

**ZÁPADOČESKÁ UNIVERZITA V PLZNI**  
**FAKULTA EKONOMICKÁ**

Bakalářská práce  
**Analýza nákladů na podporu SW produktů**  
**Cost analysis of SW product support**

Jan Kalvoda  
Plzeň 2020



ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta ekonomická

Akademický rok: 2019/2020

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Jan KALVODA**  
Osobní číslo: **K17B0409P**  
Studijní program: **B6209 Systémové inženýrství a informatika**  
Studijní obor: **Informační management**  
Téma práce: **Analýza nákladů na podporu SW produktů**  
Zadávající katedra: **Katedra ekonomie a kvantitativních metod**

### Zásady pro vypracování

1. Charakterizujte SW podporu a uveďte zásady ITIL.
2. Objasněte rozdělení jednotlivých stupňů podpory Diebold Nixdorf, jejich definice a specifikace, možné způsoby získávání dat.
3. Popište způsob sběru dat a získejte relevantní data k výpočtu.
4. Určete rozdíly v ceně při rozdílných stupních podpory, definujte statistické ukazatele a způsob jejich výpočtu.
5. Vytvořte desktopovou aplikaci k určení predikce nákladů a vizualizaci dat.
6. Zjistěte využitelnost aplikace pomocí standardizovaného dotazníku.

Rozsah bakalářské práce: **40 – 60 stran**  
Rozsah grafických prací: **neuveđen**  
Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

- AXELOS. *ITIL Foundation: ITIL 4 Edition*. TSO, 2019. IBSN 9780113316076.
- ADDY, Rob. *Effective IT service management: to ITIL and beyond!* New York: Springer, 2007. ISBN 978-3540731979.
- HENDL, J. *Přehled statistických metod zpracování dat : analýza a metaanalýza dat*. Praha: Portál, 2006. ISBN 80-7367-123-9.
- SOUČEK, Eduard. *Statistika pro ekonomy*. Praha: Vysoká škola ekonomie a managementu, 2006. ISBN 80-86730-06-9.
- *Which Factors Affect Software Projects Maintenance Cost More?* National Center for Biotechnology Information [online]. Dostupné z: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3610582/>.

Vedoucí bakalářské práce: **Doc. RNDr. Mikuláš Gangur, Ph.D.**  
Katedra ekonomie a kvantitativních metod

Datum zadání bakalářské práce: **22. října 2019**  
Termín odevzdání bakalářské práce: **22. dubna 2020**



**Doc. Ing. Michaela Krechovská, Ph.D.**  
děkanka



**Ing. Mgr. Milan Svoboda, Ph.D.**  
vedoucí katedry

V Plzni dne 22. října 2019

## Čestné prohlášení

Prohlašuji, že jsem bakalářskou práci na téma „*Analýza nákladů na podporu SW produktů*“ vypracoval samostatně pod odborným dohledem vedoucího/vedoucí bakalářské/diplomové práce za použití pramenů uvedených v příložené bibliografii.

V Plzni dne .....

Jan Kalvoda .....

## **Poděkování**

Tímto bych chtěl poděkovat vedoucímu práce panu doc. RNDr. Mikuláši Gangurovi za odborné vedení této práce a poskytnutí rad a připomínek. Dále bych rád poděkoval Ing. Davidu Krivánkovi za samotnou možnost psaní práce, poskytnutí dat a osvětlení mnohých konceptů týkající se softwarových produktů.

## Obsah

Úvod práce .....	9
I. Softwarová podpora.....	10
Charakteristika softwarové podpory .....	10
Přístupy k softwarové podpoře .....	11
Obecné rozdělení .....	11
Rozdělení do sedmi modelů.....	12
II. ITIL A ITSM.....	14
Definice ITIL .....	14
Popis zásad ITIL .....	15
ITSM.....	16
Poddruhy ITSM .....	18
III. Náklady na softwarovou podporu.....	21
Faktory ovlivňující softwarovou podporu .....	21
Metody zabývající se odhadem nákladů .....	22
Ukázka metody PERT.....	22
Matematické modely sloužící k odhadu nákladů.....	24
Základní model COCOMO.....	24
IV. Společnost Diebold Nixdorf a způsob její softwarové podpory.....	26
Popis společnosti Diebold Nixdorf.....	26
Popis softwarové podpory Diebold Nixdorf a jejích stupňů.....	27
V. Proměnné v dosavadním způsobu výpočtu a forma dat.....	29
Dosavadní způsob výpočtu .....	29
Proměnné v dosavadním výpočtu .....	30
Forma dat .....	31
Jira.....	32
Zhodnocení dosavadních poznatků, rozhodnutí o dalším postupu .....	33
VI. Zpracování dat a návrh nového způsobu výpočtu .....	34
Práce s daty .....	34
Transformace dat .....	34
Práce s proměnnými.....	35
Explorační analýza.....	36

Základní proměnné .....	37
Testování vlivu jednotlivých proměnných.....	38
Vytvoření skupin proměnných pomocí clusterů .....	39
Tvorba nového způsobu výpočtu .....	40
Popis technologie použité k tvorbě aplikace a popis základního testování .....	41
Technologie.....	41
Testování použitelnosti .....	44
VII. Získané výsledky a tvorba nástroje na predikci nákladů.....	45
Finální stav dat.....	45
Testování vlivu proměnných na čas řešení tiketu .....	47
Testování vlivu proměnných na počet tiketů .....	48
Vytvoření aplikace k predikci nákladů .....	50
Aplikace na Cloudu.....	56
Zjištění využitelnosti aplikace pomocí dotazníku.....	56
Závěr .....	58
Seznam obrázků a tabulek .....	59
Seznam použitých zkratk .....	60
Zdroje.....	61
Literatura.....	61
Vědecké články.....	61
Zdroje z Webu.....	61
Ostatní zdroje.....	63
Seznam příloh .....	64



## Úvod práce

---

Téma práce bylo nabídnutou plzeňskou pobočkou společnosti Diebold Nixdorf. Tato firma se zabývá technologickými řešeními na trhu bankovníctví a retailu. Právě z těchto činností vzniklé náklady budou hlavním předmětem této práce.

Hlavním cílem této práce je návrh a implementace nástroje sloužící k predikci nákladů na softwarovou podporu.

Podcíle této práce jsou rozděleny následovně:

- Definice softwarové podpory
- Popis systému ITIL, který slouží ke zlepšení využití IT technologií
- Popsat možné způsoby predikce nákladů na softwarovou podporu
- Popsat společnost Diebold Nixdorf a systém jejich softwarové podpory
- Prozkoumat daný systém a určit klíčové proměnné
- Zpracovat data poskytnutá společností a vytvořit způsob predikce

Práce je rozdělena do sedmi kapitol. V první kapitole je osvětlován koncept softwarové podpory jako takový. Její součástí jsou také přístupy k softwarové podpoře a obecný popis faktorů, jež ovlivňují náklady.

V další kapitole se práce zabývá knihovnou infrastruktury IT (Information Technology Infrastructure Library – zkráceně ITIL) a managementem IT služeb (IT service management – zkráceně ITSM). ITIL je důležitou metodikou pro řízení většiny velké části činností spojených s IT. ITSM je způsob řízení softwarové podpory vycházející z ITILu. Kapitola číslo tři je věnována nákladům na softwarovou podporu. Jsou zde uvedeny příklady technik a matematických modelů používaných k jejich predikci.

Kapitola čtvrtá je o společnosti Diebold Nixdorf a stupňů rozdělení jejich podpory pro zákazníky. Kapitoly pět a šest se zabývají daty a proměnnými, které vstupují do predikce nákladů. Také je zde navrhnut nový způsob výpočtu spolu s uvedením způsobu dosavadního.

Poslední kapitola je tvořena vyhodnocením získaných výsledků a tvorbou nástroje k budoucí predikci nákladů na softwarovou podporu. Je zde popsána použitá technologie a vyhodnocení aplikace pomocí standardizovaného dotazníku.

# I. Softwarová podpora

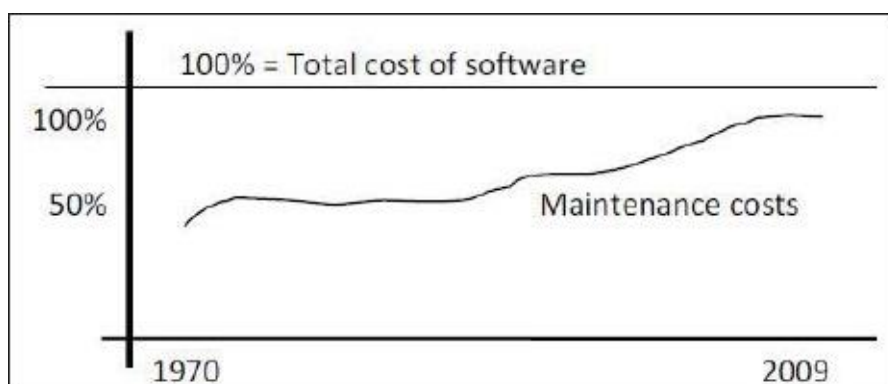
## Charakteristika softwarové podpory

Softwarová podpora je často chápána ve formě softwaru, který podporuje nějaký projekt nebo aktivitu. V této práci ale bude tento pojem chápán jako typ technické podpory, jenž podporuje a spravuje software (ADDY, Rob, 2007).

Softwarová podpora by se dala definovat mnoha způsoby, o čemž nám napovídá i předchozí odstavec. Následující definice pochází z Business Dictionary, což je slovník moderního businessu. Ten definuje softwarovou podporu jako: *“Po prodeji vzniklá služba poskytovaná vydavatelem nebo prodejcem softwaru za účelem řešení softwarových konfliktů nebo problémů s použitelností a zároveň poskytující aktualizace, opravy chyb a řešení bezpečnostních hrozeb v programu.”*(Business Dictionary,2019, vlastní překlad)

Jednoduše řečeno se tedy jedná o typ technické podpory, která se váže k softwaru. Pro účely této práce je nutnost dále definovat IT service management. Podrobná definice a popis problému jsou uvedeny v další části, ale pojem se objevuje několikrát už nyní a z tohoto důvodu je zde uveden krátký popis. Je to typ managementu, co se zabývá kvalitou služeb IT a nejlepší zkušenosti jednotlivých zákazníků s naším softwarem (ADDY, Rob, 2007).

Obrázek 1. Podíl nákladů na údržbu softwaru na celkových nákladech v letech



Total cost =  
Celkové  
náklady  
  
Maintenance costs  
= Náklady na  
údržbu

Zdroj: *The National Center for Biotechnology Information, 2013*

Nyní je softwarová podpora a s ní spojený IT service management jednou z největších částí moderního trhu s IT. *“V závislosti na části průmyslu se náklady na IT service management pohybují mezi 65 až 80% celkových výdajů na IT. [...] Většina těchto nákladů je vynaložena na mzdové náklady.”* (ADDY, Rob, 2007, s. 7, vlastní překlad)

*Cena softwarové údržby stále roste a odhady ukazují, že devadesát procent délky životního cyklu softwaru je ve vztahu s fází údržby. [...] Odhady ukazují navýšení o padesát procent v posledních dvou desetiletích.* (The National Center for Biotechnology Information, 2013, vlastní překlad)

## **Přístupy k softwarové podpoře**

### **Obecné rozdělení**

Podstatou softwarové podpory je samozřejmě poskytnutí co nejlepší zkušenosti s naším produktem. I u softwaru tvořeného jednou osobou budeme očekávat nějakou dokumentaci, jež usnadní práci s aplikací. Přestože se v podstatné části této práce budeme zabývat spíše podporou, která bude vyžadovat nějaké pravidelné aktivity ze strany firmy zodpovědné za tvorbu softwaru, právě dokumentace ať už tištěná nebo v elektronické podobě je jednou ze základních forem podpory.

Tento způsob podpory je samozřejmě jen základní kostrou, která neřeší záležitosti jako bugy a další komplexní problémy. Pokud budeme používat software k podpoře nějakého většího projektu, určitě se nebudeme moct spoléhat jen na dokumentaci. Budeme se tedy muset spoléhat na jiné formy podpory, od kterých můžeme očekávat, že už budou vyžadovat větší množství vynaložené práce a komunikaci se zákazníkem (WiseGEEK, 2003).

Ve většině případů tedy budeme muset provádět údržbu založenou na zpětné vazbě našeho klienta. Údržba by se dala rozdělit do tří kategorií (Hidden Brains, 2003).

#### *1. „Dokonalá údržba*

*Tento přístup k údržbě pomáhá při přepracování, vylepšování nebo úpravě softwarového produktu nebo aplikace při řešení problémů ovlivňující výkon a funkčnost systému.*

## 2. *Adaptivní údržba*

*Tento přístup je nejlepší použít, pokud se očekává, že náš software a hardware bude fungovat v dynamickém, novém a měnícím se prostředí.*

## 3. *Pravidelná údržba*

*Pravidelná údržba je vhodná pro stabilní produkty nebo aplikace, které požadují pravidelné aktualizace a správu jejich verzí, takovým způsobem aby byly v souladu s nejnovější technologií napřed před konkurencí. “*

(Hidden Brains, 2003, vlastní překlad)

## **Rozdělení do sedmi modelů**

Softwarová podpora se samozřejmě dá rozdělit více způsoby. Jedním takovým rozdělením je rozdělení z knihy Robba Addyho.

*“IT služby:*

- 1) *Funkčně technické*
- 2) *Systémově aplikační*
- 3) *Obchodně orientované modely*
- 4) *Modely orientované na podporu*
- 5) *Modely založené na dlouhodobém předplatném*
- 6) *Modely založené na jednorázové platbě*
- 7) *Modely založené na rolích”*

(ADDY Rob, 2007, s. 106, volný překlad)

- 1) Funkčně technickou službou můžeme chápat například službu, která se bude týkat správou databáze. Je to tedy služba zaměřená na nějakou technickou disciplínu.
- 2) Systémový přístup ke službám bude vhodný, pokud se budeme zabývat například správou emailů nějaké společnosti. Budeme tedy řídit nějaký propojený systém.
- 3) Obchodně orientovaný model by byl model podpory určený k podpoře nějakého e-shopu. Staráme se o nějakou obchodní službu.

- 4) Modely orientované na podporu považují samotnou podporu za produkt. Může se tedy jednat o nastavení a implementace firemní sítě, která ale pochází od jiné společnosti.
- 5) Ty modely, jež jsou založené na dlouhodobém předplatném, se objevují velice často a téměř každý se s tímto jevem ve svém životě setká. Je to například obnova antiviru, kdy si ročně platíme licenci, a daná firma nám poskytuje aktuální verzi softwaru.
- 6) Modely založené na jednorázové platbě jsou taky dosti rozšířené. Příkladem může být najmutí si společnosti na nákup a zprovoznění hardwaru pro naši firmu.
- 7) Poslední model je založen na myšlence, že naše pozice nás dostává do nějaké role. Pokud je někdo vedoucí prodeje očekává se od něj, že bude mít třeba dálkový přístup do firemních archivů a bude schopen udělat podstatnou část práce z mobilního zařízení. V tomto modelu tedy budeme vytvářet řešení IT podpory pro určitou roli, kterou člověk ve svém životě zastává (ADDY Rob, 2007).

## II. ITIL A ITSM

---

### Definice ITIL

Pokud firma zajišťuje nějaké IT služby, ve většině případů se očekává od jejich vedoucích zaměstnanců znalost knihovny ITIL. Za touto zkratkou se skrývá pojem “IT Infrastructure Library”.(ITIL Foundation,2019)

Web CIO je obecně definuje jako: *“ITIL je knihovna obsahující několik svazků popisující rámec nejlepších postupů pro poskytování IT služeb. ITIL ve svojí historii prošel množstvím revizí a nyní obsahuje pět knih, kdy každá z nich obsahuje různé procesy a fáze životního cyklu IT služeb. Systematický přístup ITIL může pomoci managementu IT služeb v řízení rizik, posilování vztahů se zákazníky, zavádět nové nízkonákladové postupy a stavbou stabilního IT prostředí, který umožňuje osobní růst, zvyšování rozsahu a změny.”* (CIO, 2019, vlastní překlad)

První verze ITILu vznikla v roce 1989. GITIM bylo původní jméno, které znamenalo “Government Information Technology Infrastructure Management”. Tedy management určený pro vládní IT infrastrukturu. Velké množství států v Evropě přijalo tento styl řízení své IT infrastruktury a postupně se k nim začali přidávat i soukromé společnosti. V roce 2000 použila společnost Microsoft ITIL jako základ pro stavbu svého vlastního způsobu řízení IT služeb. To pomohlo růstu a popularitě ITILu.

Od roku 2013 vlastní ITIL společnost Axelos, která se stará o distribuci knih a kurzy k tomuto způsobu řízení (ITIL Training,2018).

