



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA PODNIKATELSKÁ

FACULTY OF BUSINESS AND MANAGEMENT

ÚSTAV MANAGEMENTU

INSTITUTE OF MANAGEMENT

NÁVRH ČÁSTI WEBOVÉ APLIKACE PRO VÝPOČET REŽIJNÍCH NÁKLADŮ

A DESIGN OF A PORTION OF WEB APPLICATION FOR OVERHEAD COST CALCULATION

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Patrik Florians

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Bernard Neuwirth, Ph.D., MSc

BRNO 2021

Zadání diplomové práce

Ústav:	Ústav managementu
Student:	Bc. Patrik Florians
Studijní program:	Ekonomika a management
Studijní obor:	Řízení a ekonomika podniku
Vedoucí práce:	Ing. Bernard Neuwirth, Ph.D., MSc
Akademický rok:	2020/21

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

Návrh části webové aplikace pro výpočet režijních nákladů

Charakteristika problematiky úkolu:

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

Cíle, kterých má být dosaženo:

Cílem diplomové práce je návrh a implementace části webové aplikace pro výpočet režijních nákladů ve firmě. Navrhnuté řešení bude součástí modernizačního směru zastarávajících produktů firmy, transformovaných do podoby nového cloudového portfolia ERP softwarových aplikací. Dílčím cílem práce je provedení analýzy současného stavu používané aplikace, z jejichž výsledků bude v navrhovaném řešení vycházeno. Součástí práce bude ekonomické zhodnocení navrhovaného řešení.

Základní literární prameny:

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

DOSTÁL, Petr, Karel RAIS a Zdeněk SOJKA. Pokročilé metody manažerského rozhodování. 1. vyd. Praha: Grada, 2005. 168 s. ISBN 80-247-1338-1.

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

MOLNÁR, Zdeněk. Efektivnost informačních systémů. 2. vyd. Praha: Grada, 2001. 179 s. ISBN 80-2470-087-5.

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

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

V Brně dne 28.2.2021

L. S.

doc. Ing. Robert Zich, Ph.D.
ředitel

doc. Ing. Vojtěch Bartoš, Ph.D.
děkan

Abstrakt

Predmetom tejto práce je návrh vytvorenia webovej aplikácie pre kalkuláciu režijných nákladov, ktorá má za úlohu nahradiť súčasné riešenie, ktoré je považované za zastarané. Robí sa tak v súlade so stratégiou spoločnosti SAP SE, pre ktorú riešenie navrhujeme. Táto snaha je súčasťou ambícií o rozvoji a zlepšení cloudového portfólia už existujúcich aplikácií spoločnosti, čo by malo v dlhodobom horizonte viesť aj k lepšej pozícii na trhu pre túto spoločnosť a jej produkty. Práca je rozdelená na 3 časti. Začína popisom teoretických východísk, kde sú zhrnuté nástroje a princípy, s ktorými v práci ďalej pracujeme. Nasledujúca časť venuje pozornosť analýze súčasného stavu kde ilustratívne rozoberáme súčasné riešenie a kľúčové časti. Záverečná, teda 3. kapitola obsahuje popis implementovaného riešenia a bližšie sa venuje rozdielom oproti zisteniam zhrnutých v kapitole 2.

Kľúčové slová

informačný systém, systém, cloud, aplikácia, réžie, algoritmus, dáta, dátový tok, dátová transformácia, databáza, výkon, program

Abstract

Subject of this thesis is to design a web application for overhead calculation, whose purpose is to be a substitution for presently used solution, which is considered to be deprecated. This is being done as a part of strategy of SAP SE corporation for which the solution is designed. This ambition to develop and improve cloud portfolio of already existing applications of the company should lead to betterment of already existing applications of this type and in a long run an improvement of the company's market position as well as its products. The thesis is divided into 3 parts. It begins with a description of theoretical concepts, tools and principles, which are then utilized in further chapters. The following chapter analyzes current state of the affairs, where it is illustrated, what the current solution looks like along with key parts of it. The final, 3rd chapter is dedicated to a description of implemented solution and it also closely describes key differences mentioned in chapter 2.

Keywords

information system, system, cloud, application, overhead, algorithm, data, data flow, data transformation, database, performance, program

Bibliografická citácia

FLORIANS, Patrik. *Návrh části webové aplikace pro výpočet režijních nákladů* [online]. Brno, 2021 [cit. 2021-04-28]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/132077>. Diplomová práce. Vysoké učení technické v Brně, Fakulta podnikatelská, Ústav managementu. Vedoucí práce Bernard Neuwirth.

ČESTNÉ PREHLÁSENIE

Prehlasujem, že predložená práca je pôvodná a spracoval som ju samostatne. Prehlasujem, že citácia použitých prameňov je úplná, že som vo svojej práci neporušil autorské práva (v zmysle Zákona 121/2000 Zb., o práve autorskom a právach súvisiacich s právom autorským).

V Brne dňa 03.05.2021

.....

podpis autora

POĎAKOVANIE

Moja srdečná vďaka patrí vedúcemu mojej práce pánu Doktorovi Neuwirthovi za jeho ochotu, trpezlivosť a mnohé užitočné podnety, ktoré by ma ani nenapadli. Ďalej ďakujem Doktorovi Luhanovi za jeho oponentúru. Osobitná vďaka predstaviteľom spoločnosti SAP ČR, ktorí so mnou spolupracovali.

OBSAH

ÚVOD.....	15
CIELE PRÁCE, METÓDY A POSTUPY SPRACOVANIA	16
Ciele Práce	16
Metódy a postupy spracovania.....	17
1 TEORETICKÉ VÝCHODISKÁ PRÁCE	18
1.1 Prostredie.....	18
1.1.1 Informačný systém	19
1.2 Aplikácie informačných systémov	21
1.3 Cloud	22
1.3.1 5 kľúčových charakteristík cloudového modelu	23
1.3.2 3 Servisné modely cloudu	25
1.3.3 4 Dodacie modely.....	25
1.4 Reprezentácia technických častí práce	26
1.4.1 Modelovanie algoritmov	26
1.4.2 Modelovanie procesov biznisu	28
1.4.3 Modelovanie prúdenia toku dát	30
2 ANALÝZA SÚČASNÉHO STAVU	31
2.1 Popis a informácie o spoločnosti SAP	31
2.1.1 Vedenie spoločnosti a kultúra	32
2.1.2 Informačná stratégia spoločnosti SAP	33
2.1.3 Súhrn	38
2.2 Analýza infraštruktúry.....	39
2.2.1 Súhrn	44
2.3 Analýza požiadaviek na riešenie	45
2.3.1 Súhrn	46

2.4	Analýza súčasnej aplikácie na počítanie režijných nákladov	47
2.4.1	Konfiguračný hárok	56
2.4.2	Diagram biznis procesu behu Programov KGI2 a CJ44.....	62
2.4.3	Zjednodušený pohľad na algoritmus výpočtu.....	64
2.4.4	Algoritmus výpočtového podprocesu	65
2.4.5	Diagramy toku dát	66
2.4.6	Meranie výkonu	70
2.4.7	Súhrn.....	72
2.5	Analýza dostupných zdrojov a technológií	73
2.5.1	Nástroje vývoja	73
2.5.2	Súhrn.....	77
2.6	Záver analýzy	78
3	VLASTNÉ NÁVRHY RIEŠENIA.....	79
3.1	Vzhľad a rozhranie nového riešenia.....	79
3.2	Programová časť nového riešenia	86
3.2.1	Diagram biznis procesu novej aplikácie	87
3.2.2	Zjednodušený algoritmus novej aplikácie	88
3.2.3	Diagramy toku dát riešenia	91
3.2.4	Meranie výkonu riešenia.....	95
3.2.5	Súhrn.....	98
3.3	Ekonomické zhodnotenie	100
3.3.1	Súhrn.....	104
	ZÁVER	105
	ZOZNAM POUŽITÝCH ZDROJOV.....	107
	ZOZNAM OBRÁZKOV	112
	ZOZNAM TABULIEK	115
	ZOZNAM GRAFOV	116

ÚVOD

V súčasnej dobe je rozšírený trend cloudových technológií a už nejaký čas sme svedkami konkurenčného závodu vo svete s technológiami rozličných firiem, ktoré spolu súperia o získanie prevahy nad konkurenciou v tomto neustále sa rozvíjajúcom prostredí. Čoraz viacej softwarových spoločností sa zameriava na riešenia charakteristické vysokou flexibilitou a jednoduchou škálovateľnosťou, čo umožňuje znižovať limitácie v podobe konečnosti zdrojov na strane zákazníkov. Časy keď sa firmy museli zapodievať kúpou fyzických technických prostriedkov, prenájmom priestorov, zavádzaním infraštruktúry a komplikovanými inštaláciami rozličného software a jeho následnou bezproblémovou integráciou vo vlastnej réžii sa strácajú. Tento prístup je nahrádzaný aplikáciami s oveľa väčšou komplexnosťou, kvalitou a ďaleko lepším výkonom.

Dnes sú vyvíjané snahy zo strany firiem na získanie konkurenčnej výhody na globalizovanom trhu s využitím všetkých dostupných prostriedkov. Jeden takýto prostriedok sú typicky dáta. V súvislosti so spracovávaním dát a ťažením informácii sa stretávame s technológiami strojového učenia, digitalizácie, integrácie a automatizácie procesov na všetkých úrovniach biznisu. Tieto prominentné technológie tvoria portfóliá mnohých softwarových firiem poskytujúcich rozličné ERP riešenia.

Softwarové spoločnosti, ktoré sa na trhu s informačnými systémami vyskytujú už dlhšiu dobu čelia potrebe presunu svojho neraz početného aplikačného portfólia do modernejšej podoby s využitím novších technologických platforiem. Tu vstupuje do obrazu aj téma a predmet tejto práce, ktorým je popis vzniku aplikácie z oblasti modulu financií a kontroingu modernizovaného cloudového aplikačného portfólia spoločnosti SAP SE.

Popíšeme návrh časti modernizovanej aplikácie na počítanie režijných nákladov, ktorá by mohla byť zaradená do skupiny aplikácií, ktoré vznikajú ako súčasť úsilia nahradiť zastarané programy v súlade s plnením strategických cieľov spoločnosti pre rozvoj na cloudovom trhu. V práci budeme vychádzať z analýzy súčasného stavu, čo znamená analyzovať zastaranú aplikáciu, ktorá je momentálne na mieste, čo nám dopomôže objasniť niektoré kľúčové súvislosti a záverom sa prepracujeme k ekonomickému zhodnoteniu navrhovaného riešenia.

CIELE PRÁCE, METÓDY A POSTUPY SPRACOVANIA

Nasledovať bude definícia, objasnenie a zdôvodnenie cieľov práce spolu s metódami a postupmi použitými na ich dosiahnutie.

CIELE PRÁCE

Primárnym cieľom je návrh časti modernizovanej webovej biznis aplikácie pre kalkuláciu režijných nákladov, ktorá bude patriť pod skupinu programov vyskytujúcich sa v moduloch financií a kontroľingu nového cloudového ERP spoločnosti SAP. Návrh nebude zahŕňať samotné výpočtové jadro, ktoré už existuje. Dôvodom je, že prerobenie tak komplexnej funkcionality by bolo časovo príliš náročné a vyžiadalo by si to neprimerané zdroje.

Za ostatné, sekundárne ciele môžeme považovať súbor činností, ktoré je potreba vziať do úvahy. Patrí medzi ne dizajn vzhľadu, analýza súčasného stavu z pohľadu dát, analýza možností back-endu a front-endu a príprava testovacích dát. Toto bude obnášať návrh užívateľského rozhrania v súlade s produktovými štandardami a smernicami pre zjednotené rozhranie aplikácii ako súčasť produktových štandardov, návrh implementácie middleware a návrh implementácie časti back-endu. Pri návrhu budeme musieť brať do úvahy, že aplikácia má byť postavená na existujúcej logike, ktorá je pre nás čiernou skrinkou a nesmieme do nej zasahovať. Návrh preto musí byť neinvazívny.

Zapodievať sa nebudeme hardwarom a záležitosťami súvisiacimi s hardwarom a podobnými komponentami, keďže aplikácia je realizovaná pre cloud. Čo však do úvahy musíme vziať je domáca infraštruktúra systémov SAP, nepôjdeme však do irelevantných detailov, nakoľko hardware nehrá v tomto prípade príliš veľkú rolu.

METÓDY A POSTUPY SPRACOVANIA

Pri spracovaní začneme analýzou starej aplikácie a relevantné zistenia budeme zapisovať do tejto práce. Pre riadnu analýzu budeme študovať zdrojový kód a to aj s využitím debugovania. Pre svoju analýzu máme dostupné kontakty na niekoľkých expertov z tejto oblasti vrátane jedného z pôvodných vývojárov tohto riešenia. Čo ale nemáme je dokumentácia. Žiadna dokumentácia k pôvodnému riešeniu nebola nájdená. Našli sa náznaky návodov, no tie sú tiež veľmi staré a často písané po Nemecky. Situáciu komplikuje aj fakt, že zdrojový kód je 40 rokov starý a sú v ňom použité dnes už zakázané programovacie techniky jazyka ABAP.

Okrem štúdia zdrojového kódu a štruktúry dát budeme analyzovať aj zastarané užívateľské rozhranie, s ktorého použitím budeme simulovať fungovanie aplikácie pri rozličných podmienkach.

S využitím teoretických východísk tejto práce popíšeme niektoré aspekty nášho návrhu. Je potrebné poznamenať, že do práce neuvedieme nič čo by mohlo predstavovať intelektuálne vlastníctvo spoločnosti SAP. To znamená, že môžeme opísať čo aplikácia robí, no nesmieme popísať ako presne to robí. Pri návrhu riešenia popíšeme potrebné kľúčové časti pre implementáciu a nakoniec vykonáme ekonomické zhodnotenie riešenia. Zdrojový kód návrhu nebude súčasťou tejto práce, nakoľko sa to považuje za intelektuálne vlastníctvo spoločnosti SAP, no dotyčné dielo budeme testovať z hľadiska výkonu, čo je jeden z kľúčových ukazovateľov, ktorý musí spĺňať určité hodnoty, ktorými sa určuje kvalita softwarových produktov.

Pri teoretických východiskách budeme čerpať z viacerých literárnych prameňov, ktoré poskytujú bázu informácii slúžiacu ako podklad modelovania častí návrhu, no zároveň ide samozrejme o spôsob ozrejmenia niektorých súvislostí potrebných na pochopenie komplexného kontextu, v ktorom náš návrh figuruje.

1 TEORETICKÉ VÝCHODISKÁ PRÁCE

V tejto kapitole a jej následných podkapitolách predstavíme niektoré koncepty, či teoretické rámce, ktoré sú užitočné pre pochopenie situácie popisovanej v kapitole o analýze či návrhoch riešenia.

1.1 PROSTREDIE

Prostredie, v ktorom sa pri tvorbe práce nachádzame je prostredím veľkej nadnárodnej spoločnosti, ktorej produkty sú softwarového charakteru, a ktoré sú určené najmä iným spoločnostiam. Spoločnosť SAP je známa svojimi produktami z prostredia informačných systémov. Dôvod tvorby a potreby aplikácie, ktorá je predmetom tejto práce je aj informačná stratégia spoločnosti. Dielo prof. Ing. Zdeňka Molnára, CSc. o efektívnosti informačných systémov (1) a dielo od autorov Ing. Petr Sodomoka, Ph.D., MBA a Ing. Hana Klčová, Ph.D. „Informační systémy v podnikové praxi“ (2), poskytujú pomerne všeobecný prehľad o tom čo to informačná stratégia je a ako ju možno vnímať. Tak napríklad autor Molnár sa v 3. kapitole vyjadruje k tomu, čo je to informačná stratégia a akú hrá informačná stratégia rolu vo vzťahu k informačným systémom. Autor v tomto diele definuje informačnú stratégiu ako „...sústavu cieľov a spôsobov ich dosiahnutia“ (1). Toto je pomerne vágna definícia, ale už o čosi lepšiu predstavu vidíme v diele od Sodomoka a Klčovej, kde obaja autori v podstate rozširujú vyššie uvedené rozdelením implementácie informačnej stratégie v kontexte existencie informačných systémov na nasledujúce 3 časti:

„1. Analyzovať a zhodnotiť súčasný stav IS/ICT

2. Definovať cieľový stav IS/ICT

3. Navrhnuť postup ako dosiahnuť cieľového stavu za súčasných podmienok“ (2).

V prípade oboch diel, sa autori opierajú o jednu fundamentálnu a nezanedbateľnú skutočnosť, ktorou je rola, ktorú v podnikoch zohrávajú informácie. O dôležitosti informácii a ich kritickej úlohe v informačných systémoch sa môžeme dozvedieť od autorov Doc. Ing. Josef Basl, CSc, a inžiniera Blažičeka, ktorí vo svojej tvorbe s názvom „Podnikové informační systémy: podnik v informační společnosti.“ (3) začínajú na úrovni definície toho čo je to informačná spoločnosť a akú úlohu vo fungovaní podniku zohrávajú informácie a z toho je teda možné v širokých súvislostiach vyvodit' záver, že

informačná stratégia sa pravdepodobne bude odvíjať od povahy dotyčných informácií.
(3)

Profesor Molnár dospel k tomu, že informačná stratégia má za účel vplývať na rozvoj podniku vo svojom podnikateľskom prostredí s generovaním určitých ekonomických hodnôt ako zvýšenie produktivity, či iných výhod, ktoré by z využitia technológii mohli plynúť. (1) Podľa autora ide o ustavičný proces spojený s informačnou infraštruktúrou podniku (4), ak je teda základom existencie a pohyb kvalitných informácií vo firmách (3) a ak existuje snaha dosiahnuť istú pružnosť, tak potom existujú silné predpoklady prítomnosti nejakej formy informačnej stratégie a to v takom prípade smeruje aj k postaveniu informačných systémov.

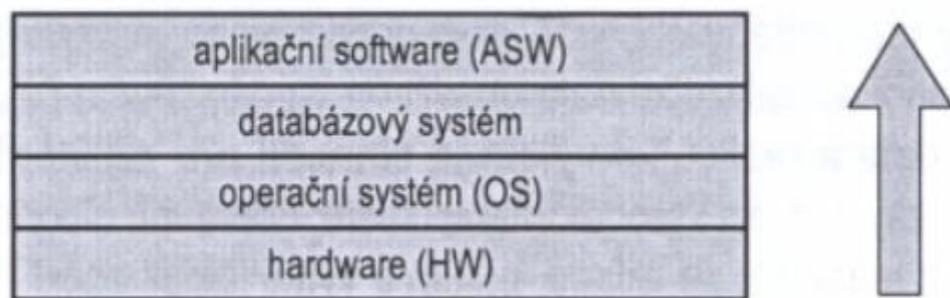
Správne využívanie informácií tvoriacich súčasť informačnej stratégie môže v kľúčových momentoch viesť k znižovaniu nákladov a zvyšovaniu príjmov (3). V dnešnej dobe nie je ťažké si predstaviť výrobnú halu s tisíckami senzorov, ktoré sú napojené na rôzne komponenty informačných systémov, ktoré musia pracovať s veľkými objemami dát v relatívne nízkom čase.

Práca s informáciami je teda dynamický a nepretržitý proces, ktorý je potrebné vykonávať za účelom neustáleho vývoja a zlepšovania spoločnosti, jej smerovania a jej postavenia v porovnaní s konkurenciou. (1) (3) (2)

Informačná stratégia môže mať rozličné podoby a môže byť braná rozlične vážne v závislosti na charaktere podniku. Príkladom je 6 možných prístupov: „Monopol, Centrálne plánovanie, Vedúcu rolu, Voľný trh, Obmedzené zdroje a Nevyhnutné zlo“ (4) , ktoré boli popísané autorom Molnárom, väčšinou však ide o kombináciu možných stratégií.

1.1.1 INFORMAČNÝ SYSTÉM

Informačný systém možno do istej miery považovať za manifestáciu informačnej stratégie. (4) (3) Na informačný systém ako taký je možné sa pozrieť z viacerých strán. Najtechnickejší pohľad je z hľadiska technológií a IT infraštruktúry.



Obrázok 1 – IT model informačného systému

(Zdroj: (3))

V minulosti platilo, že firmy si tvorili informačný systém z rôznorodého software, to však bolo v dobách kedy nie každý software mohol fungovať v ľubovoľnom prostredí. V tých časoch malo teda zmysel brať do úvahy potreby hardware, operačných systémov a pod. Z hľadiska užívateľov je možné podľa vyššie citovaných autorov možno hovoriť o organizačnej pyramíde z pohľadu skupín užívateľov:



Obrázok 2 – úrovne používateľov IS

(Zdroj: (3))

Autori správne konštatujú, že potreby pracovníkov na týchto odlišných úrovniach v rámci organizácie majú odlišné potreby a očakávania od IS pre výkon svojej práce.

Posledné hľadisko je hľadisko procesné, pričom využijeme zjednodušenú definíciu oboch autorov, ktorí tvrdia, že „proces predstavuje súhrn činností, ktoré smerujú k realizácii

určitého výstupu užitočného pre zákazníka“ (3). Dobrý príklad fungovania procesov v moderných organizáciách predstavuje aj nasledujúci obrázok:



Obrázok 3 – procesné spracovávanie objednávok

(Zdroj: (3))

Na obrázku vyššie, je pomerne dobre ilustrované fungovanie firmy, ktorá používa ERP na prácu s objednávkami, v prípade autorov, ktorých citujeme ide zrovna o SAP a na ilustrácii vidíme niekoľko procesov, s ktorými zamestnanci pomocou informačného systému interagujú a posúvajú tak dianie v spoločnosti vpred. Tu by sme si len dovolili poznamenať, že aj aplikácia, ktorá je predmetom tejto práce predstavuje jeden taký proces (prípadne tvorí súčasť iných procesov, ale o tom pojednáme o čosi neskôr).

1.2 APLIKÁCIE INFORMAČNÝCH SYSTÉMOV

Definíciu pojmu aplikácie informačných technológií čerpáme z diela „Podniková informatika“ (5) od autorov Ing. Libora Gála, Ph.D., Doc. Ing. Jana Poura, CSc. a Ing. Zuzany Šedivej, Ph.D., ktorí v 1. kapitole definovali aplikáciu informačných technológií ako aplikáciu, ktorá „v určitom kontexte poskytuje funkcie užívateľom a manipuluje s dátami s tým, že využíva softwaru, hardwaru a ľudí“ (5). „Aplikačné portfólio“ je

množina takýchto aplikácií (5), pričom v realite ide o značne heterogénne prostredie, ktoré je klasifikovateľné z viacerých hľadísk (5) a viacerými spôsobmi.

Aplikačné portfólio je často tvorené rozličnými aplikáciami (5), to sa týka nie len ich účelu ale aj technickej kompozície, čo je obvykle spojené s otázkami o možných spôsoboch integrácie aplikácií, alebo systémov.

Z literatúry od Sodomku a Klčovej vieme o viacerých zaujímavých spôsoboch integrácie. Jedným z týchto spôsobov je aj „riešenie na princípe middlewaru“ (6). Middleware, ktorého zmyslom je, „aby v prostredí existencie rôznorodých (heterogénnych) distribuovaných aplikácií, ktoré sú často autonómne (na iných aplikáciách nezávislé), bolo možné tieto aplikácie integrovať do väčšieho celku (5). Okrem toho je nutné, aby tieto aplikácie boli schopné si navzájom poskytovať svoje aplikačné služby prostredníctvom počítačovej siete.“ (5).

Sodomka a Klčová tvrdia, že stupne integrácie je možné rozdeliť do 4 úrovní: „dátová, užívateľské rozhranie, aplikačné rozhranie a obchodná logika“ (6).

V súvislosti s dátovou integráciou možno uvažovať nad zjednocovaním rozličných databázových systémov, ako perzistentnej vrstvy pod vrstvou abstrakcie, čo je typický príklad, využívaný v ERP od SAP. (3) (6) Čo sa týka užívateľského rozhrania tam je princípom integrácia starých aplikácií, teda je to akási čierna skrinka, kde sú strojom simulované vstupy a výstupy sú zachytené a prenesené do modernejšieho užívateľského rozhrania, pričom môže ísť aj o takzvané portálové riešenie. (6) Ďalší, aplikačný prístup je zase realizovaný s využitím aplikačných rozhraní výrobcov tzv. API (6).

Zatiaľ čo autori vyššie hovorili o klasickej integrácii, dnes sa stretávame s ešte jedným prístupom, ktorý je v podobe cloudu.

1.3 CLOUD

Cloudové technológie sú v súčasnej dobe aktívne diskutovanou témou a preto nie je prekvapivé, že sa v rozličných vedeckých článkoch opakovane stretávame s definíciami, ktoré sú prakticky totožné, alebo minimálne podobné v mnohých aspektoch. V nasledujúcich podkapitolách uvedieme niekoľko kľúčových pojmov súvisiacich s cloudom. Pri štúdiu skutočností sme vychádzali z viacerých akademických a vedeckých článkov. Ako úvod do problematiky sme použili článok autorov Chena, Chuanga a Nakataniho o vnímanej hodnote cloudu pre biznis vo svojej štúdií (7).

Akademická polemika na túto tému sa realizovala už začiatkom milénia, no reálne články a štúdie, z ktorých bolo a aj je dodnes čerpané sa začali objavovať vo väčšom počte okolo roku 2009, odkiaľ pochádza aj jeden z hlavných prameňov (8), ktorý sa snaží zjednotiť definície, ktoré boli do tej doby používané. Je to zdroj, ktorý pomenúva všeobecné koncepty a pojmy, ktoré sú s touto problematikou spojené.

Najrigoróznejší zdroj definície cloudu a jeho častí, či technológii sme našli v publikácii k štandardu pre ministerstvo obchodu Spojených štátov (9). Z uvedeného čerpáme definíciu cloudového výpočtového systému(cloudu) ako: „... model pre všadeprítomný, príhodný, vyžiadany sieťový prístup k zdieľanej množine konfigurovateľných počítačových zdrojov(napr. Siete, servery, úložiská, aplikácie a služby), ktorá môže byť rýchlo spojzdnená a spustená s minimálnym riadiacim úsilím, alebo interakcie poskytovateľa služieb.“ (9) Väčšina definícií potom spomína skutočnosť, že cloudový model pozostáva z 5 kľúčových charakteristík, 4 dodacích modelov a 3 servisných modelov. (9) Existujú však aj neskoršie publikácie, ktoré dodacie modely vynechávajú (7).

1.3.1 5 KĽÚČOVÝCH CHARAKTERISTÍK CLOUDOVÉHO MODELU

- Samo-obslužnosť na vyžiadanie
 - Zákazník je schopný zriadiť si podľa vlastného uváženia a potreby požadované výpočtové kapacity, alebo si nakonfigurovať relevantné služby
- Široký sieťový prístup
 - Poskytovateľ cloudového modelu musí umožňovať rozsiahle rozhranie pre sieťovú komunikáciu voči svojej(hostiteľskej) infraštruktúre
- Zhlukovanie zdrojov
 - Ide o zlúčenie technických a výpočtových kapacít
- Rýchla elasticita
 - Schopnosť rýchleho škálovania infraštruktúry a rýchleho(dynamického) získania nových zdrojov
- Merateľné služby

- Cloudové infraštruktúry často disponujú rozličnými monitorovacími nástrojmi, ktoré sa používajú na kalkuláciu vyťaženia, optimalizáciu a prípadne rovnomerné rozdeľovanie zdrojov (9)

Hore uvedené body sme prevzali priamo z normy pre ministerstvo obchodu USA. Rozličná literatúra sa spomenutým aspektom venuje v rozličnej miere, ktorá závisí na študovanej časti cloudových systémov.

V niektorej literatúre (10) (7) sme sa tiež dočítali k určitej paralele, ktorá predstavuje polemiku nad tzv. „mainframom“ (10) ako predchodcom cloudu. Zdroje, ktoré sme použili však vzápätí čitateľa informujú, a to celkom na mieste, že cloudové riešenie nie je to isté ako mainframe. Oba výpočtové modely sa líšia vo svojej architektúre (to sa obvykle týka aj rýdzo technickej stránky), ale aj použitím a možnosťami poskytovania služieb.

1.3.2 3 SERVISNÉ MODELY CLOUDU

Väčšina kvalitnej literatúry rozlišuje nasledujúce 3 typy servisných modelov, t.j. modelov poskytovania služieb zákazníkom cloudovými poskytovateľmi.

1) Software ako služba(SaaS)

- a. Softwarom ako službou môžeme rozumieť aplikácie určené k bežnému použitiu koncovými užívateľmi. V technickom žargóne ide o riešenie tenkého klienta a princípom, je že tieto aplikácie nemusia byť fyzicky prítomné na disku užívateľovho počítača. (9) Tento prístup nie je vhodný tam, kde sa vyžaduje rýchla odozva(real-time pracovanie s dátami), alebo je legislatíva nastavaná tak, že neumožňuje fyzické host'ovanie dát mimo určité geografické územie (11) (typicky Čína). Za dobrý príklad možno uviesť Outlook 365 Cloud od spoločnosti Microsoft.

2) Platforma ako služba(PaaS)

- a. V tomto prípade často hovoríme o službách rozličných zariadení, ide o istú vrstvu abstrakcie nad Hardware (8), avšak nejde nevyhnutne o klasické programy ako MS Word a pod. Môže ísť o inštaláciu webových, aplikačných, databázových, alebo iných typov Software(v podstate middleware) (12). Definícia NIST uvádza aj sprístupnenie knižníc, programovacích jazykov a ich prostredí(typicky python, Java) pod. (9)

3) Infraštruktúra ako služba(IaaS)

- a. Pod infraštruktúrou sa v kontexte cloudového modelu často rozumie tzv. Hardware, ale aj nejaká forma elementárneho Software(Firmware, operačný systém, virtualizačný hypervízor) (12)

1.3.3 4 DODACIE MODELY

Dodacie modely, je názov prevzatý z cudzojazyčnej literatúry (9), ale vyskytuje sa aj isté množstvo literatúry, ktoré sa často zameriava na konkrétny druh technickej implementácie cloudu (13) (14). Názov tejto podkapitoly v skutočnosti reprezentuje druh zavedenia cloudu po fyzickej stránke ako je ďalej ozrejmene:

- Súkromný cloud
 - Súkromný cloud je model počítačovej infraštruktúry, ktorý operuje s využitím cloudových technológií, avšak je špecifický pre jednu konkrétnu organizáciu (9) (v závislosti na modeli riadenia a IT vo firme môže, ale nemusí byť patričný hardware vo vlastníctve danej firmy). Dobrý príklad môže byť napríklad spoločnosť Barclays (13)
- Komunitný cloud
 - Tento cloud je určený špecifickej komunite užívateľov(firiem, firemných zoskupení, a pod.) (9).
- Verejný cloud
 - Toto je cloud určený širokej verejnosti. (9) Za klasické príklady možno uviesť Google cloud, Microsoft cloud(OneDrive), Amazon atď.
- Hybridný cloud

Typickou črtou je v tomto prípade kombinácia 2 a viacerých predošlých typov. (9) Klasicky sa stretávame s implementáciou súkromného a verejného cloudu.

1.4 REPREZENTÁCIA TECHNICKÝCH ČASTÍ PRÁCE

Na zrozumiteľné zachytenie, niektorých častí riešenia a pre lepšie pochopenie súvislostí pri priebehu programu použijeme dobre známe značenia uvedené v nasledujúcich podkapitolách.

1.4.1 MODELOVANIE ALGORITMOV

Na modelovanie a zachytenie správania je vhodné použiť takzvaný vývojový diagram(z Anglického Flow-chart). Tradičný vývojový diagram bol definovaný dnes už dávno zastaralým štandardom ECMA 4 a bol rozdelený na 2 hlavné druhy:

- 1) Programový vývojový diagram
- 2) Dátový vývojový diagram (15)

Vývojový diagram pozostáva podľa spomenutej normy z nasledujúcich značiek:



Obrázok 4 - Terminálová značka

(Zdroj: (15))

Predošlý obrázok zachytáva tzv. terminál, ktorý označuje začiatok a koniec diagramu.

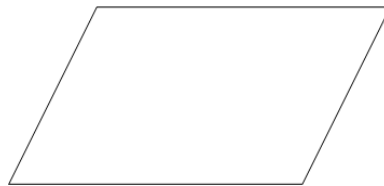
Všeobecný symbol pre akúkoľvek operáciu je nasledujúca značka v podobe obdĺžniku:



Obrázok 5 – Všeobecná činnosť

(Zdroj: (15))

Vstupno-výstupné operácie sú reprezentované lichobežníkovým útvarom:



Obrázok 6 – Vstupno-výstupná značka

(Zdroj: (15))

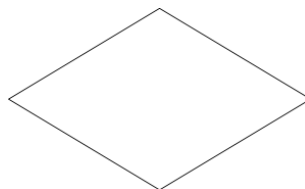
Pre potreby zachytenia prítomnosti podprogramu je použitá značka, pripomínajúca všeobecnú činnosť:



Obrázok 7 – Pod-proces

(Zdroj: (15))

Pre potreby vetvenia programu (v prípade dobre známej konštrukcie if), je použitá značka kosoštvorca:



Obrázok 8 – Značka vetvenia

(Zdroj: (15))

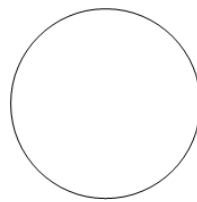
V prípade, že je potreba zachytiť v diagrame cyklicky sa opakujúcu činnosť je použité nasledujúce značenie:



Obrázok 9 - Cyklus

(Zdroj: (15))

Poslednou značkou je referencia, alebo odkaz, ktorá má za úlohu presmerovať tok programu na iné miesto.



