifikuje bezpečnostné príslušné udalosti a ako sa pristupuje ku zdrojom,
b) **Zariadenie pre kontrolu zdrojov prístupu** (RACF, The Resource Access Control Facility) – je subsystém bežiaci v adresnom priestore systému, v ktorom je priamo nadefinovaný zákaz alebo zamietnutie prístupu k zdroju. Pracuje s rolami, ktoré sú nadefinované formou skriptu a uložené v RACF databáze.



Obrázok 13: RACF komponenty (Zdroj: Kettner, 2011)

2.3.5 Databázové systémy na mainframe

Na Mainframoch sa môžeme stretnút s rôznymi databázovými systémami. Najrozšírenejšie sú (Kettner, 2011):

a) **Databázový systém DB2** – relačná databáza firmy IBM, ktorá je optimalizovaná práve pre z/OS systémy, keďže viacero programov dokáže pristupovať k rovnakým dátam simultánne, použitím SQL jazyka. Tablespace môže byť do výšky až 16 TB.

A jedna zo zaujímavých vecí, ktorou DB2 je, že disponuje XML integráciou. Je najčastejšie používaná,

b) **IMS / DB** – IBM Information Management System – spája hierarichké databázy a manažérsky informačný systém s rozsiahlymi možnosťami spracovania transakcií,
c) **Oracle for z/OS** – integračné produkty, ktoré umožňujú participovať z/OS komponenty ako CICS, IMS s aplikáciou Oracle Database.

2.3.6 Programovací jazyk REXX

Programovací jazyk REXX (REstructured eXtended eXecutor) bol vyvinutý v rokoch 1979 až 1982. Štandard REXXu bol vydaný až v roku 1996. Pôvodne to mal byť skriptovací jazyk pre ľubovoľnú systém (ako napr. dnešný Python). Má mnoho verzií, od tých pre mainframy až po varianty pre Windows, Amigy, Linux či Solaris. Od polovice deväťdesiatich rokov 20. Storočia (Hofta, 2008).

2.3.6.1 Charakteristiky REXX

Hlavnými charakteristikami REXXu sú hlavne (Hofta, 2008):

a) Intuitívnosť – ako príkazy používa bežné anglické slovíčka (SAY, DO, END ...),
b) Voľnosť formátovania – príkazy nie je nutné písať do určitých stĺpcov, je možné medzi ne vkladať medzery, je možné písať malými či veľkými písmenami atď.,
c) Mnoho zabudovaných funkcií,
d) Existencia chybových hlásení pri preklade,
e) Interpretačný jazyk – pred spustením nie je potreba programy prekladať,
f) Pokročila analýza vstupov – REXX dokáže rozlišovať na vstupe znaky a čísla a oddeliť ich,
g) Nedeklarujú sa premenné.

Programovací jazyk REXX pozostáva z inštrukcií, zabudovaných funkcií, vonkajších funkcií TSO/E a funkcií pracujúcich s dátami uloženými na zásobníku. Inštrukciami môžu byť kľúčové slová, priradenia, labely, null a príkazy (Hofta, 2008).

Programy napísané v programovacom jazyku REXX sa nazývajú exeky (execs). Zdrojové kódy exekov zapisujeme štandradne do data setov. Každý data set obsahujúci nejaky exek by mal podľa doporučení IBM začínať riadkom s následujúcim komentárom (Hofta, 2008):

/********** REXX **********/

Nasledujúci REXX program má za úlohu pravidelne kontrolovať (každý deň), či majú úlohy zapnutý FLAG, ktorý znamená, že autooperátor má kontrolu pod úlohou.

Pracujeme tu s funkciou opsvalue s parametrom „f“, ktorým skontrolujeme či existuje premenná v datasete GLOBAL.TT1.FLAGOFF. Pokiaľ existuje, funkcia má návratovú hodnotu I a načíta obsah globálnej premennej do premennej stcold. Tam sú uložené názvy started taskov (STC) z predchádzajúceho dňa.

Ďalším krokom je nadefinovať si premenné. Do started tasku (STC) budeme ukladať názvy, pre ktoré bude v tabuľke STCTBL v stĺpci MODE iná hodnota ako „ACTIVE“ – line 16. Zároveň, pokiaľ už bol názov zahrnutý v premennej STCOLD, ktorú sme si načítali na začiatku REXXu, tak naplníme túto hodnotu aj premennou alrt.

Posledným krokom je vykonanie aktualizácie premennej GLOBAL.TT1.FLAGOFF, do ktorej uložíme aktuálny zoznam STC, kde nemajú v stĺpci node „ACTIVE“. Zistíme, či je premenná alarm prázdna. Pokiaľ nie je, tak sa vytvorí ticket, do ktorého vložíme preddefinovaný text a obsah premennej obsahujúci názvy tasku, u ktorého je FLAG vypnutý.

```
 File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
VIEW        YKZVSW.CA.OPSMVS.TEGLOBAL.REXX(FLACHECK) - 01.3 Columns 00001 00072
Command ===>                                                 Scroll ===> CSR
****** ***************************** Top of Data ******************************
==MSG> -Warning- The UNDO command is not available until you change
==MSG>           your edit profile using the command RECOVERY ON.
000001 /* REXX                                                              */
000002 /*******************************************************************/
000003 /* GET LIST STC WITH OPFLA OFF FROM VARIABLE')                       */
000004 stcold = OPSVALUE('GLOBAL.TT1.FLAGOFF','F')
000005 if stcold = 'I' then pull stcold
000006
000007 /*******************************************************************/
000008 /* GET LIST STC WITH OPFLA OFF  (alrt)                               */
000009 stc = ''
000010 alrt = ''
000011 Address SQL "SELECT NAME MODE FROM STCTBL"
000012 Do A = 1 to NAME.0
000013  If mode.a = 'ACTIVE' then NOP
000014                       else
000015           Do
000016           stc = stc' 'name.a
000017           if POS(name.a,stcold) > 0 then alrt = alrt' '|| name.a
000018           End
000019   End
000020
000021 /*******************************************************************/
000022
000023 thissys = OPSINFO('SMFID')
000024
000025 ? = OPSVALUE('GLOBAL.TT1.FLAGOFF','U',stc)
000026
000027 if alrt='' then NOP
000028             Else Do
000029                  say 'STC with flag off: 'alrt
000030                  alrtdesc = ' Please check following STC - 'alrt
000031                  alrtdesc = alrtdesc 'why is flag off 24 hours'
000032                  key = alrt'LT'
000033                  alertxt = thissys' 3 'key '!!' :'alrtdesc
 F1=Help      F2=Split     F3=Exit      F4=Expand    F5=Rfind     F6=Rchange
 F7=Up        F8=Down      F9=Swap      F10=Left     F11=Right    F12=Cancel
 File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
VIEW        YKZVSW.CA.OPSMVS.TEGLOBAL.REXX(FLACHECK) - 01.3 Columns 00001 00072
Command ===>                                                 Scroll ===> CSR
000034                   Address OSF
000035                   "OI A2PEM" alertxt
000036                End
000037 /*20140526/Pavelric: FLACHECK - check the status of AOFLAG            */
****** ***************************** Bottom of Data ***************************
```