Nyní je ITIL ve svojí čtvrté fázi. Dříve byl poměrně přísný, ale nyní bere v potaz nové směry myšlení jako je třeba Lean, tedy přístup, který se snaží minimalizovat plýtvání a tím zvýšit efektivitu. Dalším takovým směrem je Agile. To je iterační styl projektového řízení neboli způsob vyvíjení po krocích (ITIL Foundation,2019).

Podrobný popis situace a problémů vzniklé z kombinace Agile a ITILu je mimo rozsah této práce.

Důležitou podstatou ITILu bylo a je donutit ostatní průmyslové odvětví začít nahlížet na IT jako na rovnocenného partnera a ne jako na nutné zlo. Často je IT necháno v izolaci, místo toho aby se díky vzájemné komunikaci vytvořila společná strategie, což by přineslo větší užitek všem (ADDY Rob, 2007).

## **Popis zásad ITIL**

Celý systém ITIL se drží sedmi základních zásad.

První z těchto zásad je, aby se vždy zaměřovalo na přidanou hodnotu pro stakeholdery. Nemusí to být hodnota peněžní, ale například vytvoření dobré zkušenosti z práce s naším produktem. Musíme si vždy provést správnou analýzu stakeholderů, zjistit jejich nynější potřeby a vybrat jaké prostředky použijeme na jejich uspokojení.

Další zásadou je začít tam kde jste. Uvnitř společnosti už pravděpodobně je obrovské množství prostředků použitelných k tomu, abychom se dostali do cíle. Nejsme povinni vždy začínat od začátku. Velmi důležité je zde zvažování rizika opětovného použití některých prostředků. Také je důležité si uvědomit, že občas neexistuje nic, co by dalo znovu použít.

Třetí zásada je obdobná jako u řízení stylem Agile. Říká nám, že bychom měli vše provádět iteračně a vždy si vyslechnout zpětnou vazbu. Až po získání informací o kvalitě naší práce bychom měli začít pracovat na něčem dalším. Pro úspěšné použití tohoto principu je nutné si pamatovat, že výstup jednotlivých cyklů iterace musí obsahovat všechny prvky, které jsou nutné k úspěchu.

Čtvrtá položka se týká týmové práce. Členové týmu by spolu měli vždy spolupracovat na co nejvyšší úrovni a neměli by se zatajovat žádné informace. Dobrá komunikace s týmem nám umožní nalezení míst, kde se plýtvá prostředky. Také nám umožní lépe pochopit jednotlivé pracovní úkony a jejich specifikace.

Položka pátá říká, že by se mělo přemýšlet nad problémem ve stylu Holismu. To znamená, že na všechno musíme nahlížet jako na celek a ne jenom jako na jednotlivý dílek. Žádný element IT služeb neexistuje ve vakuu. Měli bychom vždy zkoumat složitost jednotlivých systémů a vztahy jednotlivých dílků uvnitř nich.

Zásada šest hovoří o nutnosti řídit procesy jednoduše a prakticky. Pokud tedy nějaký proces dlouhodobě nevykazuje výsledky, je nejlepší ho eliminovat. Je svázána s první zásadou, která říká, že musíme vždy nejdříve myslet na zvýšení hodnoty. Je důležité myslet na to, že při zavádění nového procesu zvýšíme pravděpodobnost úspěchu tak, že daný proces bude jednoduchý na pochopení a naučení. Další důležitou částí této zásady je vytváření rychlých vítězství. To nám umožní demonstraci pokroku.

Položka sedm popisuje důležitost optimalizace a automatizace. Každý dostupný zdroj by měl být vytížen do svého maxima. Technologie by měla být použita všude, kde použita může být. Lidské zásahy by měly existovat jen v případech, kdy tvoří nějakou hodnotu. Opět je zde důležité myslet na to, že když se budeme snažit automatizovat komplexní procesy, výsledek nebude takový, jaký jsme zamýšleli. Vždy je dobré automatizovat ty procesy, co jsou jednoduché a nejlépe k jejich automatizaci již máme technologii. (ITIL Foundation,2019)

## ITSM

ITSM je pojem již krátce zmíněný v předchozí kapitole. Je úzce spjat se zásadami ITIL a je to velmi důležitý prvek řízení IT služeb.

*“IT Service Management (ITSM) - obvykle se nepřekládá a používá se zkratka. Je to souhrn nejlepších praxí a referenčních modelů procesů řízení služeb IT. ITSM představuje způsob řízení informačních a komunikačních technologií, jejich provozu i rozvoje, který využívá principů řízení na bázi služeb, zahrnuje tedy pohled zákazníků i poskytovatele IT služeb. Pojem IT Service management vychází z rámce ITIL, ve kterém byl koncept řízení IT pomocí služeb poprvé použit, ale není s ním výlučně spjat.”*  
(ManagementMania,2011)

Aby byla možnost dodržet pátou zásadu ITIL, jež mluví o nutnosti nahlížet na jednotlivé problémy holisticky, existuje vymezení čtyř základních dimenzí service managementu. Tím že budeme schopni správně alokovat zdroje do jednotlivých dimenzí a dodržet zásadu první, veškerá naše činnost bude tvořit hodnotu (ITIL Foundation,2019).



*”Tyto 4 dimenze jsou:*

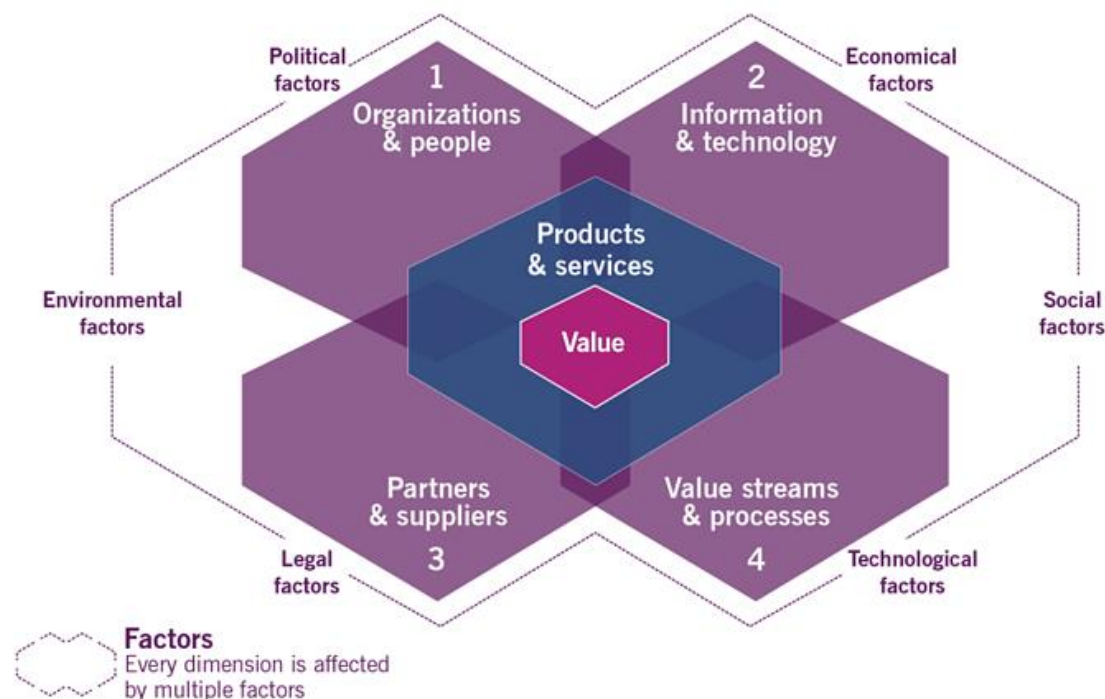
*Organizace a lidé*

*Informace a technologie*

*Partneři a dodavatelé*

*Jednotlivé procesy a toky plateb“ (ITIL Foundation,s. 15, 2019, vlastní překlad)*

Obrázek 2. Dimenze ITSM



*Zdroj: Axios Systems, 2019*

Správně implementovaný systém ITSM bude mít velké množství pozitivních dopadů na danou společnost. Umožní nám to lépe dodržovat základní zásady ITILu, což povede ke zvýšení efektivity. Zvýšení efektivity může být všechno od snížení času, který budeme potřebovat na vyřešení problému, do odstranění zbytečných procesů plýtvajících časem zaměstnanců. Dobrý ITSM také povede k vytvoření lepší zákaznické podpory. Měl by také pomoci ke stabilnějšímu systému a snížení nákladů.

Velký problém, se kterým se ITSM často potýká je špatná kompetence manažerů k vedení svých podřízených. Často se jedná o mistry ve svém oboru v IT, ale chybí jim takzvané měkké dovednosti. Znamená to, že jednotliví manažeři jsou schopni vytvořit složitý algoritmus, ale nejsou schopni motivovat a vést svůj tým vhodným způsobem.

Dalším častým neduhem je také to, že mnoho manažerů považuje vytváření obrovských množství záznamů o činnostech svého týmu za management. Místo vedení svého týmu do budoucnosti vytváří popis minulosti. To ovšem neznamená, že nevytvářet žádné záznamy je správně. Měli by se vždy archivovat informace, co jsou relevantní.

Nejdůležitější částí ITSM značí v jeho zkratce “S - Service”, tedy služba. V tomto případě se bude nejčastěji jednat o softwarovou podporu. První částí této podpory je zjištění problému, co náš zákazník má. Musíme vytvořit takový způsob, abychom mohli jednoznačně zjistit problém a začít připravovat řešení.

V IT existuje velká výhoda, že nemusíme být fyzicky u problému, abychom ho mohli vyřešit. Proto můžeme řešení zaslat našemu zákazníkovi elektronicky a nebudeme muset řešit výjezd. Některé problémy ale nebude možné takto vyřešit a zde je třeba zvažovat, jaký člen našeho týmu je vhodný na řešení daného úkolu. Musíme zvažovat věci jako je produktivita a schopnosti interakce se zákazníkem.

Je doporučováno vyjasnit si softwarovou podporu pomocí nějakého kontraktu. Ten by měl jasně a jednoznačně určit jaké povinnosti máme. Také bychom měli definovat, jaké informace nám musí zákazník dodat, abychom byli schopni dodat mu uspokojivé řešení. Dále je dobré si ujasnit, jakým způsobem vynahradíme zákazníkovi náš případný neúspěch (ADDY Rob, 2007).

### **Poddruhy ITSM**

Ze samotného ITSM je odvozeno obrovské množství poddruhů managementu. Práce se bude zabývat těmi, jež jsou pro ni nejvíce relevantní.

- 1) Change Management - Management změn
- 2) Interaction Management - Management kontaktu se zákazníkem
- 3) Service Level Management - Management správy služeb
- 4) Problem Management - Management řešení problémů (ADDY Rob, 2007)

- 1) IT služby se jenom díky svojí existenci dostávají do absurdní situace. Očekává se od nich stabilita a spolehlivost, ale také schopnost rychle se adaptovat na veškeré změny. Už po přečtení těchto dvou očekávání je jasné, že jsou protichůdné.

Úkolem managementu změn je vytvořit takové řešení, které bude správně reagovat na danou situaci, ale co nejméně ohrozí stabilitu systému.

Co se týká change managementu tak ITIL nenabízí příliš velký popis. Záleží především na daném manažerovi, aby změnu díky svojí znalosti produktu správně naplánoval a implementoval.

Testování případných změn je v IT o trochu snazší než v ostatních odvětvích. Pokud se bude jednat o změnu softwaru, můžeme ho testovat přímo v průběhu implementace změny a budeme objevovat problémy, jež by změna mohla přinést (BMC Software Blog, 2016).

- 2) Kontakt se zákazníkem je jednou z nejpodstatnějších částí IT podpory. Bez správné definice problému ho nejsme schopni správně vyřešit. Je doporučeno, aby tento typ managementu byl brán jenom jako součást managementu jiného. Pokud tedy budeme jakožto manažer změn nějakou změnu provádět, musíme být schopni zároveň komunikovat se zákazníkem a získávat jeho input na daný proces.

Znovu se objevuje už předem zmiňovaný problém chybějících měkkých schopností některých IT manažerů. Často probíhá komunikace pouze pomocí emailu a při pokusu o jiný typ komunikace dojde k neúspěchu. Velký problém je také časté spoléhání pouze na komunikaci, která nevyžaduje lidský faktor. To mohou být různé formuláře, které způsobí špatné zachycení problému (ADDY Rob, 2007).

- 3) Management správy služeb je typem managementu nadřazený všem ostatním typům ITSM. Zabývá se jak návrhem, tak pozdější správou a kontrolou celého systému IT služeb dané firmy.

SLM je rozdělen do tří komponent - Firemní, Zákaznická a úroveň dané služby

Vždy si musíme odpovědět na to otázku, co se po jednotlivých komponentách vyžaduje. Pokud naše firma bude pracovat s daty zákazníka, která jsou citlivá, bude se očekávat určitá míra zabezpečení. Úrovní dané služby se rozumí, například možnost zavolat na technickou podporu.

Pokud chceme správně určit, jak musí být vytvořené jednotlivé komponenty, musíme znát své stakeholders. Díky znalosti stakeholders se můžeme také

rozhodnout, jestli svůj model služeb zaměříme více na služby nebo na zákazníky. Model zaměřený na zákazníka vytvoří každému jeho nabídku na míru. Ten zaměřený na služby vytvoří jednu službu, co bude vyhovovat nejvíce lidem (Simplilearn,2019).

- 4) Poslední typ ITSM popsany v práci, se zabývá tím, jak nejlépe naložit s problémy. V úvodní fázi se snaží, aby žádný problém vůbec nevznikl. Poté se pokouší minimalizovat jeho následky. Dále se daný manažer zabývá detailní analýzou problému, snaží se najít jeho hlavní příčinu a zajistit, že už se nikdy nebude opakovat.

Existují dva přístupy k problémům - reaktivní a proaktivní. Reaktivní znamená, že čekáme, až se daný problém objeví a poté teprve následuje jeho řešení. Proaktivní se snaží vylepšovat jednotlivé prvky softwarové podpory, tak aby žádný významný problém nemohl nastat.

Důležitým pojmem v Problem managementu je “Workaround”. Je to krátkodobý způsob řešení problému, vytvořen za účelem získání více času na řešení hlavní. “Workaround” neřeší příčiny problému, jenom odstraňuje jeho následky (Freshservice, 2018).

### III. Náklady na softwarovou podporu

---

#### Faktory ovlivňující softwarovou podporu

Výši nákladů na softwarovou podporu ovlivňuje několik skupin faktorů a to: “

- *Charakteristika produktu*
- *Technologické omezení počítačů*
- *Charakteristika zaměstnanců*
- *Charakteristika projektu*
- *Faktory ovlivňující údržbu projektu*
- *Náklady na údržbu počítačových sítí “*

(The National Center for Biotechnology Information,2013, vlastní překlad)

- Relevantní položka patřící do první skupiny je především složitost našeho produktu. S její rostoucí složitostí budou růst i naše náklady.
- Největším faktorem ve skupině technologických omezení je čas, jenž počítač potřebuje k vykonání operace. Problém, který v této době téměř neexistuje, je hardwarová kapacita počítače na uložení dat nutných k instalaci a provozu programu a není tedy nutné se jím výrazně zabývat.
- Co se týká zaměstnanců, jsou klíčové dvě položky. První z nich je zkušenost s prací s daným produktem. Druhou je schopnost našich programátorů provádět úpravy v daném softwaru.
- U charakteristiky produktu nás budou nejvíce zajímat technologie a postupy použité k tvorbě produktu. Náklady tedy výrazně ovlivní programovací postupy použité v naší práci. Také volba nástrojů bude výraznou položkou při rozhodování.
- Faktory týkající se údržby jsou podobné jako faktory týkající se zaměstnanců. Nejdůležitější tedy bude znalost daného softwaru a kompetence našeho týmu k údržbě softwaru. Co nás zde také velmi ovlivňuje je dokumentace vytvořená při samotné tvorbě softwaru. Její nízká úroveň nám může přinést obrovské množství problémů.

- Poslední skupinou faktorů jsou náklady týkající se počítačových sítí. Především na jejich údržbu a ochranu před útočníky (The National Center for Biotechnology Information, 2013).

### **Metody zabývající se odhadem nákladů**

Pro odhad nákladů na činnost existuje velké množství postupů. Můžeme použít analogii, kdy expert vytvoří vyčíslení na základě nějakých předchozích zkušeností, dále parametrického modelu kde se snažíme získat náklady podle jednotkové ceny parametru. To by v našem případě byla třeba cena odstranění jedné chyby v programu. Tyto metody jsou sice poměrně snadné a rychlé, ale jejich přesnost nemusí být na takové úrovni, jakou potřebujeme (SKALICKÝ Jiří, JERMÁŘ Milan, SVOBODA Jaroslav, 2010).