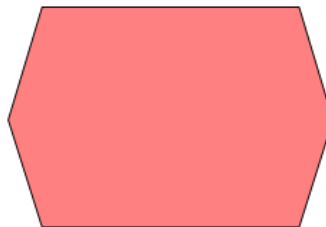
Obrázok 10 - Odkaz

(Zdroj: (15))

1.4.2 MODELOVANIE PROCESOV BIZNISU

Na zachytenie interakcie užívateľov s časťami systému, tvoriacimi tzv. biznis procesy by mohol slúžiť vývojový diagram známy ako EPC – Event Process Chain. Tento druh diagramu je vo firemnom prostredí pomerne rozšírený a aj populárny vďaka svojej jednoduchosti. (16) Čo je pre tento druh diagramu typické, je to, že je riadený udalosťami, ktoré vyjadrujú stav procesu. Diagramy EPC by mali udalosťou začínať aj končiť.

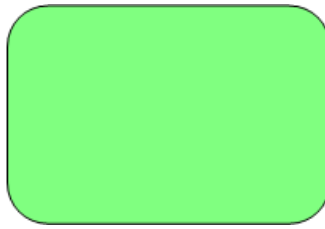
Udalosť má nasledujúcu značku:



Obrázok 11 – Udalosť v diagrame EPC

(Zdroj: (16))

Ďalšou značkou je funkcia. Funkcie predstavujú nejaký druh transformácie v rámci procesu, zväčša ide o nejakú činnosť.



Obrázok 12 – Funkcia EPC

(Zdroj: (16))

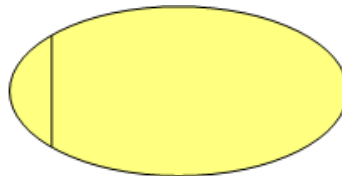
V dokumentácii modelovacieho nástroja, ktorý máme dostupný sa okrem iných ešte udáva aj značka informačných zdrojov:



Obrázok 13 – Informačná surovina, informačný zdroj

(Zdroj: (17))

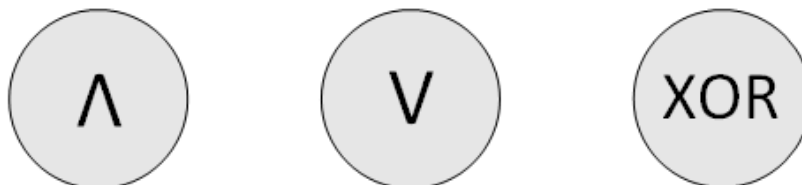
Ďalšou kľúčovou časťou sú role aktérov procesu, niekde nazývané aj Organizačné jednotky:



Obrázok 14 – organizačná jednotka

(Zdroj: (17))

Poslednou kľúčovou súčasťou sú operátory AND, OR a XOR, pričom operátor XOR sa používa ako rozhodovací operátor, čo plynie z definície jeho logickej funkcie vyjadrenej napríklad Booleovou algebrou.

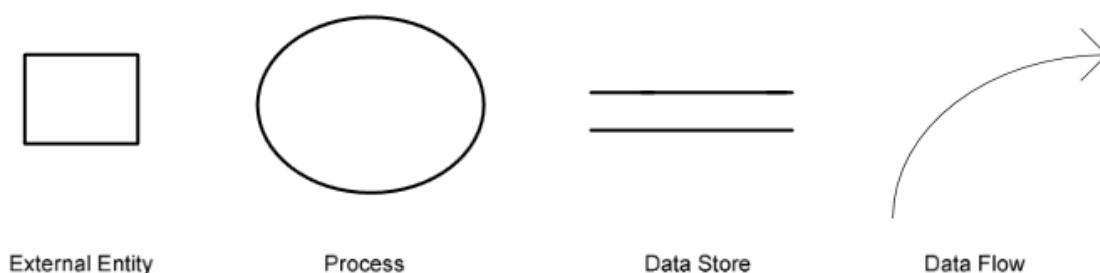


Obrázok 15 – Operátory diagramu EPC

(Zdroj: (17))

1.4.3 MODELOVANIE PRÚDENIA TOKU DÁT

Diagramy používané na zachytenie toku dát pri určitých procesoch sa nazývajú „Data flow Diagrams“, skrátene DFD. Ide o nástroj funkčného modelovania, ktorý je charakteristická absencia riadiaceho značenia ako v predošlých dvoch prípadoch. (18) Zatiaľ čo vývojový diagram popisuje dianie v programe na úrovni programu a EPC diagram zachytáva interakciu na úrovni biznis procesu, DFD predstavuje spôsob popisu toku informácii a dát v jednotlivých procesoch. DFD má viacero druhov notácie, pričom význam ani fungovanie jednotlivých značiek sa nemení. My uvedieme notáciu Yourdon-Coud, niekde označovanú aj ako Yourdon-DeMarco:



Obrázok 16 – DFD notácia Yourdon-DeMarco

(Zdroj: (19))

Zľava doprava môžeme vidieť značenie: externých entít(terminátorov), procesov, dátových úložísk a značenia toku dát. (18) V DFD vždy platí, že dátový tok môže vchádzať do niektorého z komponentov(proces, úložisko), ale mal by z neho aj vychádzať, teda nie je možné prepojiť dve dátové úložiská bez prítomnosti procesu.

V DFD diagramoch sa tiež zvykne vyskytovať určitá hierarchia, preto sa hovorí o tzv. úrovniach DFD, ktoré slúžia na popis jednotlivých úrovní. Toto je dobre uplatniteľné v prípade postupu vývoja od všeobecného po konkrétny. Na počiatku typicky nájdeme úroveň 0, kde sú zachytené len všeobecné funkcie s ktorých systém, alebo daná jednotka pozostáva. Ak platí predošlé tvrdenie tak platí aj, že diagramy ostatných úrovní popisujú vždy len nejakú časť celku. (18)

2 ANALÝZA SÚČASNÉHO STAVU

V tejto kapitole pristúpime k analýze súčasného stavu riešenia, ktoré je doposiaľ na mieste. Analýza je vykonávaná v prostredí spoločnosti SAP SE, resp. jej dcérskej vetvy SAP Labs, čo sú laboratória spoločnosti SAP patriace pod subjekt SAP ČR. Laboratória spoločnosti SAP sú charakteristické koncentráciou špecialistov v oblasti IT. Autor tejto práce je v dobe jej vzniku zamestnancom v tomto laboratóriu, ktoré sa nachádza v Brne a jeho pracovnou náplňou je okrem iného vývoj aplikácií v pridelenom teame. Jednou takouto aplikáciou je aj aplikácia, ktorá je predmetom tejto práce. Pri analýze budeme postupovať nasledujúcimi krokmi:

1. Analýza a popis spoločnosti SAP a jej informačnej stratégie – tu popíšeme príčiny a dôvody potreby modernizácie aplikácii ako je tá naša a uvedieme stratégiu spoločnosti v niektorých technologických oblastiach.
2. Analýza infraštruktúry – v tejto podkapitole sa budeme venovať infraštruktúre tvoriacej implementačné prostredie, v ktorom pracujeme.
3. Analýza požiadaviek na riešenie – tu uvedieme niektoré kritériá potrebné na realizáciu modernizačného úsilia.
4. Analýza súčasnej aplikácie na počítanie režijných nákladov – v tejto časti preberieme a ukážeme niektoré časti riešenia, ktoré má byť nahradené.
5. Analýza dostupných zdrojov a technológií – v poslednej časti analýzy predstavíme technické zdroje a prostriedky, ktoré máme pre návrh riešenia k dispozícii.

2.1 POPIS A INFORMÁCIE O SPOLOČNOSTI SAP

Spoločnosť SAP vznikla v roku 1972 a založili ju 5 zamestnanci z IBM – Dietmar Hopp, Hasso Plattner, Hans-Werner Hector, Klaus Tschira a Claus Wellenreuther (20). Spoločnosť vznikla okolo vízie vytvoriť software, ktorý je plne prispôsobený potrebám zákazníkov. Išlo o myšlienku urobiť software súčasťou výroby a fungovania podnikov samotných. SAP bola jedna z prvých spoločností, ktorá predstavila software pre riadenie biznisu. Účtovnícka funkcionálna bola jednou z prvých softwarových komponent, ktoré vznikli. Pôvodné riešenia spoločnosti SAP vznikali na diernych štítkoch (20), pričom behom niekoľkých rokov bol novovznikajúci systém prenesený aj na systém UNIX, ktorý

vznikal v tomto období, od roku 1969 (21). Spoločnosť a jej klientela sa začala rozrastať a bolo to práve kvôli tomu, že podobné riešenia doposiaľ neexistovali. Prvý systém, ktorý v SAPe vznikol sa nazýval systém RF, ktorý sa neskôr stal základom systému SAP R/1 (22). Nasledoval systém SAP R/2, ktorý rozšíril funkcionality predošlých iterácií softwaru (23). SAP sa stala priekopníkom v podnikových technológiách a informačných systémoch. V 90. rokoch minulého storočia bol vyvinutý systém R/3, ktorý bol neskôr uvedený na trh a stal s veľkým úspechom. (24) Dodnes sa môžeme v mnohých spoločnostiach a vládnych inštitúciách stretnúť práve s týmto systémom, ktorý je dnes už, ale zastaraný tiež. Názov SAP sa dnes nerozlučne spája s pojmom ERP a dlhé roky platilo, že SAP bol synonymom pre informačný systém, alebo ERP. V roku 2004 bola vydaná prvá verzia platformy NetWeaver, ktorá umožňovala integráciu veľkého množstva software, ktorý nepochádzal od spoločnosti SAP (25), čo vlastne značí dosiahnutie ambícií o multiplatformovom riešení pre každého zákazníka bez ohľadu na infraštruktúru, ktorú má na mieste. V prípade mnohých zákazníkov sa SAP stal de-facto štandardom pre správne fungovanie. Kľúčovou charakteristikou software od SAP je a vždy bolo spracúvanie dát priamo v procese výroby, behu biznisu s čo najkratšou systémovou odozvou. Toto je niečo čo formovalo moderné štandardy kladené na všetok software, ktorý spoločnosť vyprodukuje. Modernou technológiou, ktorá je teraz spoločnosťou ponúkaná, je množina riešení S4/HANA (26), ktoré sú postavené na hybridnej analytickej databáze HANA.

2.1.1 VEDENIE SPOLOČNOSTI A KULTÚRA

Vedenie spoločnosti pozostáva z dozornej a výkonnej rady. Na čele spoločnosti je najnovšie Christian Klein, ktorý bol pôvodne v tejto funkcii s Jennifer Morgan, no s príchodom svetovej pandémie a nestabilitou na niektorých burzách je dnes Christian Klein vo funkcii sám. Za technologický vývoj je v spoločnosti zodpovedný Juergen Mueller. Okrem predsedníctva spoločnosti sa na riadení angažuje aj dozorná rada, ktorej predsedá Prof. Dr. h.c. mult. Hasso Plattner. (27)

Keďže spoločnosť SAP je početná v radoch svojich zamestnancov a zahŕňa v sebe kultúry so všetkých svetových kontinentov panuje v nej firemná kultúra raziaca multi-kulturálnu harmonickú spoluprácu s vyváženými príležitosťami pre každého. Toto je niečo s čím sa možno stretnúť aj v iných veľkých spoločnostiach a hlavná výhoda je

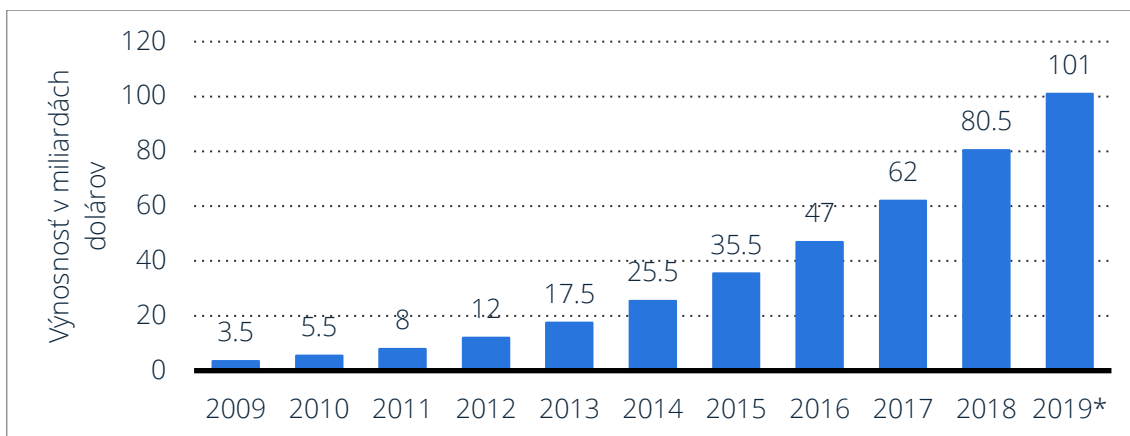
schopnosť porozumieť kultúram, ktoré sa líšia a produkujú ľudí s inými hodnotami, alebo cudzími názormi na niektoré kľúčové aspekty. Toto je dôležité pretože SAP ponúka svoje produkty v mnohých krajinách po celom svete a je nesmierne dôležité aby prístup reprezentantov spoločnosti bol adekvátne prijateľný pre diverzné národy celého sveta. To však nie je všetko, lokalizácia a technické koncipovanie produktov musí byť natoľko flexibilné, že dokáže uspokojiť pracovné požiadavky biznisu v každej jednej krajine kde SAP obchoduje a ponúka svoje produkty. Typický príklad sú zákazníci z Japonska, kde sú štandardy na kvalitu produktov neporovnateľne vyššie ako v iných krajinách sveta a pri tom tolerancia chýb v softwari, ktorý zo svojej povahy často tvorí kritickú časť podniku je v Japonsku takmer neexistujúca.

Okrem klasickej inkluzívnej politiky SAP razí aj ideológiu technického rozvoja s opatnosťou, čo znamená zavádzanie niektorých moderných technológií, ale s racionálnym skepticizmom. SAP má vlastný ekosystém technológií, ktorý je nesmierne špecifický a súčasťou tohto ekosystému je okrem iného aj komerčný programovací jazyk ABAP, ktorý sa okrem produktov spoločnosti SAP nedá použiť na žiadnej inej technológii, čo vlastne buduje isté bariéry v tom, že ide o veľmi špecializovanú technologickú doménu, ku ktorej je takmer nemožné sa dostať bez vlastnenia licencie na produkty spoločnosti SAP.

V Českej republike je na čele SAP ČR Hana Součková a SAP Labs Brno je pod vedením Martina Janáčka (28). Spoločnosť SAP vykonáva množstvo rozličných akvizícií a preto má veľké množstvo dcérskych jednotiek (29), pričom medzi najprominentnejšími z nich je Ariba (30), Concur (31), Qualtrics (32) a mnohé iné. Všetky tieto technológie boli kúpené za účelom realizácie strategických cieľov spoločnosti SAP.

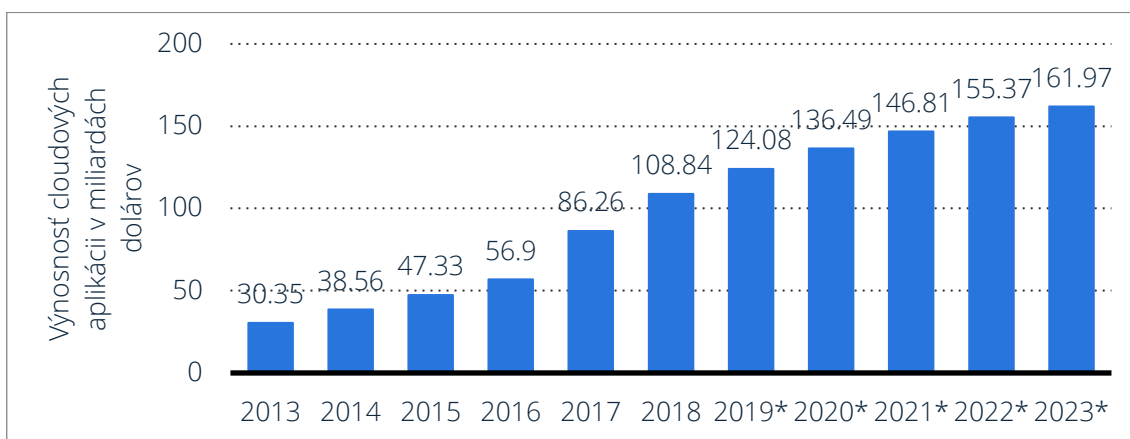
2.1.2 INFORMAČNÁ STRATÉGIA SPOLOČNOSTI SAP

Spoločnosť SAP sa snaží raziť filozofiu expanzie na trhu cloudových technológií. To sa týka všetkých druhov teda: Infraštruktúry ako služba, Platforma ako služba a hlavne Software ako služba. Keď vezmeme do úvahy výnosnosť trhu s technológiami SaaS a veľkosť tohto trhu tak je zrejmé prečo tomu tak je. Nasleduje niekoľko štatistík, ktoré pomôžu objasniť situáciu na trhu.



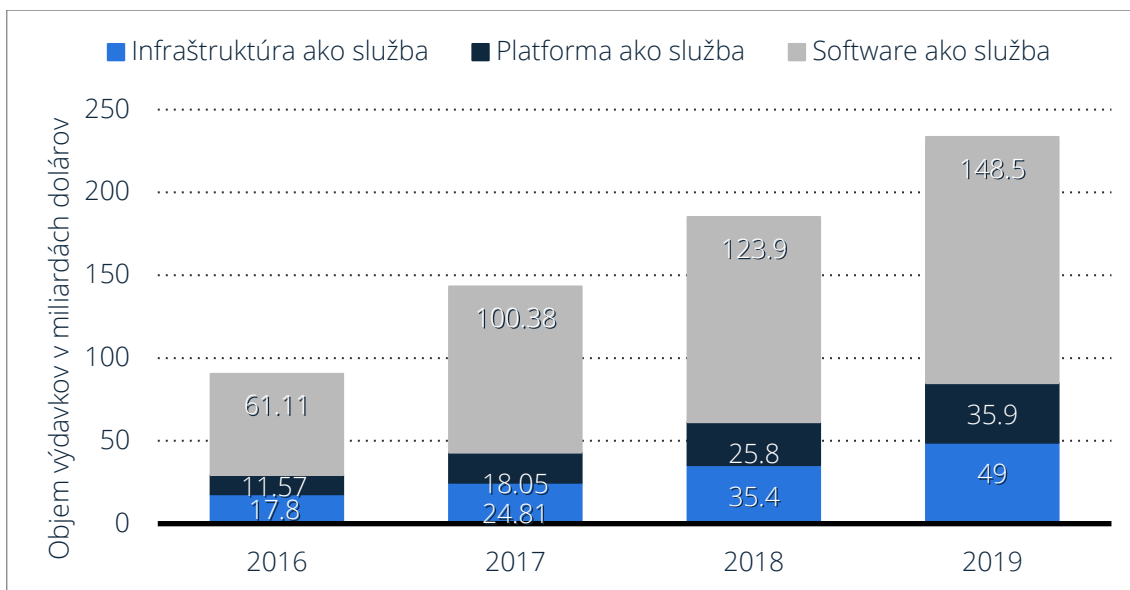
Graf 1 - Výnosnost' trhu SaaS do roku 2019 v miliardách dolarů spojených štátov

(Zdroj: (33))



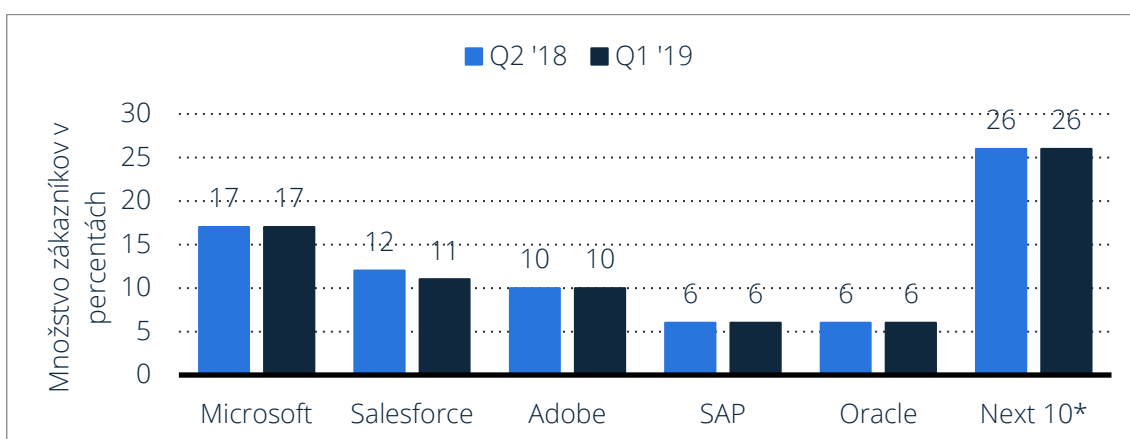
Graf 2 - Výnosnost' cloudových aplikácií v miliardách dolarů

(Zdroj: (34))



Graf 3 – Objem výdavkov na verejný cloud podľa technologického segmentu v miliardách dolárov

(Zdroj: (34))



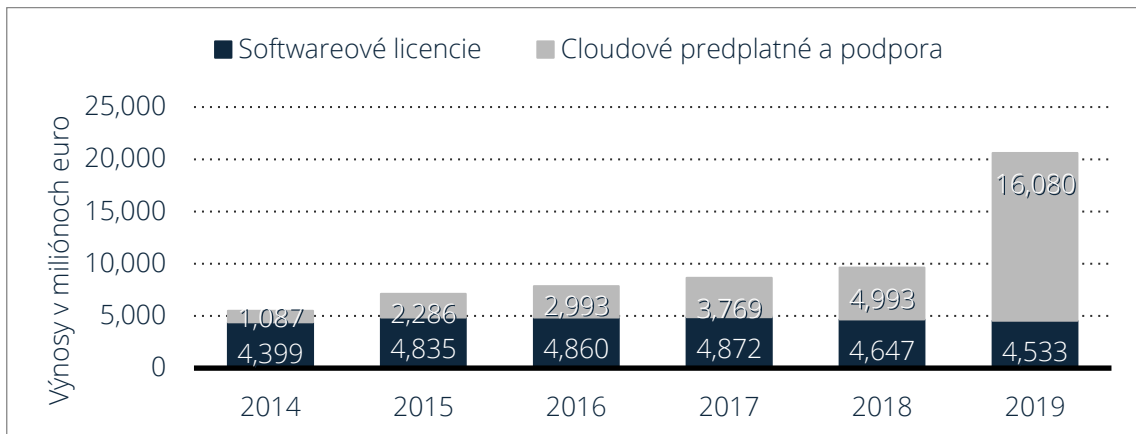
Graf 4 – objem výnosov na celosvetovom trhu so SaaS 2. kvartál 2018 a 1. kvartál 2019

(Zdroj: (35))

Z uvedených grafov vyplýva prítomnosť trendu rastu investícií do cloudových technológií, pričom podľa zdroju statistika (36) je možné očakávať v najbližších rokoch pokračovanie investícií, čo povedie k pokračovaniu rastu. Software ako služba (SaaS) je kategória cloudových služieb do ktorej ide najviac výdavkov a je to práve tento segment, v ktorom sa vyskytuje aplikácia, ktorou sa zapodieваме.

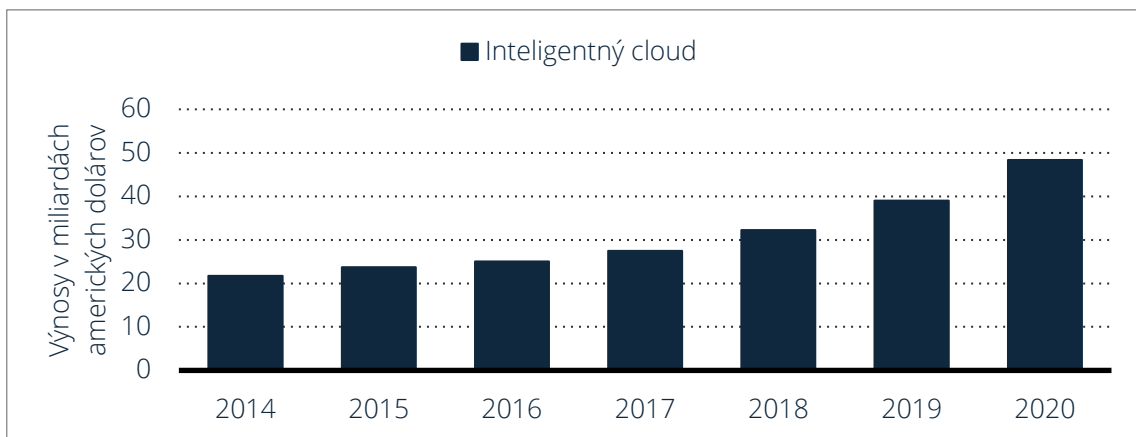
Pri porovnaní konkurentov sa SAP umiestňoval medzi prvými 5 čo nie je až taký zlý výsledok vzhľadom na náskok spoločností ako Salesforce, ktoré vznikali s biznis

modelom pre cloud, aj napriek tomu ale SAP podniká kroky aby tento náskok znížil a dosahuje toho rozličnými akvizíciami.



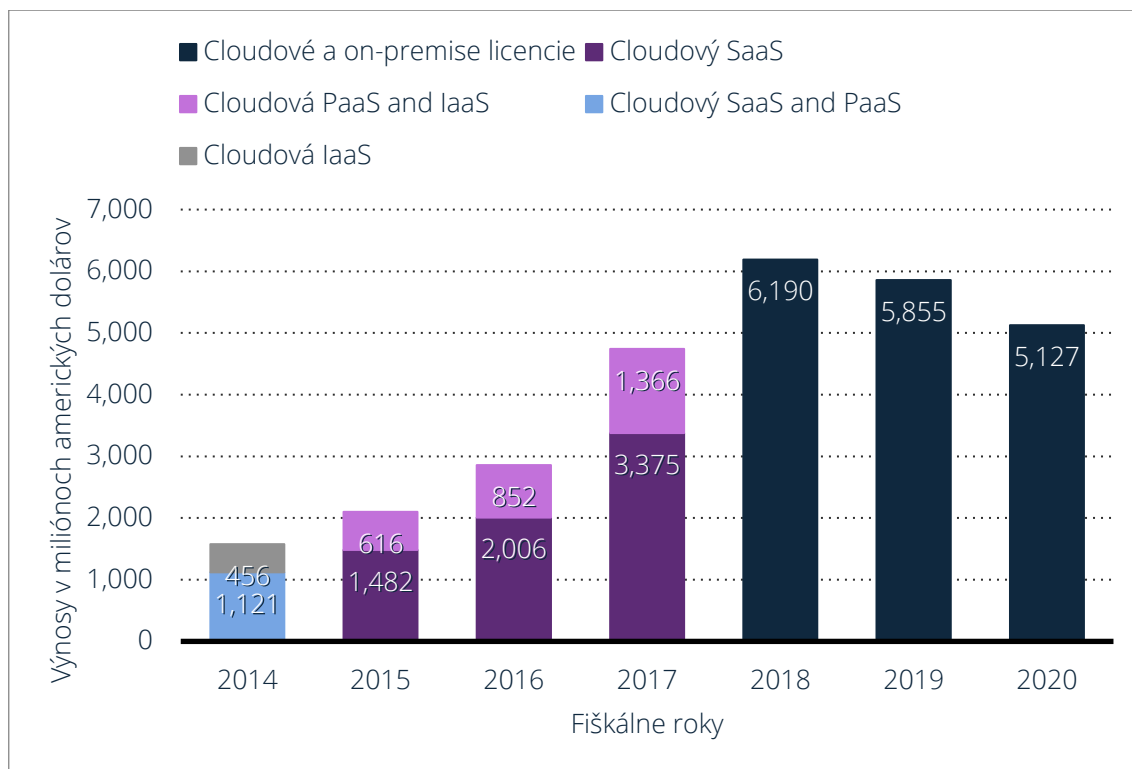
Graf 5 – Celosvetové výnosy SAPu rozdelené podľa segmentu v miliónoch euro

(Zdroj: (37))



Graf 6 – Výnosy spoločnosti Microsoft z inteligentného cloudu v miliardách dolárov

(Zdroj: (38))



Graf 7 – výnosy spoločnosti Oracle z rozličných cloudových technológií

(Zdroj: (39))

Z grafov vyššie vidno, že spoločnosť SAP zastáva agresívnu stratégiu na cloudovom trhu a svoje zastúpenie myslí naozaj vážne. Hoci treba poznamenať, že spoločnosť sa snaží svoje cloudové produkty pretlačiť na zákazníkov tým, že ukončuje podporu a vývoj staršieho software a niektorej funkcionality, ktorá sa doposiaľ vyskytovala len na on-premise (teda infraštruktúre v domácej réžii zákazníkov). Nie každému zákazníkovi to však musí vyhovovať, keďže s tak radikálne odlišným vzhľadom produktov S4 v porovnaní s R3 prichádza aj rada prekážok ako napríklad prispôbenie sa novému rozhraniu, pre usporiadanie niektorých aplikácií, ktoré začali tvoriť súčasť iných komponentov, nové neznáme technológie a premiestnenie kľúčových častí zdrojových kódov mimo dosah zákazníkov (zákazníci mali v predošlých produktoch možnosť vidieť a upravovať zdrojový kód všetkých aplikácií, ktoré si kúpili a dokonca im bolo často umožňované pridávať vlastné špecifické rozšírenia). Na druhú stranu skutočnosť, že intelektuálne dedičstvo v podobe algoritmov mohol hocikto vidieť bez väčšej námahy malo silný potenciál, že to zneužije konkurencia alebo dokonca aj implementačný partneri. Podľa novej filozofie budú zdrojové kódy viacej skryté, pokiaľ pôjde o cloud.

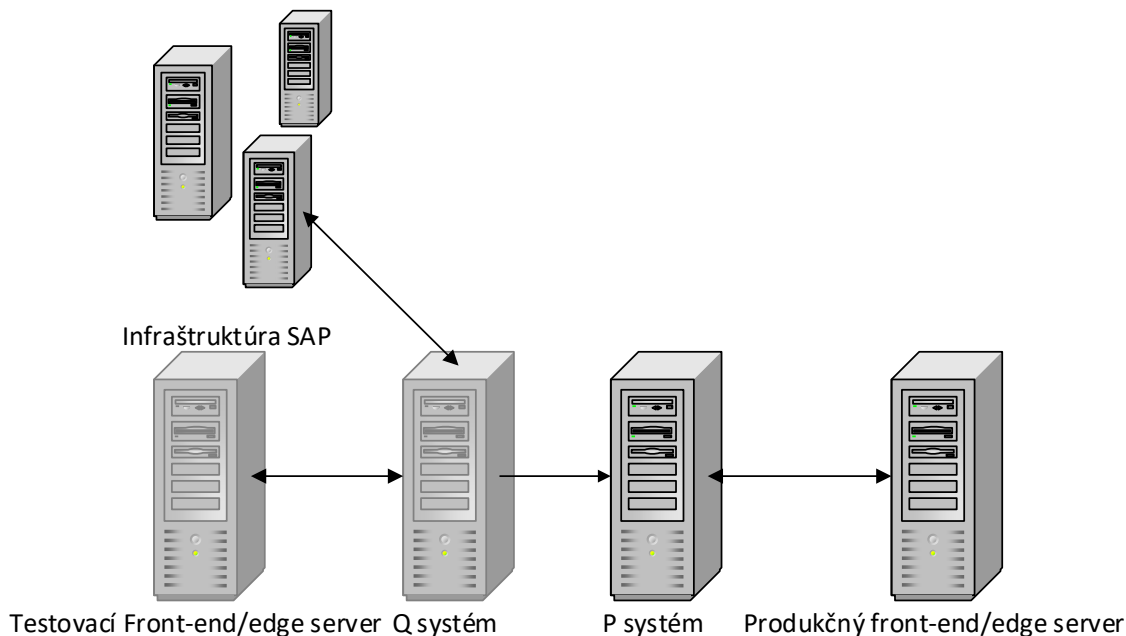
2.1.3 SÚHRN

Ako sme sa mohli v predošlých podkapitolách dozvedieť spoločnosť SAP SE je veľkým softwarovým hráčom s mnohými zamestnancami, pričom tisíce z nich sú softwarový vývojári. SAP je spoločnosť, ktorá je často prehliadaná uchádzačmi o zamestnanie a stáva sa, že jej technológie sú širokej verejnosti neznáme. Je to tým, že táto spoločnosť sa špecializuje na software, ktorý širokej verejnosti síce nie je taký povedomý ako software spoločností Apple, alebo Microsoft, ale zato plní kritické úlohy v tisíckach podnikov a používajú ho mnohí užívatelia, či už si to uvedomujú alebo nie. Je to vďaka softwaru od spoločnosti SAP, že CocaCola vie koľko presne fliaš bolo vyrobených za deň, v ktorej fabrike a aké náklady pri tom vznikli. Taktiež je to software od spoločnosti SAP, ktorý umožní distribuovať všetky vyrobené fľaše na správne miesto v správnom čase a za správnu cenu. A keď príde moment zaplatiť zamestnancom je to opäť software od SAP, ktorý sa o to postará.