Obrázok 14: Ukážka kódu v programovacom jazyku REXX (Zdroj: vlastný)

3 Analýza súčasného stavu

3.1 Proces vzniku incidentov na IBM Mainframe

Táto časť opisuje, akým spôsobom sú alerty distribuované z automatizovaného riadenia udalostí (event management) OPS/MVS do AP a následne do BEMu. Ak chceme zabrániť zvýšenému konzumovaniu sieťových prvkov novými alertami, bol vytvorený filtrovací mechanizmus alertov, tzn. že ak stovky udalostí pre ten istý problém sú vytvorené na LPARe, tak len jeden bude poslaný do BEM v danom intervale (momentálne nastavený na 30 sekúnd). Ak je vytvorených minimálne 10 eventov pre rovnaký problém, tak bude odovzdaný do BEMu. Miesto, kde sú eventy uložené na Mainframe, sa nazýva OPS RDF tabuľka – to znamená, že ak eventy nie sú posielané do BEMu, napríklad pre sieťový problém, sú stále viditeľné v systémovom logu – syslog. RDF tabuľka taktiež prežije aj IPL.

3.1.1 Tvorba incidentov na strane Mainframe

Na Mainframoch sa riešenie skladá z dvoch REXX skriptov a 4 pravidiel, ktoré manipulujú „heartbeat“ funkcie a posúvajú alerty produkované rôznymi OPS pravidlami a/alebo REXX skriptami.

3.1.1.1 Všeobecný popis pravidiel a Rexx skriptov, ktoré bežia na strane Mainframe

Tabuľka 3: Pravidlá na strane Mainframe (Zdroj: vlastný)

Meno pravidla v AORULES ruleset	Účel
ALRT2AP	TOD pravidlo – číta riadky z tabuľky SEND2BEM a volá Rexx SEND2AP
AP2OPS	MSG pravidlo – správa je posielaná Rexx-om AP2OPS, ktorý beží na strane AP – aktualizuje variabilné zložky časovou pečiatkou – ak je žiadosť prijatá AP – súčasť hertbeat funkčnosti
HEARTBEAT	TOD pravidlo – kontroluje premenné nastavené AP2OPS pravidlom – ak pripojenie nie je vyhodnotené, že je aktívne, tak následne generuje 4 časové správy na konzolu
OPS2AP	TOD pravidlio – spúšťa AP2OPS Rexx na strane AP – aktualizuje premenné časovou pečiatkou a očakáva vrátenie tej istej časovej pečiatky pravdilom AP2OPS. Je súčasťou “heartbeat”

Tabuľka 4: REXX skripty na strane Mainframe (Zdroj: vlastný)

REXX meno v TEGLOBAL.REXX	Účel
SENDALRT	Ukladá alerty do RDF tabuľky
SEND2AP	Je volaná TOD pravidlom ALRT2AP. Vybere všetky riadky z RDF tabuľky SEND2BEM. Ak počet alertov, priradený k tomu istému problému v rovnakom intervale dosiahne počet 10, tak alert je vytvorený.

3.1.1.2 Spôsob komunikácie

Ak chceme mať na nadviazané spojenie na systém Windows Server, následovné veci musia byť nakonfigurované na strane Mainframe:

a) OPS/MVS musí mať nakonfigurovaný Multi System Facility (MSF), ktorá sa používa na pripojenie k CCI aplikácii,
b) CCI aplikácia musí byť nasadená na rovnaký LPAR, na ktorom beží OPS (CCI je súčasť STC: ENF),
c) Prostredníctvom CCI sa OPS pripojí k CCI na Windows Serveri.

3.1.2 Tvorba incidentov na strane Windows Server

Následovné veci musia byť nakonfigurované k tomu aby sme dosiahli funkčnej konektivity k BEMu:

a) CA Automation Point musí byť nainštalovaný a správne nakonfigurovaný,
b) MSEND nástroj je nasadený na rovnaký server ako CA Automation Point,
c) Musíme mať otvorený firewall k BEM serverom.

3.1.2.1 CA Automation Point

CA AP zachytáva eventy z OPS/MVS. Je schopný spracovávať Rexx skripty. Skripty, ktoré musia byť prítomné na strane AP:

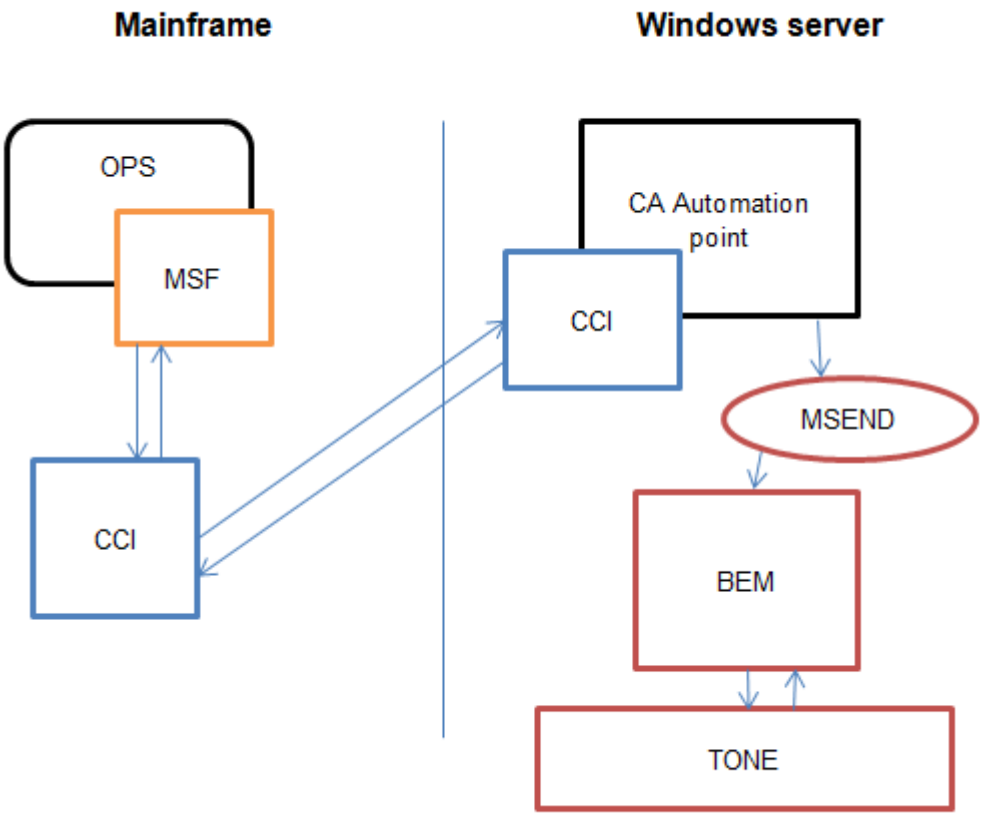
Tabuľka 5: Skripty na strane Windows Server (Zdroj: vlastný)

OPS2AP	Súčasť "HeartBeat" mechanizmu. Vracia časové známky na LPARy.
AP2BEM	Konvertuje OPS formát alertov na MSEND formát.