### **Ukázka metody PERT**

Metodou odhadu, jež nebude na úrovni matematických modelů, ale zároveň bude mít dostatečnou vypovídající hodnotu, se zabývá Petr Marounek ve svojí práci *Simplified approach to effort estimation in software maintenance*. Metoda byla zvolena jako příklad metody, která se dá použít, pokud nechceme používat komplexní matematické modelování. Jedná se o upravenou metodu PERT pocházející z projektového managementu.

Definice samotné metody PERT před úpravou:

“Metoda PERT používá statistiku k určení očekávané hodnoty doby trvání. [...] Očekávaná hodnota využívá tři odhadů doby trvání a určí se jako jejich vážený průměr a směrodatná odchylka.

$$\text{očekávaná hodnota} = (o + p + 4s) / 6$$

kde je: *o*... optimistický odhad

*p*... pesimistický odhad

*s*... nejpravděpodobnější odhad

směrodatná odchylka  $\sigma = (p-o)/6$

rozptyl  $\sigma^2$ ” (SKALICKÝ Jiří, JERMÁŘ Milan, SVOBODA Jaroslav, 2010, s. 142)

Zde se metoda zabývá odhadem času, ale pro odhad nákladů je postup stejný.

Pokud máme historické záznamy z obdobného projektu, můžeme metodu PERT dále upravovat následujícím způsobem.

$$O = \frac{\frac{Min + 4x Mod + Max}{6} + 1/n \sum Xi}{2}$$

Kde: 'O' reprezentuje vypočtený odhad vynaloženého úsilí na danou aktivitu [...]

Průměrné hodnoty z obdobné znalostní základny pro podobné relevantní úkoly se počítají jako  $1/n \sum Xi$ . “ (MAROUNEK P., 2012, s. 57, vlastní překlad)

Min je optimistický odhad nákladů. Mod je realistický odhad a max je pesimistický. Obdobně jako u obyčejné metody PERT.

Pokud máme i souhrn predikcí z minulých let, můžeme vytvořit pravděpodobnost správného expertního odhadu. K tomu se používá ukazatel velikosti relativní chyby (MRE). Ten vytvoříme následovně.

“ $MRE = (realita - odhad) / realita$ ” (MAROUNEK P., 2012, s. 57, vlastní překlad)

Vytvoříme tedy rozdíl reálných nákladů a nákladů předpokládaných, které vydělíme náklady reálnými. Čím větší množství historických dat máme, tím je daný ukazatel přesnější. Tento ukazatel můžeme pro jednotlivé aktivity použít jako přesnost expertního odhadu a násobit jím ty odhady, co sami jsme vytvořili. Ve vzorci ho budeme značit jako P (expA). Můžeme tedy říct, že  $MRE \text{ Experta A} = P (\text{expA})$ .

Finálním vzorec odhadu nákladů dle upravené metody PERT tedy bude

$$O = \frac{\frac{P(\text{expA}) * Min + P(\text{expA}) * 4x Mod + P(\text{expA}) * Max}{6} + 1/n \sum Xi}{2}$$

(MAROUNEK P., 2012, s. 58)

## Matematické modely sloužící k odhadu nákladů

„*Ekonomicko – matematický model je zjednodušené zobrazení, resp. matematický popis reálného systému, který obsahuje pouze prvky a vazby mezi těmito prvky podstatné pro zkoumaný ekonomický problém.*“ (PLEVNÝ Miroslav, ŽIŽKA Miroslav, 2010, s. 14)

Další možností jak provést predikci našich nákladů je použití některého z matematických modelů na to určených. Pro tento úkon bylo vytvořeno poměrně velké množství modelů, přičemž většina z nich používá jako svůj vstup počet řádků kódu daného softwaru.

Mnoho manažerů má velkou nedůvěru k modelům. Odhaduje se, že kolem 80% všech odhadů je tvořeno pouze odhadem experta. Často je to zapříčiněno velkou komplexností modelů. Některé z modelů například používají umělou neuronovou síť a rozeznávání vzorů. Na druhou stranu existují modely, jež jsou heuristické nebo vyžadují jenom základní početní úkony. (MAROUNEK P., 2012)

Možným problémem, se kterým se můžeme při použití modelů potýkat, je jejich stáří. IT velmi rychle se vyvíjejícím oborem a mnoho faktorů, které některé modely používají, už nemusí být relevantní. Může jít například o zvyšující se paměť počítačů, nebo o zavádění nových technik do IT sektoru (The National Center for Biotechnology Information, 2013).

### Základní model COCOMO

Model COCOMO už je poměrně starý. Vznikl v roce 1981, ale i přesto je jeden z nejpoužívanějších modelů na predikci softwarových nákladů na světě. Vzorce jsou upraveny pro demonstraci.

Pro účel zápisu rovnic  $PR =$  odhadovaný počet řádek kódu

Jeho obecný předpis pro úsilí je  $Us = a * (PR) * b$

Kde: a spolu s b jsou zvolené podle typu projektu

Us vyjadřuje nutné úsilí k vykonání aktivity. Úsilí se měří ve člověkoměsících.

Jeho obecný předpis pro čas k vykonání je  $C = c * (Us) * d$

Kde: a spolu s b jsou zvolené podle typu projektu

C vyjadřuje čas nutný k vykonání. Čas se měří v měsících.



COCOMO rozlišuje tři druhy projektů:

Organický: kdy dobře fungující tým pracuje na poměrně jednoduchém softwaru

Částečně oddělený: použijeme ho, když pracujeme s mixem zkušených a nezkušených zaměstnanců v daném odvětví

Vnořený: Pokud je daný softwarový projekt úzce spjat s nějakým hardwarem

Tabulka 1. Koeficienty COCOMO

Typ projektu	a	b	c	d
O	2,4	1,05	2,5	0,38
ČO	3,0	1,12	2,5	0,35
V	3,6	1,2	2,5	0,32

*Zdroj: upravená tabulka z webu JAVATPOINT,2011*

Po rozhodnutí jaké koeficienty použijeme, provedeme odhad počtu řádek a dosadíme do modelu. Výsledkem bude kolik člověkohodin, budeme potřebovat k úspěšnému konci projektu a celkový čas, který na projekt budeme potřebovat.

Pokud bychom chtěli přesnější výsledky, musíme do modelu ještě započítat faktory ovlivňující náklady. Tím by se základní model COCOMO stal COCOMO střední úrovně (JAVATPOINT,2011, vzorce zjednodušeny pro účel demonstrace modelu).

## IV. Společnost Diebold Nixdorf a způsob její softwarové podpory

---

### Popis společnosti Diebold Nixdorf

*„Společnost Diebold Nixdorf, s.r.o. zaujímá přední postavení mezi výrobci a dodavateli informačních systémů pro oblast bankovníctví a maloobchodu (Retailu).*

*Hlavní činností společnosti jsou návrhy a prodeje komplexních hardwarových a softwarových řešení včetně servisu a poradenských služeb.“ (Diebold Nixdorf Ostrava, 2019)*

Už od svého začátku se tato firma zabývala penězi. V roce 1859 byla založena Charlesem Dieboldem za účelem výroby sejfů a trezorů. Ve svojí historii má mnoho zajímavých předmětů, jako jsou třeba obrovské dveře trezoru vyrobené z manganu, jenž byly odolné proti trhavině TNT.

Historie společnosti byla vždy úzce svázána s bankovníctvím. Po druhé světové válce rostla spolu s počtem bank v USA i tato společnost. S růstem popularity šeků, využila firma příležitosti a stala se výrobcem několika jejich částí.

V sedmdesátých letech se snížila poptávka po bankovním zabezpečení. Diebold potřeboval diverzifikaci a proto začal s výzkumem bankomatů. Toto rozhodnutí se vyplatilo, společnost se stala expertem na danou oblast a byla schopna rozšířit svoje pobočky po celém světě. Svůj stávající název Diebold Nixdorf firma získala po spojení s německou společností Wincor Nixdorf (Encyclopedia,2019).

V současné době je Diebold Nixdorf skutečně obrovská společnost. Každý třetí bankomat na světě pochází z jejich režie. Má více jak 23 000 zaměstnanců a působí ve více než sto třiceti zemích světa. Odhadem jsou zodpovědní za 1,3 milionu transakcí denně. Také se jedná o společnost, která se snaží být společensky odpovědnou.

Firma se nyní také zabývá nových technologií pro retail. Jsou to různé nové způsoby plateb, jako jsou například samoobslužné pokladny (Diebold Nixdorf Mezinárodní, 2019).

V České republice je činnost firmy řízena z pražské pobočky založené díky činnosti Wincor Nixdorf. Přestože tuzemská část nese název Diebold Nixdorf s.r.o, jejím jediným společníkem je právě Wincor. Česká část je tudíž provázána se svým světovým protějškem tímto společníkem (Kurzy.cz, 2020).

Pro moji práci bude relevantní pobočka v Plzni, která vznikla v roce 2015. Tato pobočka se zabývá především toky peněz, což obsahuje bankovní transakce a bankomaty. Jedná se především o podporu softwarových produktů, které zajišťují správný chod celého procesu. Nyní je v ní zaměstnáno okolo 170 lidí (Diebold Nixdorf Plzeň, 2019).

### **Popis softwarové podpory Diebold Nixdorf a jejích stupňů**

Ve společnosti funguje systém založený na bázi takzvaných tiketů. Znamená to, že do systému jsou zadávány jednotlivé potíže, kterými jsou postiženi jednotliví zákazníci. Pořadí, ve kterém jsou řešeny, se vytváří pomocí prioritizace.

Problémy s nejvyšší prioritou jsou značeny 1 a s nejmenší prioritou 4. Prioritu řešení také ovlivňuje, jaký typ balíčku podpory má zákazník zakoupený. Podrobný popis prioritizace je mimo rozsah této práce.

Při vzniku problému poskytuje Diebold Nixdorf takzvanou FQA – první kvalifikovanou odpověď. Nejčastěji se jedná o telefonát, kdy se pracovník pokusí zjistit závažnost vzniklé události a vytvořit odhad doby pro vytvoření Workaround. Jedná se o jakési rychle vytvořené řešení, jež bude minimalizovat dopady problému, ale nevyřeší ho.

Společnost nabízí čtyři stupně softwarové podpory, těmi jsou:

I. **Basic** – Základní úroveň podpory, daný subjekt má jenom minimální prioritu, dostane jenom FQA a služby pro něj fungují jenom v normální pracovní dobu.

II. **Bronze** – Stupeň podpory, jenž má mnohem rychlejší časy poskytnutí služby, navíc jsou už i nabízeny workaroundy, podpora opět funguje jenom v normální pracovní době.

III. **Silver** – Podpora je zde poskytována v rámci hodin a je provozována v celých pracovních dnech spolu se sobotou, pokud se jedná o prioritu 1. Další možností je od 6:00 do 20:00 v pracovních dnech, pokud se jedná o nižší prioritu.

IV. **Gold** – Jedná se o nejvyšší stupeň, časy jsou opět nižší a navíc je poskytována nepřetržitá podpora pro problémy priority 1.

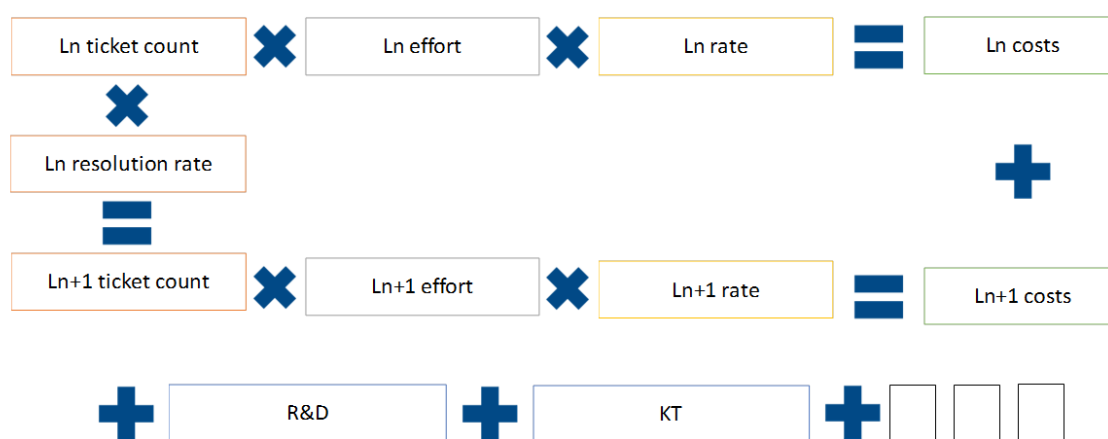
Pro všechny stupně je garantován měsíční report obsahující informace o množství vykonané práce (ENGEL Dorothea, 2017).

## V. Proměnné v dosavadním způsobu výpočtu a forma dat

### Dosavadní způsob výpočtu

Pro určení dalšího způsobu výpočtu a nalezení relevantních dat, bude v této práci popsán i dosavadní návrh predikce nákladů.

Obrázek 3. Aktuální návrh výpočtu nákladů na podporu



Zdroj: Interní zdroje Diebold Nixdorf, 2020

Nejdůležitější části výpočtu jsou „Ticket count“ (počet tiketů) a „Effort“ (vynaložená práce na tiket). Jejich správné odvození tvoří nejdůležitější část celého procesu. Jejichž odvození probíhá pomocí čtyř proměnných, jež jsou popsány v následující podkapitole.

Jedinou jistotou je položka „Rate“ (hodinová sazba). To je tabulková hodnota, tudíž je jisté, že používáme ve výpočtu správné čísla.

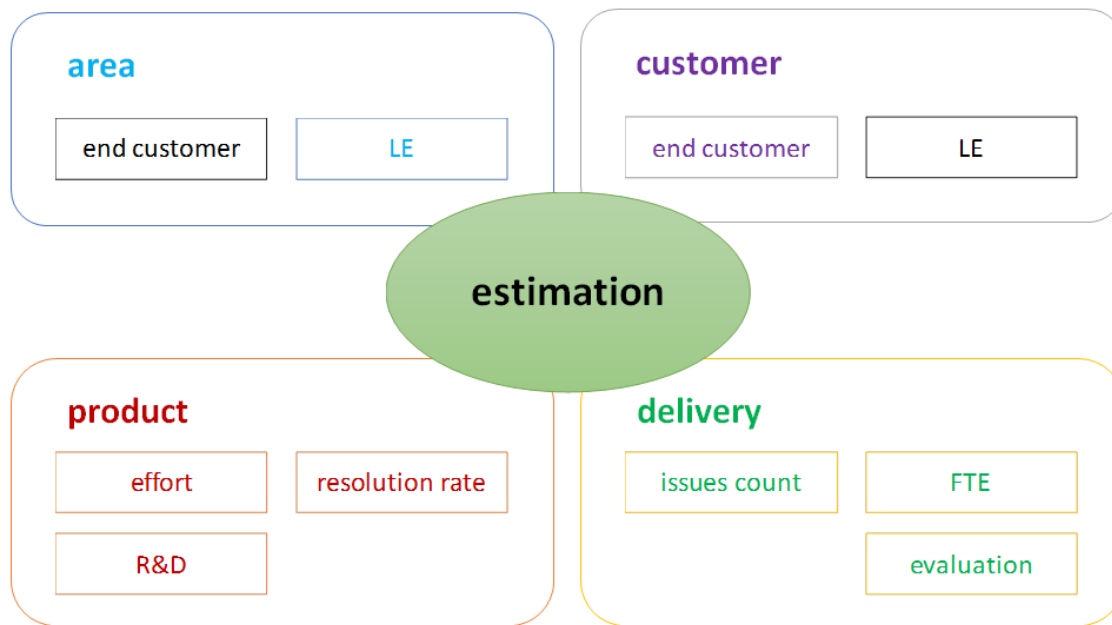
Vynásobením těchto tří položek získáme hodnotu softwarových nákladů na první úroveň podpory.

Dále se používá „Resolution rate“ (poměr vyřešených problémů). „Resolution rate“ se odvozuje z dat z minulých let. Ve vzorci uvedený „Resolution rate“ není zvolen zcela správně, je nutno ho ještě transformovat na „Failure rate“ (poměr nevyřešených problémů) pomocí jednoduchého výpočtu  $(1 - \text{„Resolution Rate“})$ . Při použití „Ticket Count“ \* „Failure rate“, získáme počet tiketů pro druhou úroveň. Zbytek výpočtu druhé úrovně probíhá obdobně jako u první.

Do nákladů se dále zahrnují dodatečné vývojové náklady, které jsou většinou pevně domluvené a dále platby pro konzultační týmy. Také se zde může objevit velké množství položek jako je například školení (KRIVÁNK A David, 2020).