Spoločnosť SAP, však už dávno nie je jediná, ktorá ponúka produkty s takou rozsiahlou funkcionalitou a komplexnosťou. Odbytisko budúcnosti na ktorom sa nachádzajú aj firmy ako Oracle, či Salesforce je cloud. So štatistik, ktoré sme ukázali je možné vydedukovať, že informačná stratégia, ktorú spoločnosť momentálne razí naznačuje, že úvaha o cloude je správna, no čaká ju ešte dlhý boj a roky rýchleho a prelomového technologického vývoja.

2.2 ANALÝZA INFRAŠTRUKTÚRY

Infraštruktúra, na ktorej je návrh postavený pozostáva z technológií SAP a teda ide o súčasť technického ekosystému.



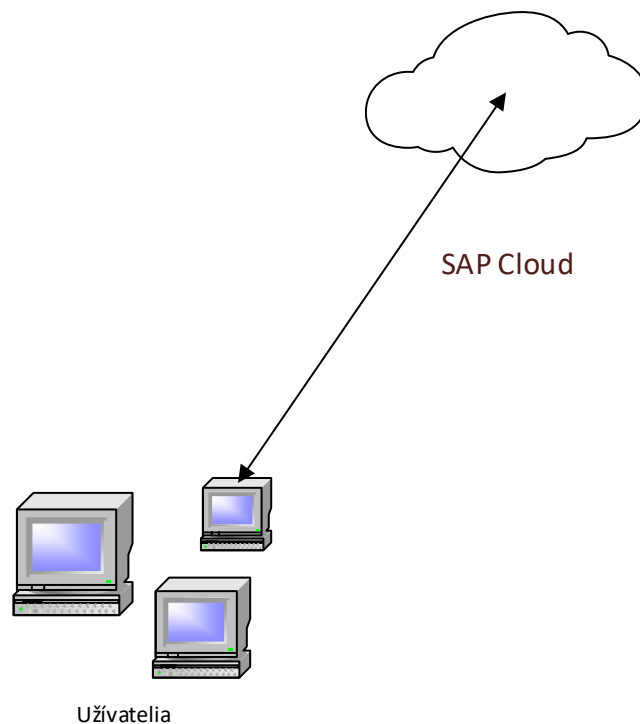
Obrázok 17 – všeobecný náhľad na infraštruktúru systémov SAP

(Zdroj: Vlastné spracovanie)

Riešenia v systémoch SAP sa dodávajú cez transportný systém, ktorý používa vlastné komunikačné protokoly. Infraštruktúra SAP vo všeobecnej rovine pozostáva z tzv. Q systémov, alebo systémov kvality, ktoré slúžia na vývoj a testovanie nových komponent predtým než prebehne transport na P, teda produkčný systém, ktorého funkčnosť je často kritická. Dôležité je spomenúť, že predtým, než je akékoľvek riešenie doručené zákazníkom na produkčné systémy, prebehne rada manuálnych aj automatizovaných testov na zákazníkových Q systémoch, kde sa simulujú rozličné nastavenia pri rozličných dátach, ktoré pochádzajú zo zákazníckych systémov. Pri vývoji na infraštruktúre SAP je prístup podobný – je popísaný nižšie v tejto podkapitole.

V prípade výskytu softwarovej chyby u zákazníka sa technická podpora obvykle pripája na testovací systém kvality, kde analyzuje situáciu a realizuje zmeny v kritických prípadoch. Niektorí zákazníci však umožňujú aj prístup na produkčné systémy, no nebýva zvykom na týchto systémoch niečo meniť. Všetky tieto systémy sú systémami na ktorých je nainštalovaný software spoločnosti SAP (obvykle NetWeaver – ten je dnes často

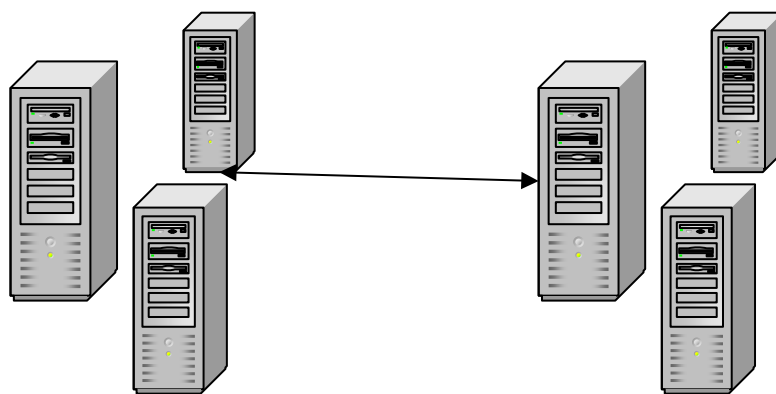
dodávaný ako súčasť upravenej distribúcie nejakého UNIXového operačného systému). Jednotlivé systémy môžu byť cloudové – teda v správe spoločnosti SAP, alebo môžu byť umiestnené na infraštruktúre danej spoločnosti. Front-end, alebo takzvaný edge server je v tomto prípade server, alebo skupina serverov, na ktorých sú umiestňované front-endové komponenty software spolu s aplikačnou servisnou infraštruktúrou, ktorá obvykle pracuje na báze OData (Open data REST formátu) (40). Sú to práve tieto typy serverov, ktoré môžu byť pripojené na internet, teda je k nim viac-menej verejný prístup, zatiaľ čo zvyšková infraštruktúra býva prístupná len na definovaných portoch a len pre konkrétny druh komunikácie. Aktualizácie software bývajú vydávané v pravidelne definovaných časových cykloch, no zákazníci, ktorí sú považovaní za obzvlášť cenných môžu dostať aktualizácie a novú funkcionálnosť skôr.



Obrázok 18 – infraštruktúra rýdzo cloudového riešenia

(Zdroj: vlastná réžia)

V prípade čisto cloudového riešenia potom vyzerá infraštruktúra z pohľadu užívateľov podobne ako na obrázku 18, pričom skutočná infraštruktúra na strane spoločnosti SAP je oveľa komplexnejšia.



Cluster vývojových systémov

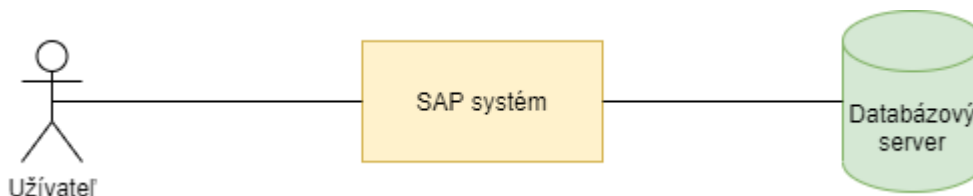
Testovacia infraštruktúra

Obrázok 19 – Infraštruktúra pre vývoj

(Zdroj: vlastná réžia)

V prípade vývoja sa obvykle používa jeden, či viacero vývojových systémov. Tu by sme len pripomenuli, že SAP systém môže v princípe znamenať skupinu, teda cluster niekoľkých fyzických zariadení, alebo môže ísť o jeden fyzický server. Testovacia infraštruktúra sa skladá z rôzne nakonfigurovaných Q systémov, na ktorých sú dátové sety pre spúšťanie automatizovaných testov, kompiláciu nového kódu a kontrolu kvality podľa rozličných ukazovateľov. Princíp vývoja sa podobá princípu dodávania produktov na zákaznícke systémy. Na rozdiel od reálnych zákazníckych systémov, na Q systémoch vo vývojárskej infraštruktúre nájdeme dáta, ktoré sú podobné ako tie, ktoré má zákazník, no podliehajú medzinárodným reguláciám o osobných dátach a citlivých údajoch, preto ide často o vymyslené hodnoty (typický príklad môžu byť mená zamestnancov).

Aby mohol byť nový produkt dodaný zákazníkovi musí byť na všetkých systémoch testovacej infraštruktúry všetko v poriadku. Pokiaľ testy odhalia chyby, vývojári sú informovaní a musí sa realizovať náprava. Po náprave sa proces opakuje.

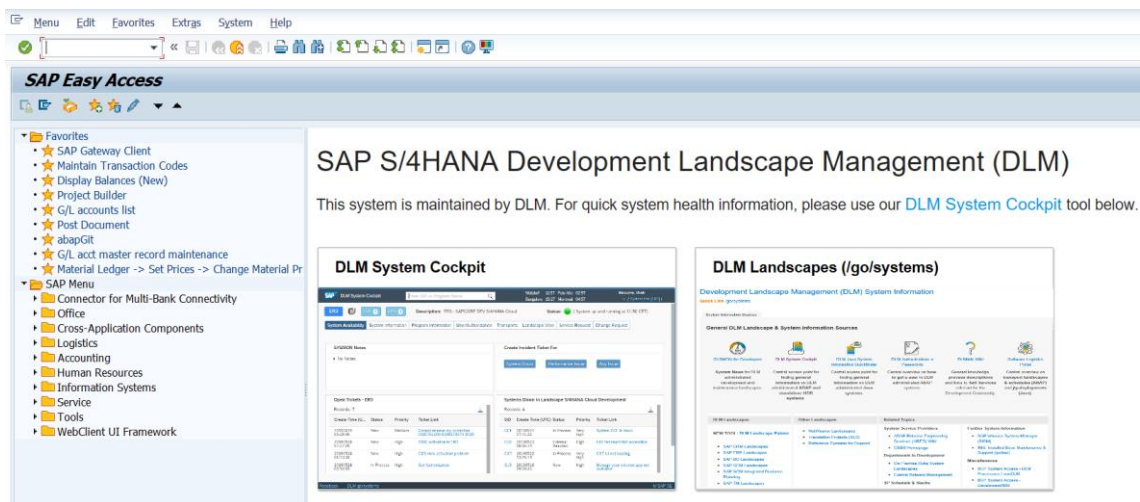


Obrázok 20 – diagram prepojenia back-endu vývojového prostredia

(Zdroj: vlastná réžia)

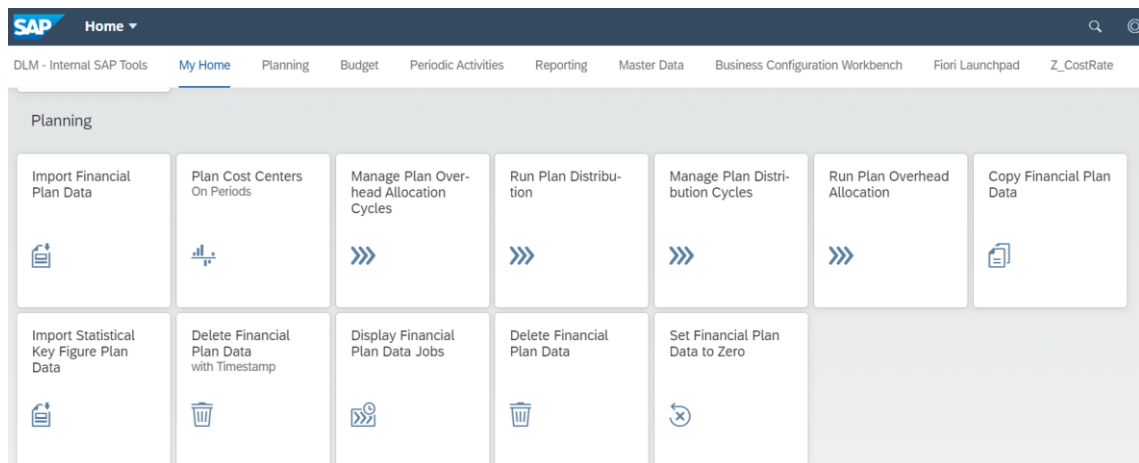
Obrázok 20 zachytáva zjednodušený náhľad na back-endovú časť infraštruktúry riešení SAP. Býva totiž zvykom, že okrem rozdelenia na front-endový a back-endový server je aj databázový server oddelený (často to tiež býva cluster), obvykle ide o HANA databázový systém, ale v podstate to môže byť väčšina známych systémov vrátane tých od spoločnosti Microsoft, alebo Oracle. Dôvod použitia HANA je, že S4/HANA je v dobe vzniku tejto práce vlajkový produkt spoločnosti SAP, ktorý je robený pre cloud aj pre domácu infraštruktúru.

Prístup k systémom môže byť dvojakým spôsobom. V prípade prvého spôsobu sa používa aplikácia SAP GUI, ktorá realizuje spojenie na všetky typy systémov bez ohľadu na to, či ide o cloudové, alebo on-premise systémy. Druhý spôsob je cez webové rozhranie, teda prehliadač.



Obrázok 21 – ukážka SAP GUI

(Zdroj: Vlastná réžia)



Obrázok 22 – ukážka webového rozhrania aplikácií SAP v technológii FIORI

(Zdroj: Vlastná réžia)

V prípade oboch typov rozhraní je možné ich upraviť podľa rozličnej farebnej témy, no FIORI poskytuje oveľa väčšiu flexibilitu – produkty v tomto riešení sú po dodaní zákazníkom často upravené podľa ich firemných farieb, potrieb a prípadne aj ďalších požiadaviek.

2.2.1 SÚHRN

V tejto kapitole sme predstavili prostredie, s ktorým sme sa pri analýze stretávali, a s ktorým budeme pri realizácii riešenia pracovať. V zjednodušenej forme sme popísali fungovanie transportného systému spoločnosti SAP, ktorý pomáha raziť filozofiu dodržiavanie kvalitatívnych štandardov, stanovených spoločnosťou, a ktorý zjednodušuje dodávanie software pre zákazníkov, ktorí sa tak môžu spoľahnúť, že ani kritická časť ich biznisu nebude ohrozená.

Záverom sme ukázali spôsoby prístupu k aplikáciám na infraštruktúre SAP, pričom za zdanlivo jednoduchým obrazom fungovania aplikácii sa skrýva veľká komplexita mnohých vzájomne spolupracujúcich a poprepájaných aplikácií, ktoré dokážu reflektovať aj tie najkomplexnejšie biznis procesy tých najnáročnejších zákazníkov. Novou tvárou spoločnosti SAP je rozhranie FIORI, ktoré tvorí moderný aplikačný štandard v zjednotenom dizajne, ktorý nadobúdajú aplikácie tvoriace cloudové portfólio spoločnosti SAP a jej rady produktov známych ako S4, fungujúcich na technológii HANA.

2.3 ANALÝZA POŽIADAVIEK NA RIEŠENIE

Pri zadávaní projektu vzniklo niekoľko hlavných požiadaviek. Pri vývoji je potreba však brať na vedomie implicitné požiadavky v podobe produktových štandardov a rozličných obmedzení, ktoré majú za úlohu zaistiť kvalitu produktov a bezproblémový chod všetkých procesov v podobe softwarových riešení. Primárne požiadavky definované v zadaní sú nasledovné:

1. Cieľom je vytvoriť novú FIORI aplikáciu pre výpočet réžie nákladových objektov typu Projekt, WBS element a výrobná objednávka
2. Zjednotiť funkcionality vizuálne harmonizovaných aplikácií s transakčnými kódmi CJ44 a KGI2 do jednej aplikácie, aby zastarané programy mohli byť vyradené z produkčného používania

Hlavné prípady používania aplikácie sú nasledovné:

- a. Simulácia kalkulácie réžií bez ovplyvňovania živých dát
- b. Kalkulácia réžií a aktualizácia živých dát podľa potreby
- c. Simulácia reverzného behu už vyrátaných réžií
- d. Výpočet reverzného behu existujúcich réžií a aktualizácia dát podľa potreby

Okrem týchto primárnych požiadaviek a prípadov použitia musíme dbať aj na už viac krát spomenuté štandardy spoločnosti. Tých je pomerne veľké množstvo, preto ich neuvádzame všetky, nakoľko mnohé z požiadaviek môžu mať riešenie, ktoré je riešené jednou technológiou – napríklad užívateľské rozhranie a dostupnosť pre osoby s postihnutím je riešené prostredníctvom frameworku pre front-end, ktorý je súčasťou dizajnového návrhu vzhľadu – FIORI. Medzi typické požiadavky back-endu patrí napríklad výkon aplikácie, ktorý je možné regulovať na rôznych úrovniach. Môže ísť napríklad o obmedzenie prístupov k databáze, limitácia používania zdrojov ako výpočtový výkon procesora, množstvo pamäte, latencia programu (toto je obzvlášť komplexná téma, keďže sa do výpočtov zachytáva aj latencia siete) a napríklad aj prítomnosť adekvátne diverzných testovacích dát. Pokrytie jednotkovými, integračnými a komponentovými testami je samozrejmosť. V prípade jednotkových testov ide minimálne o nutnosť 75% pokrytia kódu každého objektu. Poslednou požiadavkou je časové obmedzenie bežný vývoj trvá približne 8 až 12 týždňov. Predtým bývajú 2 týždne neformálnej analýzy a príprav a po konci 8 týždňov býva ešte čas maximálne do konca

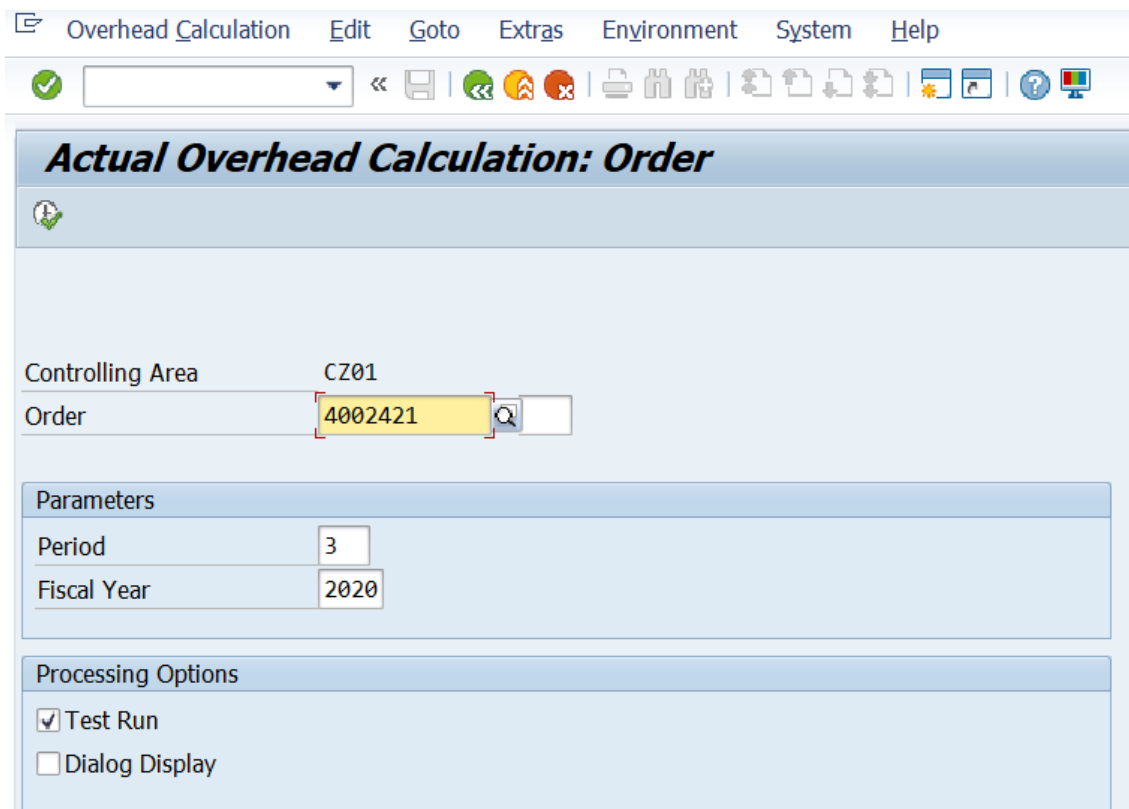
nasledujúceho mesiaca na opravy drobných chýb a bugov, ale to je pomerne premenlivé, keďže od istého momentu je potreba vypýtať si na úpravy povolenie. V čase po vývoji a na jeho konci taktiež obvykle svoju prácu vykonávajú testerí zaisťujúci kvalitu(deje sa to v období na opravu chýb pred vypustením do produkcie). Títo zamestnanci spúšťajú a konfigurujú automatizované testy pomocou rozličného skriptovania a hlásia svoje zistenia vývojovému teamu. Jeden konkrétny dátum je stanovený ako míľnik uvoľnenia produktov do prevádzkovej infraštruktúry. To je moment keď musí byť všetko hotové a na systémoch sú aktivované prístupové zámky, takže nie je možné technicky realizovať zmeny.

2.3.1 SÚHRN

V tejto podkapitole sme rozobrali požiadavky na riešenie a priblížili sme tak podmienky, v ktorých budeme pracovať a pojednali sme o tom čo sa od nás očakáva pri vývoji. Toto je dôležité z hľadiska časového, ktoré je zase dôležité pre úspech projektu. Vďaka detailným a dobre definovaným štandardom kvality kladeným na produkty v spoločnosti SAP budeme musieť mať na zreteli koncového zákazníka a jeho spokojnosť ako aj hrdosť spoločnosti SAP a jej reprezentantov, ktorí musia byť presvedčení, že produkt, ktorý chcú predat' dokáže dostať svojmu menu v každom ohľade a to prispeje k budovaniu pozitívnej reputácii spoločnosti doma aj v zahraničí.

2.4 ANALÝZA SÚČASNEJ APLIKÁCIE NA POČÍTANIE REŽIJNÝCH NÁKLADOV

Keď už sme analyzovali všeobecné požiadavky môžeme pristúpiť k analýze aplikácie ktorá má byť nahradená. To čo je potrebné na úvod povedať, je skutočnosť, že hoci pracujeme s viacerými transakciami v skutočnosti narábame s jedným balíkom zdrojových kódov.



Actual Overhead Calculation: Order

Controlling Area CZ01

Order 4002421

Parameters

Period 3

Fiscal Year 2020

Processing Options

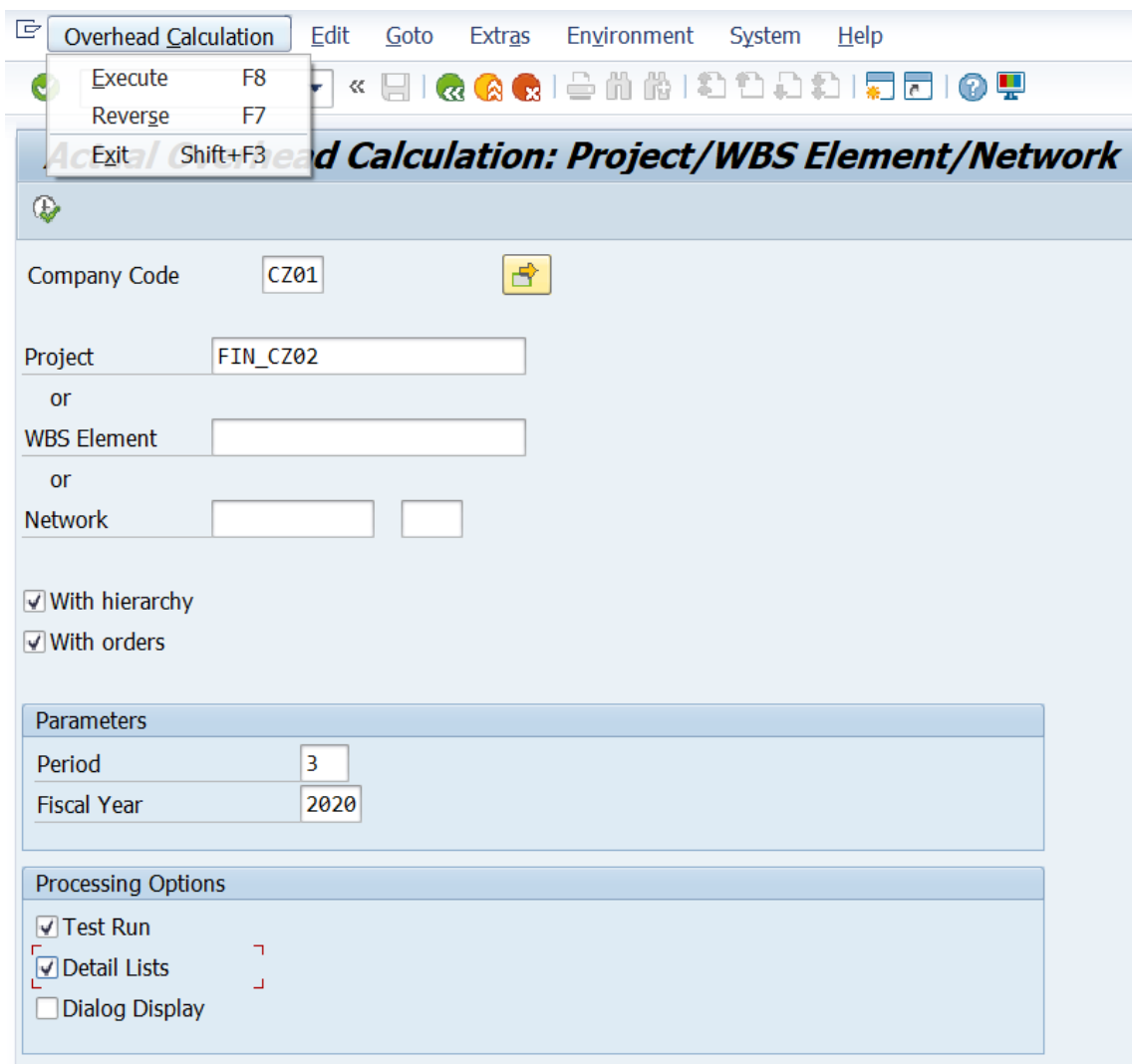
Test Run

Dialog Display

Obrázok 23 – Transakcia KGI2 – réžia objednávok

(Zdroj: vlastné spracovanie)

Ako môžeme vidieť na obrázku vyššie rozhranie pôvodnej transakcie pozostáva zo vstupných polí, kde sme zadali číslo objednávky. Následne sme nastavili parametre v podobe obdobia a fiškálneho roku a na záver sme volili testovací beh – ten používame pokiaľ chceme operáciu simulovať.



Obrázok 24 – vstupná obrazovka transakcie CJ44 – výpočet réžie zvyšných objektov

(Zdroj: vlastné spracovanie)

Obrazovka pre transakciu CJ44, ktorá počíta réžie s komplexnejšími objektami ako projekty a WBS elementy je viditeľne zložitejšia. Na rozdiel od KGI2 tu vieme nastaviť kód firmy, s ktorým sa pracuje. Podotkneme ešte, že „Network“, alebo sieť je druh nákladového objektu, ktorý sa už nepoužíva a nebude figurovať v našom návrhu. Zaškrtnuté polia v hornej časti indikujú či sa má kalkulácia robiť hierarchicky (to preto, že WBS elementy sú vo WBS štruktúre projektu usporiadané hierarchicky) a či sa majú do výpočtu brať aj objednávky (tie môžu byť pridelené k WBS elementom).

Parametre sú totožné s KGI2. Možnosti spracovania umožňujú tzv. detailné zobrazenie (to je v prípade KGI2 použité automaticky) a zobrazenie s dialógom. Kalkulácia réžii

využíva koncept konfiguračného hárku a ten je v jednotlivých krokoch výpočtu možné zobraziť aj s vypočítanými hodnotami.

V prípade oboch transakcii vyzerá zobrazenie detailov nasledovne:

The screenshot shows the SAP 'Actual Overhead Calculation: Project/WBS Element/Network Basic list' window. It is divided into several sections:

- Selection:** A table with columns 'Selection Parameters', 'Value', and 'Name'.

Selection Parameters	Value	Name
Project definition	FIN_CZ02	FIN_CZ02_0001
With orders	X	
With hierarchy	X	
Company Code	CZ01	
- Processing Options:** A table with columns 'Selection Parameters' and 'Value'.

Selection Parameters	Value
Execution Type	Overhead Calculated
Processing Mode	Test run
- Processing completed with errors:** A text box indicating the status of the calculation.
- Summary Table:** A table with columns 'Number of Messages', 'Maximum Message Type', 'Error Messages', 'Warning Messages', and 'Information Messages'.

Number of Messages	Maximum Message Type	Error Messages	Warning Messages	Information Messages
4		4		
- Statistics:** A table with columns 'Processing Category', 'Number', and a total row.

Processing Category	Number
Overhead Calculated	7
Not Relevant	1
Inappropriate Status	1
Error	2
Total	11

Obrázok 25 – detailné zobrazenie počítania réžií

(Zdroj: vlastné spracovanie)

Obrazovka, ktorú vidíme na obrázku 25 je plná údajov a nie všetky nesú dostatočnú výpovednú hodnotu. Pre nezaväšenú osobu môže byť náročné zistiť čo všetky jednotlivé časti znamenajú. To najdôležitejšie sú štatistiky na spodku obrazovky, ktoré ukazujú ako dopadol výsledok. Vidíme, že objektov v hierarchii bolo 11 no výpočet prebehol len nad 7. Ostatné objekty obsahovali rôzne chyby. Tieto chyby sme tam úmyselne vytvorili my, aby sme videli ako sa program správa, nakoľko popri analýze rozhranie sme realizovali aj analýzu zdrojového kódu.

Po kliknutí na chyby sme boli schopný dopracovať sa k, niektorým chybovým hláškam:

Actual Overhead Calculation: Project/WBS Element/Network Basic list

Messages: Display messages

Type	Message Text	LTxt
Information	Order 705341	
Error	Profit center CZ01/TEST01 does not exist for 01.03.2020	?
Error	Profit center TEST01 not found in controlling area CZ01	
Information	Order 705342	
Error	Profit center CZ01/TEST01 does not exist for 01.03.2020	?
Error	Profit center TEST01 not found in controlling area CZ01	

Processing completed with

Number of Messages	Maximum Message Type	Error Messages	Warning Messages	Information Messages
4	Error	4	0	0

Statistics

Processing Category	Number
Overhead Calculated	7
Not Relevant	1
Inappropriate Status	1
Error	2
Total	11

Obrázok 26 – chybové hlášky z výpočtu réžie

(Zdroj: vlastné spracovanie)

Chybové správy neniesli ďalšie dodatočné informácie a je pravda, že nie ku každej chybe sme správu dostali. Následne sme pokračovali na obrazovku s výsledkami výpočtu kliknutím na tlačidlo vľavo hore.

Senders	Receiver	Cost Elem.	Val/COArea Crcy	ObCur	Value in Obj. Crcy
CTR PRODUCTION	ORD 4002420	855200	30,00	CZK	750,00
CTR PRODUCTION	WBS FIN_CZ021	855100	30,00	CZK	750,00
CTR PRODUCTION		855200	30,00	CZK	750,00
CTR PRODUCTION	WBS FIN_CZ0211	855100	60,00	CZK	1.500,00
CTR SAP-DUMMY			4,76	CZK	119,00
CTR PRODUCTION		855200	60,00	CZK	1.500,00
CTR SAP-DUMMY		855400	2,38	CZK	59,50
CTR PRODUCTION	WBS FIN_CZ02111	855100	274,00	CZK	6.850,00
CTR PRODUCTION		855200	274,00	CZK	6.850,00
CTR PRODUCTION	WBS FIN_CZ022	855100	38,00	CZK	950,00
CTR SAP-DUMMY			4,80	CZK	125,00
CTR PRODUCTION		855200	38,00	CZK	950,00
CTR PRODUCTION	WBS FIN_CZ0221		10,00	CZK	250,00
			47,18	CZK	1.184,50

Obrázok 27 – tabuľka výsledkov výpočtu rézie

(Zdroj: vlastné spracovanie)

V tabuľke, ktorá sa zobrazila po kliknutí na dotyčné tlačidlo sme mohli pozorovať operácie nad objektami, ktoré boli do kalkulácie zapojené. Opäť sú tieto výsledky pomerne mäťúce pre niekoho, kto ich vidí prvý krát. V prípade jedného objektu sú výsledky často jednoduchšie.

Senders	Receiver	Cost Elem.	Val/COArea Crcy	ObCur	Value in Obj. Crcy
CTR SAP-DUMMY	ORD 4002421	855100	6,08	CZK	152,00
CTR PRODUCTION		855200	20,00	CZK	500,00
			26,08	CZK	652,00

Obrázok 28 – výsledky pre jednu objednávku

(Zdroj: vlastné spracovanie)

V prípade obrázku sme počítali len s jednou objednávkou. AK by však zákazník mal tisíce objektov zapojených do výpočtu, bola by táto výsledná obrazovka veľmi neprehľadná. Je zvykom veľké dávky dát počítať masovo a často sa to robí ako proces na pozadí systému. Tieto aplikácie sú koncipované predovšetkým na jednotlivé objekty,

alebo objekty v racionálnom počte, hoci zdrojový kód odhalil, že zvláda aj dávkové spracovávanie.