3.1.2.2 Nástroj MSEND

Jedná sa o nástroj, ktorý poskytuje presun správ do BEM a garantuje, že tieto správy budú doručené.

3.1.3 Grafické zobrazenie alert procesu



Obrázok 15: Zobrazenie alert procesu (Zdroj: vlastný)

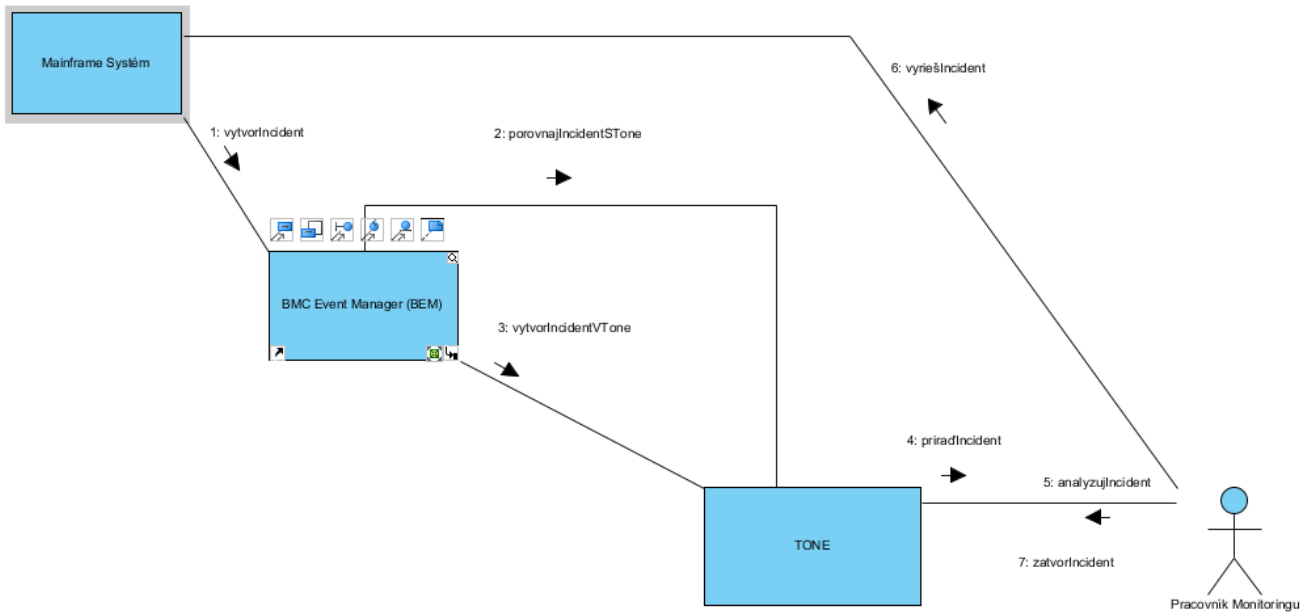
Kontrola konektivy na kritickom systéme za pomoci príkazu OPBEM STATUS.

```
000280   OPBEM STATUS
000080   OPSNOTIFY == OPBEM == SYS1.ENATOR ========================
000080   OPSNOTIFY Connection to Automation Point EVMINTE5 is ACTIVE
000080   OPSNOTIFY ================================================== 471
000080             current alerts which should be checked
000080             ----------------------------------------------
000080   OPSNOTIFY  No alerts are waiting to be processed to BEM€
```

Obrázok 16: Príkaz OPBEM STATUS (Zdroj: vlastný)

3.1.4 Komunikačný diagram vytvorenia incidentu

Nasledujúci komunikačný diagram vyjadruje životný cyklus incidentu pred optimalizáciou, ktorý začína jeho vytvorením na systéme až po jeho uzatvorenie pracovníkom v monitorovacej aplikácii. Komunikačný diagram bol vytvorený v trial verzii programu Visual Paradigm.



Obrázok 177: Komunikačný diagram vytvorenia incidentu (Zdroj: vlastný)

Postup vytvorenia incidentu:

1. Na systéme vznikol incident, ktorý sa následne vytvorí v prostredí BMC Event Manager,
2. BMC Event Manager analyzuje daný incident, či už nie je otvorený v monitorovacej aplikácii TONE, aby sa predišlo duplovaniu incidentov,
3. Pokiaľ incident nie je ešte vytvorený v monitorovacej aplikácii TONE, tak sa tu vytvorí,
4. Pracovníkovi monitoringu je následne novo vytvorený incident priradený,
5. Pracovník monitoringu analyzuje daný incident,
6. Nasleduje riešenie incidentu – pokiaľ je možné, pracovníkom monitoringu,
7. Zatvorenie incidentu v monitorovacej aplikácii TONE.

3.2 Štatistické vyjadrenie vytvorených incidentov

V tejto časti budú znázornené jednotlivé údaje o vytvorených incidentoch. Do úvahy budú brané len tie najkritickejšie LPAR-y a nie tie, ktoré sú určené na vývoj a testovanie. Pre zachovanie diskrétnosti jednotlivých zákazníkov, sú jednotlivé LPAR-y pomenované fiktívnymi kombináciami znakov a písmen.

3.2.1 Štatistika vytvorených incidentov na kritických systémoch

Tabuľka vyjadruje vytvorené incidenty za obdobie posledných šesť mesiacov. V stĺpcoch sú nadefinované jednotlivé kritické systémy, na ktorých boli incidenty vytvorené a v riadkoch sú operačné mainframe skupiny, ktorým boli dané problémy pridelené.

Pracovné skupiny, ktoré riešia problémy na IBM Mainframe:

a) **MF_OSTRAVA** – pracovná skupina, ktorá predstavuje spojený prvý a druhý level servisnej podpory a stará sa o vyriešenie prevažnej väčšiny incidentov,
b) **MF_APP_GOT** – skupina, ktorá sa zaoberá dávkovými (batch) úlohami,
c) **MF_AUTOMATION** – automation tím sa stará o automatizáciu na IBM Mainframe,
d) **MF_CAPACITY** – tím CAPACITY tvoria servisní architekti, ktorí riešia predovšetkým projekty a o servis sa starajú minimálne,
e) **MF_DBDC** – DBDC tím sa zaoberá databázami (DB2, IMS), upgradami týchto databáza na produkčných a testovacích systémoch, rieši problémy s online transakciami CICS a iné,
f) **MF_NET** – network tím sa zaoberá konektivitou a problémami s ňou spojenou na systémoch.