### Proměnné v dosavadním výpočtu

Obrázek 4. Proměnné ve výpočtu



Zdroj: Interní zdroje Diebold Nixdorf, 2020

„Area“ (Oblast) a „Customer“ (Zákazník) není potřeba nijak složitě vysvětlovat. Jedná se o oblast, kde zákazník vyžaduje dané služby a poté informace o specifickém zákazníkovi.

„Product“ (Produkt) a „Delivery“ (Dodání produktu) nemusí být už tak lehké na správné pochopení, protože dodávání softwarového produktu neobsahuje jenom jeho samotné dodání zákazníkovi, ale i zajištění jeho funkcionality. Následující problém se dá představit na příkladu instalace operačního systému Windows. „Product“ je operační systém Windows na datovém nosiči. „Delivery“ je samotná instalace systému na našem počítači, různé konfigurace, instalace ovladačů a podobně.

Čtyři dimenze zobrazené na obrázku se kombinují do položek „Ticket count“ a „Effort“ následujícím způsobem. Z kombinace informací o oblasti působení a zákaznickovy získáme počet tiketů a z kombinace dolní řádky získáme vynaloženou snahu.

Říkáme tedy:

**Area + Customer => Ticket Count**

**Product + Delivery => Effort**

Nutnost kombinace daných položek se dá uvést na příkladu: Firma X má pobočky ve dvou oblastech Kanada a Karibik. V obou těchto oblastech má svůj vlastní expertní IT tým, co se pokusí vyřešit problém předtím, než ho odešle do nějaké z poboček společnosti Diebold Nixdorf. Zatímco pobočka v Kanadě je jednou z nejlepších a posílá naprosté minimum, jejich kolegové z Karibiku se vyznačují tím, že generují obrovské množství problémů.

Uvedený příklad je samozřejmě extrémní možnost, ale dokazuje nám, že pro správnou predikci nemůžeme sledovat jen jednu věc, ale musíme brát v potaz co nejvíce relevantních informací.

Individuální přístup také bude vyžadovat každý z produktů. Může existovat extrémně jednoduchý produkt, jehož implementace a případné úpravy zaberou velké množství času, což zvýší úsilí extrémním způsobem. Nebo může být opačná situace, kdy může být opravdu rozsáhlý, ale optimalizovaný produkt, se kterým při práci nebudou žádné problémy.

Při uvažování nad položkou „Effort“ je také třeba uvažovat nad tím, že pokud se objevuje velmi malé množství tiketů, jako se projevovalo v příkladu s pobočkou v Kanadě, je velmi pravděpodobné že tikety, které od nich společnost dostane, budou velice složité. Položka „Effort“ by v těchto případech měla být nějakým způsobem mírně navýšena, aby získané výsledky co nejvíce odpovídaly realitě (KRIVÁNKA David, 2020).

## **Forma dat**

Data se vyskytují ve dvou formách. První z forem je jednoduchý seznam tiketů v sešitu aplikace Excel. Tento seznam obsahuje název tiketu v systému Jira (popsaným dále), spolu s dobou trvání řešení problému, jímž se tiket zabývá. Seznam byl autorovi poskytnut společností, přičemž byl vytvořen zaměstnancem, jenž se zabývá sledováním těchto tiketů.

Tyto názvy zároveň slouží i jako unikátní klíče k jednotlivým záznamům v Jiře. Záznamy se tedy stávají naší druhou formou dat.

## **Jira**

Jira je plánovací nástroj, vytvořený společností Atlassian, využívaný především v softwarovém inženýrství. Často se využívá ve spojení s dalším produktem od dané společnosti a to je Confluence, sloužící jako zdroj informací pro softwarové týmy. Jira umožňuje koordinaci i velmi komplexních mezinárodních projektů. Jednotlivým členům týmu je zde umožněno zadávat progres svých úkolů, problémy ve formě tiketů, dále se například umožňuje kontrolovat úkoly jiných.

Pro účely práce jsou nejdůležitější tikety. Z nich jsme schopni získat informace o problémech, které softwarová podpora řeší a vyvodit z nich případné náklady (Atlassian,2020).

Pro získání informací se používá speciální příkazový jazyk. Ten nazýváme Jira Query Language(JQL) – v překladu dotazovací jazyk pro Jira. Správně formulovaný dotaz nám umožní získat souhrn všech relevantních položek ze systému v jednom kroku. Dotazy mají následující formu: „*reporter = jankalvoda AND project =BP*“.

Tento dotaz by nám umožnil najít seznam všech tiketů vytvořený uživatelem „jankalvoda“ na projektu „BP“. Byl by to tedy jakýsi přehled možných problémů v projektu, který nahlásil daný uživatel.

Po zadání správného příkazu nám umožňuje systém jejich hromadné stažení (Atlassian, 2019).

Data, která takto získáme, mají nejčastěji formát CSV. Comma Separated Values, neboli čárkou oddělené hodnoty. S tímto formátem dokáže pracovat jakýkoliv tabulkový procesor (OpenData,2020).



## Zhodnocení dosavadních poznatků, rozhodnutí o dalším postupu

---

Autorem zvolené téma je v tuzemském prostředí poměrně málo prozkoumané. Při většině práce je nutno se spoléhat na zahraniční zdroje, především na společnost Axelos a její materiály. Zde se jistě ukazuje přínos v popsání alespoň základních pilířů tématu moderního IT service managementu a jeho zásad ITIL v českém jazyce. ITIL je velmi problematický ve velkých změnách ve své čtvrté iteraci, ve své snaze zahrnout iterační přístup Agile do softwarové podpory. Popis kombinace ITILu a Agile je velmi problematický a je mimo rozsah této práce.

Softwarová podpora a údržba je jedna z nejrychleji rostoucích částí IT z hlediska nákladů. Práce se nejdříve zajímá o její obecné vymezení a poté jak její náklady vypadají v praxi, konkrétně ve společnosti zadavatele. Práce se z velké části zabývá predikcí těchto nákladů. Existuje podstatné množství matematických metod a modelů vytvořených k tomuto účelu, ale jde především o odhady podle řádků kódu nebo názorů odborníků.

Práce se proto zabývá vytvořením nástroje, který bude schopný predikovat náklady na základě minulých dat získaných z firemních zdrojů. Data získáme pomocí získání unikátních klíčů záznamů ve firemním systému a následovněm stažením dat, která jsou s jednotlivými klíči svázána. Vzhledem k velkému množství dat bude nutnost vytvořit hromadný dotaz, což nám umožní získat všechna data v jednom kroku.

Původním nápadem autora práce bylo z podstatné části založit nástroj na základním modelu nákladů získaným od společnosti, kde by se také využila pozměněná metoda PERT vycházející z projektového managementu.

Po práci s daty se ukázalo, že tato metoda nebude použitelná ve svojí celé podobě. Pro predikci budoucích nákladů bude použita její upravená verze. Kde místo predikce budeme brát určité položky ze seznamu dat.

Velmi důležitý prvek tvorby jsou získané informace z rozhovoru s manažerem společnosti, jenž sloužil jako klíčový zdroj pro pátou kapitolu. Díky němu budeme moci správně určit všechny důležité proměnné.

## VI. Zpracování dat a návrh nového způsobu výpočtu

---

### Práce s daty

#### **Transformace dat**

Prvním krokem bude získání všech položek z excelové tabulky a zajištění jejich transformace na dotaz pro systém JIRA. Příkaz má následující syntaxi.

```
„issuekey in (SUPPORT-1, SUPPORT-2...SUPPORT-N)“
```

Je zde velké množství položek a manuální přepsání do příkazu by nám zabral velké množství času. Připravíme si skript v jazyce Visual basic, jenž nám umožní získat všechny klíče najednou a převede je to textové souboru.

Jako základ použijeme skript z webu Extend Office. Ten obsahuje cyklus s příkazem:

```
„Print #1, Cells(xFRNum, xFCNum).Value“
```

V tomto tvaru bychom získali výpis, kdy by na každé řádce byl právě jeden klíč. Potřebujeme ale formu, kterou můžeme vložit do závorek JIRA dotazu. Tudíž upravíme příkaz do následující formy:

```
„Print #1, Cells(xFRNum, xFCNum).Value + ",";“
```

Příkaz v této formě obsahuje dva nové prvky a to „+ “, ““ a „;“. První z nich přidá na konec každé položky „,“. Středník na konci nám zajistí, že celý výpis bude na jedné řádce (Extend Office, 2009). Po použití spuštění skriptu, můžeme tedy data vložit do JIRA Query a získáme souhrn všech relevantních dat.

Správně položený dotaz nám tedy umožní hromadné stažení ze systému. Data budou v již zmiňovaném formátu CSV. Počet buněk v získaném souboru bude přesahovat jeden milion. Musíme tedy najít a získat data, jež jsou relevantní pro naši práci.

## **Práce s proměnnými**

Pokud chceme filtrovat data, musíme si nejdříve ujasnit, co přesně hledáme.

Hledáme v dostupném souboru proměnné, které zásadním způsobem ovlivňují buď čas na vyřešení jednoho tiketů, nebo ty proměnné, jež ovlivňují počet nově vzniklých tiketů v daném roce.

Z dat budeme schopni najít informace, zdali existuje obchodní partner, který asistuje s podporou pro daného zákazníka u daného tiketů. Dále budeme schopni zjistit, o jakého zákazníka se jednalo. Také můžeme vyhledat, v jaké zemi se problém vyskytl. Samozřejmostí je název produktu, u kterého se problém vyskytl. Můžeme dohledat informace o tom, v jaké fázi projektu tiket vznikl.

Také se musí v potaz vzít sazba. Sazba je informace, kterou firma nechce sdělit, tudíž bude nástroj nastaven tak, aby se dala implementovat později kompetentním zaměstnancem společnosti. Z nástroje tedy zjistíme základní informace o časech, ke kterým bude eventuálně dodána průměrná sazba, a díky tomu nalezneme peněžní náklady.

V našich údajích by mělo být dostatečné množství dat na správné určení potřebných statistických ukazatelů. Jedná se o základní populaci tiketů roku 2019 pro plzeňskou pobočku (úplně všechny objekty co splňují zadané kritéria). Pokud budeme například hledat údaje o všech pobočkách zákazníka X, za jejichž podporu je zodpovědná plzeňská pobočka v našich záznamech bychom je měli nalézt za daný rok všechny (HENDL Jan, 2015). Vzhledem k tomu, že jednotlivé položky v systému JIRA jsou často kontrolovány managementem, data by měla být relativně spolehlivá.

Mírný problém bude tvořit proces získání informací o počtu vzniklých tiketů pro danou kombinaci. Máme jejich celkový seznam a potřebujeme z něj získat informace například o tom, kolik daný zákazník vytvoří za rok tiketů. Různí zákazníci mají ale různé počty produktů. Mohlo by se nám stát, že relativně bezproblémový zákazník bude zařazen do kategorie problematických.

Vytvoříme kontingenční tabulku, kde vytvoříme součet všech tiketů, co zákazník vytvoří a zároveň součet produktů, jež zákazník využívá. Součet tiketů vznikne jednoduše vložení funkce POČET do této tabulky. Pokud bychom to samé udělali pro produkty,

vyjde nám, že produktů má zákazník stejné množství jako tiketů. Jedná se o problém vycházející z formy dat. Kontingenční tabulka v tomto případě počítá i prázdné sloupce.

Zadáme produkt jako popis sloupců a využijeme funkce „=COUNTIF“ v následující formě, kdy oblastí dat bude řádek našeho zákazníka:

„=COUNTIF(OBLAST-DAT;“>0“)“

Funkce nám vrátí počet produktů, jaké zákazník používá. Obdobný postup použijeme pro všechny zkoumané proměnné a získáme vždy počet tiketů na jeden produkt, pro který daná skutečnost platí.

### **Explorační analýza**

Prvním krokem průzkumu dat bude nalezení základních vlastností souboru dat. Klíčová pro nás bude normalita dat. Podle jejího stavu budeme vybírat statistické testy. Pro normalitu dat je typické symetrické rozdělení dat kolem Gaussovy křivky (SOUČEK Eduard, 2006).

Pro ověření normality dat použijeme statistický test, jenž se nazývá Shapiro-Wilkův. Tento test nalezneme v SW Statistica a umožní nám jednoduše ověřit existenci normality v datech. Nulová hypotéza tohoto testu nám říká, že data jsou normálně rozdělena. Alternativní hypotéza nám tvrdí opak. Budeme testovat na hladině významnosti 5%. O platnosti hypotéz budeme rozhodovat pomocí p-hodnoty (SVOBODA Milan, GANGUR Mikuláš, MIČUDOVA Kateřina, 2019).

*„P-hodnota je vlastně nejnižší hladina významnosti, na které lze nulovou hypotézu zamítnout, a současně nejvyšší hladina, na které se nulová hypotéza ještě nezamítá. Lze tedy říci, že p-hodnota je hraniční hodnota významnosti, která dělí nezamítnutí („přijetí“) nulové hypotézy od jejího zamítnutí.“* (SVOBODA Milan, GANGUR Mikuláš, MIČUDOVA Kateřina, 2019, str. 34).

Pokud tedy výsledná hodnota p bude menší než 0,05, zamítneme normalitu. Pro nalezení ostatních základních vlastností souboru použijeme statistickou analýzu z Excelu. Budeme hledat především průměrné hodnoty pro různé skupiny dat, abychom je mohli porovnat s výstupy našeho nástroje. Také budeme testovat odchylky v různých regionech.

Dále můžeme najít ukazatele napomáhající nám k testu databáze, což jsou celkový průměrný čas a průměrný počet tiketů. Pomocí těchto ukazatelů můžeme ověřit, zdali je naše databáze naplněná správnými daty a zdali se převedli správným způsobem.

Pokud se prokáže normalita, tak další zkoumanou vlastností souboru bude shoda rozptylů. ANOVA předpokládá danou vlastnost a před použitím této metody ji tedy musíme otestovat. Použijeme opět SW Statistica, konkrétně typ testu, jenž je nazýván Levenův. Nulová hypotéza tohoto testu nám tvrdí, že data mají shodné rozptyly. Alternativní hypotéza tvrdí, že nulová neplatí (SVOBODA Milan, GANGUR Mikuláš, MIČUDOVÁ Kateřina, 2019).

### **Základní proměnné**

Při uvažování nad proměnnými v našem výpočtu je vhodné uvažovat i nad modelem navržený společností Diebold a vytvořit úvahu, co je z něj možné reálně použít. Proměnné stanovené tímto modelem jsou AREA, CUSTOMER, PRODUCT a DELIVERY. Tedy oblast, zákazník, produkt a jeho implementace. Budeme zkoumat vliv těchto proměnných na čas řešení tiketu a jejich počet.

Soubor dat neobsahuje žádné informace o složitosti implementace a jiných vlastností, které by se dali spojit s DELIVERY. Ve výpočtu jí tedy nelze použít. Pro její použití by bylo data nutné sesbírat pomocí například měření časů práce jednotlivých zaměstnanců. Tato aktivita je mimo rozsah této práce, zároveň je otázkou zdali by firma ocenila, kdyby někdo stál za zády jejich zaměstnanců a měřil jejich výkon.

Proměnná PRODUCT a její vazba na čas je v datech obsažena. Umožní prozkoumání vlivu produktu na čas vyřešení tiketu. Pokud se test ukáže jako pravdivý, bude vazba použita ve formě clusterů popsaných dále.

CUSTOMER obdobně jako PRODUCT je v datech jednoduchý k nalezení. Obsahuje kompletní informace o tom, u jakého zákazníka se vyskytl problém. Stejně jako PRODUCT vybízí k použití clusterů.

Poslední teoretická proměnná AREA, se vyskytuje v datech dokonce ve dvou formách. A to ve formě regionu (například EMEA – Europe, Middle East and Africa – Evropa, Střední Východ a Afrika) nebo jednotlivých států (například Německo). Využije se zde

obou možností. Oblast EMEA má pravděpodobně velkou směrodatnou odchylku počtu tiketů, proto bude vhodné rozdělit danou oblast na Afriku, Evropu a Střední východ a otestovat, zdali je průkazný rozdíl v počtu tiketů na zákazníka. Pomocí informací o státu poté můžeme region rozdělit.

Další proměnné použitelné jako vstupní, jsou: zdali se jednalo o spolupráci s obchodním partnerem, který zároveň sloužil jako zprostředkovatel (BUSINESS PARTNER) a také jakými fázemi projekt ve sledovaném období projde (PROJECT PHASE). Kdy obě proměnné budeme testovat ve vztahu s počtem vzniklých tiketů i vlivu na čas řešení jednoho tiketu.

### **Testování vlivu jednotlivých proměnných**

K určení existence vztahu proměnných k našemu výpočtu použijeme statistické testy analýzy rozptylu. Podle normality a homoskedasticity (ve výběrech jsou stejné rozptyly) rozhodneme ve prospěch ANOVY nebo Kruscal-Wallisova testu. ANOVU použijeme při splnění zmíněných dvou vlastností. Ke Kruscal-Wallisovu testu se uchýlíme, pokud jedna z vlastností není naplněna.