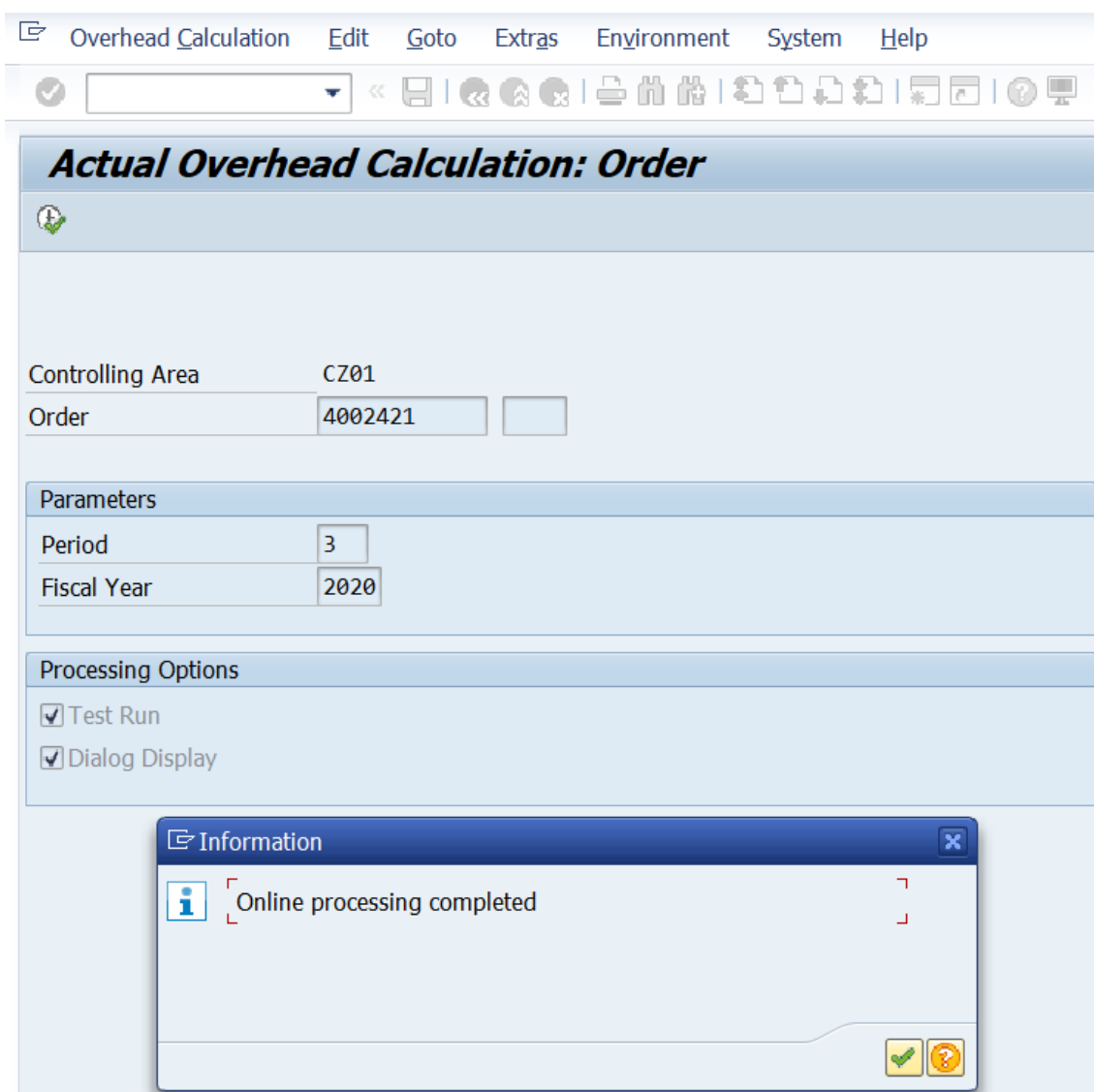
Pokiaľ by sme pred spustením zvolili aj možnosť dialógového zobrazovania konfiguračného hárku vyzeralo by to nasledovne:

I..	Cn...	Description	Amount	Crcy	per	Condition Value	Curr.	Status	Num...	ATO/MTS Component	OUn	CCon...	Un	Cond
	B000	Material				152,00	EUR						0	
	C000	Material OH	4,000	%		6,08	EUR						0	
		Material usage	0,00	EUR	1	158,08	EUR						0	
	B010	Production				100,00	EUR						0	
	C300	Hpers/quantity base	50,00	CZK	1H	20,00	EUR			1			1H	

Obrázok 29 – konfiguračný hárak s vypočítanými hodnotami

(Zdroj: vlastné spracovanie)

O konfiguračnom hárku budeme pojednávať ešte neskôr, keďže jeho konfigurácia je veľmi dôležitá pre beh a fungovanie niekoľkých aplikácií. Záverom v prípade, že sme použili dialógové zobrazenie bude na hlavnej obrazovke vyskakovacie okno informujúce o dokončení spracovania.



Obrázok 30 – dokončenie spracovávaní pri dialógovom zobrazovaní

(Zdroj: vlastná réžia)

Keď je beh transakcií spustený naživo, realizuje sa zápis dokumentov do databázy a tieto sú potom viditeľné aj s použitím iných účtovníckych transakcií. My sme použili transakciu KOB1, aby sme si overili výsledky behu aplikácie na dokumentoch.

Display Actual Cost Line Items for Orders

Document Master Record

Layout 1SAP Primary cost posting
 Order 4002421 Aggregate crushing
 Report currency EUR Euro

Cost Element	Cost element name	Val.in rep.cur.	Total quantity	PUM	O...	Offsetting Account	Name of offsetting account
501100	Cons.mater.d.3000	152,00				S 211000	Cash
821000	DAA work costs	100,00	10	H			
855100	Adm bonus direct mat	6,08					
855200	Adm personnel costs	20,00					
Order 4002421 Aggregate crushing		278,08					
		278,08					

Obrázok 31 – prehľad účtovníckych operácií na nákladovom objekte v transakcii KOB1

(Zdroj: vlastné spracovanie)

V tejto transakcii to vyzeralo už o čosi zrozumiteľnejšie. V prvom riadku sme mohli pozorovať, že na stranu má dať sme účtovali účet 501100 a na stranu dal 211000. Dôkaz toho sme našli v dokumentoch, ktoré vie táto transakcia zobrazit’.

Display Actual Cost Line Items for Orders

Document Master Record

Layout 1SAP Primary cost posting
 Order 4002421 Aggregate crushing
 Report currency EUR Euro

List of Documents in Accounting

Cost Element	quantity	PUM	O...	Offsetting Account	Name of offsetting account
501100				S 211000	Cash
821000	10	H			
855100					
855200					
Order 4002421					

Documents in Accounting

Document	Object type text
0100000023	Accounting document
A000393700	Controlling Document

Separate Original document

Obrázok 32 – zoznam dokumentov vygenerovaných operáciou výpočtu réžie

(Zdroj: vlastné spracovanie)

Patričné dokumenty sa delia na oblasť klasického finančného účtovníctva a oblasť kontrolingu, alebo vnútorného účtovníctva.

Display Document: Data Entry View

Taxes Display Currency General Ledger View

Data Entry View

Document Number: 110000023 Company Code: CZ01 Fiscal Year: 2020
 Document Date: 18.03.2020 Posting Date: 18.03.2020 Period: 3
 Reference: Cross-Comp.No.: Texts Exist: Ledger Group:

Currency: CZK

D.	CoCd	Value Date	Item	Clrng doc.	Key SG	Account	Description	Amount	Curr.	Cost Center	Profit Center	Venture
S	CZ01		1		40	501100	Cons.mater.d.3000	3.800,00	CZK		PRODUCTION	
H			2		50	211000	Cash	3.800,00-	CZK			

Obrázok 33 – dokumenty zachytávajúce účtovnícke operácie – finančné účtovníctvo

(Zdroj: vlastné spracovanie)

Na obrázku 34 si môžeme všimnúť operácie, ktoré sa vykonali, vidíme tiež, že pred sebou máme dokument, ktorý je identifikovateľný číslom dokumentu, ktoré je unikátne a má stanovený číselný interval.

Display Actual Cost Documents

Document Master Record

Layout: 1SAP Primary cost posting
 COarea currency: EUR EUR
 Valuation View/Group: 0 Legal Valuation

DocumentNo	Doc. Date	Document Header Text	RT	RefDocNo	User Name	Rev	RvD
PRw	OTy	Object	CO object name	Cost Elem.	Cost element name	Val	COArea Crncy
300004600	03.12.2020				FLORIAN		
1	ORD 4002421	Aggregate crushing	855100	Adm bonus direct mat		6,08	
2	CTR SAP-DUMMY	SAP DUMMY	855100	Adm bonus direct mat		6,08-	
3	ORD 4002421	Aggregate crushing	855200	Adm personnel costs		20,00	
4	CTR PRODUCTION	Production	855200	Adm personnel costs		20,00-	

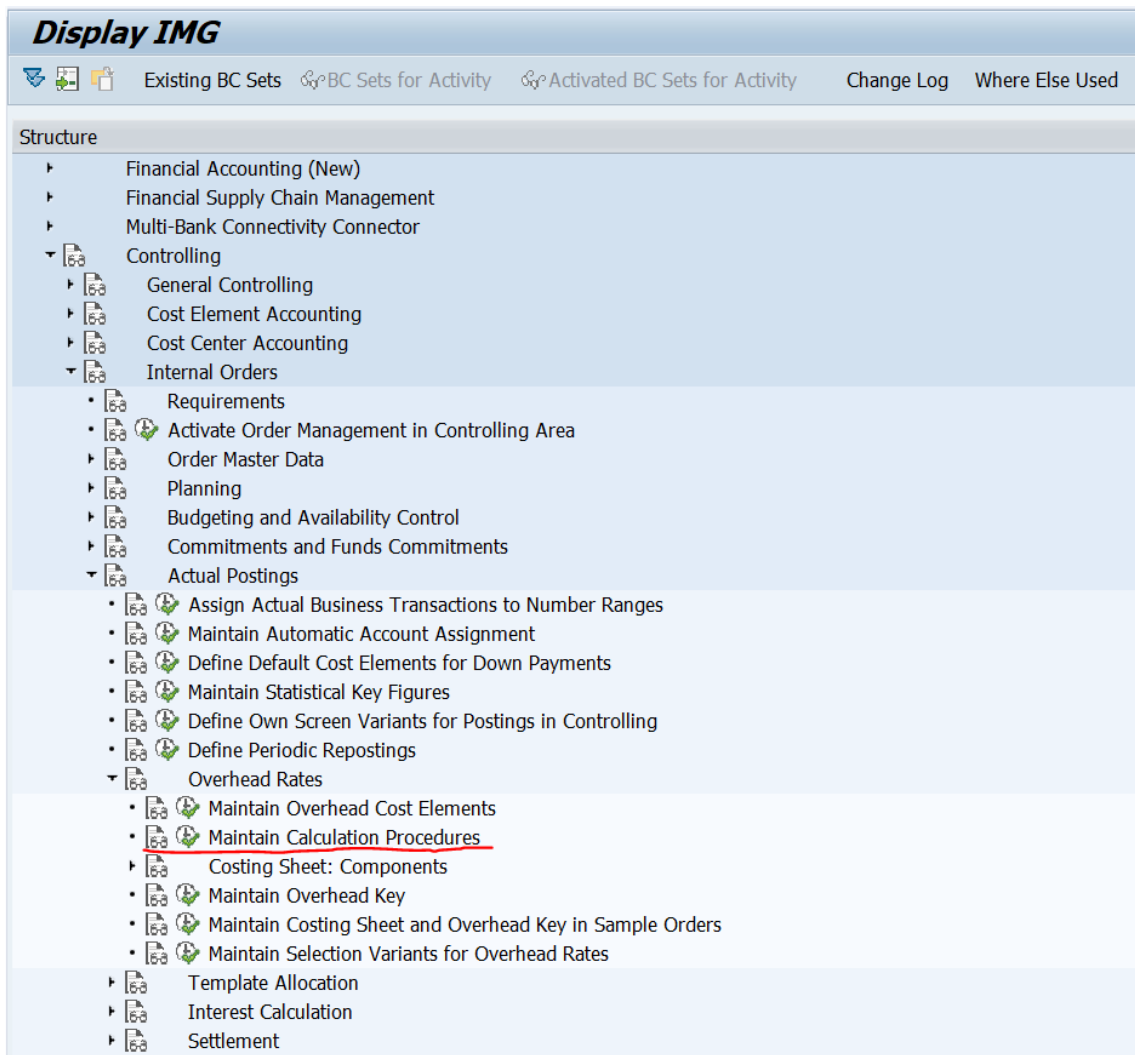
Obrázok 34 – kontrolingový dokument, ktorý vznikol z výpočtu réžie

(Zdroj: vlastné spracovanie)

Kontrolingový dokument už vyzerá odlišne a má aj odlišné operácie, takže dojem môže byť do istej miery podobný no pri podrobnejšom skúmaní je cítiť rozdiely. Číslo dokumentu je prítomné aj v tomto prípade a je vidno, že sa nachádza v odlišnom časovom intervale, čo je tiež dôležitý spôsob diferenciacie, keď je potreba analyzovať nespracované dáta z databázy a my potrebujeme vedieť s akým typom dokumentu máme dočinenia.

2.4.1 KONFIGURAČNÝ HÁROK

Konfiguračný hárok, ako sme už vyššie spomenuli tvorí neoddeliteľnú súčasť fungovania programov počítajúcich réžie. Tento hárok je súčasťou konfiguračnej hierarchie každého systému od SAPu. Ide spravidla o časť „customizácie“ (alebo úprav pre prispôsobenie riešenia, aby slúžilo potrebám toho ktorého zákazníka) a kmeňových dát.



Obrázok 35 – aktivita úprav konfigurácie v transakcii SPRO

(Zdroj: vlastné spracovanie)

Cez transakciu SPRO je možné získať ku konfiguračnej časti riešenia prístup, toto je miesto, s ktorým obvykle pracujú konzultanti, ktorí sú poverení konfiguráciou riešenia po jeho dodaní a obvykle je to súčasťou implementácie riešenia. Výnimkou sú samozrejme úpravy počas cyklu fungovania riešenia IS, keď sa napríklad zmenia či rozšíria potreby spoločnosti zákazníka, ktorý riešenie používa.

Change View "Costing sheets": Overview

New Entries

Dialog Structure

- Costing sheets
 - Costing sheet row
 - Base
 - Overhead rate
 - Credit

Costing Sheets

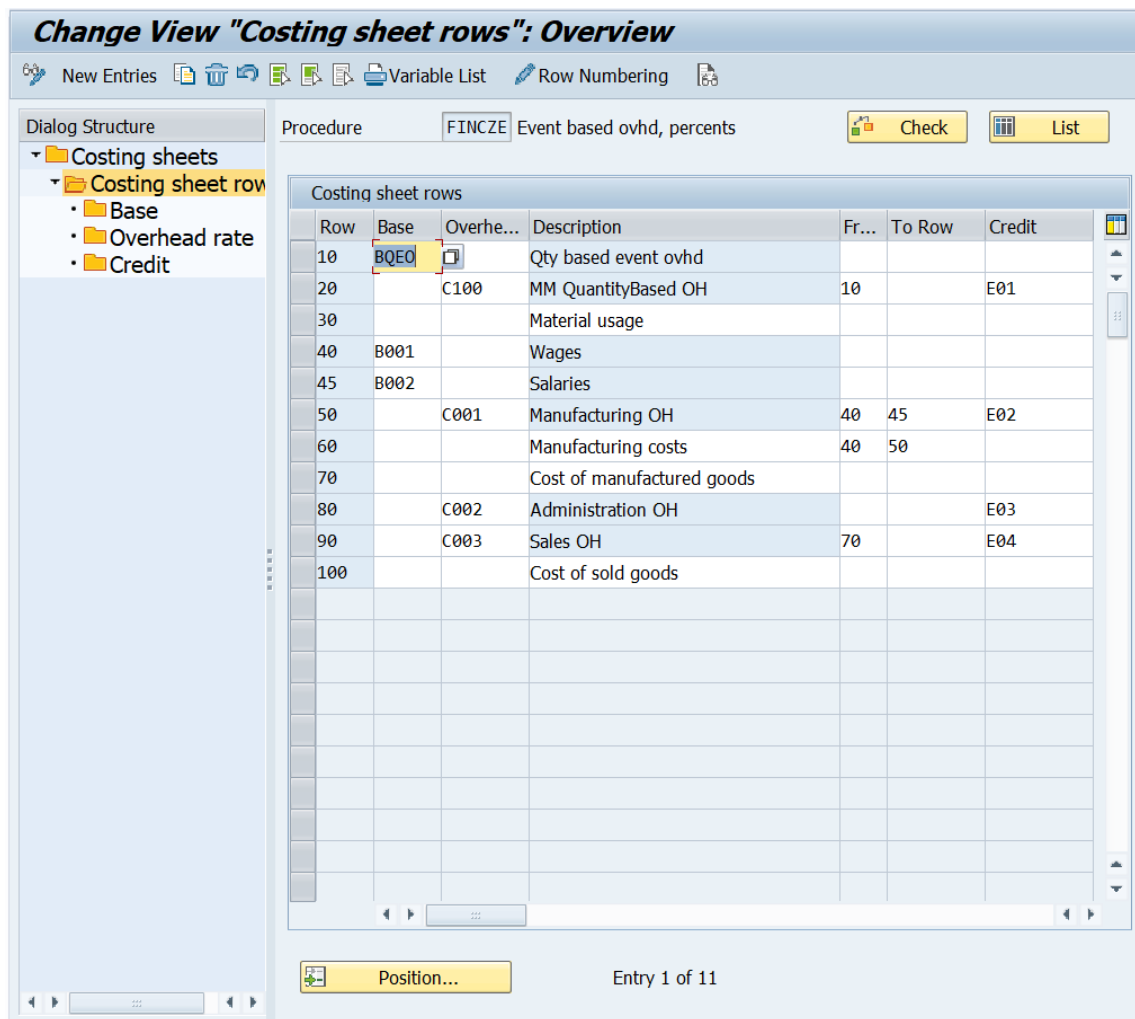
Costing Sheet	Description	Evt. based
A00000	Standard	<input type="checkbox"/>
A00001	Standard/Surcharge Key	<input checked="" type="checkbox"/>
A00002	Standard/Plant	<input type="checkbox"/>
A00003	Standard/Company Code	<input type="checkbox"/>
A00005	Standard/Order Type	<input type="checkbox"/>
A00007	Standard Base Cost Ctr/ActTyp	<input checked="" type="checkbox"/>
B001	Test BO	<input type="checkbox"/>
CO-QB	Quantity Based Overhead Calcul	<input type="checkbox"/>
CO_QB2	Quantity-based for SHA CO	<input type="checkbox"/>
CUMUL	Standard Cumulated Overheads	<input checked="" type="checkbox"/>
DAVECS		<input type="checkbox"/>
EMPTY	Keep this empty	<input checked="" type="checkbox"/>
FIN000	Standard FINCZ1	<input type="checkbox"/>
FINCZE	Event based ovhd, percents	<input checked="" type="checkbox"/>
JTH01		<input type="checkbox"/>
JTH02		<input type="checkbox"/>
MLK-01	PP-PC Standard w/o Admin Overh	<input type="checkbox"/>
OC01	Overhead calculation - example	<input type="checkbox"/>
PP-PC1	PP-PC Standard	<input checked="" type="checkbox"/>
PP-PC2	PP-PC Surch.on CostOfGoodsManu	<input type="checkbox"/>
PP-PC3	Surcharges for manuf. costs	<input type="checkbox"/>
PP-PC4	PP-PC standard for SHA CO2	<input checked="" type="checkbox"/>

Position... Entry 1 of 35

Obrázok 36 – zoznam konfiguračných hárkov

(Zdroj: vlastné spracovanie)

Po spustení úprav vidí užívateľ tabuľku s definovanými hárkami, ktoré pomocou navigačného menu možno prechádzať v rôznej granularite.

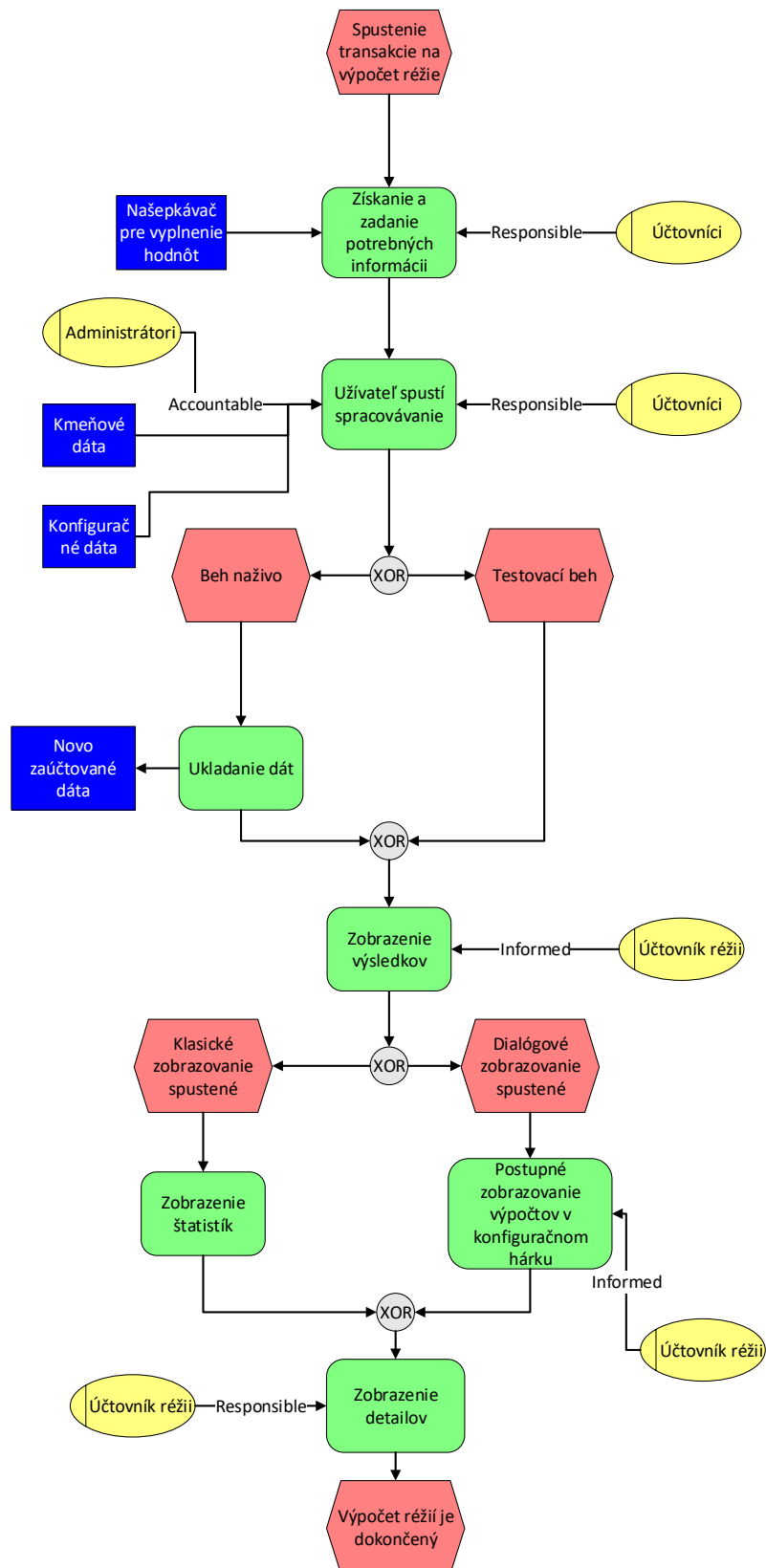


Obrázok 37 – riadku konfiguračného hárku

(Zdroj: vlastné spracovanie)

Po vykonaní voľby vidíme riadky konfiguračného hárku v ktorých je definovaná hodnota bázy(to je druh intervalu, v ktorom sa definujú nákladové objekty, prípadne ich skupiny), hodnota režijných mier(tu sa definujú konkrétne hodnoty pre výpočet réžie) a konfigurácia objektu, ktorý má byť na strane dal v prípade výpočtu réžie.

Výpočet réžie môže prebiehať v dvoch variantoch. Prvým je na základe percentuálnej kalkulácie a druhým je to na základe množstva, čo je vyjadrené ako počet peňazí v nejakej mene za nejakú definovanú jednotku. Okrem toho konfiguračný hárak obsahuje aj príznak či má byť réžia počítaná ako súčasť reťaze činností, teda či má byť výpočet spustený automaticky po tom, ako prebehne nejaká iná biznis transakcia ako napríklad alokácia nákladov ručne.

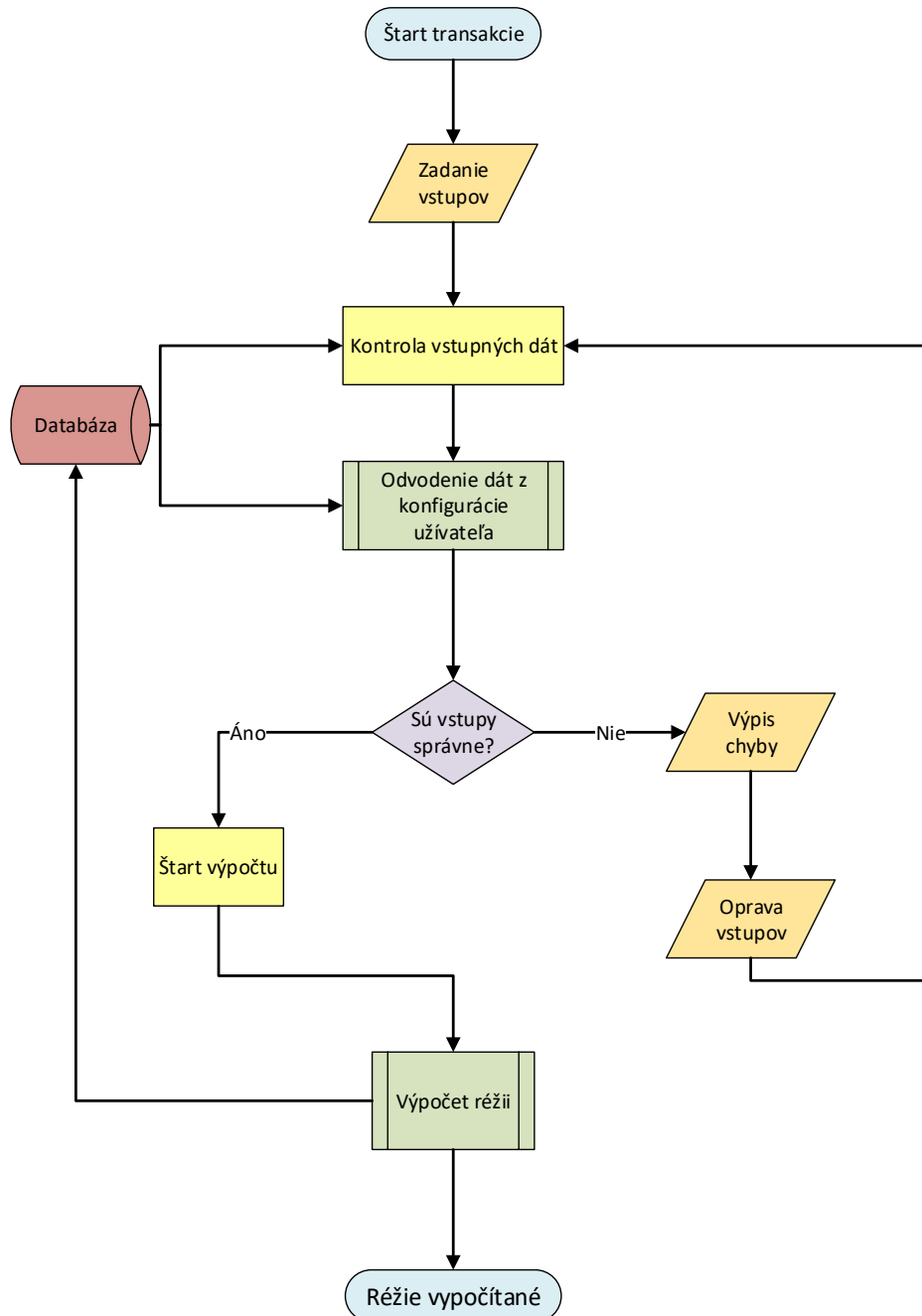


Obrázok 41 - Diagram behu procesu výpočtu réžii

(Zdroj: vlastné spracovanie)

2.4.3 ZJEDNODUŠENÝ POHĽAD NA ALGORITMUS VÝPOČTU

Nasledujúca ilustrácia predstavuje veľmi zjednodušený pohľad na algoritmus, ktorý zhruba nasleduje ten uvedený na [obrázku 41](#). Tu je ešte vhodné objasniť, že niektoré nepotrebné detaily sú pre jednoduchosť vynechané, pričom detaily, týkajúce sa intelektuálneho vlastníctva sú tiež opomenuté.

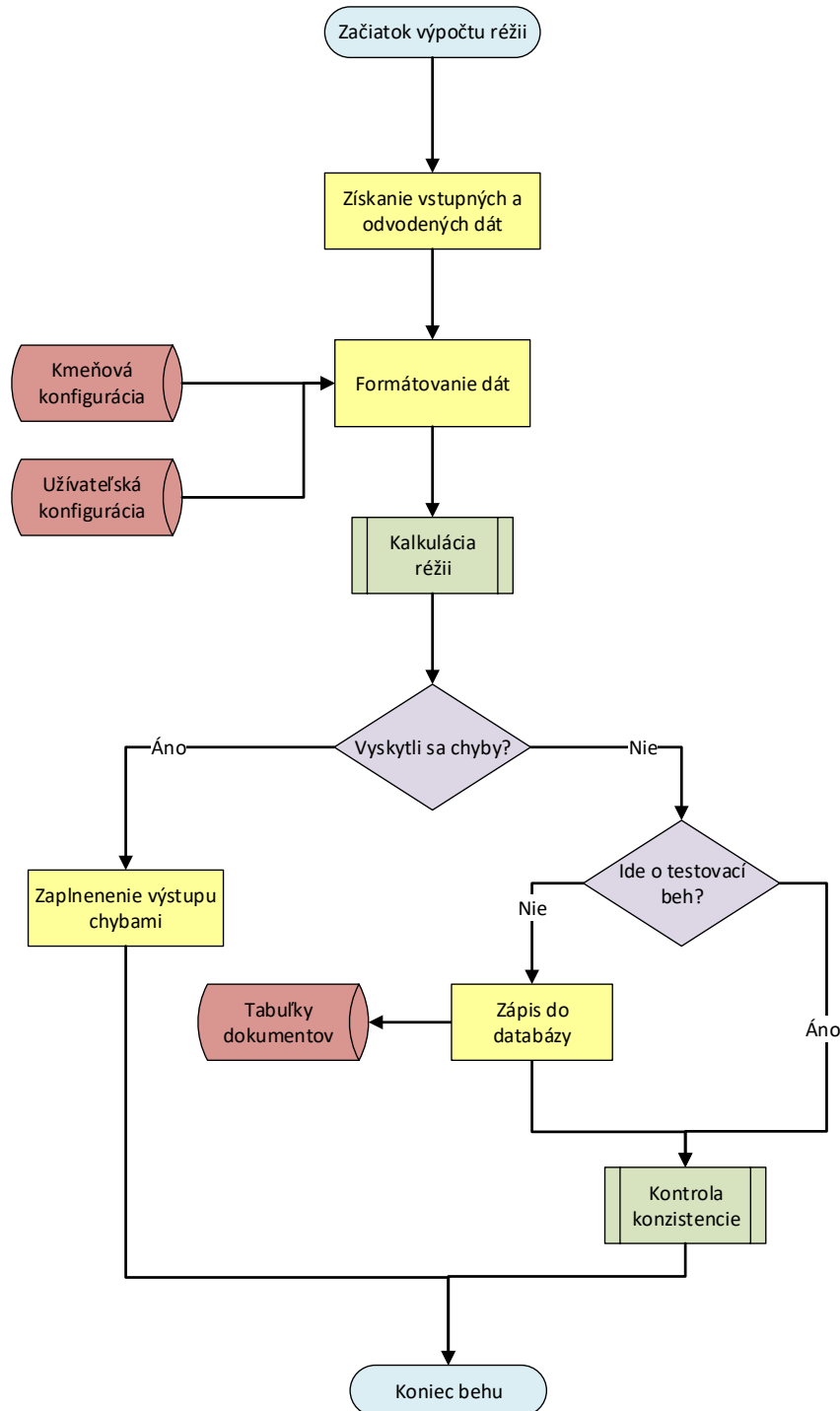


Obrázok 42 – Zjednodušený algoritmus výpočtu réžii

(Zdroj: vlastné spracovanie)

2.4.4 ALGORITMUS VÝPOČTOVÉHO PODPROCESU

V tomto prípade ide taktiež o jednoduchý pohľad na pod-proces, ktorý je na obrázku 42 označený ako výpočet réžii.