Z nasledujúcej tabuľky vidíme, že najviac incidentov bolo pridelených skupine MF_OSTRAVA a to z dôvodu, že táto skupina je v ITIL systéme uvedená ako druhý level, to znamená, že 99% všetkých incidentov prechádza práve touto skupinou. Ostatné skupiny (MF_APP_GOT, MF_AUTOMATION, MF_CAPACITY, MF_DBDC, MF_NET, MF_OS) sú skupiny tretieho levelu a tu sa riešia incidenty, ktoré neboli uzavreté na druhom leveli. Môže to byť z dôvodu nutnosti vyššej kvalifikácie na vyriešenie daného problému, alebo z dôvodu, že pracovník na druhom leveli nemal bezpečnostný prístup k aplikácií, ktorá by videla na vyriešenie problému.

DP_CreatedTicketToGroups

Assignment group	Affected CI													
	A084.ENATOR	asys.plawm1	beta.enator	csy2.ilmplex	csys.ilmplex	p001.enator	pr22.enator	sya1.tita	sys1.enator	sysa.plx1	tok1.tokplex	tok2.tokplex	Other	Total
MF_APP_GOT	5	0	0	0	0	2	9	0	943	0	0	0	0	959
MF_AUTOMATION	13	8	5	13	9	7	28	6	13	10	4	10	1	127
MF_CAPACITY	2	1	0	0	1	1	3	0	0	1	1	0	0	10
MF_DBDC	35	4	5	1	1	29	60	0	7	10	0	1	0	153
MF_NET	0	0	2	0	0	0	0	0	0	0	2	0	0	4
MF_OS	6	1	3	0	2	4	9	0	1	9	2	0	0	37
MF_OSTRAVA	638	659	723	397	616	858	1,807	1,164	2,425	311	1,145	719	324	11,786
Total	699	673	738	411	629	901	1,916	1,170	3,389	341	1,154	730	325	13,076

Obrázok 18: Počet vytvorených ticketov u jednotlivých skupín (Zdroj: vlastný)

Tabuľkové vyjadrenie vytvorených incidentov a počtové a percentuálne vyjadrenie pracovnej skupiny, na ktorú boli priradené.

Assignment group	Count	Percentage of Incidents
MF_OSTRAVA	11,787	90.14%
MF_APP_GOT	959	7.33%
MF_DBDC	153	1.17%
MF_AUTOMATION	127	0.97%
MF_OS	37	0.28%
MF_CAPACITY	10	0.08%
MF_NET	4	0.03%
Total	**13,077**	

Obrázok 19: Počet vytvorených incidentov (Zdroj: vlastný)

Následujúca tabuľka vyjadruje počet vytvorených incidentov na jednotlivých kritických systémoch, ich daný počet a percentuálne vyjadrenie.

Affected CI	Count	Percentage of Incidents
sys1.enator	3,391	25.93%
pr22.enator	1,916	14.65%
sya1.tita	1,170	8.95%
tok1.tokplex	1,154	8.82%
p001.enator	901	6.89%
beta.enator	738	5.64%
tok2.tokplex	730	5.58%
A084.ENATOR	699	5.34%
asys.plawm1	673	5.15%
csys.ilmplex	630	4.82%
csy2.ilmplex	411	3.14%
sysa.plx1	341	2.61%
Other	325	2.48%
Total	**13,079**	

Obrázok 20: Tabuľkové vyjadrenie počtu vytvorených incidentov na jednotlivých systémoch (Zdroj: vlastný)

3.2.2 Štatistika najviac vytváraných incidentov na kritických systémoch

Keďže je denne vytvorených veľké množstvo incidentov na každom systéme, či už kritickom alebo testovacom prípadne vývojárskom, pre potreby svojej štatistiky najviac vytváraných problémov som si vybral tri celkovo najvytváranejšie problémy, ktoré budem ďalej analyzovať, riešiť a optimalizovať.

Najčastejšie incidenty, ktoré sú vytvárane sú problémy s následovným popisom:

a) CICS - CICSMEN1 - DFHSM0133 P001MEN1 CICS IS UNDER STRESS (SHORT ON STORAGE ABOVE 16MB),
b) GSVX321W JOBMIPS = 776.5 (LIM 740.0) FOR JOB DB2PDBM1 MILLION INSTRUCTIONS PER SECOND (MIPS) LAST OVER 12 MIN,
c) DAYCHECK PLEASE CHECK BBMPAS - STATE IS NOT UP - VERIFY WHY IT IS NOT.

3.2.2.1 Incident „GSVX321W JOBMIPS = 776.5 (LIM 740.0) FOR JOB DB2PDBM1 MILLION INSTRUCTIONS PER SECOND (MIPS) LAST OVER 12 MIN“

Tento problém vzniká prekročením úlohy limitovaný počet JOB MIPS-ov. MIPS (Million Instruction Per Second) je jednotka výkonnosti počítača, ktorá udáva počet spracovaných inštrukcií za sekundu. Keďže MIPS-y sú predplatené zákazníkom, je nutné strážiť ich prekročenie. V tomto prípade databázová úloha pri vytváraní reportov spotrebovala väčšie množstvo MIPS-ov len po dobu niekoľko sekúnd, takže dopad nie je žiadny.

Daný problém nie je možné zautomatizovať, z toho dôvodu, že je vždy nutné „ľudským okom“ zohľadniť všetky okolnosti, za ktorých daný problém vznikol.

3.2.2.2 Incident „CICS - CICSMEN1 - DFHSM0133 P001MEN1 CICS IS UNDER STRESS (SHORT ON STORAGE ABOVE 16MB).“

CICS sú životne dôležité pre zákazníka. Problémy na produkčných systémoch, najmä v úradných hodinách, musia byť okamžite hlásene osobám, ktoré sú priamo zodpovedné za daný CICS. Podnety, na základe ktorých môže daný problém vzniknúť sú: vysoké využitie CPU, zle naprogramovaná transakcia v danom CICS, problémy s úložiskom, niekoľko dump-ov (kopírovanie na disky) v krátkom čase, atď.

Náš problém, je založený na nedostatku pamäti na spustenie transakcie a teda, že transakcie nemôžu byť spracované.

Riešenie tohto problému môže byť dvomi spôsobmi:

a) Dynamickým zvýšením alokácie pamäti pre spustenie danej transakcie
b) Manuálnym identifikovaním transakcie, ktorá problém spôsobuje a následné zistenie, či to je spôsobené zlým naprogramovaním alebo len dočasný výpadok.