Obě možnosti jsou tedy testy srovnávající libovolný počet průměrů. Nulovou hypotéza je tvrdí, že jednotlivé průměry jsou si rovny. Alternativní hypotéza nám obdobně jako u Levenova testu říká, že nulová hypotéza neplatí. V alternativní hypotéze se tedy vyskytuje alespoň jeden výskyt položky, která je statisticky významně odlišná od ostatních. Pokud tedy zamítneme nulovou hypotézu, znamená to, že danou proměnnou má ve výpočtu cenu použít. K provedení poté opět použijeme program Statistica, kdy použijeme různé proměnné jako grupovací.

Základem ANOVY a i jiných způsobů analýzy rozptylu je ukazatel, jenž rozhoduje, zdali se určené proměnné liší díky náhodnému kolísání, nebo zdali je zde nějaký významný rozdíl, což znamená, že faktory mají vliv. Tento ukazatel se nazývá F-poměr nebo F-statistika. Po jeho dosazení do Fisherova-Snedecorova rozdělení, které odečteme od jedničky, získáme p-hodnotu (HENDL Jan, 2015).

Vypočteme ho následovně:

$$„F = \frac{\text{vážený rozptyl mezi průměry hodnot ve skupinách}}{\text{rozptyl mezi hodnotami ve stejné skupině}}“ \text{ (HENDL Jan, 2015, s. 349).}$$

Kruskal-Wallis je test, který funguje na bázi vytvoření pořadí prvků. Vezmeme všechny vstupní data a seřadíme je. V případech výskytu dvou identických hodnot, vytvoříme průměrné pořadí. Dalším krokem je vytvoření součtů pořadí z jednotlivých skupin.

Pro provedení této operace pro všechny naše položky, můžeme dosadit do vzorce, ze kterého získáme H-statistiku. Tu dosadíme do chí kvadrát rozdělení, jež odečteme od jedničky, a získáme naši p-hodnotu (HENDL Jan, 2015).

$$H = \left( \frac{12}{n(n+1)} * \sum_i \frac{(SR_i)^2}{n_i} \right) - 3 * (n + 1)$$

*Kde: n                      počet prvků*

*SR<sub>i</sub>                      součet pro ítou skupinu“*

(HENDL Jan, 2015, s. 357).

Pokud budeme zkoumat proměnnou, která má jenom dvě varianty, jako je například zdali máme v daném případně obchodního partnera, tak použijeme Mann-Whitneyův test. Ten je obdobou Kruskal-Wallise pro dva výběry. Hypotézy budou stanoveny stejně jako u předchozích testů (SVOBODA Milan, GANGUR Mikuláš, MIČUDOVA Kateřina, 2019).

### **Vytvoření skupin proměnných pomocí clusterů**

Některé proměnné mají menší počet výskytů. Pokus o jejich budoucí predikci tedy může být značně zkreslená. Data tedy bude vhodné seskupit do nějakých skupin, kdy pro skupinu bude platit určitá podobnost mezi členy. Můžeme poté vzít i ostatní členy této skupiny a získáme lepší obraz situace (Statsoft, 2014).

Nejvíce problematické jsou skupiny proměnných PRODUCT a CUSTOMER, tedy seznamy produktů a zákazníků. U zákazníků se dále vyskytuje problém v zadávání názvu. Často jsou zde různé překlepy, především od zákazníků z rozvojových zemí, kde je počítačová gramotnost na nižší úrovni.

Při průzkumu dat se například objevil zákazník, jenž měl pokaždé jinou chybu v zadání svého názvu. Proto opět i tyto data seskupíme do skupin a budeme brát v potaz celou skupinu, čímž se pokusíme vyřešit tyto neduhy.

Pro automatické vytvoření clusterů využijeme nástroj Power BI. Z něj si můžeme exportovat tabulku vztahů názvů a clusterů. Ta se bude dát jednoduše použít v našem nástroji.

### Tvorba nového způsobu výpočtu

Po prozkoumání vlastností dat se pokusíme vytvořit vlastní jednoduchý nástroj na predikci nákladů. Nástroj bude implementován ve formě třívrstvé aplikace. Způsob výpočtu bude vycházet z upravené metody PERT pro softwarové náklady.

Sazba bude do výpočtu zahrnuta jenom teoreticky, firma si dané informace nepřeje sdělit. Většinou se používá průměrná, tudíž pokud nám náš výpočet vrátí počet normohodin, bude zbytek procesu už bezproblémový.

Vzorec:

$$CN = PT * VP * Sazba$$

Vysvětlivky:

VP = vynaložená práce

CN = celkové náklady

PT = počet tiketů

Sazba = sazba na hodinu práce

Pro získání základního PT a VP využijeme upravenou metodu PERT.

$$PT = \frac{\frac{OPT_{pt} + 4 * REAL_{pt} + PES_{pt}}{6} + \text{Celkový průměr z minulosti PT}}{2}$$

$$VP = \frac{\frac{OPT_{vp} + 4 * REAL_{vp} + PES_{vp}}{6} + \text{Celkový průměr z minulosti VP}}{2}$$

Pro nalezení jednotlivých položek v modelu PERT využijeme našeho zpracovaného datového souboru. Pokud budeme hledat například čas řešení tiketu, vytvoříme podsoubor, který bude tvořen jen informacemi relevantními k danému problému. Pokud se tedy ukáže, že čas řešení



tiketů je ovlivňován produktem a zákazníkem a tvoříme predikci k produktu1 a zákazníkovi1, vytvoříme zkombinovaný podsoubor obsahující všechny možné časy clusterů těchto dvou subjektů.

Následně hledáme jeho tři základní části. Optimistickou predikci nahradíme nalezením nejlepšího času v souboru. Pesimistickou provedeme stejně, jenom nyní budeme hledat na opačném konci souboru. Realistickou provedeme pomocí nalezení mediánu.

Celkový průměr z minulosti vypočteme následovně: Vytvoříme si pro každou proměnnou, která ovlivňuje výpočet, průměr z jejího datového výběru. Následně tyto průměry sečteme a vydělíme počtem proměnných. Je to způsobeno tím, že například v jednom světovém regionu je mnohem méně zemí, než je v jednom clusteru zákazníků. Vytvořili bychom vlastně jen průměr pro cluster zákazníků s minimálním vlivem druhého vstupu.

## **Popis technologie použité k tvorbě aplikace a popis základního testování**

### **Technologie**

Jak již bylo řečeno, výpočet bude implementován jako lokální webová aplikace. Aplikace bude tvořena webovými technologiemi. Bude se jednat o aplikaci tvořenou ze tří vrstev. Prezenční bude složena z HTML a CSS. Aplikační vrstva bude tvořena PHP a Javascriptem. Datová vrstva je MYSQL databáze obsahující zpracovaná data.

Při popisu použitých technologií, začneme těmi nejsnazšími na uchopení, jež jsou HTML a CSS. Nejčastěji tvoří již zmíněnou prezenční vrstvu webových aplikací, kdy toto tvrzení platí i pro náš nástroj.

HTML, celým názvem Hypertext Markup Language, je nejpoužívanější jazyk na vytvoření rozložení webových stránek. Funguje na bázi párových tagů. Těmito základními tagy jsou hlavička a tělo, tedy tagy „<head>“ „<body>“. Vzhled HTML nejvíce určuje CSS (jakpsatweb,2017).

CSS, neboli kaskádové styly, jsou od roku 2015 nástrojem tvořící vzhled webu. Dříve se stylizování vytvářelo jen pomocí HTML, ale nyní je stylizování téměř plně v rukou této technologie. CSS můžeme zapisovat do sešitu, který si v našem hypertextovém dokumentu

načteme, nebo ho můžeme psát přímo do jednotlivých tagů díky příkazu „style“. Základní stylizování je velmi snadné, pokud chceme například červenou barvu, stačí použít příkaz „color: red;“ (jakpsatweb,2017).

Druhá vrstva stručně popsána v této kapitole, bude vrstva aplikační. Zde používané technologie nejsou tak snadné jako ty z vrstvy prezenční. Procesy probíhající v aplikační vrstvě mohou probíhat buď na počítači uživatele, jenž se k nám připojil, nebo na serveru kde je web umístěn. Kdy první z případů je typický pro Javascript a druhý pro PHP. Technologii Javascript ale můžeme použít i na straně serveru (FreeCodeCamp,2020).

V práci bude Javascript zastoupen minimálně. Nutnost jeho existence je podmíněna použitím souboru šablon Bootstrap, které nám ulehčí tvorbu vzhledu aplikace. Zde bude obsažen, aby nám zajistil, že různé interaktivní grafické prvky správně fungují.

Všechny podstatné výpočty a zároveň zpracování dat z databáze nám bude zajišťovat jazyk PHP. Samotný název tohoto jazyka je rekurzivní. Znamená to, že první „P“ v jeho názvu znamená PHP. Celý název je tedy "PHP: Hypertext Preprocessor". Kód má formu „`<?php echo („<p>Hello World </p>“); ?>`“. Zde uvedená ukázka kódu vytvoří HTML paragraf s textem „Hello World“.

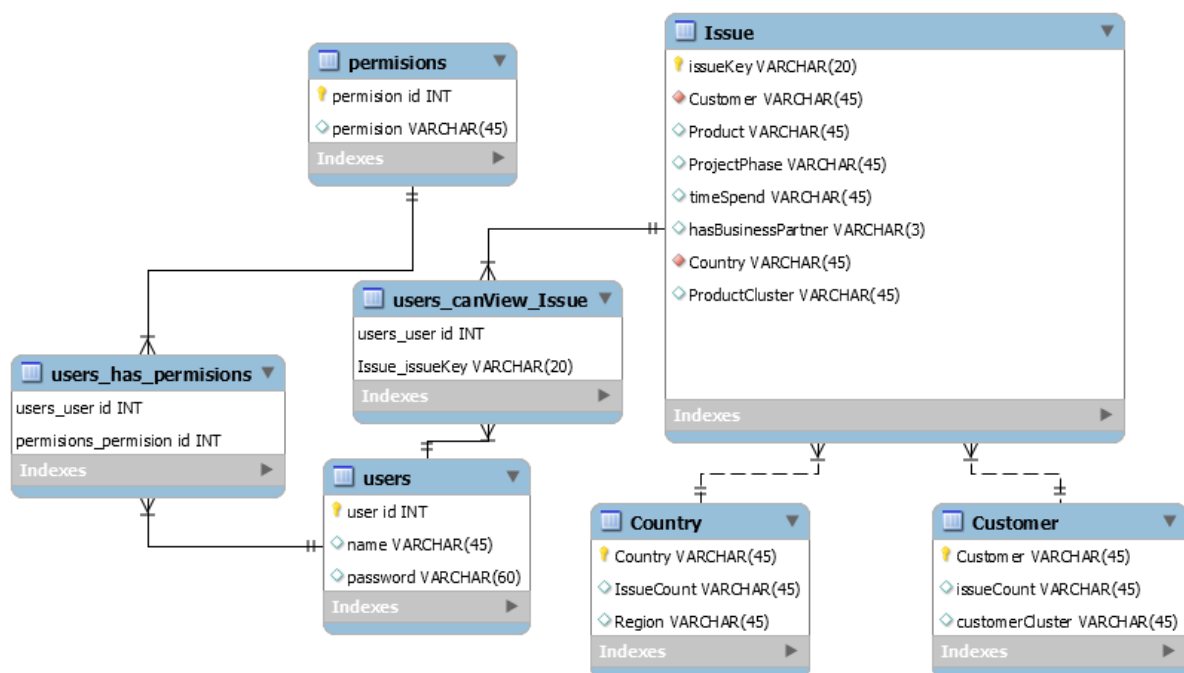
V PHP vytvoříme třídu, která bude zastávat funkci PERTu. Tato funkce bude brát jako argumenty všechny zpracované pole dat. Poté z nich vytvoří pomocí vnitřních funkcí naše požadované výsledky. Tvoří je čas na vytvoření jednoho tiketu pro danou kombinaci, počet tiketů pro danou kombinaci a celkový normočas nutný k podpoře dané kombinace. Ty program vypíše a porovná s jinými hodnotami, jako jsou například celkové průměrné hodnoty.

Velmi důležité bude také použití knihovny MySQLi, což umožní získávat nezpracovaná data z naší databáze. Umožňuje používat jazyk SQL přímo v PHP. (PHP manual,2020).

Datovou vrstvu stvoříme z MySQL databáze, na které budeme pokládat SQL dotazy. SQL je dotazovací jazyk, jenž nám umožňuje, abychom získávali data pomocí příkazů z naší databáze. Příkazy budeme tvořit jako proměnné v PHP. Ty nám poté vrátí vyžadovaná data. SQL sice povoluje různé datové typy, my ale budeme pracovat s textovým typem. Ten si sami zpracujeme až uvnitř aplikace. S SQL se dají data samozřejmě i vkládat (W3SCHOOLS,2020).

Naše databáze bude vygenerována z ERA modelu. Tento ERA model byl vypracován až po zpracování dat, tudíž už obsahuje připravené tabulky. K tvorbě modelu byl použit MySQL workbench. Model obsahuje tabulky vytvořené tak, abychom z nich pomocí SQL dotazů mohli získat potřebná data k vytvoření výpočtu podle již zmíněného vzorce. Tento vzorec již bude implementován pomocí PHP jako funkce a budeme do něj jen dosazovat data.

Obrázek 5. ERA model aplikace



Zdroj: *Vlastní tvorba, 2020 (Vytvořeno pomocí MySQL Workbench)*

Era model obsahuje i tabulky určené spíše k zabezpečení, jako jsou users (uživatelé) a jejich různé permissions (práva). Také je připravena M:N vazba k vytvoření možnosti zobrazení jenom určitých tiketů daným uživatelem. Práva nebudou přiřazována v aplikaci, ale budou určována správcem databáze.

Dále jsou zde vytvořeny tabulky pro jednotlivé vstupující proměnné. Issue (hlavní tabulka pro tikety) nám umožní získat data o projektové fázi a o produktu. Údaje o počtech tiketů pro danou zemi nebo pro daného zákazníka, jsou ve vlastních tabulkách. Data jsou uložena tímto způsobem z důvodu jednoduchého importu souborů CSV, vytvořený po získání a transformaci dat.

K vývoji budou využity následující základní nástroje:

PhpStorm – Ke psaní kódu využijeme tento program, jeho licence je bezplatná pro studenty, čehož využijeme při naší tvorbě. I přes název programu, jenž napovídá možnost práce jen s PHP, ho můžeme využít k editaci libovolného jazyka týkající se webových technologií.

XAMPP – Pro simulaci databáze na lokálním počítači spolu s interpretací PHP využijeme tento program. Tato aplikace nám umožní si u sebe lokálně vytvořit vlastní server, což umožňuje všechny funkce očekávané od takového serveru (ApacheFriends, 2020).

MySQL Workbench – Využijeme ho k vytvoření ERA modelu a následné vygenerování kódu pro vytvoření databáze.

### **Testování použitelnosti**

Pro testování byl zvolen způsob výběru několika respondentů, kteří budou požádáni, aby v aplikaci vykonali několik úkonů. Pro základní otestování budou vhodné alespoň tři respondenti. Poté sdělí svůj názor na grafické rozhraní, a zdali se jim povedl daný proces vykonat. Bude se jednat o několik jednoduchých úkonů. Těmito úkony budou:

1. Vybrat libovolnou konfiguraci ze seznamu a předložit hodnoty programu.
2. Zkusit se přihlásit se špatnými uživatelskými údaji.
3. Vybrat zemi Egypt s produktem CMD V4.
4. Přihlásit se správnými údaji.

Aplikace bude dána k otestování na autorově notebooku. Bude se jednat jen o pár respondentů ke zjištění základního názoru na použitelnost uživatelského rozhraní této aplikace.

## VII. Získané výsledky a tvorba nástroje na predikci nákladů

### Finální stav dat

Výsledný skript do systému JIRA musel být rozdělen na třináct částí. Položky si stáhneme ve formátu CSV, kdy oddělovačem bude středník. To nám umožní dále s nimi pracovat v Excelu. Zadaná data je nutné ještě zkombinovat do jednoho souboru a připojit k nim údaje o času z druhé tabulky. Ke kombinaci byl použit program POWER BI, konkrétně jeho query editor. Unikátní klíč použit ke kombinaci je klíč daného tiketu.

Pro usnadnění práce se souborem dat odstraníme nadbytečná data. Jedná se o různé komentáře a poznámky vývojářů v systému JIRA, která nejsou pro naši práci relevantní. Tyto data odstraníme nejlehčím možným způsobem a to manuálním odstraněním sloupců tabulek v programu Excel.