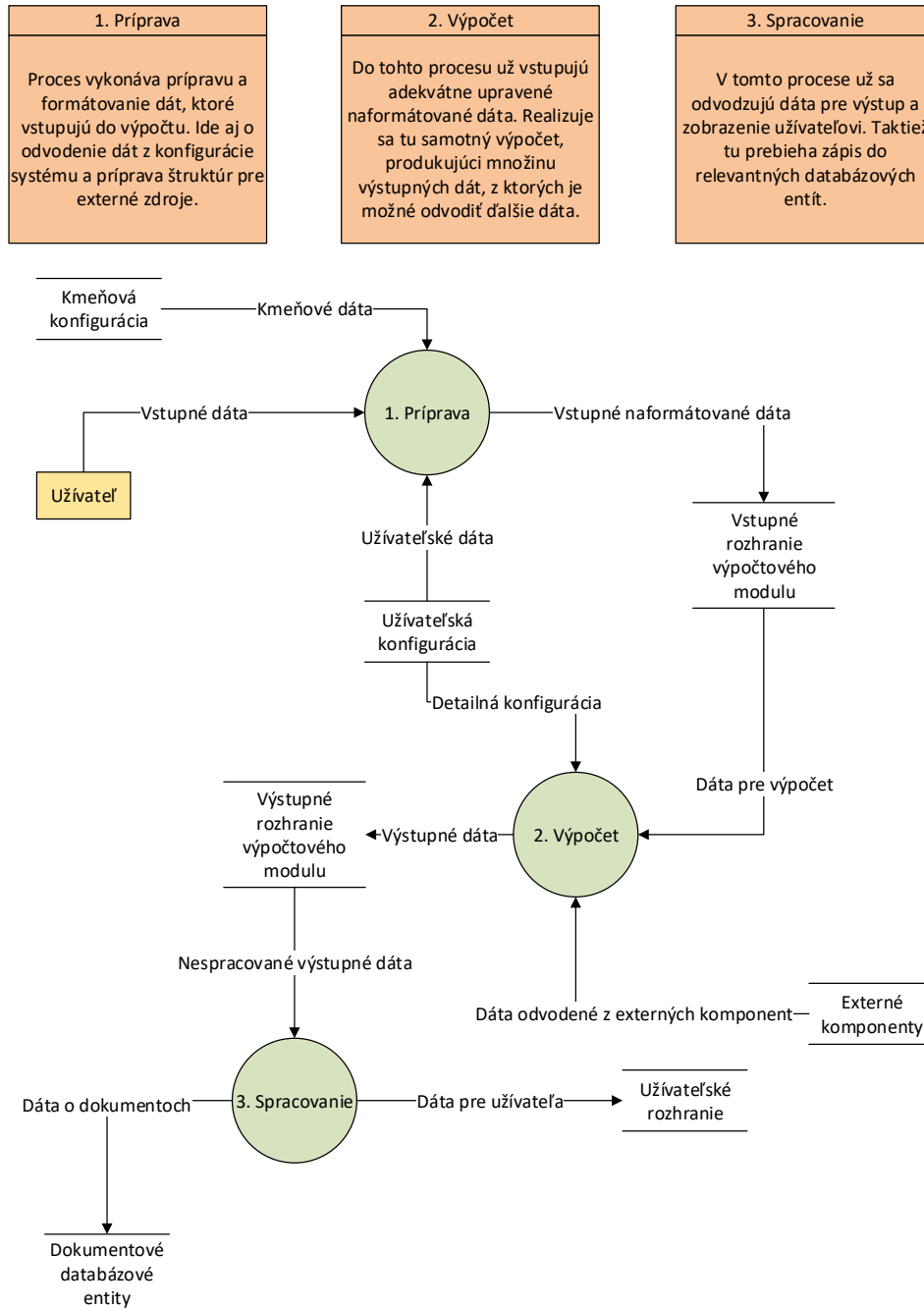


Obrázok 43 – Zjednodušený algoritmus výpočtu réžii

(Zdroj: vlastné spracovanie)

2.4.5 DIAGRAMY TOKU DÁT

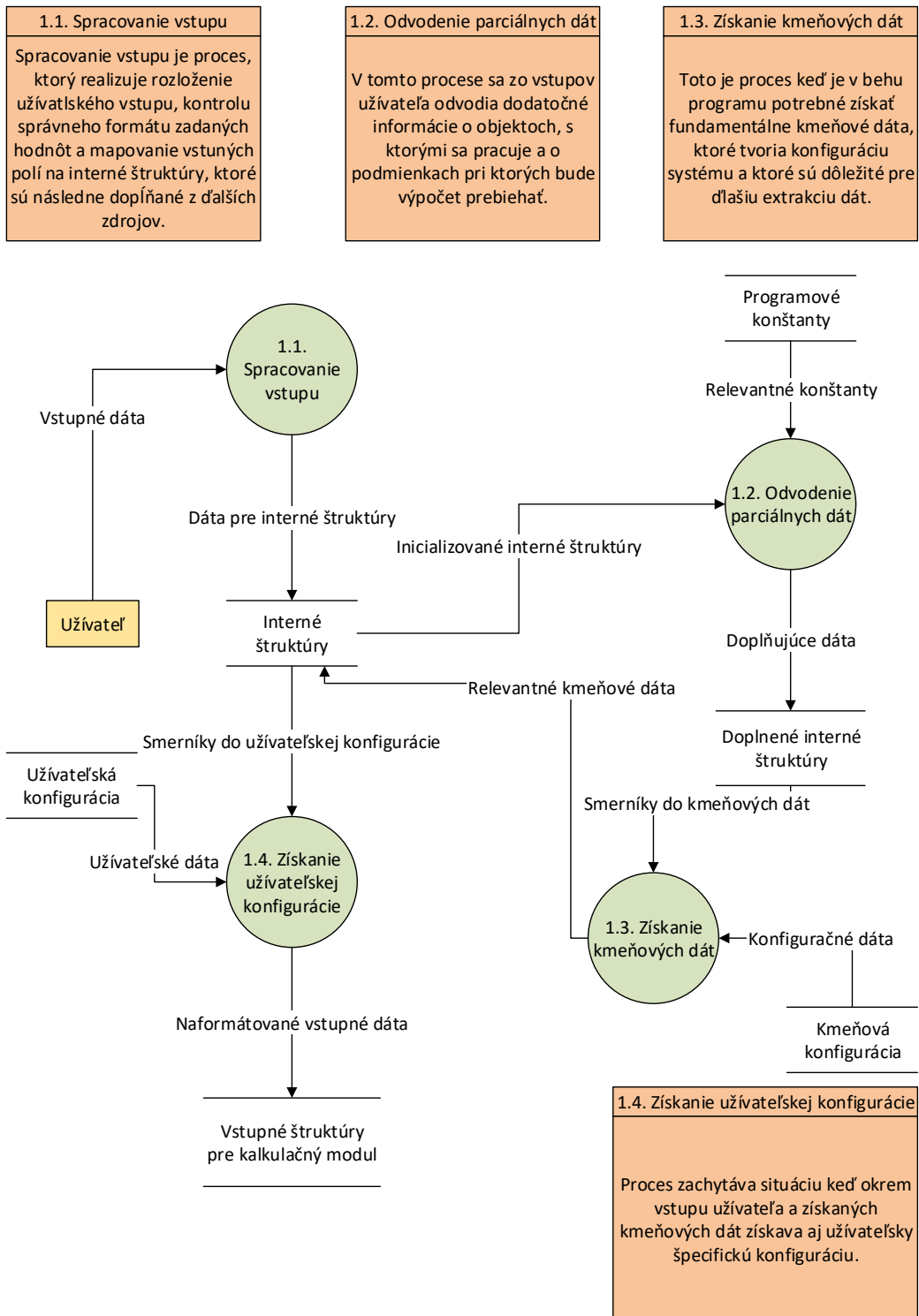
Veríme, že doposiaľ uvedené diagramy poskytujú primeraný náhľad do technických skutočností, s ktorými budeme pri spracovávaní riešenia pracovať. Teraz nasledujú diagramy, ktoré sú nemenej dôležité, keďže zachytávajú prúdenie dát.



Obrázok 44 – Diagram prúdenia dát, úroveň 0

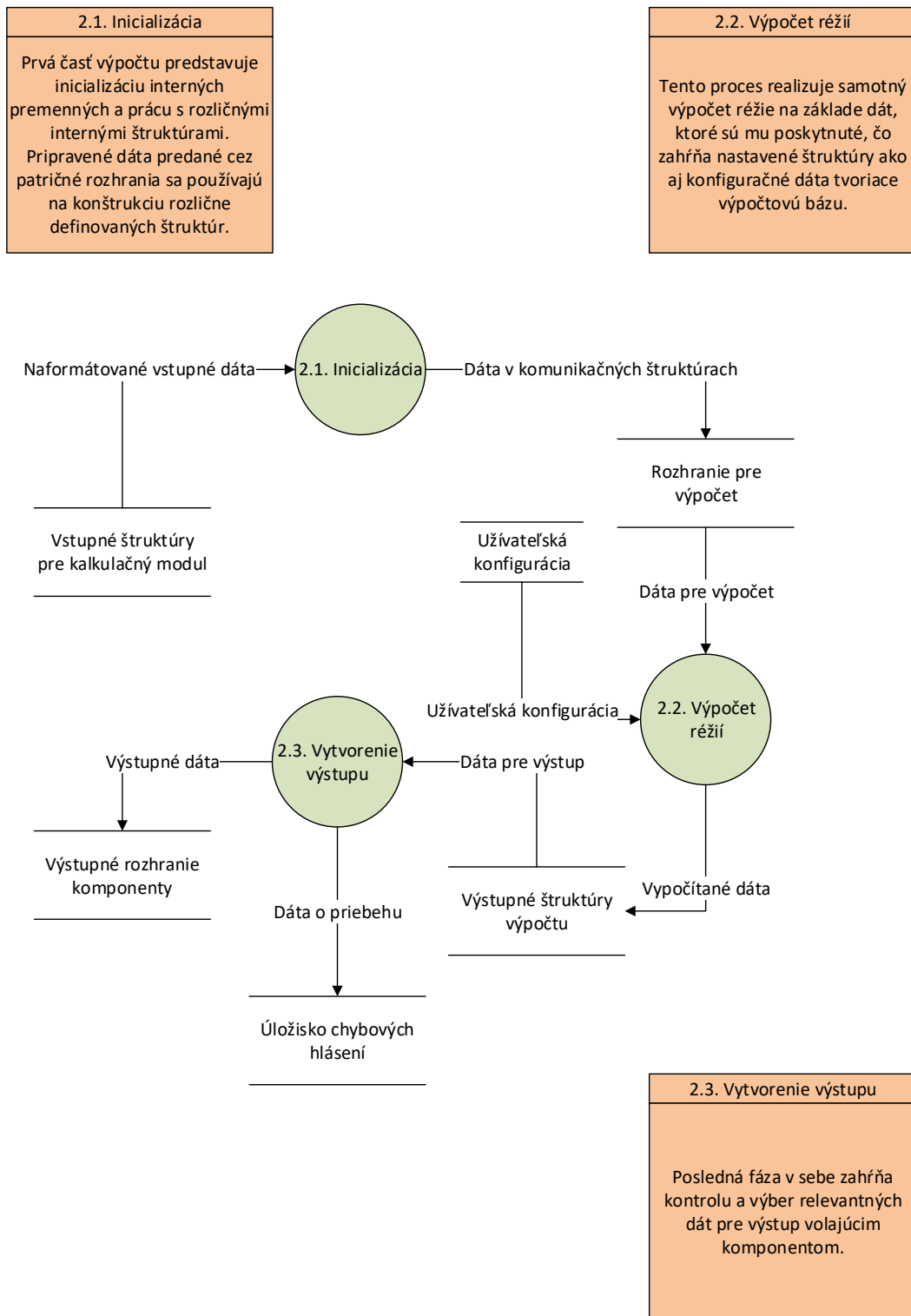
(Zdroj: vlastné spracovanie)

3 hlavné fázy, ktoré z predošlého obrázku môžeme pozorovať sú príprava, výpočet a spracovanie, ktoré opäť reflektujú skutočnosti z predošlých podkapitol, no tento krát je to z pohľadu všeobecného prúdenia dát v programe.



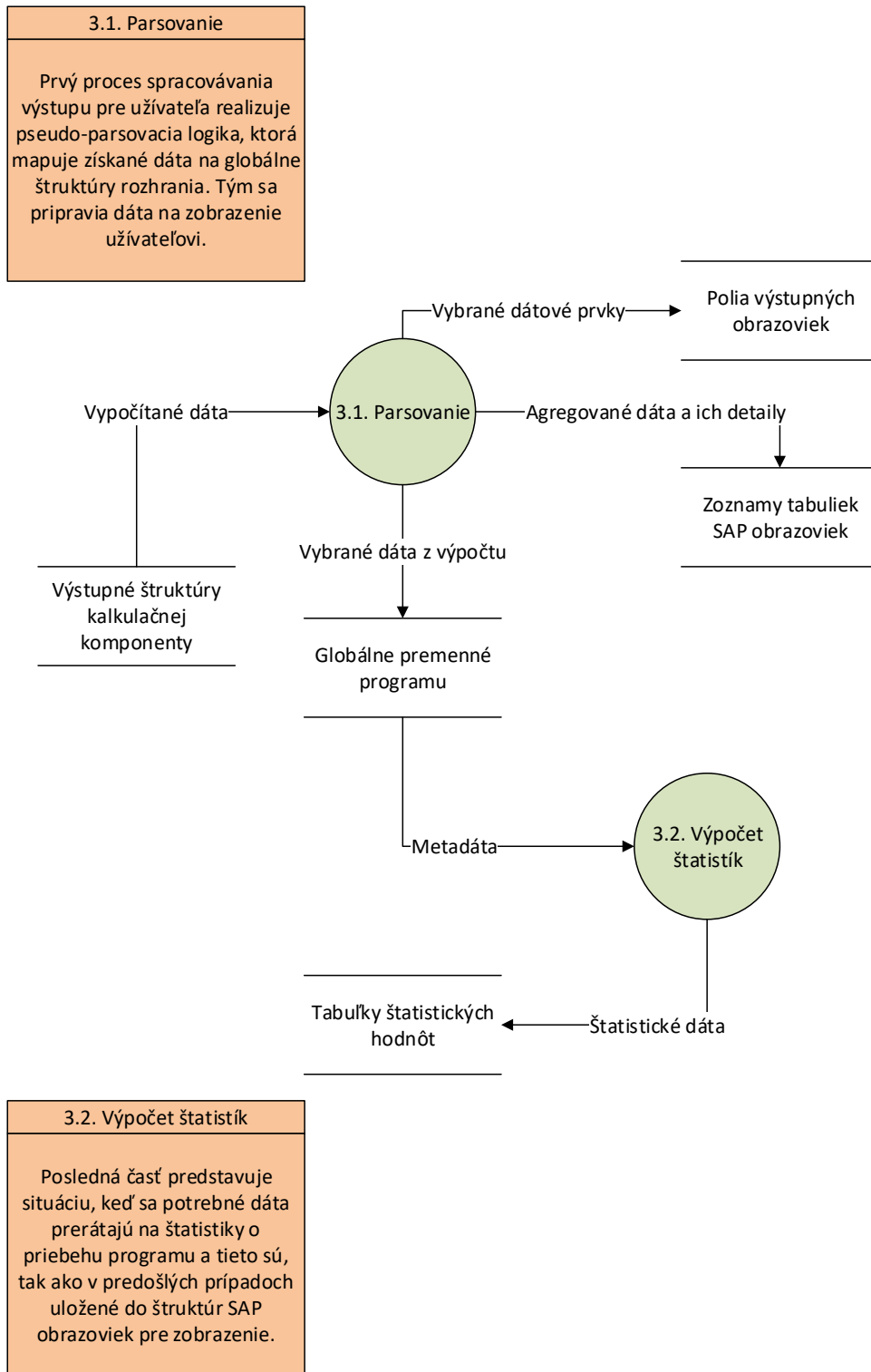
Obrázok 45 – Diagram toku dát 1. úrovne, proces prípravy dát

(Zdroj: vlastné spracovanie)



Obrázok 46 – Diagram toku dát 1. úrovne, proces výpočtu réžii

(Zdroj: vlastné spracovanie)

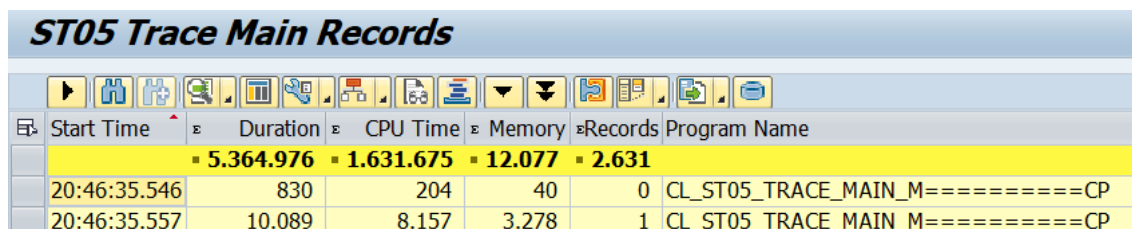


Obrázok 47 – Diagram prúdenia dát 1. úrovne, proces spracovania

(Zdroj: vlastné spracovanie)

2.4.6 MERANIE VÝKONU

Výkon bol meraný na behu programu KGI2 na jednom z klasických testovacích systémov. Išlo o serverový cluster, využívajúci databázu HANA. Ide o cloudové prostredie.

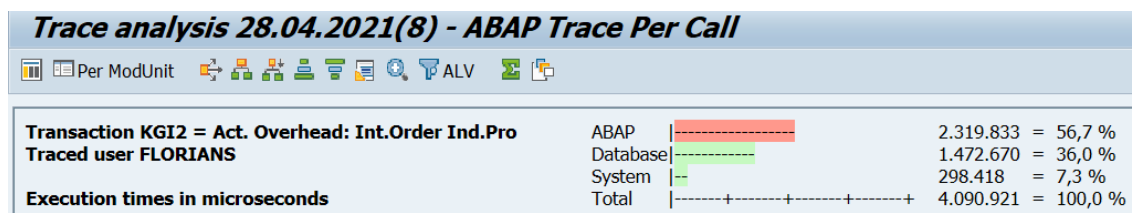


Start Time	Duration	CPU Time	Memory	Records	Program Name
	5.364.976	1.631.675	12.077	2.631	
20:46:35.546	830	204	40	0	CL_ST05_TRACE_MAIN_M=====CP
20:46:35.557	10.089	8.157	3.278	1	CL_ST05_TRACE_MAIN_M=====CP

Obrázok 48 – Meranie výkonu transakciou ST05

(Zdroj: vlastné spracovanie)

Na predošlom obrázku je možné vidieť výstup merania, v transakcii ST05, ktorá bola spustená v rámci nástrojov transakcie ST12. Tu je potrebné uviesť niektoré kľúčové informácie. Meranie sa týka zásadne prístupov do databázy ide o tzv. SQL trace. Sumačné časy uvedené vo svetložltom riadku(prvom od vrchu) sú uvedené v mikrosekundách. Ide o výkon databázy, resp. databázového servera. Ako je vidno ide o veľmi malé čísla a jediná informácia, ktorú sa dozvedáme, je, že práca s databázou je v relatívnom poriadku. Samotný zoznam údajov je príliš dlhý, ale z našej inšpekcie sme nezistili žiadne úzke miesta, ani zbytočné dopyty voči databáze. Dohromady išlo o približne 5 sekúnd.



Transaction KGI2 = Act. Overhead: Int.Order Ind.Pro		ABAP	2.319.833 = 56,7 %
Traced user FLORIAN S		Database	1.472.670 = 36,0 %
Execution times in microseconds		System	298.418 = 7,3 %
		Total	4.090.921 = 100,0 %

Obrázok 49 – Meranie kódu ABAPu

(Zdroj: vlastné spracovanie)

Predošlý obrázok zachytáva meranie výkonnosti kódu ABAPu, ktorý sa vykonal pri behu KGI2. Celkový čas výkonu bol približne 4 sekundy, hodnoty vyššie sú opäť v mikrosekundách. Pri detailnejšej analýze modularizovaných jednotiek bol zistený potenciál pre niekoľko úzkych miest.

Tabuľka 1 – Kľúčové metriky výkonu KGI2

(Zdroj: vlastné spracovanie)

Index	Počet okruhov	Okruh v ms	Databázové dopyty v ms	Požadovaný objem dát DB (Bajty)	Počet volaných DB záznamov	Pamäť v kB	Maximálna a použitá pamäť (kB)	Použitie rozšírenej pamäte kB
1	2	158	6	4,336	4	15,783	82,781	15,783
2	1	204	84	3,876	21	3,596	167,153	3,596
3	4	3,141	1,648	241,731	1,978	12,303	167,153	12,303
4	8	1,015	136	31,712	90	15,751	167,153	15,751
5	4	1,375	323	68,516	356	15,783	82,781	15,783
6	7	734	78	30,012	80	15,783	82,781	15,783
7	3	886	111	33,898	102	15,783	82,781	15,783
8	2	142	5	398	3	15,783	82,781	15,783

Predošlá tabuľka obsahuje kľúčové metriky merania, ide najmä o prístupy do databázy a prácu s pamäťou servera. Jednotlivé riadky reprezentujú volania rozličných podprogramov, tým sa nemyslia funkcie, metódy a pod., ale iné transakcie, či programy ako celky. Okruhy znamenajú koľko krát muselo GUI rozhranie volať back-end pre výpočet, alebo dáta, prípadne môže ísť o komunikáciu paralelných procesov.

Tabuľka 2 – Doplnkové metriky KGI2

(Zdroj: vlastné spracovanie)

Index	CPU čas v ms	Čas na pracovný proces v ms	GUI a sieťový čas v ms
1	60	60	98
2	130	204	0
3	1,010	2,572	569
4	290	431	584
5	570	841	534
6	230	283	451
7	250	346	540
8	40	54	88

Uvedená tabuľka je doplnkom k tabuľke 1 a obe spolu tvoria malú časť skutočne nameraných metrick. V prípade tejto, druhej tabuľky môžeme vidieť čas jednotlivých podprogramov na procesore a na pracovných procesoch (to sú vlákna aplikačného servera, ktoré možno vnímať ako podprocesy operačného systému).

2.4.7 SÚHRN

V tejto rozsiahlej podkapitole sme predstavili sadu detailov, z ktorých aplikácia, ktorá je predmetom tejto práce pozostáva. Z obrázkov, ktoré sme uviedli je možné pozorovať značnú komplexitu danej úlohy, na ktorej sme sa podieľali aj pri implementácii. Dotyčné obrázky sú súčasťou analýzy, ktorú sme vykonali aj v našom pracovnom prostredí, a ktorá slúži ako podklad pre analýzu požiadaviek pred plnohodnotným spustením vývoja. Je zrejmé, že sme v tejto práci nešli do tých najmenších detailov. Okrem rozhrania bola vykonaná aj hĺbková analýza zdrojového kódu, ktorý je za užívateľským rozhraním a hlavne za funkčnosťou, ktorú možno v uľahčenej miere pozorovať na obrázkoch s diagramami. Analyzovaný zdrojový kód je veľmi vysokej sofistikovanosti, no zároveň je aj veľmi starého dáta, takže rozumieť jazyku ABAP v tomto prípade znamená rozumieť programovacím koncepciám v jazyku ABAP z pred 40 rokov.

So záveru tejto časti analýzy plynie, že implementácia, ktorú budeme realizovať (a to je dané aj pôvodnými požiadavkami na projekt) je pre počítanie jednotlivých nákladových objektov kontroľingu (nepôjde o aplikáciu na masové spracovávanie) a ide o druh periodického výpočtu v prípade kvantitatívnej réžie ako aj finančných hodnôt na základe konfiguračného hárka. Funkčnosť voči, ktorej budeme programovať zaisťuje prepojenie na všetky relevantné moduly a komponenty (počítame s komunikáciou s modulom Financí).

Dotyčnú funkčnosť sme odmerali pomocou relevantných nástrojov a výsledky sme uviedli v podkapitole 2.4.6 o meraní výkonu. Celkové zoznamy tabuliek merania sme kvôli pragmatickosti neuvádzali, ale uviedli sme kľúčové riadky. S jadrovou funkčnosťou budeme pracovať aj pri našej implementácii takže je potreba počítat s minimálne takými hodnotami aké boli namerané. To čo nahrádzame sú hodnoty týkajúce sa SAP GUI, čo so sebou nesie aj redundantné volanie zvyškovej logiky.

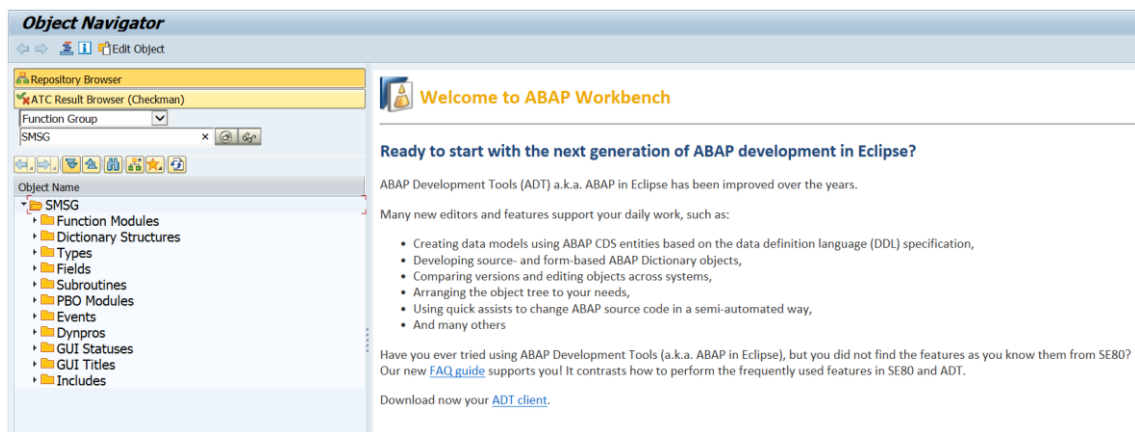
Patričné merania rozšírené o niektoré detaily budeme realizovať aj v prípade implementovaného riešenia.

2.5 ANALÝZA DOSTUPNÝCH ZDROJOV A TECHNOLOGIÍ

Vývoj budeme realizovať v určenom testovacom prostredí, ktoré predstavuje jeden systém rozložený na zhluk viacerých zariadení. Tento zhluk má pridelené rozsiahle technické zdroje a slúži na vývoj a testovanie aplikácie pre stovky vývojárov po celom svete. Verzie pre cloud aj riešenia na domácu infraštruktúru sa v prvom rade programujú a testujú tu, pričom transportný mechanizmus ich na žiadosť vývojárov prenáša na testovacie prostredie, ktoré pozostáva z mnohých systémov na ktorých sú rozličné podmienky v podobe diverzifikovaných dát a realistických podmienok, ktoré môžu nastať na systémoch zákazníka. Po tom ako riešenie prejde všetkými testami na všetkých potrebných systémoch môže byť doručené zákazníkovi, ktorí si ho kúpili. Tento istý mechanizmus využívajú aj situácie, pri ktorých sa robia drobné opravy už aktívnych programov.

2.5.1 NÁSTROJE VÝVOJA

Medzi nástroje vývoja patria štandardné vývojárske programy, ktoré poskytuje spoločnosť SAP a ide o rozličné vývojárske prostredia prispôbené pre SAP. Takýto typický nástroj je transakcia SE80, ktorá bola dlhé roky to, čo sa na akýkoľvek vývoj používalo.

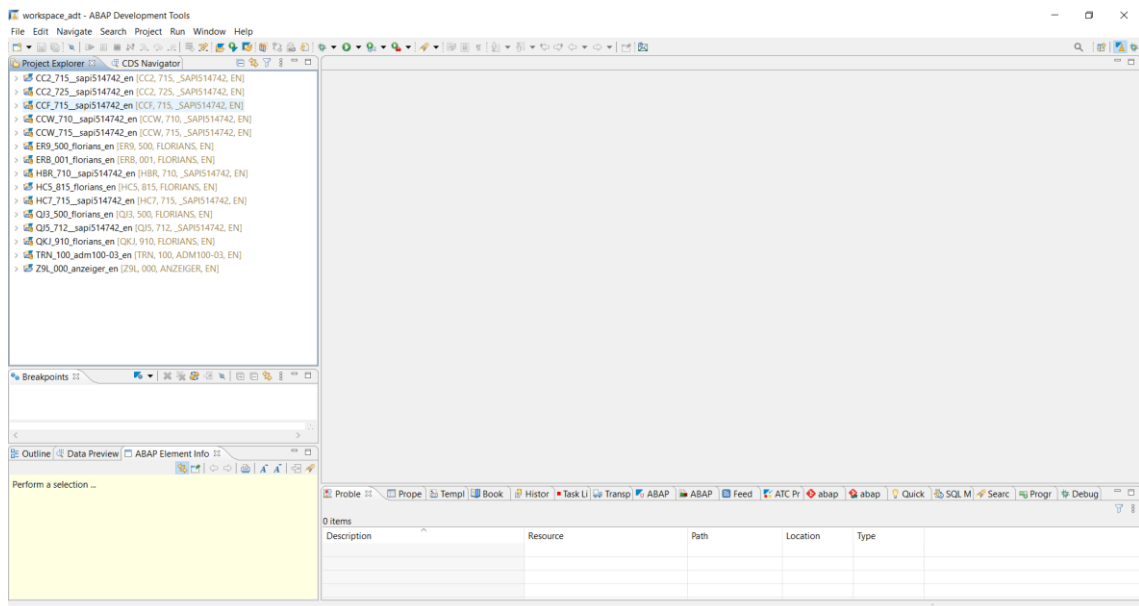


Obrázok 50 – Ukážka transakcie SE80

(Zdroj: vlastné spracovanie)

Ďalším nástrojom, ktorý sa dnes používa oveľa častejšie je tzv. ADT, alebo „ABAP development tools“, ktoré je postavené na prostredí Eclipse, pričom ADT je modifikáciou tohto open-source software. Zásadný rozdiel ADT oproti bežnému Eclipse IDE tkvie

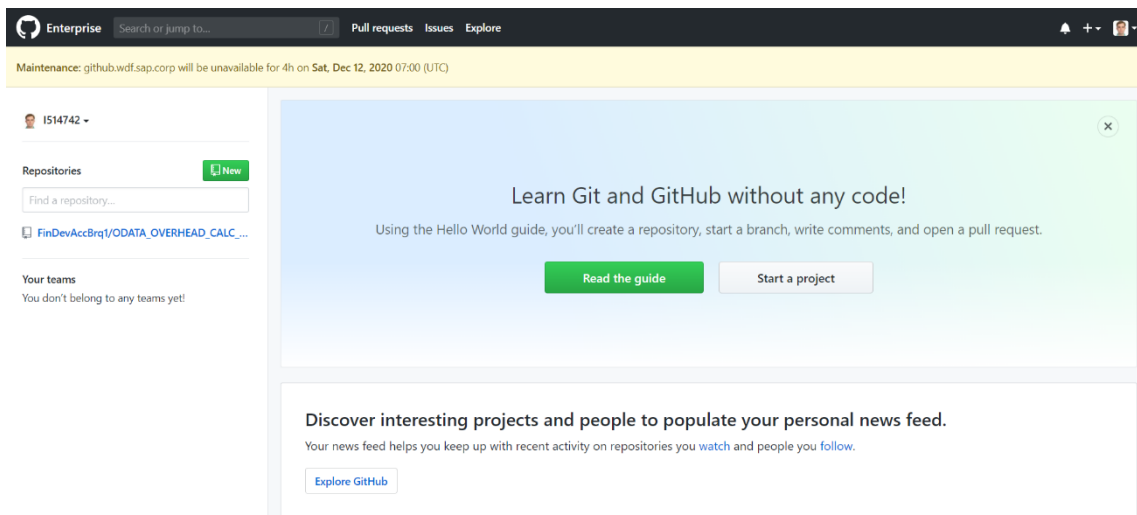
v tom, že ADT integruje aj potrebné nástroje na debuggovanie SAP programov a hlavne funkcionality umožňujúcu pripojiť sa na SAP systémy. Okrem toho to ponúka aj mnoho iných výhod, ale v princípe sa dá povedať, že ADT dokáže všetko to čo klasické SAP GUI, avšak oveľa prehľadnejším a jednotnejším spôsobom. Okrem toho tu nájdeme aj funkcionality dopĺňovania kódu a dynamickej analýzy rozhraní objektov, ktorým majú byť predané parametre, vrátane štruktúry dát s ktorými pracujeme pri písaní kódu.