3.2.2.3 Incident „DAYCHECK PLEASE CHECK ... - STATE IS NOT UP - VERIFY WHY IT IS NOT“.

Incidentov, ktorých bolo na systémoch vytvorených veľké množstvo, sú tzv. Daychecky. Jedná sa o vopred nadefinovaný REXX skript, ktorý každý deň v tú istú hodinu kontroluje preddefinovaný started task (STC), rôzne monitory a iné, či sú aktívne, pretože sú kritické pre mainframy systémy. Pokiaľ nie sú aktívne, čiže sú v nekorektnom statuse, tak obdržíme vytvorený incident a musíme diagnostikovať, prečo sa tak stalo. Korektným statusom máme na mysli statusy ACT / UP / UP alebo INA / DOWN / DOWN. Nekorektný status v našom prípade môže mať viacero dôvodov:

a) **INA / DOWN / DOWN** – situácia, ktorá vznikla manuálnym zadaním STOP príkazu,
b) **INA / FAILED / UP** – tento status nastáva pokiaľ sa užívateľovi vyskytla chyba (abend) počas štartovania,
c) **INA / DOWN / UP** – v tomto prípade užívateľ čaká na naštartovanie nášho rodičovského tasku.
d) **ACT / STARTING / UP** – situácia, kedy užívateľ čaká na naštartovanie danej štartovacej úlohy.

Práve tento incident budem v nasledovnej kapitole racionalizovať.

4 Návrh racionalizácie procesu riadenia incidentov na IBM Mainframe

4.1 Výber racionalizovaného incidentu

Pri detailnejšom skúmaní vytvorených incidentov na kritických systémoch sme zistili, že veľmi veľa problémov je vytvorených kvôli daychecku, a teda denná kontrola. Preto sme odfiltrovali zo všetkých ticketov len incidenty, ktorých popis problému obsahuje slovo „daycheck“. Vyšlo nám, že na kritických systémoch za posledných 6 mesiacov bolo spolu vytvorených 967 problémov, ktoré obsahovali nami vyhľadávané slovo „daycheck“. Na prvý pohľad sa môže zdať, že toto číslo pri celkovom počte 13076 vytvorených ticketov (tab. 6) nie je vysoké, ale treba brať do úvahy, že na systémoch vzniká vysoké množstvo rozdielnych typov chybových hlásení, správ atď.

Následovná tabuľka vyjadruje počet vytvorených incidentov, ktoré obsahovali slovo „daycheck“ a sú na kritických systémoch.

	Affected CI					
Assignment group	A084.ENATOR	p001.enator	pr22.enator	sya1.tita	sys1.enator	Total
CD_24MF	8	5	12	1	12	38
MF_AUTOMATION	1	0	2	0	0	3
MF_OSTRAVA	207	189	128	81	320	925
SD_BANK	1	0	0	0	0	1
Total	217	194	142	82	332	967

Obrázok 21: Daycheck filter (Zdroj: vlastný)

4.2 Racionalizácia incidentu „DAYCHECK Please check DDTPROC1 - STATE is not UP - verify why it is not!“.

Za pomoci filtrovania sme zistili, že najvyšší počet „daycheck“ incidentov je vytvorených na kritickom systéme SYS1 a dotazuje sa na komponentu DDTPROC1. V našej tabuľke môžeme vidieť, že z celkového množstva 967 ticketov, ktoré obsahovali slovo „daycheck“ bolo 177 prípadov práve pre nami racionalizovanú komponentu DDTPROC1.

	Affected CI	
Assignment group	sys1.enator	Total
CD_24MF	9	9
MF_OSTRAVA	168	168
Total	177	177

Obrázok 22: DDTPROC1 na systéme SYS1 (Zdroj: vlastný)

Pre bližšiu diagnostiku komponenty DDTPROC1, bolo nutné zadať na našom kritickom systéme sys1, príkaz OPLIST DDTPROC1, pre zobrazenie presnej informácie o danej komponente. V našom prípade sa jedná o Opertune for DB2, čo je komponenta, ktorá ponúka komplexné riešenie pre zmenu parametrov databáze DB2 a stará sa takisto o korekciu prevádzkových problémov, ktoré môžu ovplyvňovať výkon celej DB2 na operačnom systéme z/OS.

```
OPLIS DDTPROC1
TT1LI OPLIST
TT1LI NAME   CmdCnt   Real  Currnt   Desired Description
TT1LI +DDTPROC1   0   INA   DOWN     DOWN    Opertune for DB2
TT1LI OK, showing 1 of 100
```

Obrázok 23: Príkaz OPLIST na komponentu DDTPROC1 v systéme IBM Mainframe (Zdroj: vlastný)

Ako bolo uvedené v kapitole 3.2.2.3 Incident „DAYCHECK PLEASE CHECK ... - STATE IS NOT UP - VERIFY WHY IT IS NOT“ dôvodov pre korektný status, avšak neaktívnu komponentu môže byť niekoľko. V našom prípade ide o to, že komponenta Opertune for DB2 patrí a je vyvíjaná americkou spoločnosťou BMC Software. Keďže firma, v ktorej je moja diplomová práca vypracovávaná rozviazala kontrakt s touto spoločnosťou a prešla pod inú americkú spoločnosť CA Technologies, ktorá danú komponentu neobsahuje pod názvom DDTPROC1, ale jej názov je DB2MSTR. Táto komponenta je už plne aktívna a funkčná a vykonáva jej stanovenú činnosť. Z tohto dôvodu je nutné našu komponentu DDTPROC1 odstrániť z kritického systému SYS1, pretože je na systéme už zbytočná, nevykonáva žiadnu funkciu a kvôli nej sú neustále vytvárane incidenty zo systému.

O odstránenie komponenty DDTPROC1 zo systému sa postará automation team, z dôvodu, že ako jediný majú dostatočné bezpečnostné práva na jej odstránenie.

Na nasledujúcom obrázku č. 17, a teda po opätovnom zobrazení reportu o vytvorených ticketoch na našich kritických systémoch, môžeme vidieť, že celkový počet je 790 incidentov. Toto číslo je zoptimalizované, čiže zmenšené práve o počet 177 incidentov, ktoré predstavovala komponenta DDTPROC1 na kritickom systéme SYS1 a bola zbytočná na našom systéme.