Obrázek 6. Ukázka formy dat

2 Project Phase	3 Čas na tiket	4 Product
Production	10,75	ProFlex4/DDC
Production	1,93333333	ProFlex4/DDC
Production	1,31666667	Vynamic Security Server
Production	2,48333333	ProCash/NDC-DDC
Certification	2,48333333	ProCash/NDC-DDC
Production	4,3	ProTopas/WebExtensions
Certification	1,91666667	ProCash/NDC-DDC
Certification	5,05	ProCash/NDC-DDC
Production	3,58333333	ProView/Server
Internal QA	2,31666667	ProCash/NDC-DDC

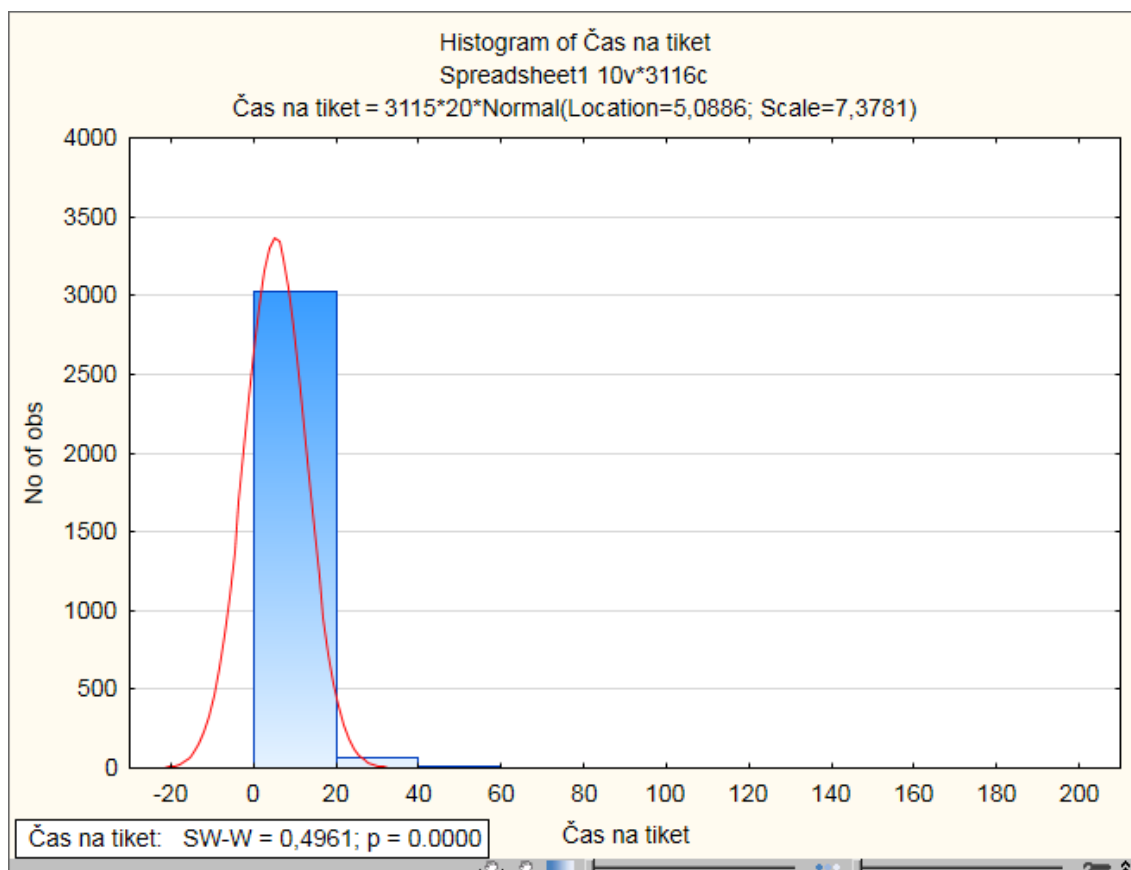
Zdroj: *Vlastní tvorba, 2020*

Po otestování vlastností jednotlivých regionů ve formě proměnné REGION, se ukazuje problém s regionem EMEA. Tento region má velmi velkou odchylku počtu tiketů i času na vyřešení tiketu. Obsahuje velké množství různorodých zemí a zároveň ve srovnání s ostatními zahrnuje velkou plochu. Afrika, Střední východ a Evropa proto dostanou vlastní regiony ve formě A, ME a E. Díky tomu vytvoříme novou transformovanou proměnnou REGION-2.

Proměnné týkající se počtu tiketů byly testovány vždy v ohledu na počet produktů, jenž jest pro zákazníka provozován. Pokud tedy má zákazník dva tikety a dva produkty, má zákazník jeden tiket na produkt.

Provedený test normality času na vyřešení nám nebyl schopný zamítnout nulovou hypotézu díky p-hodnotě, jenž se pohybuje v kruhovém okolí nuly. Možnost ANOVY je tímto vyloučena a data musí být testována pomocí Kruskal Wallisova testu, případně Mann Whitneyho testu v případě testování BUSINESS PARTNER.

Obrázek 7. Výsledky testování normality dat pro čas



Zdroj: *Vlastní tvorba, 2020(Vytvořeno pomocí SW Statistica)*

Testování zdali mají skupiny shodné rozptyly pomocí Levenova testu, dopadlo stejně jako test normality. Použití ANOVY už bylo zamítnuto díky neexistující normalitě a tento test jenom potvrdil použití Kruskal Wallisova testu. Příkladem mohou být testování s proměnnými regionu a projektové fáze, které vyšly s p-hodnotami 0,0094 a 0,000010. Testování rozptylu je ale pro výběr testu již nerelevantní.

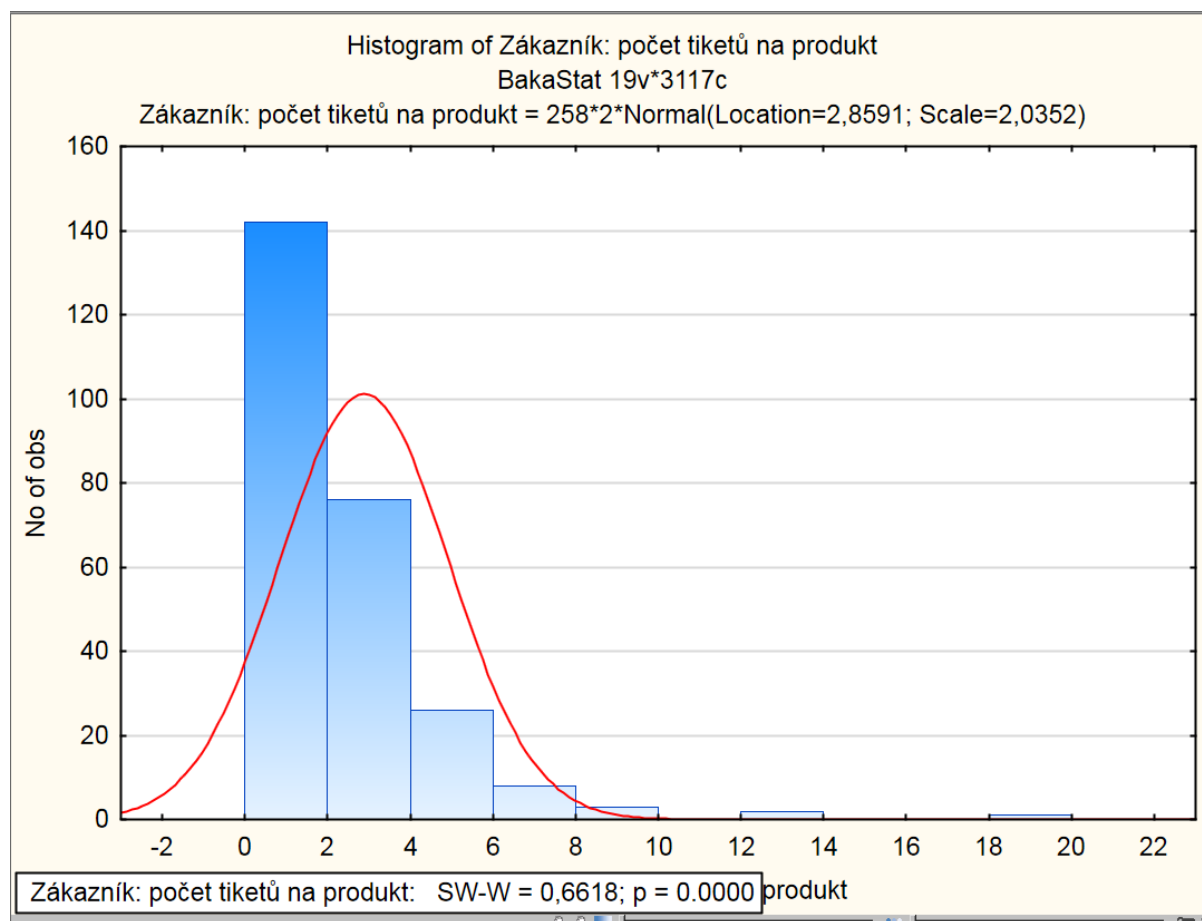
Tabulka 2. Vliv na čas

<b>Testování vlivu proměnných na čas řešení tiketu</b>		
Proměnná	p-hodnota testu	Zahrnout do výpočtu
Region-2	0,2223	NE
Project Phase	0,039	ANO
Product	0,00	ANO
Business Partner	0,5991	NE

Zdroj: Vlastní tvorba, 2020

Testování normality pro data týkající se počtu tiketů dopadla obdobně jako testování dat času. Zde je ukázka testování počtu tiketů u jednotlivých zákazníků. Ostatní proměnné spojené s počtem jsou ve stejné situaci a jejich p-hodnota je v kruhovém okolí nuly.

Obrázek 8. Výsledky testování normality dat pro počet tiketů



Zdroj: Vlastní tvorba, 2020 (Vytvořeno pomocí SW Statistica)

U počtů tiketů testování dopadlo téměř stejně jako v předchozím případě. U zkoumaných dat můžeme na dané hladině tedy říct, že jednotlivé skupiny mají rozdílné rozptyly. Opět může být příkladem testování projektové fáze a zákazníka, kdy p-hodnoty jsou 0,0062 a 0,00. Objevila se zde výjimka ve formě regionu. Ten má shodné rozptyly s p-hodnotou 0,194. Testování není i zde z důvodu neexistující normality příliš relevantní. Dle výsledků testování znovu použijeme Kruscall Wallise a Mann Whitneyho.

Tabulka 3. Vliv na počet

<b>Testování vlivu proměnných na počet tiketů</b>		
Proměnná	p-hodnota testu	Zahrnout do výpočtu
Region-2	0,0394	ANO
Business Partner	0,688921	NE
Customer	0,00	ANO
Project Phase	0,2884	NE

Zdroj: *Vlastní tvorba, 2020*

V programu POWER BI si tedy vytvoříme clustery, exportujeme a propojíme je s naším datasetem pomocí Excelu. Pro propojení použijeme funkci „SVyhledat()“. Ta nám umožní najít ve vybrané tabulce hodnotu patřící k jiné hledané hodnotě. Můžeme tedy zadat název produktu „A“ a pomocí této funkce najít do jakého clusteru patří a zároveň ho vypsát. Na obrázku jsou uvedeny clustery pro produkty jako ukázka procesu.

V analýze clusterů se u zákazníků vyskytla již zmíněná chyba s překlepy. Všechny chybné zadání se seskupili do jednoho clusteru, konkrétně do clusteru 2. Je pro ně typické přesně jeden tiket na produkt za celé období. V programu tedy budeme tento cluster vyřazovat z důležitých výpočtů. Zároveň vytvoříme upozornění, že pokud se tento jev vyskytne ve výpočtu, výsledky mohou být zkreslené.

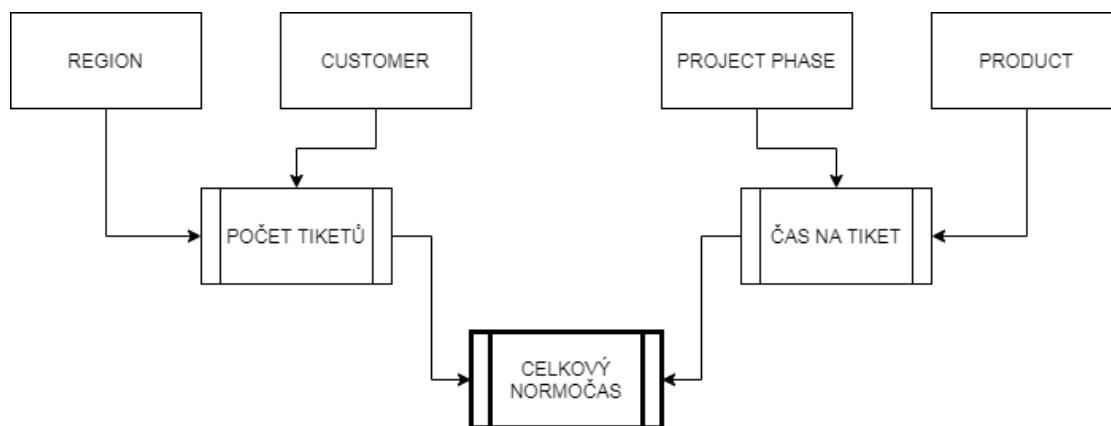


Obrázek 9. Clustery produktů

Custom field (Product)	Spent Time Number	Custom field (Product) (clustery)
278 ECR	0,77	Cluster8
7750 ATM	4,82	Cluster8
ActivMedia - MMA - CCDM	7,95	Cluster8
AFD 2.0	12,90	Cluster2
Agilis 91X SW	25,18	Cluster2
Agilis Empower (NSL)	1,42	Cluster8
Agilis XFS SW	144,25	Cluster4
BA80-TFT Display	0,25	Cluster8
Basic Endpoint Security	35,85	Cluster10
CCDM V1	1,63	Cluster8
CCDM V2	0,92	Cluster8
CINEO C1010	3,22	Cluster8
CINEO C2040 FL	2,68	Cluster8
CINEO C2060 Frontload	23,53	Cluster2

Zdroj: *Vlastní tvorba, 2020(Vytvořeno pomocí PowerBI)*

Obrázek 10. Vztahy proměnných



Zdroj: *Vlastní tvorba, 2020(Vytvořeno pomocí Diagram Drawio)*

Nástroj tedy bude uvažovat čtyři základní proměnné vstupu, po jejich zadání je transformuje do následující podoby.

Region (Country) – do jakého regionu patří země, v níž bude produkt implementován

Customer – zde využijeme cluster námi zvoleného zákazníka a budeme pracovat s ním

Project Phase – jednoduše zadáme fázi, ve které se daný rok bude produkt nacházet

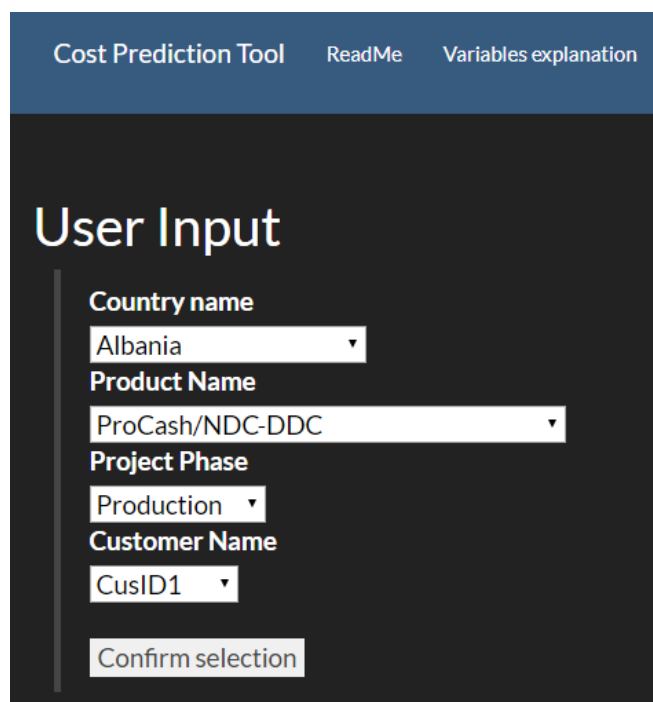
Product – vybereme cluster, jenž má stejné vlastnosti jako náš produkt

Kdy počet tiketů bude tvořen kombinací regionu a zákazníka. Čas na tiket bude tvořit projektová fáze a produkt. Jak již bylo zmíněno, aby se dosáhlo celkových nákladů, potřebovali bychom průměrnou hodinovou sazbu. Firma se ale danou informací střeží a tudíž jí nyní nebudeme brát v potaz a naším výstupem bude celkový normočas na podporu.

### Vytvoření aplikace k predikci nákladů

Nejdříve si pomocí zmíněných technologií HTML, CSS a Javascript vytvoříme vzhled našeho nástroje. Použijeme knihovnu šablon Bootstrap jako základ, konkrétně její podobu Darkly. Pod tímto odstavcem je ukázka vzhledu jak ve formě grafické, tak ve formě kódu. Nástroj je v angličtině z důvodu mezinárodní povahy společnosti.

