

VYSOKÁ ŠKOLA BÁŇSKÁ – TECHNICKÁ UNIVERZITA OSTRAVA
EKONOMICKÁ FAKULTA

KATEDRA APLIKOVANEJ INFORMATIKY

Návrh a implementácia odmeňovacieho systému
Design and Implementation of The Reward System

Student: Bc. Rastislav Golier
Vedoucí diplomové práce: Ing. Vítězslav Novák, Ph.D.

Ostrava 2012

Zadání diplomové práce

Student: **Bc. Rastislav Golier**

Studijní program: N6209 Systémové inženýrství a informatika

Studijní obor: 1802T001 Aplikovaná informatika

Téma: **Návrh a implementácia odmeňovacieho systému
Design and Implementation of the Reward System**

Zásady pro vypracování:

1. Úvod
 2. Teoretické východiska
 3. Analýza současného stavu
 4. Návrh řešení a jeho realizácie
 5. Zhodnotenie navrhovaného riešenia
 6. Záver
- Zoznam použitej literatúry
Zoznam skratiek
Prehlásenie o využití výsledkov diplomovej práce
Zoznam príloh
Prílohy

Seznam doporučené odborné literatury:

- AMSTRONG, Michael. Odměňování pracovníků. Přeložil Josef KOUBEK. 2. vyd. Praha: Grada Publishing, 2009. ISBN 978-80-247-2890-2.
- HELD, Bernd. Access VBA: Velká kniha řešení. Přeložil Jiří KOLÁŘ. Brno: Computer Press, 2006. ISBN 80-251-1112-1.
- RITCHIE, Colin. Relational database principles. 2nd ed. London: Cengage Learning EMEA, 2002. ISBN 0-8264-5713-4.

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

Vedoucí diplomové práce: **Ing. Vítězslav Novák, Ph.D.**

Datum zadání: 25.11.2011

Datum odevzdání: 27.04.2012



Ing. Petr Rozehnal, Ph.D.
vedoucí katedry



prof. Dr. Ing. Dana Dluhošová
děkanka fakulty

„Chcel by som sa mimoriadne poďakovať môjmu vedúcemu diplomovej práce Ing. Vítězslavovi Novákovi, Ph.D. za ochotu, pomoc a cenné rady pri vypracovaní mojej diplomovej práce. Ďalej by som chcel poďakovať mojej rodine, priateľom a všetkým zainteresovaným za podporu pri realizácii mojej diplomovej práce.“

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

V Ostravě 27. 04. 2012

.....
Bc. Rastislav Golier

Obsah

1	Úvod	5
2	Teoretické východiska	6
2.1	Odmeňovanie pracovníkov	6
2.1.1	Základná finančná odmena	9
2.1.2	Systém bonifikácie	14
2.2	Informačný systém	17
2.2.1	Štruktúra informačného systému	18
2.2.2	Vzťah informačného systému k systému riadenia	18
2.3	Softwarové inžinierstvo	20
2.3.1	Metodiky, životné cykly a procesy vývoja softwaru	21
2.3.2	Agilné metodiky	26
2.4	UML	28
2.4.1	Stavebné bloky	29
2.4.2	Všeobecná mechanika jazyka UML	32
2.4.3	Architektúra	33
2.5	Microsoft Access 2003	34
3	Analýza súčasného stavu	36
4	Návrh riešenia a jeho realizácie	41
4.1	Návrh riešenia bonusového systému	41
4.1.1	Spracovanie užívateľských požiadaviek	41
4.1.2	Diagram prípadu užitia Bonusového systému	44
4.1.3	Matica RTM (matica sledovanosti požiadaviek)	55
4.1.4	Analytický model tried	55
4.1.5	Realizácia prípadu užitia	57
4.2	Realizácia bonusového systému v prostredí Microsoft Access	60
4.2.1	Relačný model	61
4.2.2	Tvorba formulárov	62
4.2.3	Tvorba reportov	65
4.2.4	Implementácia systému	67
5	Zhodnotenie navrhovaného riešenia	68
6	Záver	69
	Zoznam použitej literatúry	70

Zoznam skratiek	72
Prohlášení o využití výsledků diplomové práce.....	73
Zoznam príloh	74

1 ÚVOD

Odmeňovanie pracovníkov je jednou z najpopulárnejších techník ako zamestnanca motivovať k lepšiemu výkonu a prístupu k práci. Túto metodiku by sme si nemali predstavovať len ako vidinu finančných odmien, ale aj ako systém, ktorý dokáže pri správnom nastavení priniesť spoločnosti lepšie výsledky, spokojnejších zákazníkov a v neposlednom rade finančný zisk.