Obrázok 51 – SAP ADT

(Zdroj: vlastné spracovanie)

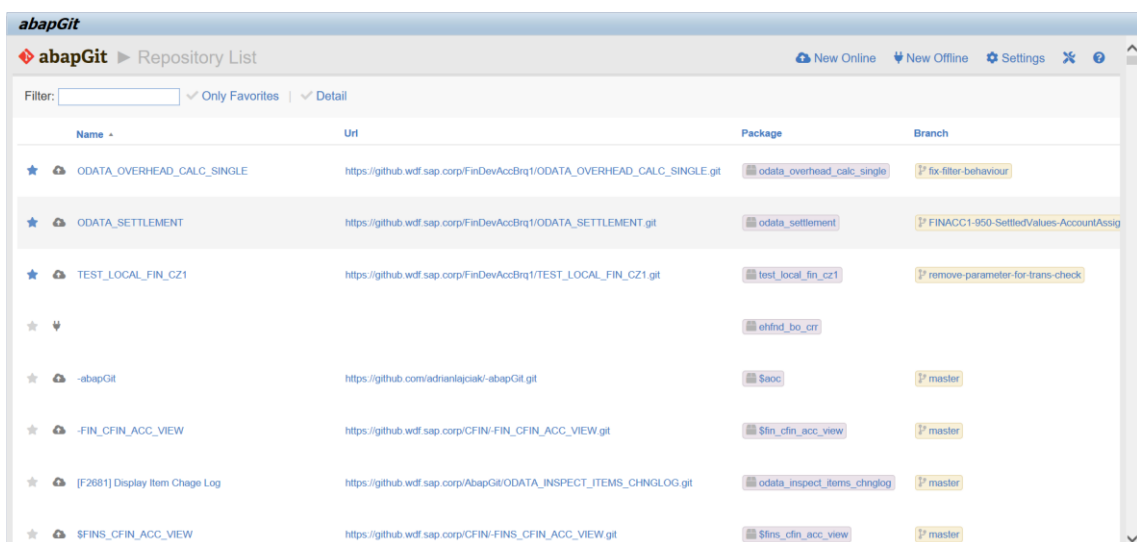
Ako systém kontroly nad pracovnou činnosťou vývojárov sa používa vlastný variant GitHubu.



Obrázok 52 – Intranetový GitHub

(Zdroj: vlastné spracovanie)

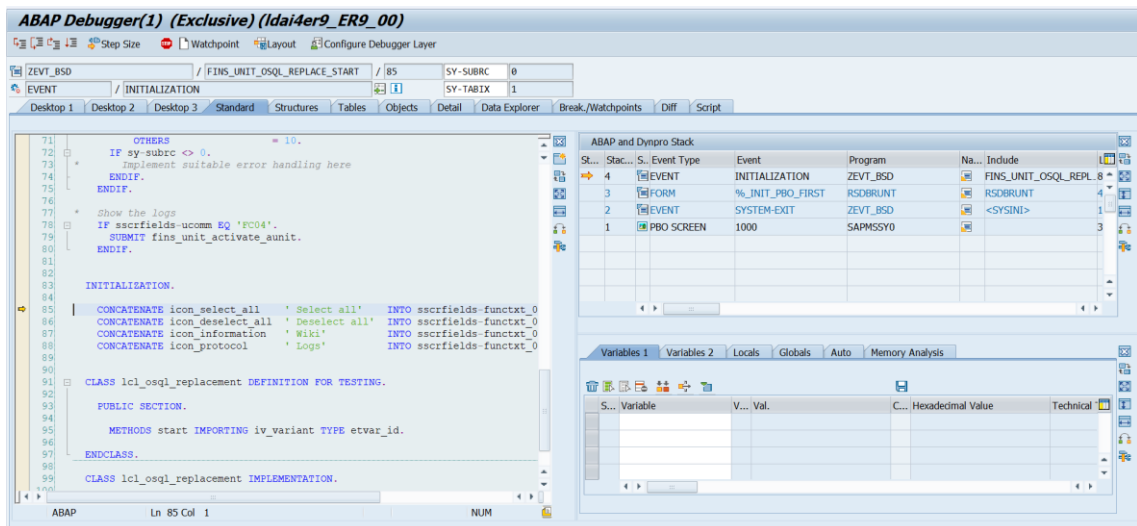
Do GitHubu sa robia príspevky cez transakciu ZABAPGIT, alebo cez ADT.



Obrázok 53 – Transakcia ZABAPGIT

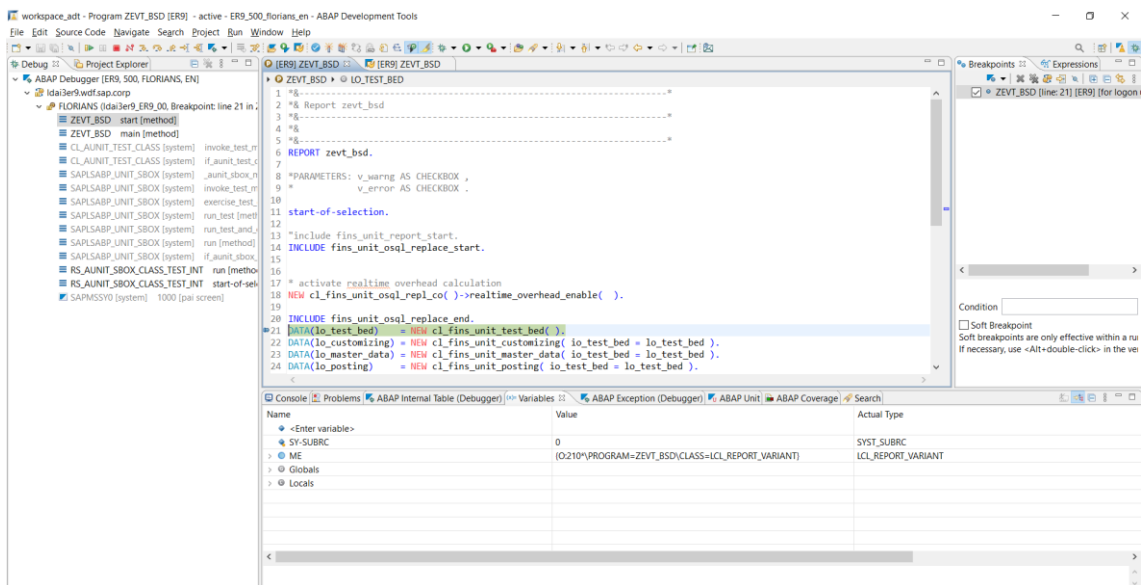
(Zdroj: vlastné spracovanie)

Tá najdôležitejšia časť je však debugger, na odstraňovanie chýb a ladenie programov. Debugger spoločnosti SAP je možné použiť opäť dvoma spôsobmi. Prvým je klasické rozhranie, spúšťané cez SAP GUI a druhým je použitie ADT.



Obrázok 54 – ABAP debugger – klasické rozhranie

(Zdroj: vlastné spracovanie)



Obrázok 55 – ABAP debugger – použité ADT

(Zdroj: vlastné spracovanie)

Posledným nástrojom je JIRA, kde sa realizuje a zaznamenáva pokrok teamu na projektoch a kde sa zaznamenávajú a rozdeľujú úlohy pre vývojárov a všetkých ostatných členov teamu.

2.5.2 SÚHRN

Ako poslednú časť analýzy sme predstavili nástroje, ktoré sú nám pri implementácii k dispozícii. Predstavili sme staršiu transakciu a novšie vývojové rozhranie, ktoré môžeme použiť. My používame prostredie ADT, keďže toto je koncipované na používanie novo zavedených technologických objektov realizujúcich rôzne úrovne abstrakcie ako súčasť nových frameworkov pre vývoj v rámci cloudových technológií a všeobecne S4/HANA.

2.6 ZÁVER ANALÝZY

V závere konštatujeme, že sme analyzovali prostredie ako aj podmienky projektu modernizácie aplikácie na kalkuláciu režijných nákladov. Svoju analýzu sme začali popisom predmetnej spoločnosti a predstavili sme jej informačnú stratégiu, pričom svoje tvrdenia a konštatovania sme podkladali viacerými grafmi a štatistikami zo zdrojov s dobrou kredibilitou. Následne sme prešli k ozrejmeniu fungovania technologickej infraštruktúry, s ktorou sa v prípade produktov a systémov od spoločnosti SAP potýkame. Išlo pri tom o zjednodušený pohľad na to, čo je v skutočnosti komplexný technologický systém s vysokou spoľahlivosťou.

Ďalej sme pokračovali analýzou požiadaviek na projekt, ktoré sme v dotyčnej časti rozobrali a predstavili sme úvahu o tom, na čo je potrebné pri implementácii dbať a aké sú v skutočnosti očakávania kladené na tento projekt.

Najobsiahlejšia časť bola venovaná predstavovaniu aplikácie, ktorá reprezentuje súčasný stav. Prebrali sme jej funkcionality, z ktorej niektoré časti majú byť zachované a iné majú byť zase ponechané minulosti. Čo určite zachované nebude je vzhľad aplikácie, ktorý bude predstavovať asi najradikálnejšiu zmenu. Táto zmena znamená prechod od starého SAP GUI rozhrania na nové FIORI. Výpočtového jadra sa v tomto projekte dotýkať nebudeme.

Záverom analýzy sme predstavili nástroje, ktoré sú k realizácii našich úloh pri implementácii k dispozícii, pričom sme konštatovali, že použijeme modernejšie vývojové prostredie, nakoľko poskytuje rozšírenú funkcionality lepšie koncipovanú na moderné frameworky a technológie, s ktorými budeme pracovať.

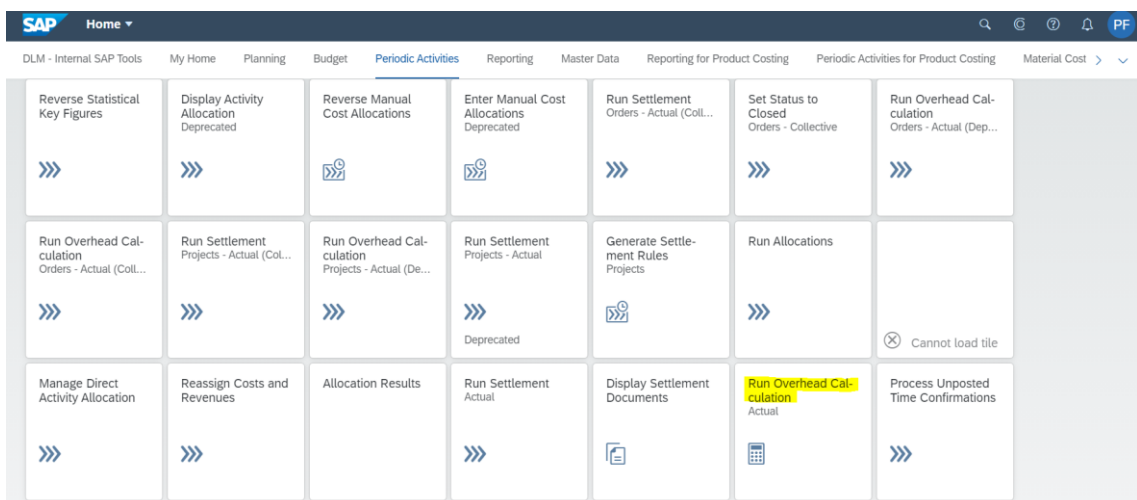
Popri analýze a uvedenej v tejto práci sme realizovali aj hĺbkovú analýzu na pracovisku, pričom táto v sebe obnášala rozsiahle štúdium zdrojových kódov ako aj prípravu testovacích dát a prototypovanie, či drobné experimentovanie s niektorými technológiami a prístupmi, ktoré by sme pri implementácii mohli zvoliť.

3 VLASTNÉ NÁVRHY RIEŠENIA

V tejto kapitole sa budeme venovať implementovanému riešeniu a jeho porovnaniu s aplikáciou, ktorú strieda. Započneme popisom rozhrania, pokračovať budeme ukázkou diagramov, ktoré sa v dôsledku novej implementácie zmenili, alebo boli doplnené.

3.1 VZHLAD A ROZHRANIE NOVÉHO RIEŠENIA

Nový vzhľad zodpovedá rámcu technológii FIORI. Ide o konceptuálne zjednotené užívateľské rozhranie s definovaným správaním pre cloudové produkty spoločnosti SAP. Novú aplikáciu je možné nájsť na uvítacej obrazovke po prihlásení užívateľa s relevantnou autorizáciou. Platí, že pre každú sadu aplikácií, ku ktorej užívateľ potrebuje pristupovať, je potrebná autorizácia, ktorá sa realizuje pomocou tzv. autorizačnej role. To akú rolu je potreba použiť je možné dohľadať v oficiálnej dokumentácii spoločnosti SAP. Role zjednocujú prístupy k aplikáciám podľa biznis procesov. V prípade našej aplikácie to je rola SAP_BR_OVERHEAD_ACCOUNTANT (41).



Obrázok 56 – Dlaždice FIORI na systéme SAP po prihlásení

(Zdroj: vlastné spracovanie)

Po prihlásení na systém za predpokladu pridelenia správnych rolí je užívateľ privítaný dlaždicami FIORI, pričom každá z dlaždíc reprezentuje jednu webovú aplikáciu. Na obrázku je žltou farbou zvýraznená predmetná aplikácia.

Standard* ▾

Search Cost Object Type: WBS Element Project: Company Code: Adapt Filters (1)

WBS Element	WBS Element Name	Project	Company Code
<input type="radio"/> RR011	WBS Element RR011	RR01 (Test ralř)	F001 (SAP A.G. F001)
<input type="radio"/> FINS_UNIT_F002	sFIN Unit Testing - Do not change!!	FINS_UNIT_F002 (sFIN Unit Testing - Do not change!!)	F002 (Tools Inc. America)
<input type="radio"/> FINS_UNIT_F002.1	sFIN Unit Testing - Do not change!!	FINS_UNIT_F002 (sFIN Unit Testing - Do not change!!)	F002 (Tools Inc. America)
<input type="radio"/> JLA-1	JLA-1 - test curr res/val - accr. rev.	JLA (Test EB RevRec)	0001 (SAP DEMO Company 0001)
<input type="radio"/> AK011	WBS Element RR011	AK01 (Test)	F001 (SAP A.G. F001)
<input type="radio"/> JLA-2	JLA-2 - test curr res/val - loss	JLA (Test EB RevRec)	0001 (SAP DEMO Company 0001)
<input type="radio"/> AK021	WBS Element RR011	AK02 (Test ralř)	F001 (SAP A.G. F001)
<input type="radio"/> JLA-3	JLA-3 - test curr res/val - def. cogs	JLA (Test EB RevRec)	0001 (SAP DEMO Company 0001)
<input type="radio"/> JLA-4	JLA-4 - test curr res/val - accr. rev.	JLA (Test EB RevRec)	0001 (SAP DEMO Company 0001)
<input type="radio"/> STAT1		STAT (Statistical Project)	0001 (SAP DEMO Company 0001)

Obrázok 57 – Záznamy po otvorení aplikácie

(Zdroj: vlastné spracovanie)

Po kliknutí na dlaždicu sa otvorí aplikácia, a po otvorení je pre užívateľov pripravený základný filter, ktorý možno vidieť v hornej časti a pod filtrom sa nachádza listový zoznam objektov, ktoré môže užívateľ po jednom vybrať a tlačidlami, ktoré sú nad zoznamom (na obrázku vyššie šedé) môže spúšťať rozličné činnosti. Záznamy v zobrazenej tabuľke obsahujú v niektorých stĺpcoch vyskakovacie okno s dodatočnými informáciami, odkazmi na ďalšie aplikácie, alebo je to samotný hypertextový odkaz bez informačnej karty, ktorý môže užívateľa previesť do ďalších aplikácií. Typicky ide o navigáciu do programov, ktoré pracujú s metadátami, alebo všeobecnou konfiguráciou.

WBS Element	WBS Element Name	Project	Run	Reverse
<input type="radio"/> RR011	WBS Element RR011	RR01 (Test ralř)	<input type="button" value="Run"/>	<input type="button" value="Reverse"/>
<input type="radio"/> FINS_UNIT_F002	sFIN Unit Testing - Do not change!!	FINS_UNIT_F002 (sFIN Unit Testing - Do not change!!)		
<input type="radio"/> FINS_UNIT_F002.1	sFIN Unit Testing - Do not change!!	FINS_UNIT_F002 (sFIN Unit Testing - Do not change!!)		

RR01
Test ralř

[Display Line Items - Cost Accounting](#)

[Projects - Actuals](#)

[More Links](#)

Obrázok 58 – Vyskakovacia informačná bublina s odkazmi do ďalších aplikácií a dodatočnými informáciami

(Zdroj: vlastné spracovanie)

Spomínaný filter aplikácie sa dá skryť aj odkryť tlačidlom v podobe šípky a výsledok je zachytený na obrázku nižšie:

Standard*
Filtered By (1): Cost Object Type

WBS Elements (115.597)

WBS Element	WBS Element Name	Project	Company Code
RR011	WBS Element RR011	RR01 (Test ral)	F001 (SAP A.G. F001)
FINS_UNIT_F002	sFIN Unit Testing - Do not change!!	FINS_UNIT_F002 (sFIN Unit Testing - Do not change!!)	F002 (Tools Inc. America)
FINS_UNIT_F002.1	sFIN Unit Testing - Do not change!!	FINS_UNIT_F002 (sFIN Unit Testing - Do not change!!)	F002 (Tools Inc. America)
JLA-1	JLA-1 - test curr res/val - accr. rev.	JLA (Test EB RevRec)	0001 (SAP DEMO Company 0001)

Obrázok 59 – Aplikácia po skrytí filtra

(Zdroj: vlastné spracovanie)

Na obrázku vyššie už možno vidieť otočený smer spomínanej šípky a stav aplikácie po skrytí filtrovacej hlavičky. Vo filtrovacej hlavičke je okrem jednotlivých identifikátorov rozličných nákladových objektov možné zvoliť napríklad aj všeobecné filtrovanie podľa typu objektu.

Standard*
Filtered By (1): Cost Object Type

Search

Cost Object Type: WBS Element

WBS Element: Project: Company Code: Go Adapt Filters (1)

WBS Elements (115.597)

WBS Element	WBS Element Name	Project	Company Code
RR011	WBS Element RR011	RR01 (Test ral)	F001 (SAP A.G. F001)
FINS_UNIT_F002	sFIN Unit Testing - Do not change!!	FINS_UNIT_F002 (sFIN Unit Testing - Do not change!!)	F002 (Tools Inc. America)
FINS_UNIT_F002.1	sFIN Unit Testing - Do not change!!	FINS_UNIT_F002 (sFIN Unit Testing - Do not change!!)	F002 (Tools Inc. America)

Obrázok 60 – Voľba objektov podľa typu

(Zdroj: vlastné spracovanie)

Standard*

Search

Cost Object Type: Order

Order: Description: Company Code: Go Adapt Filters (1)

Orders (66.669)

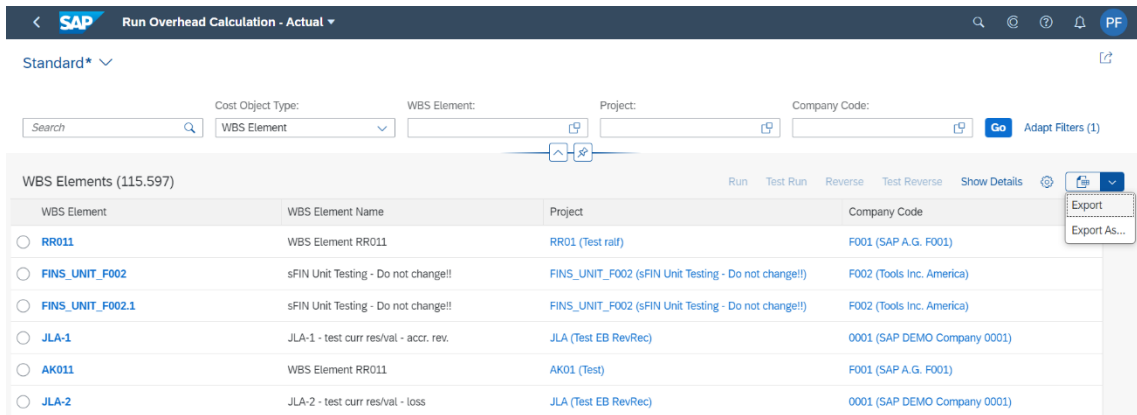
Order	Description	Company Code
S1	for KOKRS CZ01 - FIORI	CZ01 (Country Template CZ)
SCR11	cProjects	0001 (SAP DEMO Company 0001)
SCR11	cProjects	0001 (SAP DEMO Company 0001)
SVM01	test VM	
12564	FINS Unit Testing	0001 (SAP DEMO Company 0001)
100000	Wilson Test	0001 (SAP DEMO Company 0001)
100020	Sprint2 Test	0001 (SAP DEMO Company 0001)

Obrázok 61 – Nastavenia filtra na nákladový typ objednávka

(Zdroj: vlastné spracovanie)

Po voľbe odlišného typu môžeme pozorovať patričné zmeny v zobrazených údajoch.

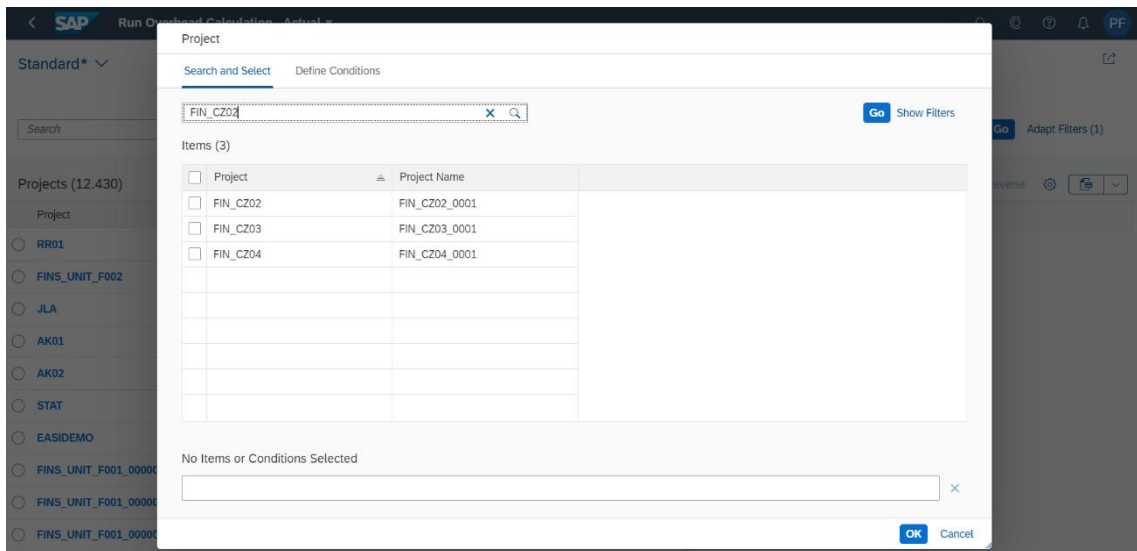
Medzi sekundárne možnosti takejto aplikácie patrí napríklad aj možnosť exportu získaného zoznamu dát do rozličných formátov, klasicky napríklad pre kompatibilitu s aplikáciou MS Excel.



Obrázok 62 – Možnosti exportu vyfiltrovaných hodnôt

(Zdroj: vlastné spracovanie)

Po pomoc s nastavovaním hodnôt filtra slúži špeciálne vyskakovacie okno, ktoré reprezentuje modernejšiu obdobu, toho, ktoré bolo spomenuté v kapitole 2, a ktoré slúži pre ponúkajú rôznych možností hodnôt, ktoré by potenciálne mohli do daného filtrovacieho poľa patriť.

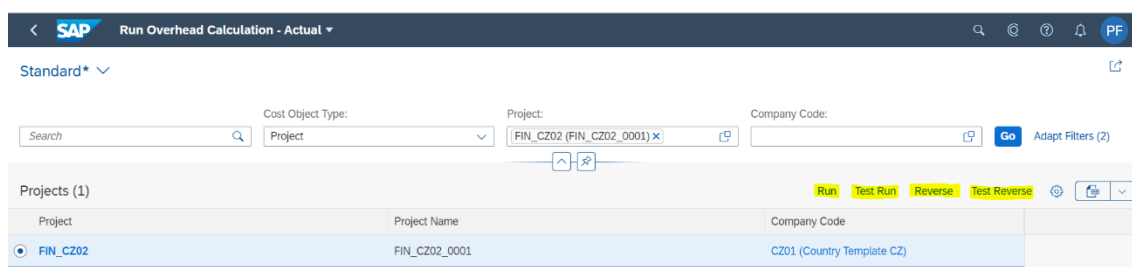


Obrázok 63 – Pomocná obrazovka pre výber hodnôt

(Zdroj: vlastné spracovanie)

V prípade ako na obrázku vyššie si môžeme všimnúť, ďalšie vyhľadávacie pole(ktoré je tiež prítomné v hlavičke filtra), využívajúce fuzzy logiku pre zobrazovanie potenciálnych výsledkov, ktoré by užívateľa mohli zaujímať.

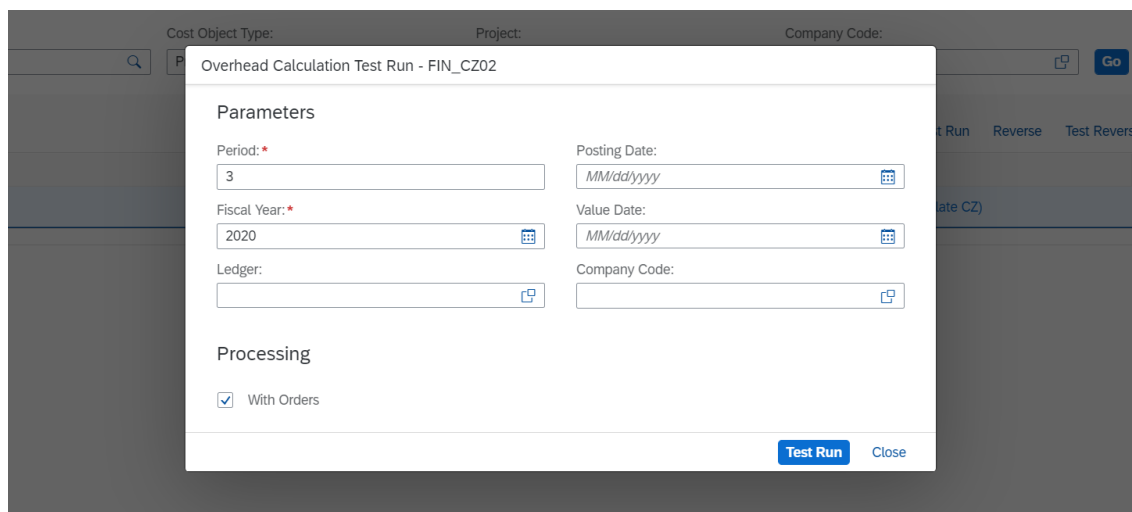
Po nastavení filtra, je čas zvoliť si objekt, s ktorým bude užívateľ pracovať. Hneď ako tak užívateľ učiní, sú mu sprístupnené hlavné funkcie tejto aplikácie, ktorými sú: výpočet réžie pre zvolený záznam, testovací výpočet réžie, reverz vypočítaných réžií, testovací reverz vypočítaných réžií. Testovacie režimy sa používajú v prípade, že užívateľ si nie je úplne istý, že výsledky dopadnú podľa očakávaní, nakoľko ostré behy môžu v prípade bezchybného behu spôsobiť zápis do databázy.



Obrázok 64 – Tlačidlá pre spustenie hlavných funkcií

(Zdroj: vlastná réžia)

Kliknutím na tlačidlo s funkciou sa objaví vyskakovacie okno, do ktorého je možné vyplniť dodatočné detaily pre výpočet. To je v podstate ekvivalent starej obrazovky pred spustením aplikácie, s tým rozdielom, že táto obrazovka a aj celá aplikácia zjednocujú funkcie 2 programov.



Obrázok 65 – Dialógové okno pre zadanie parametrov výpočtu

(Zdroj: vlastné spracovanie)

Po spustení napríklad výpočtu alebo testovacieho výpočtu sa zobrazí nová stránka, ktorá obsahuje vyrátané hodnoty.

Sender	Sender Type	Receiver	Receiver Type	Cost Element	Amount in Global Currency	Amount in Cor
PRODUCTION (Production)	Cost Center	FIN_CZ021 (FIN_CZ02_0002)	WBS element	855100	-30,00 EUR	
PRODUCTION (Production)	Cost Center	FIN_CZ021 (FIN_CZ02_0002)	WBS element	855200	30,00 EUR	
PRODUCTION (Production)	Cost Center	FIN_CZ0211 (FIN_CZ02_0002_01)	WBS element	855100	-60,00 EUR	
PRODUCTION (Production)	Cost Center	FIN_CZ0211 (FIN_CZ02_0002_01)	WBS element	855200	60,00 EUR	
PRODUCTION (Production)	Cost Center	FIN_CZ02111 (FIN_CZ02_0002_01_01)	WBS element	855100	-274,00 EUR	
PRODUCTION (Production)	Cost Center	FIN_CZ022 (FIN_CZ02_0003)	WBS element	855100	-38,00 EUR	
PRODUCTION (Production)	Cost Center	FIN_CZ022 (FIN_CZ02_0003)	WBS element	855200	38,00 EUR	

Obrázok 66 – Obrazovka s výsledkami

(Zdroj: vlastné spracovanie)

Výsledky obsahujú rôzne údaje o priebehu kalkulácie, no dôležité sú jednotlivé vyrátané množstvá. Aj v tomto prípade ide o obdobu obrazovky s výsledkami, ktorá bola popísaná v kapitole 2 venujúcej sa analýze zastaranej aplikácie.

V hornej časti výsledkov užívateľ nájde informáciu o priebehu operácie a štatistiky spracovania vybraných objektov. Tu treba pripomenúť, že i keď užívateľ zvolil 1 nákladový objekt, napríklad projekt, tak tento pod sebou môže obsahovať niekoľko ďalších objektov v hierarchii (objednávky, WBS elementy) – toto správanie sa dá zapnúť na už spomenutom dialógovom okne.

000000200080
Overhead Calculation Test Run

Period: 3 Result:
Fiscal Year: 2021 **Calculated Successfully**
Company Code: 1010

⏪

Results (0) Costing Sheet Messages (0)

Statistics

Overhead Calculated: 1
Not Relevant: 0
Inappropriate Status: 0
Error: 0

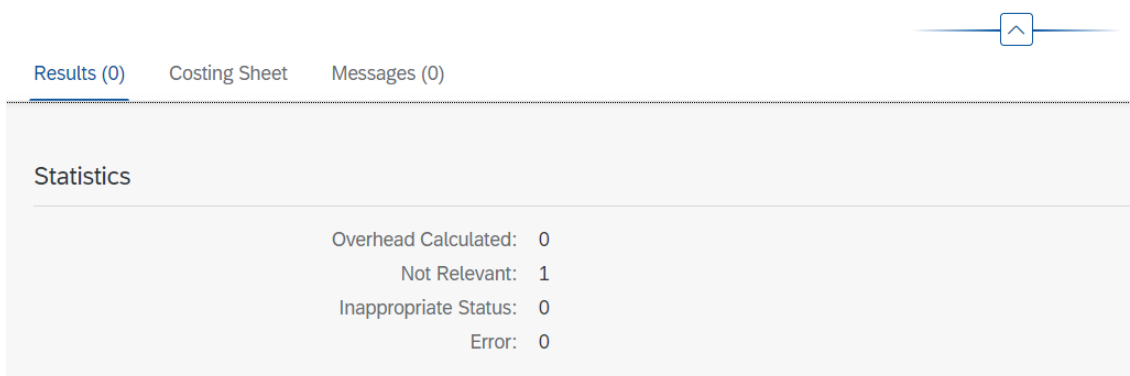
Obrázok 67 – Informácie a štatistiky v prípade úspešného výpočtu

(Zdroj: vlastné spracovanie)

00000200080
Overhead Calculation Test Run

Period: 3
Fiscal Year: 2021
Company Code: 1010

Result:
Nothing Calculated



The screenshot shows the SAP interface for an Overhead Calculation Test Run. At the top, the ID '00000200080' and the title 'Overhead Calculation Test Run' are displayed. Below this, the period is set to 3, the fiscal year to 2021, and the company code to 1010. The result is 'Nothing Calculated'. A navigation bar includes 'Results (0)', 'Costing Sheet', and 'Messages (0)'. A 'Statistics' section is visible, containing the following data:

Overhead Calculated:	0
Not Relevant:	1
Inappropriate Status:	0
Error:	0

Obrázok 68 – Chyba kalkulácie s miernou závažnosťou

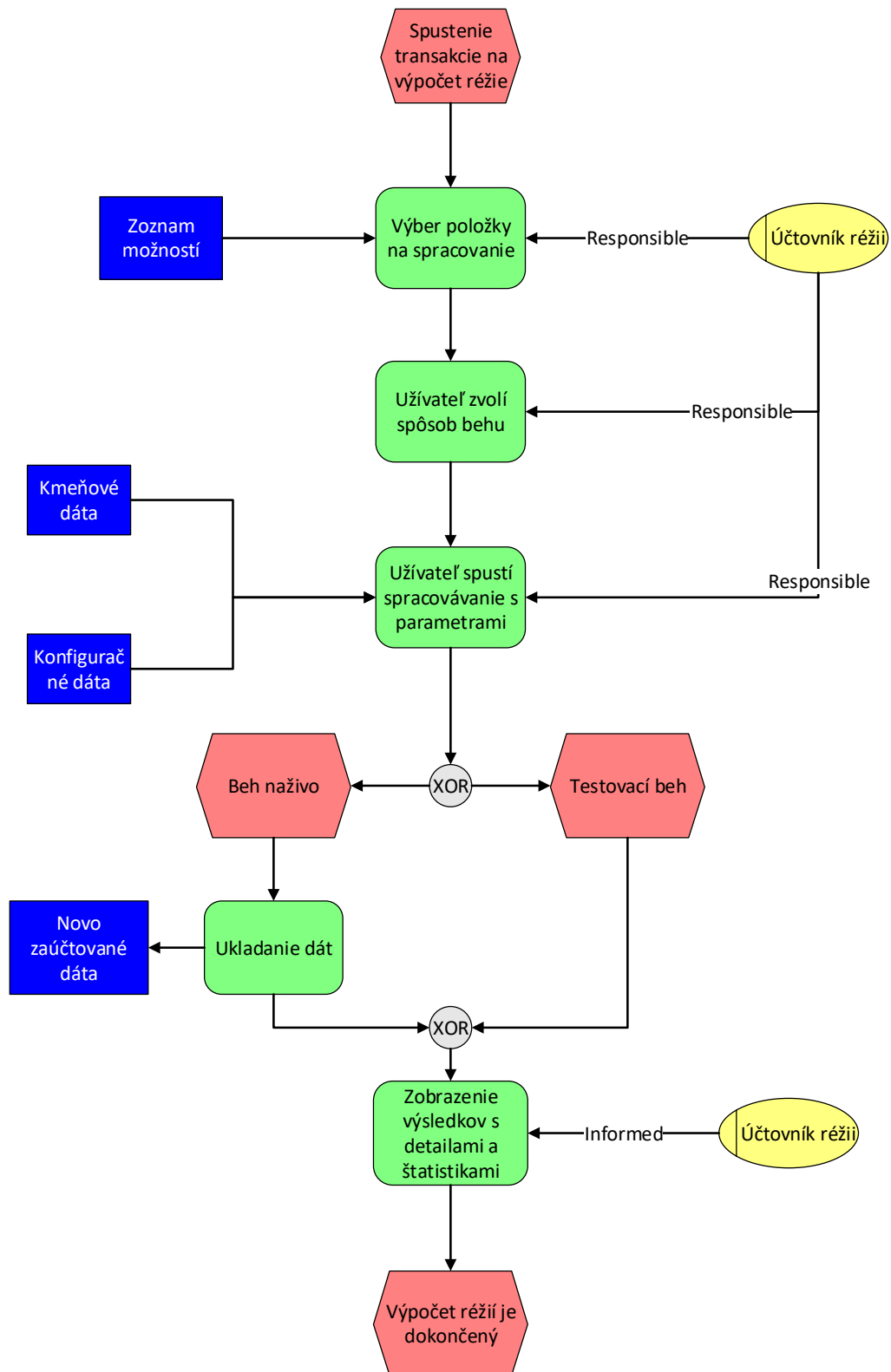
(Zdroj: vlastné spracovanie)

Na obrázkoch vyššie vidíme situácie keď sa vyskytne chyba pri spracovávaní a keď kalkulácia prebehne v poriadku. V prípade, že by počas kalkulácie nastali závažnejšie chyby vďaka ktorým by kalkulácia nemohla prebehnúť, na karte „Messages“ by boli viditeľné chyby s niektorými ďalšími detailmi. Okrem toho tu nájdeme aj kartu s informáciami o konfiguračnom hárku a v prípade živého behu aj kartu so záznamami vytvorenými v databáze.