Obrázek 11. Ukázka vzhledu nástroje



The image shows a screenshot of a web application titled "Cost Prediction Tool". At the top, there are three navigation links: "Cost Prediction Tool", "ReadMe", and "Variables explanation". The main content area is titled "User Input" and contains four dropdown menus and one button. The first dropdown is labeled "Country name" and has "Albania" selected. The second is "Product Name" with "ProCash/NDC-DDC" selected. The third is "Project Phase" with "Production" selected. The fourth is "Customer Name" with "CusID1" selected. Below these is a button labeled "Confirm selection".

Zdroj: *Vlastní tvorba, 2020*

Tabulka 4. Ukázka vzhledu ve formě kódu

```
<label class="control-label" for="CountryName">Country name</label>
</br>
<select style="color:black" id="CountryName" name="CountryName">...</select>
</br>
```

Zdroj: *Vlastní tvorba, 2020*

Dalším krokem v naší tvorbě bude vytvoření databáze, což nám umožní dodat naší aplikaci dostatečné množství dat k jejím výpočtům. Kód pro tvorbu databáze si necháme vygenerovat aplikací MySQL workbench. Bude založen na ERA modelu ukázaném v minulé kapitole.

Námi vytvořený kód bude mít následující formu:

```
„CREATE TABLE IF NOT EXISTS `mydb`.`permissions` (
  `permission id` INT NOT NULL,
  `permission` VARCHAR(45) NULL,
  PRIMARY KEY (`permission id`));“
```

Databázi naplníme daty ve formě CSV. Data, u kterých jsme zjišťovali cluster, budou CSV oddělené čárkou. Zbylá data budou CSV oddělena středníkem.

Obrázek 12. Uložení dat

278 ECR,0.7666666666666666,Cluster8
7750 ATM,4.8166666666666666,Cluster8
ActivMedia - MMA - CCDM,7.95,Cluster8

Zdroj: *Vlastní tvorba, 2020*

Do naší aplikace dostaneme data pomocí kombinace SQL a PHP. Nejdříve vytvoříme připojení k naší aplikaci pomocí PHP knihovny MySQLi. Příkaz pro naše připojení bude mít následující formu.

```
„$conn = new mysqli($servername, $username, $password, $dbname);“
```

Pro připojení k databázi už můžeme použít SQL k tvoření dotazů na získání potřebných materiálů k našemu výpočtu. Příkladem takových příkazů mohou být například tyto:

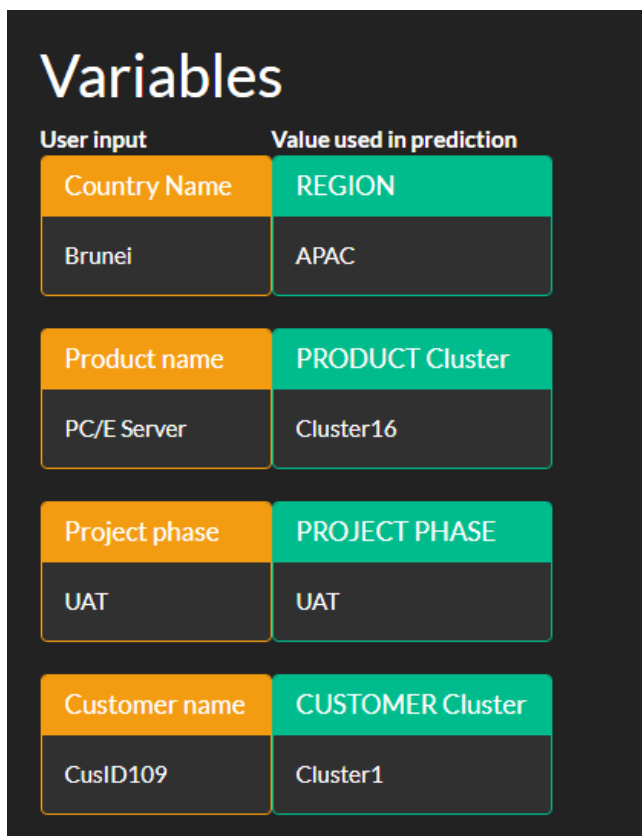
```
„SELECT DISTINCT productcluster FROM issue WHERE product = 'navez'“
```

```
„SELECT timespend FROM issue WHERE productcluster = 'cluster'“
```

Tento příkaz nejdříve nalezne cluster produktu podle jeho názvu. Pomocí „DISTINCT“ zajistíme, že získáme jenom jednu položku, tedy název jednoho clusteru. Cluster si v PHP upravíme dle potřeby a dosadíme ho do nového příkazu, jenž nám vytáhne z databáze všechny časy pro daný cluster. Data následně zpracujeme, aby se s nimi dalo pracovat v dalších operacích. Účel práce není podrobné osvětlení práce s PHP, tudíž popis této činnosti nebude zahrnut.

Výsledky daného procesu nám aplikace pro kontrolu vypíše. Uvažuje se, že s aplikací bude pracovat uživatel kompetentní s informační technologií. Proto je vhodné napsat názvy, tak jak jsou v databázi, aby si případně o nich mohl dohledat dodatečné informace podle potřeby.

Obrázek 13. Vztah vstupů s hodnotami použitých v predikci



User input	Value used in prediction
Country Name	REGION
Brunei	APAC
Product name	PRODUCT Cluster
PC/E Server	Cluster16
Project phase	PROJECT PHASE
UAT	UAT
Customer name	CUSTOMER Cluster
CusID109	Cluster1

Zdroj: *Vlastní tvorba, 2020*

Obdobným způsobem jako v předchozím kroku získáme všechny data pro naše čtyři proměnné ve formě zpracovaných datových polí.

Funkce nám vrátí hodnoty sloužící jako predikce budoucnosti pomocí upravené metody PERT. Její fungování již bylo popsáno v návrhu našeho výpočtu. Nyní si popíšeme její implementaci.

Z předchozího zpracování dat budeme mít čtyři datové pole. Prvním z nich je počet tiketů pro daný region. Dalším je délky všech času na tiket z daného produktového clusteru spolu se všemi časy pro projektovou fázi. Posledním prvkem bude počty pro cluster, do kterého řadíme aktuálně zvoleného zákazníka.

Nikdy nemůžeme mít jistotu, že program vše provedl správně a proto datové pole ještě naposledy ověříme funkcí. Ta ověří, zdali jsou všechny pole větší než nula. Pokud je splněna tato podmínka, můžeme pokračovat k samotnému výpočtu.

Tabulka 5. Funkce pro ověření

```
function testArrays($timeSpendProduct, $timeSpendPhase, $issueCountRegion,
$issueCountCustomerGroup)
{
    if (count($timeSpendProduct) && count($timeSpendPhase) &&
count($issueCountRegion) && count($issueCountCustomerGroup)) {
        return true;
    } else {
        return false;
    }
}
```

Zdroj: *Vlastní tvorba, 2020*

Po ověření deklarujeme třídu, jejímž úkolem bude celý výpočet. Jako argumenty dáme konstruktoru této třídy naše čtyři pole. Po vytvoření zavoláme dvě hlavní funkce třídy, těmi jsou `getTicketTime()` a `getTicketCount()`. Obě tyto funkce fungují obdobně. Vezmou si dvě relevantní datová pole, spojí je dohromady a poté je seřadí. Zároveň pomocí funkce `returnPertVals()` získají hodnoty do PERTu.

Poté z datových polí získají nejmenší a největší hodnoty. Dalším procesem bude nalezení mediánu. Tyto tři nově vzniklé hodnoty jen dosadíme do základního PERT. Takto získanou hodnotu ještě zkombinuje s vytvořeným celkovým průměrem počtu nebo času pro danou skupinu. Poté jednoduše vydělí dvěma a získáme čas na jeden tiket, nebo roční počet tiketů. Funkce pro čas je pro ukázkou uvedena dále. Vynásobením těchto dvou hodnot získáme celkový normočas na softwarovou podporu.

Tabulka 6. Funkce pro čas na tiket

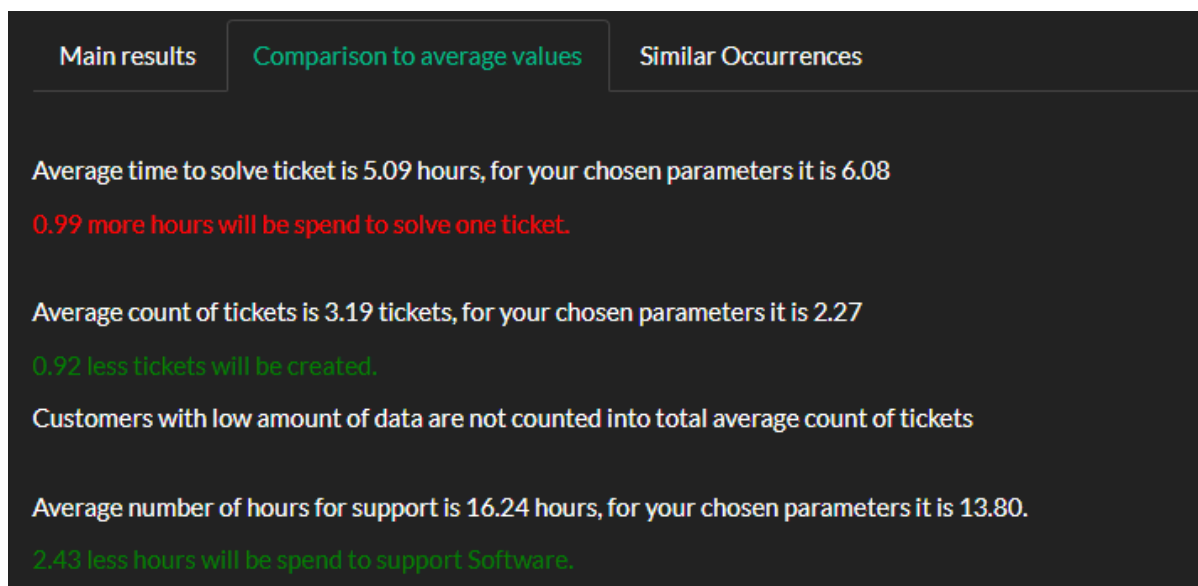
```
public function getTicketTime()
{
    $merged = array_merge($this->timeSpendPhase, $this->timeSpendProduct);
    $pertArr = $this->returnPertVals($merged);
    $output = ($pertArr["opt"] + (4 * $pertArr["real"]) + $pertArr["pes"]) / 6;

    $average = (returnAvg($this->timeSpendProduct) + returnAvg($this->timeSpendPhase))/2;
    $output = ($output + $average) / 2;
    return $output;
}
```

Zdroj: *Vlastní tvorba, 2020*

Finální výstup tedy bude tvořen třemi hlavními výsledky bez jakéhokoliv srovnání. Dále zahrneme možnost pro přepnutí na srovnání s celkovými průměrnými hodnotami. Do celkového průměru počtu hodnot nebudeme počítat zákazníky z druhého clusteru. Jejich nevhodné vlastnosti už byly zmíněny dříve. Třetí podobou výstupu bude srovnání s obdobnými případy. Bude se tedy jednat o časy pro stejný cluster produktů a stejnou projektovou fázi. Dále půjde o počty tiketů pro clustery zákazníků a regiony.

Obrázek 14. Ukázka výstupu



Zdroj: *Vlastní tvorba, 2020*

Tabulka 7. Ukázka kódu pro výstup

```
$differenceTime = $timeRes - $issueTotalAverageTime;  
$differenceTimeOut = number_format((float)$differenceTime, 2, '.', '');  
if ($differenceTime > 0) {  
    echo("<p style='color: red'>" . $differenceTimeOut . " more hours will be spend  
to solve one ticket. </p>");  
} else {  
    echo("<p style='color: green'>" . ($differenceTimeOut * -1) . " less hours will  
be spend to solve one ticket. </p>");  
}
```

Zdroj: *Vlastní tvorba, 2020*

Všechny hodnoty jsou vybírány z dat z celé databáze. Umožní to i pozdější využití nástroje, kdy při přidání dalších dat ve stejném formátu je nástroj bude brát automaticky v potaz. Příkladem by mohlo být využití v roce 2021, zatímco se do nástroje přidalo i množství dat z roku 2020. Data za dva roky by zajisté zvýšila přesnost našich výpočtů.

## Aplikace na Cloudu

Pro správnou funkčnost aplikace je potřeba její hostování na serveru nebo využití utility XAMPP. Je to z důvodu vlastností PHP. Aplikace bude v případě použití na intranetu společnosti, nyní je nutné následující řešení:

1. Stáhnout si program XAMPP z webu společnosti:  
<https://www.apachefriends.org/index.html>
2. Stáhnout si aplikaci z Google Drive (Nezáleží na typu komprimace):  
<https://drive.google.com/drive/folders/1pG-jifilsERDScOCAKITKDCITKYiKqSg?usp=sharing>
3. Nainstalovat nástroj XAMPP
4. Spustit nástroj a kliknout na tlačítko Explorer, to nám otevře systémový prohlížeč, zde najít složku „htdocs“ a přesunout do ní složku „BPCLOUD“ z komprimovaného souboru staženého z Google Disku
5. Vrátit se do rozhraní nástroje a stisknout na tlačítka start u prvních dvou modulů Apache a MySQL
5. Jít do preferovaného internetového prohlížeče a zadat zde localhost/BPCLOUD do políčka do kterého bychom normálně zadávali url adresu

## Zjištění využitelnosti aplikace pomocí dotazníku

Respondentům byla aplikace dána k otestování osobně, spolu se zadanými úkoly, které měli splnit. Později se měli vyjádřit k použitelnosti této aplikace z hlediska designu a případně dát zpětnou vazbu za účelem změny aplikace. Design práce testovali 4 respondenti.

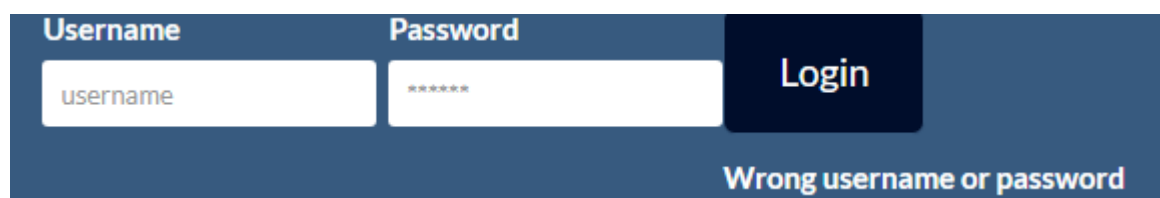
Jak již bylo řečeno, úkoly byly tvořeny těmito úkony:

1. Vybrat libovolnou konfiguraci ze seznamu a předložit hodnoty programu.
2. Zkusit se přihlásit se špatnými uživatelskými údaji.
3. Vybrat zemi Egypt s produktem CMD V4.
4. Přihlásit se správnými údaji.



Respondent 1 – Existence vstupů a výstupů na jedné stránce, kdy možnost nového vstupu na stránce už proběhlé analýzy zmátla respondenta, dále kritizoval chybějící chybovou hlášku při přihlašování. Žádné jiné problémy z hlediska funkčnosti respondent neshledal a byl schopen splnit všechny zadané úkoly. Autor se zde zadanou kritikou souhlasí a aplikaci rozdělil na rozdílné stránky pro vstupy a výstupy s možností vrácení se stránky výstupů na stránku vstupu. Také je přidána chybová hláška při zadání špatných údajů.

Obrázek 15. Přidání chybové hlášky

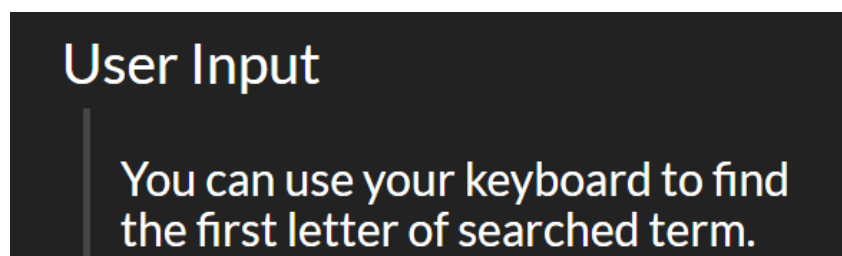


Zdroj: *Vlastní tvorba, 2020*

Respondent 2 – Splnil všechny úkoly bez jakýchkoliv problémů. Kritizoval nutnost přihlášení, ale jedná se o vlastnost programu, tudíž není nutné opravovat.