Cieľom mojej diplomovej práce je navrhnúť, vytvoriť a implementovať bonusový systém pre jednu z pobočiek najúspešnejších českých bánk. Aplikácia by mala vymeniť predchádzajúci systém, ktorý nevyhovoval množstvu informáciám a nespĺňal moderné predpoklady pre racionálny IS. Aplikácia by mala sledovať dochádzku a výkon pracovníkov, z ktorých sa bude počítať bonus pre zamestnanca k príslušnému obdobiu a bude možné sledovať a analyzovať ich pracovný výkon.

Diplomová práca sa člení do viacerých hlavných kapitol. Druhou kapitolou je teoretická časť k problematike. Na začiatku rozoberám podstatu odmeňovania v personalistike, ktorá ozrejmi hlavnú myšlienku spomínanej aplikácie. Ďalej sa zaoberám definíciou informačného systému, jeho rozdelením a náväznosťou na systém riadenia. Na podloženie teoretického základu návrhu informačného systému, stručne popisujem význam a metodiky softwarového inžinierstva a UML jazyka. Na záver teoretickej časti popisujem vývojové prostredie, v ktorom bude aplikácia vytvorená.

Tretia kapitola vyhodnocuje analýzu súčasného stavu spomínanej pobočky a ich súčasného systému. V kapitole rozoberám výhody a nevýhody súčasného systému. Jeho prínos a prečo sa stáva zastaraným a neefektívnym v pobočke.

Štvrtá kapitola využíva poznatky z predchádzajúcich dvoch kapitol na vytvorenie návrhu aplikácie podľa požiadaviek zadávateľa. Pomocou UML jazyka zachytím požiadavky, ktoré by mali byť vyhotovené v aplikácii. Na základe vyhotovených diagramov v návrhu vytvorím aplikáciu v prostredí Microsoft Access 2003.

Piata kapitola vyhodnocuje celý návrh a implementáciu aplikácie, kde popíšem možné návrhy na vylepšenie aplikácie.

2 TEORETICKÉ VÝCHODISKA

2.1 Odmeňovanie pracovníkov

Definícia riadenia odmeňovania

Odmeňovanie sa zaoberá stratégiou, politikou a procesmi nutných k zabezpečeniu toho, aby bolo, čím ľudia prispievajú organizácii, uznané a odmenené. Je zamerané na udržanie systému odmeňovania, usilujúcich o uspokojovanie potrieb organizácie aj osôb na nej zainteresovanej. Hlavným cieľom je odmeniť zamestnancov slušne, spravodlivo a dôsledne na základe ich hodnôt pre organizáciu, aby to slúžilo dosahovaniu strategických cieľov organizácie. [2]

Riadenie odmeňovania sa netýka len finančných odmien a zamestnaneckých výhod. Môže sa týkať aj nepenažných a nehmotných výhod odmeňovania, ako je uznanie, pochvala, možnosti vzdelávania, rozvoja, rastu kompetencií a zodpovednosti pri vykonávaní práce.

Ciele riadenia odmeňovania

- Odmeňovať ľudí podľa hodnôt, ktoré vytvárajú,
- Koordinovať postupy v odmeňovaní s cieľmi podniku a podnikateľskú činnosť s hodnotami a potrebami pracovníkov;
- Odmeňovať správne veci, aby vyplynulo čo je jasné, a čo je dôležité;
- Získavať a udržiavať si vysoko kvalitných pracovníkov;
- Motivovať ľudí a získavať ich oddanosť a angažovanosť;
- Vytvoriť kultúru vysokého výkonu;

Dosahovanie cieľov

Cieľov riadenia odmeňovania sa dosahuje pomocou vytvárania a realizácie stratégie, politiky, procesov a postupov. Vychádzajú z filozofie odmeňovania a sú v súlade s princípmi distribučnej spravodlivosti a práva. Fungujú slušne, spravodlivo, dôsledne, logicky a transparentne. [2]

Filozofia odmeňovania

Riadenie odmien je založené na dobre sformulovanej filozofii – sústave presvedčení a vodcovských princípov, ktoré odpovedajú hodnotám organizácie a pomáhajú ich zapojiť do systému.