3.2 PROGRAMOVÁ ČASŤ NOVÉHO RIEŠENIA

V tejto časti sa bližšie pozrieme na zmeny, ktoré nová implementácia priniesla s využitím ilustrácii diagramov podobne ako v kapitole 2. Pre objasnenie zmien poskytneme komentár, ktorý poskytne širší situačný kontext.

3.2.1 DIAGRAM BIZNIS PROCESU NOVEJ APLIKÁCIE



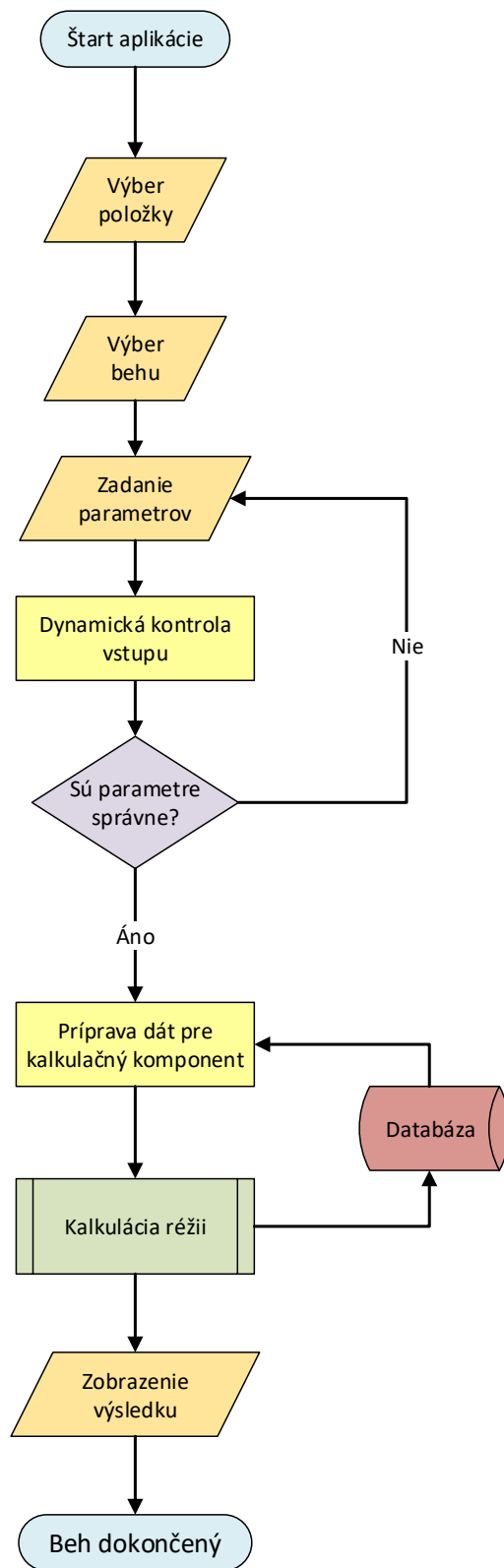
Obrázok 69 – EPC diagram nového biznis procesu

(Zdroj: vlastné spracovanie)

Keď porovnáme predošlý obrázok s tým z kapitoly 2. uvidíme, že nový proces je o čosi kratší. Konkrétne ide o časť keď sa realizovalo dodatočné zobrazovanie a užívateľ musel klikat' na rôzne vyskakujúce a blokujúce okná, aby proces mohol ďalej prebehnúť. Pôvodná funkcionálna behu naživo a testovacieho behu však ostala zachovaná, rola administrátora tu nie je ďalej potrebná a interakcia užívateľa pri zadávaní vstupov je zredukovaná na úplné minimum, konkrétne 3 kroky (úplné minimum sú 3 kliknutia a to je podmienka diktovaná produktovým štandardom), čo značí snahu zjednodušiť celú aplikáciu a jej ovládanie. Menšia nutnosť interakcie znamená jednoduchšiu automatizáciu v budúcnosti a rýchlejší spôsob dosiahnutia identických záverov ako v prípade zastaranej aplikácie.

3.2.2 ZJEDNODUŠENÝ ALGORITMUS NOVEJ APLIKÁCIE

Zo zrejmých dôvodov obchodného tajomstva tu neuvedieme celý komplexný algoritmus, ale na nasledujúcom obrázku zachytíme zjednodušenú verziu fungovania tejto novej webovej aplikácie.

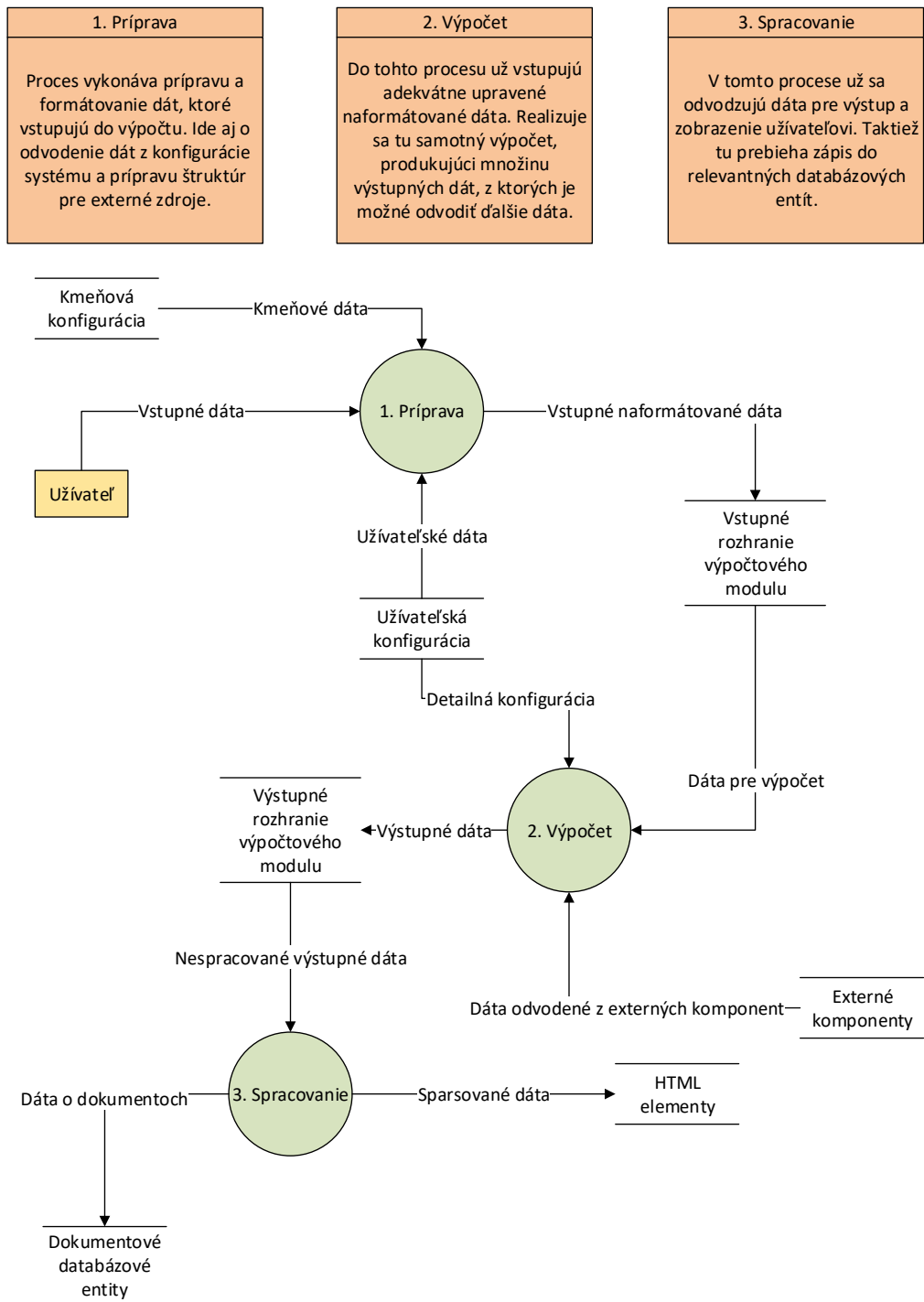


Obrázok 70 – Zjednodušený pohľad na beh webovej aplikácie

(Zdroj: vlastné spracovanie)

V úvode algoritmu si môžeme všimnúť vstupy užívateľa 3 krát po sebe, čo aj zodpovedá situácii z predošlej podkapitoly a na konci je situácia, keď sú výsledky zobrazené užívateľovi. Je potreba poznamenať, že odvodzovanie a kontrola parametrov, ktorá prebiehala v rozsiahlej miere v prípade zastaranej aplikácie je úplne odstránená a je nahradená veľmi rýchlym skrátaným mapovaním v middleware časti aplikácie. Samotná kalkulácia réžii sa v ničom nemení, je identická so spôsobom popísaným v kapitole 2., čo značí dodržanie podmienky o ponechaní výpočtového jadra.

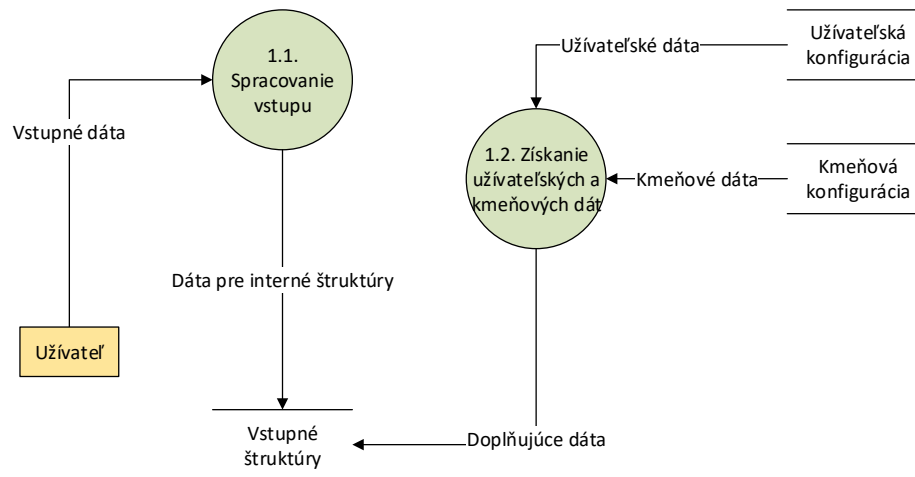
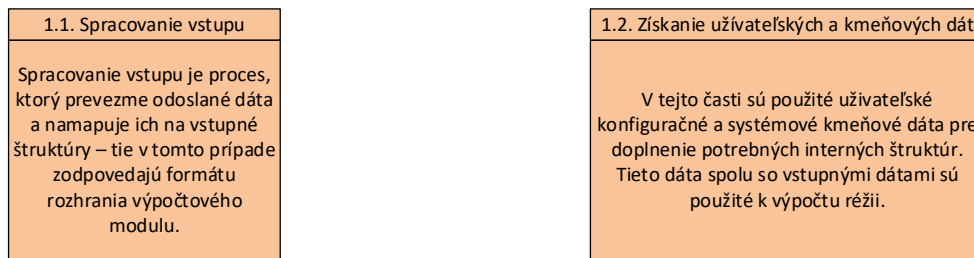
3.2.3 DIAGRAMY TOKU DÁT RIEŠENIA



Obrázok 71 – DFD diagram novej aplikácie - úroveň 0

(Zdroj: vlastné spracovanie)

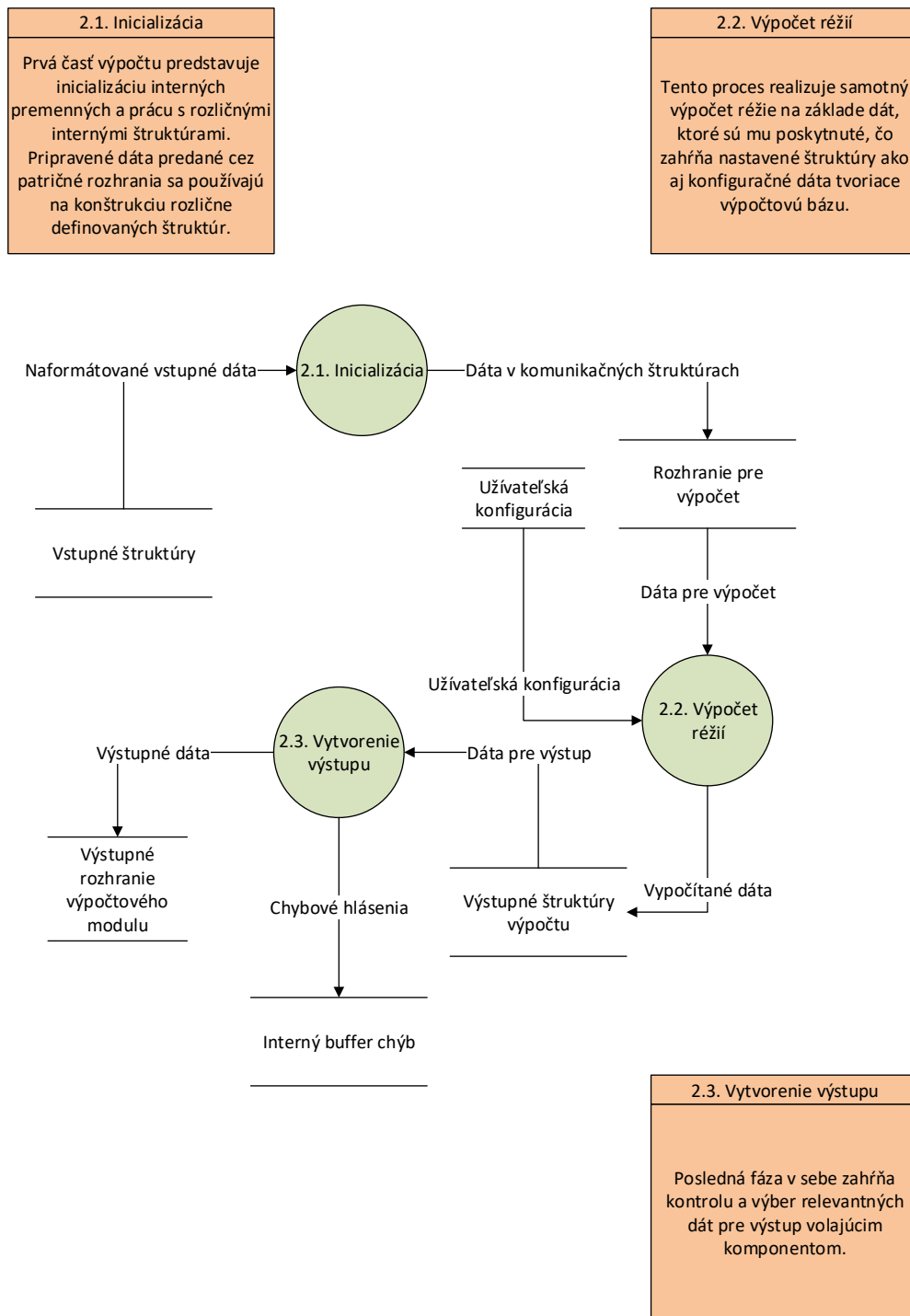
Ako možno vidieť vyššie, nultá úroveň toku dát je takmer identická s tou, ktorú možno vidieť v kapitole 2, dôvodom je, že táto úroveň nie je dostatočne detailná.



Obrázok 72 – DFD diagram – 1. úroveň, príprava

(Zdroj: vlastné spracovanie)

Prvá úroveň popisujúca prípravu dát je už citeľne rozdielna oproti tej z 2. kapitoly. V tomto prípade ide o mohutné zjednodušenie celého procesu. Nepotrebné dátové transformácie sú úplne odstránené, potrebné časti sú implementované efektívnejšie, akákoľvek logika, ktorá bola predtým pochybným spôsobom robená priamo v užívateľskom rozhraní je odstránená, alebo presunutá na vhodné miesto. Kód ktorý tento proces zaisťuje je úplne nový, písaný objektovo a má viac než 95% pokrytie jednotkovými testami. Proces generuje vstupné dátové štruktúry pre komponentu(modul) výpočtu réžii.

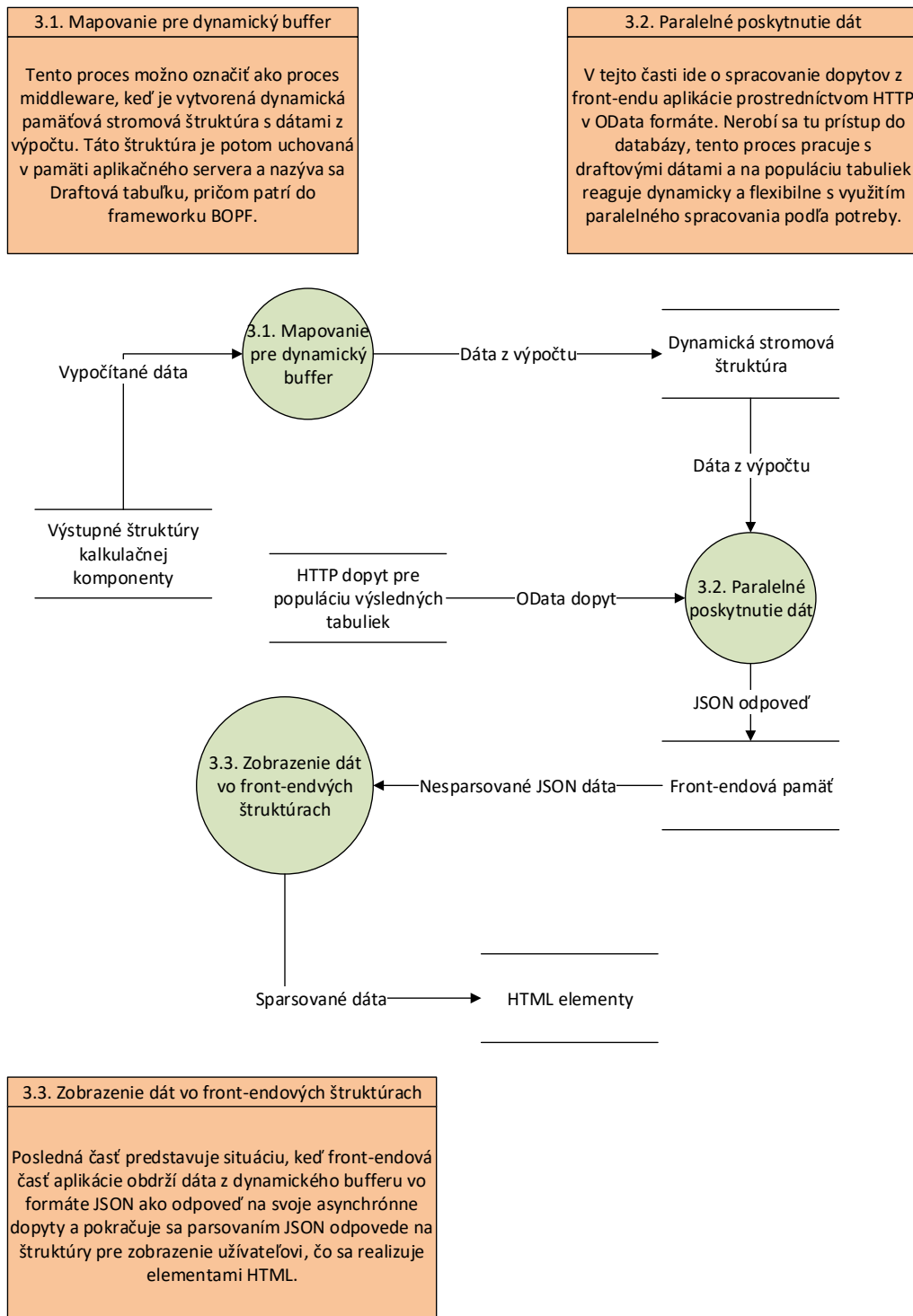


Obrázok 73 – DFD diagram – 1. úroveň, výpočet

(Zdroj: vlastné spracovanie)

Prúdenie dát v prípade výpočtu je opäť takmer identické, keďže pôvodné výpočtové jadro je nedotknuté. Výstup a jeho formát je však mierne odlišný, keďže nový program, ktorý je okolo pôvodného výpočtového jadra je objektovo orientovaný a využíva napríklad aj

vlastný chybový buffer(ide o dynamicky alokovanú pamäť). Kvôli lepšej spätnej kompatibilite sme ponechali aj pôvodný formát komponenty pre výpočet.



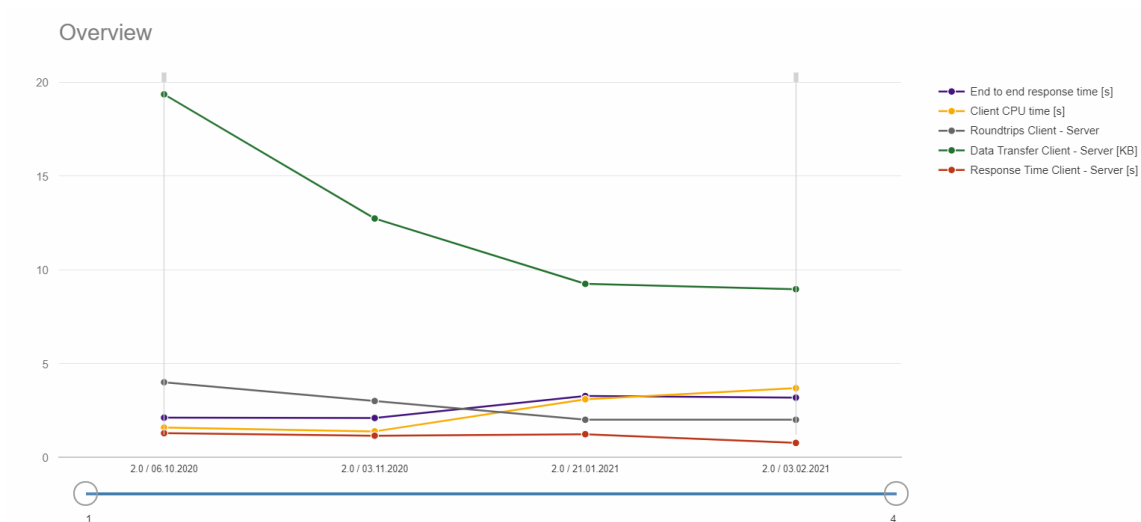
Obrázok 74 – DFD diagram – 1. úroveň, spracovanie

(Zdroj: vlastné spracovanie)

V prípade posledného procesu spracovania vidíme zdanlivo väčšiu komplexitu, ale to je dané povahou webovej aplikácie, ktorá má svoje vlastné špecifiká oproti klasickej GUI aplikácii. V prípade webovej aplikácie sa pracuje s JavaScriptom a dátovým protokolom JSON, ktorého odozva sa získava pomocou HTTP vo formáte OData. Jednoducho povedané, architektonický koncept, ktorý sme aplikovali spočíva v pred-výpočte dát a ich uložení do dynamickej stromovej pamäťovej štruktúry, ktorá existuje v pamäti aplikačného servera. Túto štruktúru je možné dynamicky vyžiadať, server umožňuje paralelné dopyty a odozvy. Okrem toho je tento typ bufferu možné upraviť na front-ende bez nutnosti databázového zápisu. Je možné robiť ďalšie úkony s takto ukladanými dátami, pohybovať sa medzi výsledkami, analyzovať ich, súbežne spúšťať a porovnávať výsledky iných aplikácií a zmeny stále ostanú v medzi-pamäti, tohto takzvaného draftu. Pokiaľ išlo o testovací beh a užívateľ je s výsledkami výpočtu spokojný, môže spustiť ostrý beh, alebo sa vrátiť a výsledky v medzi-pamäti zahodiť. V prípade že išlo o ostrý beh, je možné urobiť to isté ako reverznú akciu, ktorá môže opäť byť testovací beh, alebo ostrý beh.

3.2.4 MERANIE VÝKONU RIEŠENIA

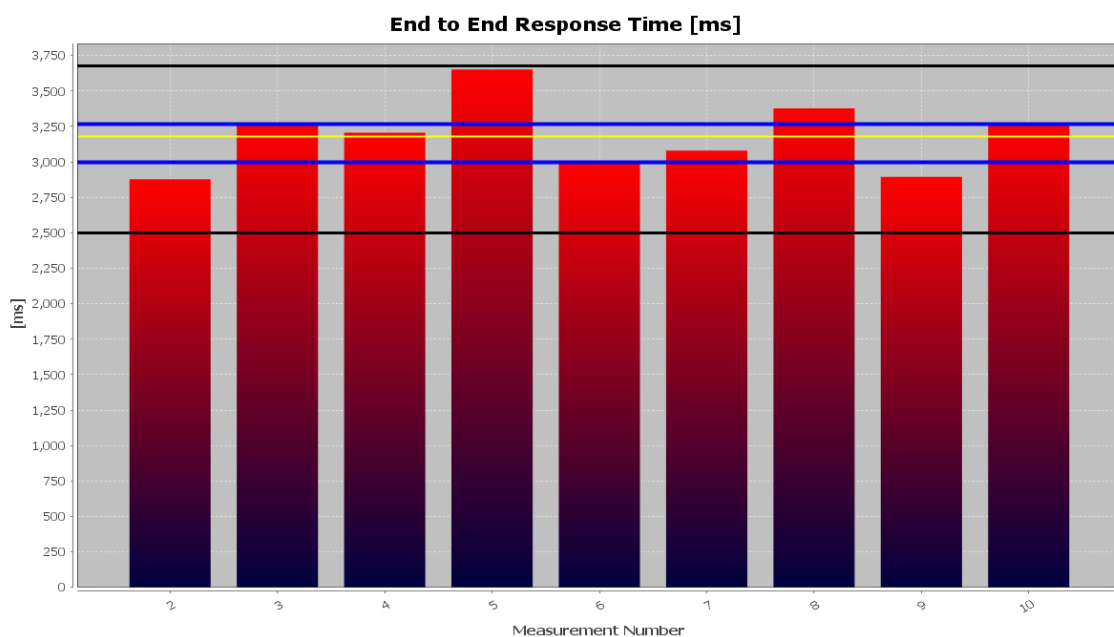
Na meranie sme použili komplexné nástroje, ktoré vytvorili testerí spoločnosti SAP. Ide hlavne o nástroj SUPA – „Single user Performance Analysis“ a IPA – „Interactive Performance Analysis“. Tieto nástroje využívajú sadu knižníc a programov, či prostriedkov na testovanie a meranie aplikácii ako Krypton, Selénium, ST12, ST05 a pod. V priebehu niekoľkých mesiacov prebiehali opakované testovania a na základe nich sme zhromaždili potrebné údaje.



Graf 8 – Merania výkonu za 4 mesiace

(Zdroj: vlastné spracovanie)

Na obrázku vyššie je zachytený graf vývoja výkonu pri implementácii. Najvyššie položená krivka zelenej farby predstavuje prenos dát medzi klientskou stanicou a serverom. Dáta boli prenášané po firemnej sieti bez využitia VPN. Ďalší dôležitý údaj je čas odozvy od začiatku do konca, tzv. „End to End“ beh.



Graf 9 – Zmeny a oscilácia času odozvy v „End to End“ meraní

(Zdroj: vlastné spracovanie)

Stĺpcový graf vyššie zachytáva vývoj času odozvy, čo je kľúčový ukazovateľ reakcie programu pri 10 meraniach. Hodnota oscilovala okolo 3 sekúnd, čo spĺňa minimálne

požiadavky na produkt v spoločnosti SAP, ale v súčasnej dobe sa plánuje ďalšia optimalizácia a v blízkej budúcnosti by tieto hodnoty mohli klesnúť niekde na úroveň 2 sekúnd.

Overview	Overview	2.0	2.0	2.0	2.0
		06.10.2020	03.11.2020	21.01.2021	03.02.2021
KPIs in Chart					
##End to end response time [s]		2.111	2.094	3.265	3.180
##Client CPU time [s]		1.582	1.381	3.094	3.687
##Roundtrips Client - Server		4.000	3.000	2.000	2.000
##Data Transfer Client - Server [KB]		19.353	12.730	9.246	8.962
##Response Time Client - Server [s]		1.285	1.151	1.230	0.766

Obrázok 75 – Prehľad kľúčových indikátorov výkonu

(Zdroj: vlastné spracovanie)

Na predošlom obrázku vidíme opäť niektoré z najdôležitejších metrick ako čas trvania na procesore klienta, celkový počet obehov(závisí od komplexnosti testu, ale je veľmi dôležité aby front-endové kódovanie zbytočne neprevolávalo back-end) a odozvu serveru medzi inými.

Tabuľka 3 – Tabuľka dôležitých metrick riešenia

(Zdroj: vlastné spracovanie)

Krok	End to End Odozva [s]	Čas na UI [s]	Sieťový čas [s]	Vykonáva nie služby a aplikácie [s]	Databázový čas [s]	Čas procesoru klienta [s]	Čas procesoru servera [s]	Čas procesora HANA databázového servera [s]
Navigácia, beh	3.206	2.425	0.119	0.550	0.112	3.577	0.460	0.125

V kroku štandardného použitia aplikácie, čo v sebe zahŕňa navigáciu v aplikácii, výber hodnôt a realizáciu výpočtu sme namerali hodnoty, ktoré vidno v tabuľke vyššie. Čas procesoru klienta je čas ktorý potreboval laptop, na ktorom bolo meranie robené(išlo o klasický laptop Lenovo Thinkpad T580, operačný systém Windows 10, prehliadač Google Chrome). Dôležité je upriamiť pozornosť na databázový čas, ktorý je oveľa menší, ako ten nameraný v kapitole 2, kde to bolo dohromady 5 sekúnd, tu je to menej ako sekunda. Najviac času bolo stráveného na strane front-endu, ktorý je písaný

v JavaScripte (užívateľské rozhranie možno vidieť v podkapitole 3.1). Čas vykonávania samotného jadra aplikácie bez GUI je skoro rovnaký, aj keď o niečo väčší, to je spôsobené webovou službou webového serveru umiestnenou medzi výpočtové jadro a front-end, bohužiaľ to je časť, ktorá sa nedá optimalizovať pri klasickej tvorbe aplikácie, nakoľko tvorí súčasť samotného webového servera. Poznamenajme ešte, že dáta v tejto tabuľke sú zjednodušením výstupu ST12 a ST05 s niektorými doplnkami.