Respondent 3 – Respondent splnil všechny úkoly bez problému. Opět se vyskytla kritika chybějící chybové hlášky stejně jako u předchozího. Dále byla kritika selektů, kdy respondentovi chybělo upozornění, že může zadat první písmeno slova pro urychlení hledání. Výtka byla opět vzata v potaz a aplikace změněna.

Obrázek 16. Přidání nápovědy



Zdroj: *Vlastní tvorba, 2020*

Respondent 4 – Respondent měl problém s nalezením požadovaných položek v seznamu. Jinak s asistencí byl schopný splnit všechny úkoly. Neprojevilo žádnou kritiku.

## Závěr

---

Dlouhodobá podpora softwaru je s mírou jeho využití v dnešní informační společnosti poměrně významnou otázkou. Efektivním způsobem řízení této činnosti ať už za účelem minimalizace nákladů nebo vytvoření co největší hodnoty pro zákazníka se budou muset zabývat IT společnosti v nadcházejícím období stále více.

Dílčí cíle této práce byly právě popsáním této činnosti, způsobem jejího řízení a poté konkrétním příkladem ve formě společnosti Diebold Nixdorf. Problematikou se zabývá malé množství českých autorů, tudíž bylo nutné velké využití zahraničních zdrojů ať už ve formě literatury nebo internetových článků. Velmi cennými byly materiály od společnosti Atlassian, jež jsou bohužel napsány velmi korporátním jazykem.

Hlavním cílem této práce bylo návrh a implementace nástroje sloužící k predikci nákladů na softwarovou podporu. Autorovi byla poskytnuta data za rok 2019. Rozsah těchto dat byl značný, ale i přesto by především pro některé zkoumané lokality bylo potřeba větší množství dat pro přesnější určení všech klíčových faktorů.

Cíl práce byl splněn, nástroj byl vytvořen a nyní záleží na společnosti Diebold Nixdorf v jaké míře se ho rozhodne použít. Data vyskytující se uvnitř datové vrstvy této aplikace jsou značně zcenzurována. Například nejsou uvedena jména zákazníků, ale jejich pořadí v celkovém seznamu zákazníků. V celkovém výpočtu nejsou uvedeny sazby, firma se je také nepřála sdělit. Celý výpočet je implantován tak, aby se tyto položky daly do nástroje jednoduše zařadit.

Problematika má jistě velký potenciál ke zkoumání do budoucna. Ať už z pohledu analýzy nákladů, kde by se jistě dali použít moderní techniky například ve formě strojové učení. Nebo z pohledu druhého, jímž je podrobné zkoumání vedení informačního týmu z pohledu managementu IT služeb, nebo jakékoliv odrůdy tohoto povolání.

## Seznam obrázků a tabulek

---

Obrázek 1.	Podíl nákladů na údržbu softwaru na celkových nákladech v letech.....	10
Obrázek 2.	Dimenze ITSM.....	17
Obrázek 3.	Aktuální návrh výpočtu nákladů na podporu.....	29
Obrázek 4.	Proměnné ve výpočtu.....	30
Obrázek 5.	ERA model aplikace .....	43
Obrázek 6.	Ukázka formy dat.....	45
Obrázek 7.	Výsledky testování normality dat pro čas.....	46
Obrázek 8.	Výsledky testování normality dat pro počet tiketů .....	47
Obrázek 9.	Clustery produktů.....	49
Obrázek 10.	Vztahy proměnných .....	49
Obrázek 11.	Ukázka vzhledu nástroje .....	50
Obrázek 12.	Uložení dat .....	51
Obrázek 13.	Vztah vstupů s hodnotami použitých v predikci .....	52
Obrázek 14.	Ukázka výstupu .....	55
Obrázek 15.	Přidání chybové hlášky .....	57
Obrázek 16.	Přidání nápovědy.....	57
Tabulka 1.	Koeficienty COCOMO .....	25
Tabulka 2.	Vliv na čas.....	47
Tabulka 3.	Vliv na počet.....	48
Tabulka 4.	Ukázka vzhledu ve formě kódu .....	51
Tabulka 5.	Funkce pro ověření .....	53
Tabulka 6.	Funkce pro čas na tiket .....	54
Tabulka 7.	Ukázka kódu pro výstup .....	55

## Seznam použitých zkratk

---

IT – Informační Technologie

ITIL – Information Technology Infrastructure Library – Knihovna pojmů týkajících se softwarové podpory obsahující nejlepší praktiky

ITSM – Information Technology Service Management- Management řízení IT služeb

SLM – service level management – management, který se zabývá službami a budováním jejich infrastruktury

LEAN – manažerská filozofie snažící se omezit plýtvání

AGILE – iterační způsob řízení projektů

WORKAROUND – dočasné řešení, pojem využívaný především v softwarovém inženýrství

PERT – Program Evaluation and Review Technique, metoda vytvořená pro síťovou analýzu

COCOMO – matematický model sloužící k odhadu vynaloženého úsilí na aktivitu v softwarovém inženýrství

# Zdroje

---

## **Literatura**

ADDY, Rob. *Effective IT service management: to ITIL and beyond!*. New York: Springer, c2007. ISBN 978-3-540-73197-9.

ITIL Foundation: *ITIL 4 Edition*. TSO, 2019. ISBN 9780113316076.

PLEVNÝ Miroslav, ŽIŽKA Miroslav. *Modelování a optimalizace v manažerském rozhodování*. Plzeň: Západočeská univerzita, 2010. ISBN 978-80-7043-933-3.

SKALICKÝ Jiří, JERMÁŘ Milan, SVOBODA Jaroslav. *Projektový management a potřebné kompetence*. Plzeň: Západočeská univerzita, 2010. ISBN 978-80-7043-975-3.

Hendl, J. *Přehled statistických metod zpracování dat : analýza a metaanalýza dat*. Praha: Portál, 2015. ISBN 978-80-262-0981-2

SVOBODA Milan, GANGUR Mikuláš, MIČUDOVÁ Kateřina. *Statistické zpracování dat*. Plzeň: Západočeská univerzita, 2019. ISBN 978-80-261-0883-2.

SOUČEK, Eduard. *Statistika pro ekonomy*. Praha: Vysoká škola ekonomie a managementu, 2006. ISBN 80-86730-06-9.

## **Vědecké články**

Marounek P. Simplified approach to effort estimation in software maintenance, Vysoká škola ekonomická, Praha, Fakulta informatiky a statistiky. Journal of systems integration. 2012. Dostupné z: <http://www.si-journal.org/index.php/JSI/article/view/123>

ZÍMA Martin. SQL 92. Plzeň: Západočeská univerzita, Katedra informatiky a výpočetní techniky, 1992 Dostupné z: <https://www.kiv.zcu.cz/studies/predmety/db1/sql-1992.html>

Which Factors Affect Software Projects Maintenance Cost More?. National Center for Biotechnology Information [online].,2013, Dostupné z: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3610582/>

## **Zdroje z Webu**

Apache Friends, XAMP, Dostupné z: <https://www.apachefriends.org>

Approaches and Types of Software Product Support and Maintenance Service - Hidden Brains Blog. Enterprise Web & Mobile App Development Company | IT Services & Software Solutions [online]. Copyright © 2003 [cit. 25.12.2019]. Dostupné z: <https://www.hiddenbrains.com/blog/approaches-and-types-of-software-product-support-and-maintenance-service.html>

CSS - Kaskádové styly. Jak psát web, návod na html stránky [online]. Dostupné z: <https://www.jakpsatweb.cz/css/>

Diebold Inc | Encyclopedia.com. Encyclopedia.com | Free Online Encyclopedia [online]. Copyright © 2019 [cit. 30.12.2019]. Dostupné z: <https://www.encyclopedia.com/social-sciences-and-law/economics-business-and-labor/businesses-and-occupations/diebold-inc>

Diebold Nixdorf - Bank Innovation – Retail Technology | Diebold Nixdorf. Diebold Nixdorf [online]. Dostupné z: <https://www.dieboldnixdorf.com/en-us>

Diebold Nixdorf s.r.o. , Praha IČO 25784528 - Obchodní rejstřík firem | Kurzy.cz. Obchodní rejstřík firem - vazby a vztahy z justice.cz | Kurzy.cz [online]. Copyright © 2000 [cit. 28.02.2020]. Dostupné z: <https://rejstrik-firem.kurzy.cz/25784528/diebold-nixdorf-sro/>

Diebold Nixdorf. Diebold Nixdorf [online]. Dostupné z: <http://www.tpomm.cz/AboutUs.html>

Export Query Output to Excel in Oracle SQL Developer - Data to Fish. Data to Fish: Data Science Tutorials using Python, SQL and more! [online]. Copyright © 2020 [cit. 24.02.2020]. Dostupné z: <https://datatofish.com/excel-oracle-sql-developer/>

Formát CSV - Otevřená data. Otevřená data v ČR: Portál pro poskytovatele - Otevřená data [online]. Dostupné z: <https://opendata.gov.cz/standardy:csv>

HTML příručka, přehled HTML tagů. Jak psát web, návod na html stránky [online]. Dostupné z: <https://www.jakpsatweb.cz/html/>

How to export multiple columns into individual text files in Excel?. ExtendOffice - Best Office Productivity Tools [online]. Copyright © 2009 [cit. 15.03.2020]. Dostupné z: <https://www.extendoffice.com/documents/excel/5355-export-excel-columns-to-individual-text-files.html>

ITIL's History from the Late 1980s to Today. ITIL Training 2018 [online]. Dostupné z: <https://www.italtraining.com/blog/2018/11/06/itil-history/>

ITIL® Change Management – BMC Blogs. BMC Software – Run and Reinvent [online]. Copyright © Copyright 2005 [cit. 28.12.2019]. Dostupné z: <https://www.bmc.com/blogs/itil-change-management/>

ITIL® Training and Preparation: SLM, Designing SLA Structures, and SLA Content. Online Certification Training Courses for Professionals | Simplilearn [online]. Copyright © 2009 [cit. 28.12.2019]. Dostupné z: <https://www.simplilearn.com/designing-sla-structures-sla-content-article>

ITSM (IT Service Management) - ManagementMania.com. [online]. Copyright © 2011 [cit. 27.12.2019]. Dostupné z: <https://managementmania.com/cs/it-service-management>

JQL: The most flexible way to search Jira. Atlassian | Software Development and Collaboration Tools [online]. Copyright © 2019 Atlassian [cit. 14.03.2020]. Dostupné z: <https://www.atlassian.com/blog/jira-software/jql-the-most-flexible-way-to-search-jira-14>

Jazyk SQL - Vojtěch Hordějčuk. Vojta Hordějčuk aka voho - Software Engineer and Bedroom Music Producer [online]. Copyright © 2008 [cit. 22.02.2020]. Dostupné z: <http://voho.eu/wiki/sql/>

Jira | Software pro sledování požadavků a projektů. Atlassian | Software Development and Collaboration Tools [online]. Copyright © 2020 Atlassian [cit. 14.03.2020]. Dostupné z: <https://www.atlassian.com/cs/software/jira>

Jsme Global Center Pilsen | Diebold Nixdorf. GC | Diebold Nixdorf [online]. Copyright © 2019 Diebold Nixdorf. [cit. 29.12.2019]. Dostupné z: <https://www.dnpilsen.cz/o-nas/>

PHP manual, PHP: Hypertext Preprocessor [online]. Dostupné z: <https://www.php.net/manual>

PHP vs JavaScript: How to Choose the Best Language for Your Project. Learn to code at home | freeCodeCamp.org [online]. Dostupné z: <https://www.freecodecamp.org/news/php-vs-javascript-which-technology-will-suit-your-business-better/>

SQL Tutorial. W3Schools Online Web Tutorials [online]. Dostupné z: <https://www.w3schools.com/sql/>

Software Engineering | COCOMO Model - javatpoint. Tutorials List - Javatpoint [online]. Copyright © Copyright 2011 [cit. 29.12.2019]. Dostupné z: <https://www.javatpoint.com/cocomo-model>

StatSoft – Shlukování podobných. Návod k software Statistica 2014 Dostupné z: [http://www.statsoft.cz/file1/PDF/newsletter/2014\\_10\\_08\\_StatSoft\\_Shluikovani\\_podobnych\\_v\\_softwaru\\_statistica.pdf](http://www.statsoft.cz/file1/PDF/newsletter/2014_10_08_StatSoft_Shluikovani_podobnych_v_softwaru_statistica.pdf)

Úvod do jazyka SQL. Největší český web zaměřený na .NET framework [online]. Copyright © 2020 [cit. 22.02.2020]. Dostupné z: <https://www.dotnetportal.cz/clanek/50/Uvod-do-jazyka-SQL>

What Are the Different Types of Computer Software Support?. wiseGEEK: clear answers for common questions [online]. Copyright © 2003 [cit. 26.12.2019]. Dostupné z: <https://www.wisegeek.com/what-are-the-different-types-of-computer-software-support.htm#didyouknowout>

What are the four Dimensions of ITIL 4?. [online]. Copyright © [cit. 29.12.2019]. Dostupné z: <https://info.axiossystems.com/blog/what-are-the-four-dimensions-of-itsil-4>

What is ITIL? Your guide to the IT Infrastructure Library | CIO. CIO.com - Tech News, Analysis, Blogs, Video [online]. Copyright © 2019 [cit. 26.12.2019]. Dostupné z: <https://www.cio.com/article/2439501/infrastructure-it-infrastructure-library-itsil-definition-and-solutions.html>

What is Problem Management | Try Freshservice ITSM Tool. Freshservice ITSM System | ITIL-aligned service desk software [online]. Copyright © Freshworks Inc. All Rights Reserved [cit. 28.12.2019]. Dostupné z: <https://freshservice.com/itsm/problem-management>

What is software support? definition and meaning - BusinessDictionary.com. Online Business Dictionary - BusinessDictionary.com [online]. Copyright © 2019 [cit. 18.12.2019]. Dostupné z: <http://www.businessdictionary.com/definition/software-support.html>

### **Ostatní zdroje**

Dorothea Engel, *Contract Template Maintenance and Support Banking*, Diebold Nixdorf 2017, Získaná ukázka kontraktu na vyžádání od společnosti Diebold Nixdorf

Téma: Softwarová podpora Diebold Nixdorf, Náklady na její podporu David Krivánka, 2020, Osobní rozhovor na dané téma. Plzeň 13.2.

## Seznam příloh

---

**Příloha A:** Aplikace na predikci nákladů na SW podporu.

Upravená verze dostupná z odkazu:

<https://drive.google.com/drive/folders/1pG-jifilsERDScOCAKITKDCITKYiKqSg?usp=sharing>



## **Abstrakt**

Kalvoda Jan (2020), *Analýza nákladů na podporu SW produktů* (Bakalářská práce), Plzeň: Západočeská univerzita v Plzni

**Klíčová slova:** Softwarová podpora, Software, Náklady, ITIL, ITSM, Analýza nákladů

Bakalářská práce se zabývá podporou softwarových produktů, především náklady vzniklémi z této činnosti. Nejdříve práce zkoumá softwarovou podporou jako takovou. Je zde definována a také je popsáno její základní členění. Další část práce se týká knihovny postupů ITIL(Knihovna infrastruktury informačních technologií). Tento soubor, jehož celý název je Information Technology Infrastructure Library, se snaží najít nejlepší možný způsob vedení softwarové podpory pro firmu i její zákazníky. Spolu s knihovnou je popsán i způsob managementu z ní odvozený, jímž je ITSM(Management IT služeb). Práce pokračuje zkoumáním faktorů, které nejvíce ovlivňují náklady softwarové podpory. Spolu s popisem těchto faktorů je zde uveden popis základních dvou skupin metod používaných k predikci nákladů na software. V práci zkoumaná data pocházejí ze společnosti Diebold Nixdorf, jež je popsána včetně svojí softwarové podpory. Následně je popsán celý proces zpracování dat od získání ze systému po statistickou analýzu. Výsledkem práce je zpracovaná datová analýza spolu s nástrojem na predikci nákladů ve formě aplikace.

## **Abstract**

Kalvoda Jan (2020), *Cost analysis of SW product support* (Bachelor thesis), Pilsner: University of West Bohemia

**Keywords:** Software support, Software, Cost, ITIL, ITSM, Cost analysis

The Bachelor thesis is focused on software product support; especially cost arising from this activity. First the work describes the basics of software support. There is its definition and then the work lists the different types of it. The next part of the work is about library ITIL. This framework, which full name is Information Technology Infrastructure Library, tries to find the best possible solution of software support, both for customer and company. Together with this library, there is an explanation of the type of management with close ties to ITIL, which is ITSM. The work continues with examining different factors that affect the costs of software support. Together with these factors there is the research of two groups of methods used for software cost prediction. The data used in the work comes from company Diebold Nixdorf, which is also described together with its software support. Then the whole process of data analysis is defined, from the data collection to statistical analysis. The result of work is a statistical analysis of the data together with the tool to predict the costs. The tool is created as an application.