	Affected CI					
Assignment group	A084.ENATOR	p001.enator	pr22.enator	sya1.tita	sys1.enator	Total
CD_24MF	8	5	12	1	3	29
MF_AUTOMATION	1	0	2	0	0	3
MF_OSTRAVA	207	189	128	81	152	757
SD_BANK	1	0	0	0	0	1
Total	217	194	142	82	155	790

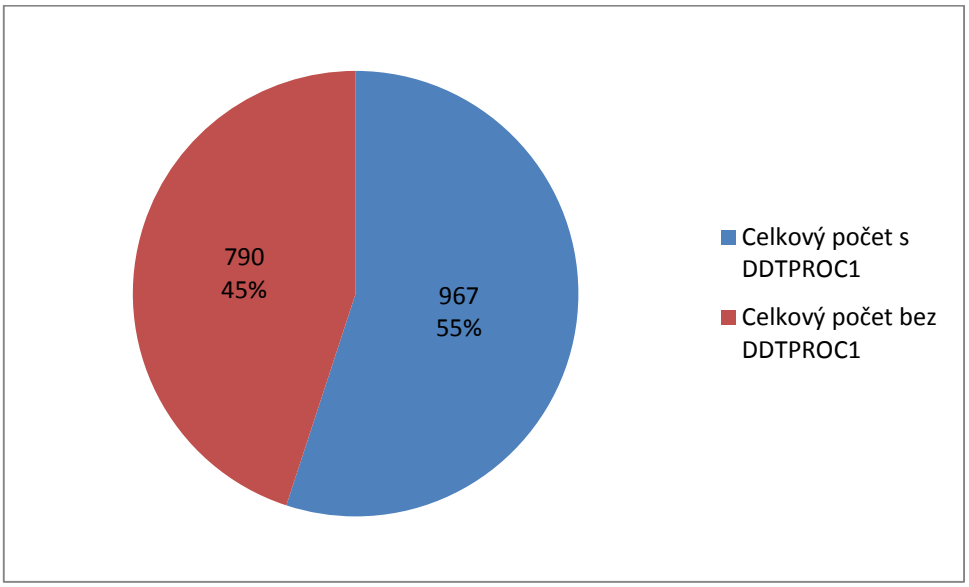
Obrázok 24: Počet "daycheck" incidentov po odstránení DDTPROC1 (Zdroj: vlastný)

Po zadaní príkazu OPLIST na systéme SYS1 vidíme, že daná komponenta sa už nezobrazuje, čo znamená, že bola zo systému vymazaná automation tímom.

```
OPLIS DDTPROC1
TT1LI OPLIST
TT1LI NAME   CmdCnt   Real Currnt   Desired Description
TT1LI OK, showing 0 of 90
```

Obrázok 25: Zobrazenie komponenty DDTPROC1 na systéme SYS1 (Zdroj: vlastný)

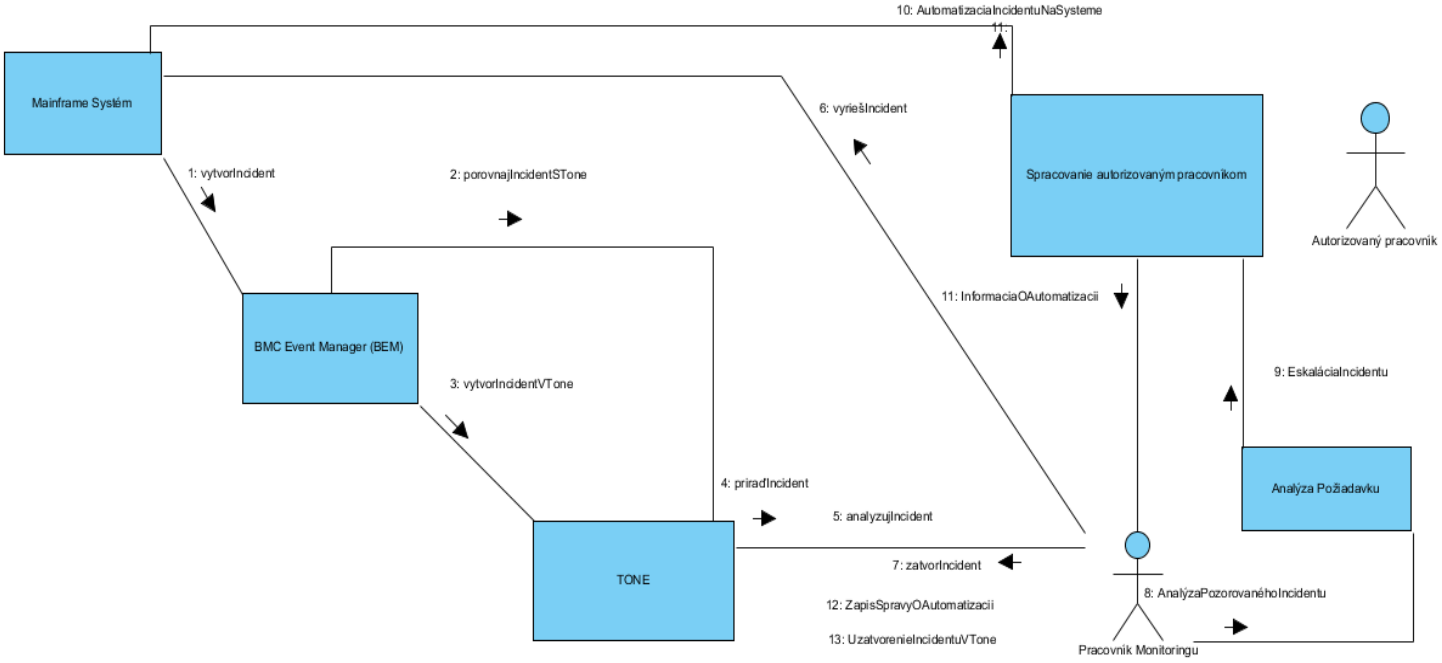
Nasledovné koláčové zobrazenie nám udáva celkový počet „daycheck“ incidentov s komponentou DDTPROC1 a bez komponenty DDTPROC1. Z grafu môžeme vyčítať, že pred odstránením našej komponenty bolo 55%, resp. 967 incidentov. Po jej odstránení zo systému, nám počet incidentov klesol na číslo 790 incidentov.



Obrázok 26: Grafické koláčové zobrazenie pred a po odstránení komponenty DDTPROC1 (Zdroj: vlastný)

4.2.1 Komunikačný diagram vytvorenia incidentu a jeho optimalizácia

Nasledujúci komunikačný diagram vyjadruje postup vytvorenia incidentu a jeho následná optimalizácia. Komunikačný diagram je vytvorený v programe Visual Paradigm.



Obrázok 27: Komunikačný diagram vytvorenia incidentu a jeho optimalizácia (Zdroj: vlastný)

Postup vytvorenia incidentu:

1. Na systéme vznikol incident, ktorý sa následne vytvorí v prostredí BMC Event Manager,
2. BMC Event Manager analyzuje daný incident, či už nie je otvorený v monitorovacej aplikácii TONE, aby sa predišlo duplovaniu incidentov,
3. Pokiaľ incident nie je ešte vytvorený v monitorovacej aplikácii TONE, tak sa tu vytvorí,
4. Pracovníkovi monitoringu je následne novo vytvorený incident priradený,
5. Pracovník monitoringu analyzuje daný incident,
6. Nasleduje riešenie incidentu – pokiaľ je možné, pracovníkom monitoringu,
7. Zatvorenie incidentu v monitorovacej aplikácii TONE.