Tabuľka 4 – Meranie výkonu na sieti

(Zdroj: vlastné spracovanie)

Krok	Čas odozvy [s]	Odhadovaný čas odozvy WAN [s]
Navigácia, beh	3.009	3.480

Posledná tabuľka predstavuje rovnaký krok, teda navigáciu a výpočet z hľadiska analýzy sieťovej prevádzky a sieťových metrík, ktoré sú notoricky problematické, keďže sieťové pripojenie sa môže často správať nevyspytateľne a to je dôvod prečo sa k metrike merania WAN siete prirátava 200 milisekúnd na každú obchádzku (teda na každé volanie back-endovej časti). Keď vezmeme do úvahy 4 volania je to 800 milisekúnd.

3.2.5 SÚHRN

V tejto podkapitole sme ukázali implementované riešenie z viacerých hľadísk. Išlo o čisto technické pohľady. Kľúčové prvky sme najprv ukázali modelovaním s pomocou EPC diagramu a poukázali sme na časti, ktoré sa zmenili v porovnaní s tým, čo sme prezentovali kapitole 2.

Podobný postup nasledoval aj pri zjednodušenom algoritme, avšak to najdôležitejšie sme ukázali v prípade diagramu toku dát, kde sme uviedli diagramy, ktoré boli odlišné od situácie v kapitole 2 a venovali sme sa najdôležitejším rozdielom, ktoré nastali a tomu čo to pre riešenie znamená.

Podkapitolu sme zakončili predstavením niekoľkých metrík, resp. kľúčových ukazovateľov výkonnosti, ktoré sme namerali v prípade novej implementácie. Tieto sme porovnali so závermi v kapitole 2 a usúdili, sme že nová aplikácia je po stránke výkonu lepšia a je aj lepšie optimalizovaná. Nedostatky, ktoré sme vo výkone zachytili plynuli najmä z externých knižníc a frameworkov, na ktoré naša práca nemá vplyv, preto sú takmer nemenné. Aplikácia z technického hľadiska spĺňa požiadavky na produktovú

kvalitu stanovené spoločnosťou SAP, no aj napriek tomu už je plánovaná optimalizácia, niektorých častí v ďalšom vývojovom cykle, ale táto bude závisieť od technologického pokroku v dostupných technológiách.

3.3 EKONOMICKÉ ZHODNOTENIE

Vývoj novej aplikácie trval 1 vývojový cyklus, čo je približne 6 vývojových šprintov, to je v prepočte 12 týždňov. Testovanie prebiehalo dohromady 4 týždne podľa potreby. Do vývoja boli zahrnutí 2 vývojári a 1 tester zodpovedný za kvalitu. Do vývoja vstupoval v nepravidelných intervaloch aj architekt riešenia a produktový vlastník SCRUM teamu, ktorí tlmočili požiadavky klientov.

Tabuľka 5 – Prepočet na pracovné hodiny použité na vývoj

(Zdroj: vlastné spracovanie)

Zamestnanec	Počet zamestnancov	Počet týždňov	Počet dní	Počet hodín
Vývojár	2	24	120	960
Tester	1	2	10	80
Architekt	1	2	10	80
Produktový vlastník	1	2	10	80

Potrebné informácie pre kalkuláciu nákladov v podobe miezd je možné vidieť v predošlej tabuľke. Samotné náklady neuvádzame, vyžiadalo by si to informácie o mzdách a tie na pozícii vývojára nemáme, firma ich ani neposkytne, ale s určitosťou môžeme tvrdiť, že kalkulujeme s 8 hodinovou pracovnou dobou a klasickým 5 dňovým pracovným týždňom.

Celkovo by sa dalo tvrdiť, že naše riešenie prináša aj dodatočnú hodnotu po estetickej stránke, po stránke prístupu, teda jednoduchosti použitia a rýchlosti interakcie. Zatiaľ čo technické porovnanie sme vykonali v predošlých kapitolách, interakcia a použiteľnosť patrí na odlišnú úroveň hodnotenia.

Pre získanie orientačných informácií o použiteľnosti aplikácie sme sa spýtali niekoľkých subjektov čo si myslia o dizajne novej aplikácie a ako by dizajn ohodnotili na stupnici od 1 do 10, pričom 1 znamená najhoršie hodnotenie a 10 najlepšie. Šetrenie sme vykonali krátkym dotazníkom zaslaným prostredníctvom elektronickej pošty s odkazom na dotyčnú aplikáciu, potrebnými informáciami a krátkym popisom toho o čo ide a v akej súvislosti. Dotazník obsahoval 3 otázky. Prvou otázkou bolo už spomínané hodnotenie dizajnu, zatiaľ čo druhá otázka sa týkala spôsobu interakcie, išlo o to zistiť či sa interakcia s novým riešením zdá jednoduchšia v porovnaní s predošlým riešením, odpoveď mohla byť áno, alebo nie. Posledná otázka sa venovala tomu, či si respondent myslí, že nové

riešenie predstavuje zlepšenie oproti pôvodnému, pričom odpoveď mohla byť opäť dichotomická – áno, nie.

Tabuľka 6 – Výsledky hodnotenia dizajnu novej aplikácie

(Zdroj: vlastné spracovanie)

Počet respondentov	Hodnotenie
0	0
0	1
0	2
0	3
1	4
2	5
4	6
8	7
5	8
3	9
2	10

Respondenti, ktorí sa zapojili pochádzali z radov našich kolegov v rámci teamu aj pobočky. Celková účasť bola v počte 25 ľudí, pričom sa dá konštatovať, že hodnotenie je skôr pozitívne ako negatívne.

Tabuľka 7 – Hodnotenie otázky či sa interakcia s novým riešením zdá jednoduchšia v porovnaní s tým predošlým

(Zdroj: vlastné spracovanie)

Počet respondentov	Hodnotenie
21	Áno, interakcia je jednoduchšia
4	Nie, interakcia nie je jednoduchšia

Výsledok otázky o zjednodušení interakcie je relatívne jednoznačný, väčšina respondentov sa zhodla na tom, že práca s novým riešením je jednoduchšia.

Tabuľka 8 – Vyhodnotenie otázky, či v prípade nového riešenia ide o zlepšenie oproti pôvodnému riešeniu

(Zdroj: vlastné spracovanie)

Počet respondentov	Hodnotenie
25	Áno, ide o zlepšenie
0	Nie, o zlepšenie nejde

V prípade poslednej otázky, bola odpoveď jednoznačné áno, podľa opýtaných respondentov ide o zlepšenie oproti pôvodnému. Domnievame sa, že tento fakt možno

vysvetliť aj tým, že ako už bolo uvedené v predošlých podkapitolách, sme zjednotili funkcionality 2 nezávislých programov do 1 aplikácie.

Hodnotenie z nášho krátkeho dotazníku možno popísať ako pozitívnu odozvu na dodanú aplikáciu, takže ostáva dúfať, že riešenie bude zaujímavé aj pre existujúcich aj potenciálnych zákazníkov a že budú mať obdobný názor ako bol prezentovaný v prípade našich kolegov.

Tabuľka 9 – Počet tiketov za obdobie od septembra 2020 po marec 2021

(Zdroj: vlastné spracovanie)

Mesiac	Počet tiketov	Počet vývojárov	Doba trvania riešenia (h)
September	7	2	22.5
Október	3	1	6
November	0	1	0
December	0	1	0
Január	2	1	5
Február	0	1	0
Marec	0	1	0

Z predošlej tabuľky je možné sa dozvedieť o množstve tiketov za obdobie nasadenia nového riešenia od septembra minulého roka po marec súčasného roka. Tikety predstavujú sťažnosti, problémy, alebo požiadavky na vylepšenia, alebo zmeny. Môže ísť o potrebu opraviť závažnejšiu chybu, alebo len zmeniť nadpis v aplikácii. Neprekvapivo sa počet tiketov postupne znižoval od septembra. Takýto vývoj je pomerne typický pri dobre urobenej implementácii. V januári možno pozorovať ďalšie tikety, ktoré sa objavili v súvislosti s ďalším vývojovým cyklom, pretože niekedy sa stane, že zmena zdanlivo nesúvislej komponenty si vyžiada zmeny iných komponent. Taktiež by sme uviedli do pozornosti, že s počiatku boli na riešenie problémov v súvislosti s touto aplikáciou pridelený 2 vývojári, neskôr bolo rozhodnuté, že bude postačovať ak zodpovednosť poniesie len jeden vývojár.

Náklady na údržbu je teoreticky možné vyčíslieť na základe dôb trvania riešení na vzniknuté tikety. Ak by tabuľka 9 mala udávať trend blízkej budúcnosti, dalo by sa diskutovať o tom, že výskyt možných problémov v budúcnosti bude minimálny, pokiaľ sa nezačnú realizovať ďalšie vylepšenia, priamo sa týkajúce tejto aplikácie, v opačnom prípade je treba rátať s klasickou osciláciou okolo niekoľkých tiketov na konci

vývojových cyklov, ale aj táto by mala časom ustať. Celkovo by sa už teraz dalo tvrdiť, že aplikácia je relatívne stabilná a je pripravená na doručenie zákazníkovi.

3.3.1 SÚHRN

Záverom tejto podkapitoly sa na základe získaných údajov a informácií prikláňame k názoru, že nové riešenie je v konečnom dôsledku ekonomicky výhodnejšie. Riešenie lepšie zapadá do portfólia sady nových cloudových produktov spoločnosti SAP a predstavuje jeden zo spôsobov inovácie v súvislosti s investíciami realizovanými práve do tohto softwarového odvetvia a jeho technológie, čo reflektuje celkovú stratégiu firmy. Okrem toho by sa dalo konštatovať, že riešenie je technicky efektívnejšie keďže v jednej aplikácii sú realizované funkcie 2 predošlých programov v klasickom rozhraní SAP GUI. Svoje tvrdenia pritom podkladáme názormi respondentov z radov našich kolegov, ktoré sme získali krátkym dotazníkom o stave novej aplikácie v porovnaní so zastaraným riešením. Účastníci, teda respondenti dostali príležitosť tak vyjadriť svoj názor, pričom tento bol prevažne pozitívny.

Čo sa týka nákladov, tie sme vyčíslili s využitím človekohodín a údajov o zložení pracovného tímu pozostávajúceho z pracovníkov v rozličných úlohách projektového tímu vyvíjajúceho agilnou metodikou SCRUM. Záverom podkapitoly sme pojednali aj o frekvencii a výskyte tiketov po nasadení aplikácie na produkčné systémy, čo je opäť obdobný údaj o generovaní nákladov. Ide však o náklady na prevádzku riešenia, ktoré vznikali spoločnosti SAP za sledované obdobie 7 mesiacov. V nadväznosti na tieto informačné údaje sme polemizovali, že výskyt tiketov, resp. problémov či požiadaviek bližšie neurčeného charakteru by mohol kmitať okolo úrovne nižších počtov, pričom odhad neuvádzame, išlo by o čistú špekuláciu, no v zásade sa dá tvrdiť, že aplikácia je v súčasnosti stabilizovaná. Náklady v korunách sme neuviedli, keďže na to by nám musela firma poskytnúť informácie o platových skutočnostiach ostatných kolegov a to je údaj, ktorý je dôverný.

ZÁVER

Cieľom tejto práce bolo zvenovať sa existujúcej aplikácii na výpočet režijných nákladov, ktorá mala byť nahradená novým riešením, ktoré by patrične zapadlo do cloudového portfólia softwarových produktov spoločnosti SAP. Začali sme teoretickými východiskami v kapitole 1, kde sme sa oboznámili s faktormi a súvislosťami, relevantnými pre nadchádzajúce kapitoly. Popísali sme pojmy týkajúce sa všeobecných pojmov z prostredia informačných systémov, vo veľkej miere sme sa venovali aj pojmom týkajúcich sa cloudových technológií a kapitolu sme ukončili pojednaním o nástrojoch pre modelovanie a zachytávanie technických skutočností.

V kapitole 2. sme sa v rozsiahlej miere venovali analýze súčasného stavu, čo znamenalo analyzovať aplikáciu, alebo skôr aplikácie na výpočet réžii. Presnejšie povedané, keďže ide o aplikácie v prostredí SAP možno hovoriť transakciách či programoch, to všetko sme v tejto časti ozrejmili.

Analýza prebiehala na viacerých úrovniach, prvou triviálnou úrovňou bolo posúdenie spôsobu akým sa jednotlivé transakcie používajú. To si vyžadovalo oboznámenie sa s mechanizmami týkajúcimi sa konfigurácie a nastavovanie kmeňových, prípadne užívateľských dát. To nás tým pádom priviedlo aj k časti skúmania samotných dát a ich kľúčových častí.

Keď už boli zrejmé patričné metódy použitia a mali sme k dispozícii aj správnu konfiguráciu, prešli sme ku skúmaniu technickej stránky aplikácii a dospeli sme k záveru, že sa používa jednotné výpočtové jadro, resp. sa dá tvrdiť, že sme overili hypotézu. V súvislosti s takouto technickou analýzou sme realizovali aj inšpekciu zdrojového kódu v statickej forme ako aj vo forme dynamickej s využitím debugovacieho nástroja. Pri dynamickej analýze kódu sme odhalili aj závislosti na iné programy a zvláštnu pozornosť sme venovali rozsiahlym dátovým transformáciám, ktoré sa pri behu vykonávali.

Všetky relevantné poznatky, ktoré nám firma umožnila použiť sme v primeranej miere reflektovali pri uvedení diagramov, prvý krát popísaných v teoretickej časti. Týmito diagramami sme modelovali biznis proces, algoritmus a prúdenie dát, pričom diagram prúdenia dát sme označili za najdôležitejší a to vďaka už spomenutým náročným dátovým transformáciám.

Ďalej sme uviedli aj niekoľko nameraných metrík, ktoré sme získali spúšťaním transakcii a poukázali sme na kľúčové detaily.

Kapitolu sme zavřšili popisom nástrojov, ktoré sme mali po dobu vývoja k dispozícii.

V tretej kapitole sme popísali implementované riešenie v podobe webovej aplikácie. Po celú dobu prechádzania touto kapitolou sme poukazovali na rozdiely oproti analýze uvedenej v kapitole 2. Kapitolu riešenia sme podobne ako v prípade kapitoly 2 začali popisom nového užívateľského rozhrania, čo sme ilustrovali patričnými obrázkami. Následne sme prešli k popisu programovej časti, kde sme predložili diagramy, tak ako tomu bolo aj v prípade kapitoly 2 a opäť sme popisovali relevantné skutočnosti, pričom tieto boli doplnené o porovnanie oproti analýze. Túto časť sme tiež uzavřeli prezentáciou niektorých nameraných indikátorov, ktoré sme za obdobie implementácie stihli nazbierať. Išlo o časový horizont niekoľkých mesiacov, pričom sme dospeli k záveru, že nová aplikácia je technicky efektívnejšia a všeobecne lepšia ako staré riešenia.

Záver kapitoly sme venovali ekonomickému zhodnoteniu, kde sme rozobrali prínosy nového riešenia a podiel práce projektového tímu.

Záverom práce konštatujeme, že zadanie na tvorbu novej aplikácie na kalkuláciu režijných nákladov sme splnili úspešne a dodali sme fungujúci produkt, ktorý je v súčasnej dobe aktívne zaradený do portfólia cloudových aplikácií spoločnosti SAP a je predávaný v rámci servisných softwarových produktov. Podľa našich informácií je tento nový produkt používaný už mnohými zákazníkmi, čo považujeme za ďalší úspech, náš návrh bol prijatý.

ZOZNAM POUŽITÝCH ZDROJOV

- (1) MOLNÁR, Zdeněk. *Efektivnost informačních systémů*. 2. rozš. vyd. Praha: Grada, 2001. Management v informační společnosti. ISBN 80-247-0087-5.
- (2) SODOMKA, Petr a Hana KLČOVÁ. *Informační systémy v podnikové praxi*. 2., aktualiz. a rozš. vyd. Brno: Computer Press, 2010. ISBN 978-80-251-2878-7.
- (3) BASL, Josef a Roman BLAŽÍČEK. *Podnikové informační systémy: podnik v informační společnosti*. 3., aktualiz. a dopl. vyd. Praha: Grada, 2012. Management v informační společnosti. ISBN 978-80-247-4307-3.
- (4) MOLNÁR, Zdeněk. *Efektivnost informačních systémů*. 2. rozš. vyd. Praha: Grada, 2001. Management v informační společnosti. ISBN 80-247-0087-5.
- (5) GÁLA, Libor, Jan POUR a Zuzana ŠEDIVÁ. *Podniková informatika: počítačové aplikace v podnikové a mezipodnikové praxi*. 3., aktualizované vydání. Praha: Grada Publishing, 2015. Management v informační společnosti. ISBN 978-80-247-5457-4.
- (6) SODOMKA, Petr a Hana KLČOVÁ. *Informační systémy v podnikové praxi*. 2., aktualiz. a rozš. vyd. Brno: Computer Press, 2010. ISBN 978-80-251-2878-7.
- (7) CHEN, Thomas, CHUANG a Kazuo NAKATANI. The Perceived Business Benefit of Cloud Computing: An Exploratory Study. *Journal of International Technology*. 2016, **25**(4), 101-121. ISSN 15435962.
- (8) VAQUERO, Luis, Luis RODERO-MERINO, Juan CACERES a Maik LINDNER. A break in the clouds. *ACM SIGCOMM Computer Communication Review*. 2008, **39**(1), 50-55. ISSN 0146-4833. Dostupné z: doi:10.1145/1496091.1496100
- (9) MELL, Peter a Timothy GRANCE. *The NIST Definition of Cloud Computing*. 1. National Institute of Standards and Technology Gaithersburg: Computer Security Division Information, Technology Laboratory National Institute of Standards and

- Technology Gaithersburg, 2011. Dostupné také z:
<https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>
- (10 FURHT, Borko, Armando ESCALANTE, ed. *Cloud Computing Fundamentals*. 1.
) Boston, MA: Springer, Boston, MA, 2010. ISBN 978-1-4419-6523-3.
- (11 MARKOVIĆ, Dragan, Irina BRANOVIĆ, Dejan ŽIVKOVIĆ a Violeta
) TOMAŠEVIĆ. OVERVIEW OF CLOUD COMPUTING IN BUSINESS. *Proceedings of the 1st International Scientific Conference - Sinteza 2014* [online]. Belgrade, Serbia: Singidunum University, 2014, , 673-677 [cit. 2020-04-26]. ISBN 978-86-7912-539-2. Dostupné z: doi:10.15308/sinteza-2014-673-677
- (12 ASMUS, Steve, Ahmed FATTAH a Chris PAVLOVSKI. Enterprise Cloud
) Deployment: Integration Patterns and Assessment Model. *IEEE Cloud Computing* [online]. 2016, 3(1), 32-41 [cit. 2020-04-26]. ISSN 2325-6095. Dostupné z: doi:10.1109/MCC.2016.11
- (13 FLINDERS, Karl. How organisations use private cloud to cut development time and
) costs. *Computer Weekly*. 2013, 1(1). ISSN 0010-4787.
- (14 LINTHICUM, David S. Emerging Hybrid Cloud Patterns. *IEEE Cloud Computing*
) [online]. 2016, 3(1), 88-91 [cit. 2020-10-27]. ISSN 2325-6095. Dostupné z: doi:10.1109/MCC.2016.22
- (15 *Standard ECMA-4: Flow Charts*. 2nd edition. 114 Rue du Rhône - 1204 Geneva
) (Switzerland): European Computer Manufacturers Association, 1966.
- (16 Towards EPC standardization. *Www.epc-standard.org* [online]. Německo:
) www.epc-standard.org, 2020 [cit. 2020-11-11]. Dostupné z: https://www.epc-standard.org/collaborate/Towards_EPC_standardization
- (17 EPC diagram. *Visual-paradigm* [online]. Hong Kong: visual-paradigm.com, 2020
) [cit. 2020-12-01]. Dostupné z: <https://www.visual-paradigm.com/VPGallery/bpmodeling/epc.html>
- (18 ŘEPA, Václav. *Analýza a návrh informačních systémů*. Praha: Ekopress, 1999.
) ISBN 80-861-1913-0.

- (19 Overview: Data Flow Diagrams (DFDs). In: *MX Computer Tuition* [online]. 465,
) Triq il-Kbira Mosta, Malta: MX Computer Tuition, 2020 [cit. 2020-11-12].
Dostupné z:
http://www.matthewxuereb.com/uniFiles/CIS1107_1207/DFDs_Notes_And_Exercises.pdf
- (20 SAP History. *SAP* [online]. Walldorf, Nemecká spolková republika: SAP SE, 2020
) [cit. 2020-11-29]. Dostupné z:
<https://www.sap.com/corporate/en/company/history.html>
- (21 History and timeline. *UNIX* [online]. Spojené štáty Americké: The Open group,
) 2020 [cit. 2020-11-29]. Dostupné z:
http://www.unix.org/what_is_unix/history_timeline.html
- (22 The early years. *SAP* [online]. Walldorf, Nemecká spolková republika: SAP SE,
) 2020 [cit. 2020-11-29]. Dostupné z:
<https://www.sap.com/corporate/en/company/history/1972-1980.html>
- (23 The SAP R/2 era. *SAP* [online]. Walldorf, Nemecká spolková republika: SAP SE,
) 2020 [cit. 2020-11-29]. Dostupné z:
<https://www.sap.com/corporate/en/company/history/1981-1990.html>
- (24 The SAP R/3 era. *SAP* [online]. Walldorf, Nemecká spolková republika: SAP SE,
) 2020 [cit. 2020-11-29]. Dostupné z:
<https://www.sap.com/corporate/en/company/history/1991-2000.html>
- (25 Real-time data where and when you need it. *SAP* [online]. Walldorf, Nemecká
) spolková republika: SAP SE, 2020 [cit. 2020-11-29]. Dostupné z:
<https://www.sap.com/corporate/en/company/history/2001-2010.html>
- (26 In-memory, cloud computing, and business network support record results. *SAP*
) [online]. Walldorf, Nemecká spolková republika: SAP SE, 2020 [cit. 2020-11-29].
Dostupné z: <https://www.sap.com/corporate/en/company/history/2011-present.html>

- (27 SAP SE Supervisory Board. *SAP* [online]. Walldorf, Nemecká spolková republika:
) SAP SE, 2020 [cit. 2020-11-29]. Dostupné z:
<https://www.sap.com/investors/en/governance/supervisory-board.html>
- (28 SAP ČR, spol. s r.o. , Praha IČO 49713361 - Obchodní rejstřík firem. *KurzyCZ*
) [online]. Česká republika: Kurzy.cz, spol. s r.o., AliaWeb, spol. s r.o., 2000 [cit.
2020-11-20]. Dostupné z: <https://rejstrik-firem.kurzy.cz/49713361/sap-cr-sro/>
- (29 *Databáza Orbis* [online]. Európa: Bureau van Dijk, 2020 [cit. 2020-10-31].
) Dostupné z: <https://www.orbis.org>
- (30 *Ariba* [online]. Walldorf, Nemecká spolková republika: SAP SE - Ariba, 2020 [cit.
) 2020-11-29]. Dostupné z: <https://www.ariba.com/>
- (31 *SAP Concur* [online]. Walldorf, Nemecká spolková republika: SAP SE - Concur,
) 2020 [cit. 2020-11-29]. Dostupné z: <https://www.concur.com/>
- (32 *Qualtrics XM* [online]. Spojené kráľovstvo Veľkej Británie a Severného Írska:
) Qualtrics, 2020 [cit. 2020-11-29]. Dostupné z: <https://www.qualtrics.com/uk/>
- (33 Enterprise SaaS market size worldwide from 2009 to 2019 (in billion U.S dollars).
) In: *Statista* [online]. Európa: Statista, 2020 [cit. 2020-11-29]. Dostupné z:
[statista.com/statistics/1114856/enterprise-saas-market-size/](https://www.statista.com/statistics/1114856/enterprise-saas-market-size/)
- (34 Statista dossier about Cloud computing. In: *Statista* [online]. Európa: Statista, 2020
) [cit. 2020-11-29]. Dostupné z: www.statista.com/study/15293/cloud-computing-statista-dossier/
- (35 Enterprise SaaS vendor market revenue share worldwide in 2nd quarter 2018 and
) 1st quarter 2019. In: *Statista* [online]. Európa: Statista, 2020 [cit. 2020-11-29].
Dostupné z: www.statista.com/statistics/209482/saas-vendors-by-customer-number/
- (36 *Statista* [online]. Európa: Statista, 2020 [cit. 2020-11-29]. Dostupné z:
) www.statista.com

- (37 SAP's global revenue from 2009 to 2019, by segment. In: *Statista* [online]. Európa: Statista, 2020 [cit. 2020-11-29]. Dostupné z: www.statista.com/statistics/263850/saps-global-revenue-by-segment/
- (38 Microsoft's revenue from 2012 to 2020 financial years, by segment. In: *Statista* [online]. Európa: Statista, 2020 [cit. 2020-11-29]. Dostupné z: www.statista.com/statistics/273482/segment-revenue-of-microsoft/
- (39 Oracle's revenue by business segment from FY2008 to FY2020*. In: *Statista* [online]. Európa: Statista, 2020 [cit. 2020-11-29]. Dostupné z: www.statista.com/statistics/269728/oracles-revenue-by-business-segment-since-2008/
- (40 *OData* [online]. Spojené štáty Americké: OData, 2020 [cit. 2020-11-30]. Dostupné z: <https://www.odata.org/>
- (41 Run Overhead Calculation - Actual. *SAP Fiori Apps Reference Library* [online]. Walldorf, Baden-Württemberg, Nemecko: SAP SE, 2021 [cit. 2021-02-11]. Dostupné z: [https://fioriappslibrary.hana.ondemand.com/sap/fix/externalViewer/#/detail/Apps\('F4857'\)/S23](https://fioriappslibrary.hana.ondemand.com/sap/fix/externalViewer/#/detail/Apps('F4857')/S23)

ZOZNAM OBRÁZKOV

Obrázok 1 – IT model informačného systému	20
Obrázok 2 – úrovne používateľov IS	20
Obrázok 3 – procesné spracovávanie objednávok	21
Obrázok 4 - Terminálová značka	26
Obrázok 5 – Všeobecná činnosť	27
Obrázok 6 – Vstupno-výstupná značka	27
Obrázok 7 – Pod-proces	27
Obrázok 8 – Značka vetvenia	27
Obrázok 9 - Cyklus	28
Obrázok 10 - Odkaz	28
Obrázok 11 – Udalosť v diagrame EPC	28
Obrázok 12 – Funkcia EPC	29
Obrázok 13 – Informačná surovina, informačný zdroj	29
Obrázok 14 – organizačná jednotka	29
Obrázok 15 – Operátory diagramu EPC	29
Obrázok 16 – DFD notácia Yourdon-DeMarco	30
Obrázok 17 – všeobecný náhľad na infraštruktúru systémov SAP	39
Obrázok 18 – infraštruktúra rýdzo cloudového riešenia	40
Obrázok 19 – Infraštruktúra pre vývoj	41
Obrázok 20 – diagram prepojenia back-endu vývojového prostredia	41
Obrázok 21 – ukážka SAP GUI	42
Obrázok 22 – ukážka webového rozhrania aplikácií SAP v technológii FIORI	43
Obrázok 23 – Transakcia KGI2 – réžia objednávok	48
Obrázok 24 – vstupná obrazovka transakcie CJ44 – výpočet rézie zvyšných objektov	49
Obrázok 25 – detailné zobrazenie počítania rézií	50
Obrázok 26 – chybové hlášky z výpočtu rézie	51

Obrázok 27 – tabuľka výsledkov výpočtu réžie.....	52
Obrázok 28 – výsledky pre jednu objednávku.....	52
Obrázok 29 – konfiguračný hárok s vypočítanými hodnotami	53
Obrázok 30 – dokončenie spracovávanía pri dialógovom zobrazovaní.....	54
Obrázok 31 – prehľad účtovníckych operácií na nákladovom objekte v transakcii KOB1.....	55
Obrázok 32 – zoznam dokumentov vygenerovaných operáciou výpočtu réžie	55
Obrázok 33 – dokumenty zachytávajúce účtovnícke operácie – finančné účtovníctvo	56
Obrázok 34 – kontrolingový dokument, ktorý vznikol z výpočtu réžie.....	56
Obrázok 35 – aktivita úprav konfigurácie v transakcii SPRO.....	57
Obrázok 36 – zoznam konfiguračných hárkov	58
Obrázok 37 – riadku konfiguračného hároku	59
Obrázok 38 – konfigurácia percentuálnej bázy pre výpočet	60
Obrázok 39 – konfigurácia pre používanie aktivít a množstevných jednotiek pre výpočet	61
Obrázok 40 – objekt, ktorý má byť na strane dal pri účtovaní	62
Obrázok 41 - Diagram behu procesu výpočtu réžii	63
Obrázok 42 – Zjednodušený algoritmus výpočtu réžii.....	64
Obrázok 43 – Zjednodušený algoritmus výpočtu réžii.....	65
Obrázok 44 – Diagram prúdenia dát, úroveň 0	66
Obrázok 45 – Diagram toku dát 1. úrovne, proces prípravy dát.....	67
Obrázok 46 – Diagram toku dát 1. úrovne, proces výpočtu réžii.....	68
Obrázok 47 – Diagram prúdenia dát 1. úrovne, proces spracovania	69
Obrázok 48 – Meranie výkonu transakciou ST05	70
Obrázok 49 – Meranie kódu ABAPu	70
Obrázok 50 – Ukážka transakcie SE80.....	73
Obrázok 51 – SAP ADT	74
Obrázok 52 – Intranetový GitHub	75

Obrázok 53 – Transakcia ZABAPGIT	75
Obrázok 54 – ABAP debugger – klasické rozhranie	76
Obrázok 55 – ABAP debugger – použité ADT	76
Obrázok 56 – Dlaždice FIORI na systéme SAP po prihlásení.....	79
Obrázok 57 – Záznamy po otvorení aplikácie.....	80
Obrázok 58 – Vyskakovacia informačná bublina s odkazmi do ďalších aplikácií a dodatočnými informáciami	80
Obrázok 59 – Aplikácia po skrytí filtra	81
Obrázok 60 – Voľba objektov podľa typu	81
Obrázok 61 – Nastavenia filtra na nákladový typ objednávka	81
Obrázok 62 – Možnosti exportu vyfiltrovaných hodnôt	82
Obrázok 63 – Pomocná obrazovka pre výber hodnôt	82
Obrázok 64 – Tlačidlá pre spustenie hlavných funkcií	83
Obrázok 65 – Dialógové okno pre zadanie parametrov výpočtu	83
Obrázok 66 – Obrazovka s výsledkami	84
Obrázok 67 – Informácie a štatistiky v prípade úspešného výpočtu	84
Obrázok 68 – Chyba kalkulácie s miernou závažnosťou	85
Obrázok 69 – EPC diagram nového biznis procesu	87
Obrázok 70 – Zjednodušený pohľad na beh webovej aplikácie.....	89
Obrázok 71 – DFD diagram novej aplikácie - úroveň 0.....	91
Obrázok 72 – DFD diagram – 1. úroveň, príprava.....	92
Obrázok 73 – DFD diagram – 1. úroveň, výpočet.....	93
Obrázok 74 – DFD diagram – 1. úroveň, spracovanie	94
Obrázok 75 – Prehľad kľúčových indikátorov výkonu	97

ZOZNAM TABULIEK

Tabuľka 1 – Kľúčové metriky výkonu KGI2	71
Tabuľka 2 – Doplnkové metriky KGI2	71
Tabuľka 3 – Tabuľka dôležitých metrík riešenia.....	97
Tabuľka 4 – Meranie výkonu na sieti.....	98
Tabuľka 5 – Prepočet na pracovné hodiny použité na vývoj	100
Tabuľka 6 – Výsledky hodnotenia dizajnu novej aplikácie.....	101
Tabuľka 7 – Hodnotenie otázky či sa interakcia s novým riešením zdá jednoduchšia v porovnaní s tým predošlým.....	101
Tabuľka 8 – Vyhodnotenie otázky, či v prípade nového riešenia ide o zlepšenie oproti pôvodnému riešeniu	101
Tabuľka 9 – Počet tiketov za obdobie od septembra 2020 po marec 2021.....	102

ZOZNAM GRAFOV

Graf 1 - Výnosnosť trhu SaaS do roku 2019 v miliardách dolárov spojených štátov	34
Graf 2 - Výnosnosť cloudových aplikácií v miliardách dolárov.....	34
Graf 3 – Objem výdajov na verejný cloud podľa technologického segmentu v miliardách dolárov.....	35
Graf 4 – objem výnosov na celosvetovom trhu so SaaS 2. kvartál 2018 a 1. kvartál 2019	35
Graf 5 – Celosvetové výnosy SAPu rozdelené podľa segmentu v miliónoch euro..	36
Graf 6 – Výnosy spoločnosti Microsoft z inteligentného cloudu v miliardách dolárov	36
Graf 7 – výnosy spoločnosti Oracle z rozličných cloudových technológií	37
Graf 8 – Merania výkonu za 4 mesiace.....	96
Graf 9 – Zmeny a oscilácia času odozvy v „End to End“ meraní	96