Táto filozofia tvrdí, že ak je riadenie ľudských zdrojov o investovaní do ľudského kapitálu, od ktorého sa vyžaduje „rozumová návratnosť“, v tom prípade je v poriadku odmeňovať ľudí odlišne na základe ich výsledkov a prínosu (tzn. podľa návratnosti investície).

Filozofia riadenia odmeňovania tiež uznáva, že musí byť strategická v tom zmysle, že je zameraná na dlhodobjšie záležitosti, ktoré sa týkajú toho, ako by mali byť zamestnanci ohodnotený a ocenený za to, čo robia a čoho dosahujú. Stratégia a procesy odmeňovania, ktoré je nutné realizovať, musia vyplývať z podnikovej stratégie.

Riadenie odmeňovania uplatňuje prístup „celkovej odmeny“, ktorý zdôrazňuje význam uznávania všetkých aspektov odmien ako súdržného celku zlúčeného s ostatnými iniciatívami riadenia ľudských zdrojov zameraných na motiváciu, oddanosť, angažovanosť a zlepšenie pracovníkov. To vyžaduje integráciu stratégie odmeňovania s ostatnými stratégiami ľudských zdrojov, vrátane riadenie talentov a rozvoja ľudských zdrojov. [2]

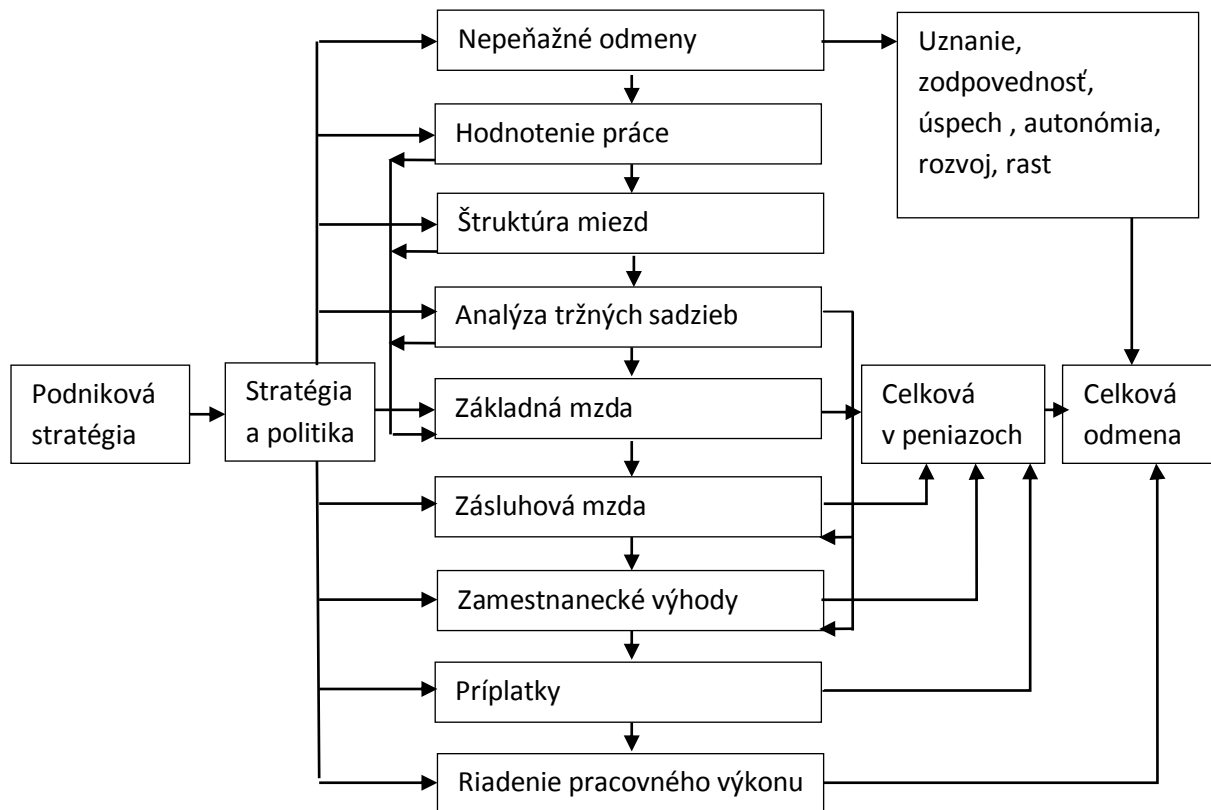
Systém odmeňovania

Prístup k plneniu dosahovania cieľov riadenia sú vtelené do systému odmeňovania organizácie. Systém odmeňovania tvorí:

- Stratégia odmeňovania, ktorá určuje, čo spoločnosti z dlhodobého hľadiska zamýšľajú vykonať v oblasti tvorby a realizácie politiky, praxe, procesov a postupov odmeňovania podporujúcich dosahovanie ich podnikateľských cieľov. Pr.: spoločnosť môže mať stratégiu na udržanie konkurencie schopnosti mzdových sadzieb.
- Politika odmeňovania, ktorá poskytuje návod na rozhodovanie a potrebné kroky. Pr.: spoločnosť môže mať politiku, ktorej úroveň finančných odmien v spoločnosti budú odpovedať priemerným tržným sadzbám.
- Prax odmeňovania, ktorá tvorí štruktúry stupňov a sadzieb, metódy, ako sú hodnotenie práce, a programy, ako je zásluhové odmeňovanie používané k realizácii stratégie a politiky odmeňovania. Pr.: politika týkajúca sa finančných odmien vedie k zhromažďovaniu a analyzovaniu údajov o tržných sadzbách.