Následná optimalizácia:

8. Pracovník monitoringu analyzuje incident, ktorý je možno zoptimalizovať,
9. Z dôvodu, že pracovník monitoringu nemá dostatočné autorizačné oprávnenie pre zautomatizovanie incidentu, je nutné escalovat daný proces pracovníkovi tretieho

levelu, ktorému sú predané dôležité informácie o systéme, na ktorom sa daný incident nachádza, problém vzniku, riešenie a spôsob zoptimalizovania daného incidentu,

10. Autorizovaný pracovník, ktorý už je oboznámený s problémom a takisto má dostatočné autorizačné práva, vyrieši problém na danom systéme,
11. Následne podáva autorizovaný pracovník informácie pracovníkovi monitoringu,
12. Pracovník monitoringu napíše zápis o optimalizácii v monitorovacej aplikácii TONE,
13. Pracovník monitoringu uzatvára incident v aplikácii TONE.

4.3 Ekonomické zhodnotenie – výpočet návratnosti investície

Návratnosť investície, skratka ROI – Return on Investment, vyjadruje čistý zisk alebo čisú stratu voči počiatočnej investícii a obvykle sa udáva v percentách. Okrem aritmetickej návratnosti investície existuje taktiež logaritmická návratnosť, ktorá sa používa pre výpočty vo vedeckých výzkumoch. Pre naše potreby bude vhodnejšia aritmetická návratnosť investície.

Pre výpočet návratnosti investície je potrebné počiatočnú investíciu a čistý zisk ktorý priniesla, vyjadriť konkrétnou peňažnou čiastkou. Návratnosť investície sa vzhľadom k počiatočnej investícii počíta buď pre jednotlivé obdobie alebo ako priemer z viac období.

Návratnosť investície bude vypočítaná podľa nasledujúceho vzorca:

$$\mathbf{ROI} = \frac{(\mathbf{čistý\ zisk} - \mathbf{počiatočné\ investície})}{\mathbf{počiatočné\ investície}}\ \mathbf{x\ 100}$$

Obrázek 28: Výpočet ROI (Zdroj: Adaptic 2015)

Nasledujúca tabuľka vyjadruje ekonomické vyhodnotenie na oddelení Mainframe. Keďže sa jedná o interné ukazovatele, čísla v tabuľke boli skreslené. Čistý zisk predstavuje sumu trojmesačného platu pracovníka, ktorá by bola ušetrená pri vykonanej optimalizácii. Investícia na optimalizáciu je takisto vyjadrená v tabuľke

Tabuľka 6: Výpočet ROI (Zdroj: vlastný)

	Obdobie za posledné 3 mesiace
Čistý zisk (external revenue)	100 000 CZK
Investícia	33 000 CZK

$$\text{ROI} = \frac{(100\,000 - 33\,000)}{33\,000} \text{ x } 100$$

$$\mathbf{ROI = 203\ \%}$$

Návratnosť investície je približne 203 %, táto hodnota je kladná a väčšia ako 100 % , to znamená, že optimalizácia incidentu sa nám vypláca.

5 Záver

Cieľom diplomovej práce bolo analyzovať proces vytvárania incidentov na kritických systémoch, ktoré fungujú na systémoch IBM Mainframe od spoločnosti IBM. Vybranou organizáciou, kde bola práca spracovávaná bola medzinárodná softwarová firma, ktorej názov je z dôvodu uchovania anonymity fiktívny. Spoločnosť v týchto dňoch zamestnáva len v Ostrave približne 2000 pracovníkov a mainframe oddelenie z tohto celkového počtu predstavuje približne 50 technických odborníkov.

Na začiatku diplomovej práce bolo potrebné čitateľa oboznámiť zo základnými pojmami z oblasti mainframov a ITIL-u. Tejto problematike sa venuje druhá časť diplomovej práce a to časť s názvom teoretické východiská servisnej podpory pri riešení incidentov na IBM Mainframe.

Ďalším krokom bolo potrebné zanalyzovať súčasný stav procesu tvorby incidentov na kritických systémoch a štatistické vyjadrenie vytvorených incidentov na týchto systémoch. Ako bolo vyobrazené v tabuľke 6, počet vytvorených incidentov u jednotlivých pracovných skupín vo firme T. C. za posledných šesť mesiacov, dosiahol čísla 13 076 incidentov.

Na základe analýzy vytvorených incidentov sme vyselektovali tri najčastejšie vytvárané incidenty, ktoré sme hlbšie analyzovali a zistili sme, že len jeden z týchto incidentov bolo možné ďalej skúmať a navrhnúť určitú možnosť jeho racionalizácie.

Incident, ktorý sme sa po vyselektovaní rozhodli analyzovať, je problém ktorý sa týka pravidelných kontrol, tzv. daycheckov. Tento problém predstavoval z celkového počtu 13 076 incidentov, presne 967 incidentov, čo je XX %. Najviac vytvorených problémov bolo na kritickom systéme SYS1 a to presne 322 incidentov. Po hlbšom preskúmaní sme zistili, že najviac problémov sa dotazuje na programovú komponentu DDTPROC1, ktorá po následnom odfiltrovaní dosiahla počet 177 incidentov na kritickom systéme SYS1.

Po diagnostikovaní danej programovej komponenty DDTPROC1 bolo zistené, že je na systéme zbytočná z dôvodu ukončenia kontraktu s firmou BMC Software a prechod na inú technológiu, ktorú začala poskytovať americká spoločnosť CA Technologies. Následné bolo nutné zadať podnet automation teamu, pre odstránenie danej komponenty zo systému, z dôvodu jej nadbytočnosti a nefunkčnosti a preto, že jej funkciu zastáva už iná aktívna programová komponenta.

Po následnej opätovnej analýze už po odstránenie komponenty DDTPROC1, sme zistili, že počet vytvorených incidentov za posledného pol roka klesol na počet 790 incidentov. Toto číslo je práve zmenšené o počet incidentov, ktoré boli vytvorené na kritickom systéme SYS1 a predstavovali práve našu analyzovanú a racionalizovanú komponentu DDTPROC1.

Z ekonomického hľadiska po vypočítaní ekonomického ukazovateľa návratnosti investície, ktorá nám vyšla 203 %, môžeme tvrdiť, že optimalizácia je zisková a firme sa vyplatí.

Na záver môžeme konštatovať, že všetky ciele, ktoré sme si v úvode diplomovej práce stanovili boli splnené. Hodnotu práce zvyšuje takisto fakt, že bola konzultovaná s odborníkmi z praxe, ktorí sa pohybujú v problematike IBM Mainframov a ITIL-u už dlhé roky a prichádzajú s nimi dennodenne do styku.

Zoznam použitej literatúry

Odborné knihy:

EBEL N. et al. (2012). *ITIL 2011*, Brno: Computer Press, 2012. 216 s. ISBN 978-80-251-3732-1.

KETTNER, J. et al.(2011). *Introduction to the New Mainframe: z/OS Basics*, New York: Redbook IBM, 2011. 792 s. ISBN 978-07-384-3534-3.

ROGERS Paul a Alvaro SALLA. (2010). *ABCs of z/OS System Programming*, New York: Redbook IBM, 2010. 200 s. ISBN 978-07-384-3383-7.

Internetové zdroje

ASPG. (2014). *Mainframes vs. Supercomputers* [online]. 2014 [cit. 2015-01-17]. Dostupné z: http://aspg.com/mainframes-vs-supercomputers/#.VRgA2PmUdHM

BESTPRACTICE.CZ. (2014). *Vztah ITIL® a CobiT* [online]. 2014 [cit. 2015-01-17]. Dostupné z: http://www.bestpractice.cz/cs/Best-practice/-ITSM-ITIL-/-Vztah-ITIL-a-dalsich-pristupu/Vztah-ITIL-a-CobiT.alej

COMPUTERWORLD. (2014). *Procesor z196, zřejmě nejrychlejší počítačový čip, uvádí na trh IBM* [online]. 2014 [cit. 2014-11-19]. Dostupné z: http://computerworld.cz/aktuality/procesor-z196-zrejme-nejrychlejsi-pocitacovy-cip-uvadi-na-trh-ibm-7641

ELLIOTT, Jim. (2014*). IBM Mainframes – 45+ Years of Evolution* [online]. 2014 [cit. 2014-10-14]. Dostupné z: http://www.vm.ibm.com/devpages/jelliott/pdfs/zhistory.pdf

FEEDIT.CZ. (2014). *Revoluční technologii IBM Mainframe denně využívame již 50 let* [online]. 2014 [cit. 2014-10-12]. Dostupné z: http://www.feedit.cz/wordpress/2014/04/08/revolucni-technologii-ibm-mainframe-denne-vyuzivame-jiz-50-let/

ADAPTIC. (2015). *Roi*. [online]. 2015 [cit. 2015-04-19]. Dostupné z: http://www.adaptic.cz/znalosti/slovnicek/roi/

.HOFTA, Jan. (2008). *Úvod do mainframe*. Praha, 2008 Diplomová práce. České vysoké učení technické v Praze, Fakulta jaderná a fyzikálně inženýrská, Katedra matematiky.

Zoznam skratiek a symbolov

ABEND - Abnormal end

ACID - Atomicity-consistency-isolation-durability

APF - Authorized program facility

API - Application Programming Interface

ASID - Address space ID

BEM – BMC Event Management

BLKSIZE - Block Size

CICS - Customer Information Control System

CLIST - Command list

COBOL - Common Business Oriented Language

CP - Central processor

CPC - Central processor complex

CPU - Central processing unit

DASD - Direct access storage device

DBMS - Database management system

DD - Data definition

DDL - Data definition language

DFSMS - Data Facility Storage Management Subsystem

FTP - File Transfer Protocol

GUI - Graphical user interface

HFS - Hierarchical file system

HMC - Hardware Management Console

HTML - HyperText Markup Language

HTTP - Hypertext transfer protocol

HTTPS - HyperText Transfer Protocol Secure

I/O - Input/Output

IMS - Information Management System

IP - Internet Protocol

IPL - Initial program load

ISPF - Interactive System Productivity Facility

JCL - Job control language

JES - Job entry subsystem

LAN - Local area network

LDAP - Lightweight Data Access Protocol

LLA - Library lookaside

LPAR - Logical partition

MIPS - Million informations per second

MVS - Multiple Virtual Storage

NFS - Network file system

OS - Operating system

PCHID - Physical Channel ID

PGID - Process group identifier

PTF - Product Temporary Fix

QSAM - Queued sequential access method

RACF - Resource Access Control Facility

RAID - Redundant array of independent disks

RAS - Reliability, availability, serviceability

RC - Return code

RDBMS - Relational database management system

REXX - Restructured Extended Executor Lanaguage

SAF - Security access facility

SAP - System Assistance Processor

SDSF - System Display and Search Facility

SLA - Service level agreement

SMF - System management facility

SQL - Structured Query Language

TCP/IP - Transmission Control Protocol/Internet Protocol

TSO - Time Sharing Option

TSO/E - Time Sharing Option/Extensions

URL - Uniform Resource Locator

VPN - Virtual private network

VSAM - Virtual Storage Access Method

VTAM - Virtual telecommunications access method

WLM - Workload Manager

WTOR - Write to operator with reply

XML - Extensible Markup Language

Prohlášení o využití výsledků diplomové (bakalářské) práce

Prohlašuji, že

- byl(a) jsem seznámen(a) s tím, že na mou diplomovou (bakalářskou) práci se plně vztahuje zákon č. 121/2000 Sb. – autorský zákon, zejména § 35 – užití díla v rámci občanských a náboženských obřadů, v rámci školních představení a užití díla školního a § 60 – školní dílo,
- beru na vědomí, že Vysoká škola báňská – Technická univerzita Ostrava (dále jen VŠB-TUO) má právo nevýdělečně ke své vnitřní potřebě diplomovou (bakalářskou) práci užít (§ 35 odst. 3),
- souhlasím s tím, že jeden výtisk diplomové (bakalářské) práce bude uložen v Ústřední knihovně VŠB-TUO k prezenčnímu nahlédnutí a jeden výtisk bude uložen u vedoucího diplomové (bakalářské) práce. Souhlasím s tím, že údaje o diplomové (bakalářské) práci, obsažené v Záznamu o závěrečné práci, umístěném v příloze mé diplomové (bakalářské) práce, budou zveřejněny v informačním systému VŠB-TUO,
- bylo sjednáno, že s VŠB-TUO, v případě zájmu z její strany, uzavřu licenční smlouvu s oprávněním užít dílo v rozsahu § 12 odst. 4 autorského zákona,
- bylo sjednáno, že užít své dílo – diplomovou (bakalářskou) práci nebo poskytnout licenci k jejímu využití mohu jen se souhlasem VŠB-TUO, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly VŠB-TUO na vytvoření díla vynaloženy (až do jejich skutečné výše).

V Ostravě dne ……..

……..

Bc. Marián Brtko

Adresa trvalého pobytu studenta:

Lánska 926/3-43

017 01 Považská Bystrica

Slovensko