- Procesy odmeňovania, ktoré tvoria spôsoby realizácie politiky vykonávanie praktickej stránky odmeňovania, napríklad spôsob, ako sú využívané a uplatnené výsledky šetrenia a ako manažér riadi proces kontroly a úpravy miezd.
- Postupy (procedúry) odmeňovania, ktoré sú používané v záujme udržania systému a k zabezpečeniu toho, že bude fungovať účinne a flexibilne a bude za vynaložené peniaze prinášať zodpovedajúcu hodnotu.[2]

Prvky systému odmeňovania



Obrázok 1: Prvky systému odmeňovania a ich vzájomné vzťahy[2]

Podniková stratégia

Východiskom systému odmeňovania je podniková stratégia organizácie. Tá identifikuje hnacie sily podnikania a stanovuje firemné ciele. Hnacie sily sú v každej organizácii jedinečné, ale často majú spoločný základ ako vysoký výkon, ziskovosť, produktivita, inovácia, kvalita a potreba uspokojiť zainteresované strany – investorov, akcionárov, pracovníkov. [2]

Stratégia a politika odmeňovania

Vyplýva z analýzy hnacích síl podnikania. Otázka: „Ako sa dá v záujme dosahovania firemných cieľov tieto hnacie sily podporovať pomocou odmien?“ Stratégia odmeňovania

definuje dlhodobé ciele v takých oblastiach, ako sú mzdové štruktúry, zamestnanecké výhody, kroky smerujúce k zvýšeniu angažovanosti a oddanosti a uplatňovaniu prístupu založeného na celkovej odmene.

Politika odmeňovania bude obsahovať také záležitosti, ako sú úrovne finančných odmien, plnenie úloh uplatňovať zásadu rovnakej odmeny za prácu rovnakej hodnoty a flexibilné zamestnanecké výhody. [2]

2.1.1 Základná finančná odmena

Základná sadzba je množstvo peňazí (pevný plat alebo mzda), ktorá tvorí sadzbu za určitú prácu alebo pracovné miesto. Môže sa meniť podľa úrovne práce. U manuálnych pracovníkov podľa úrovne požadovaných schopností.

Finančná odmena bude ovplyvnená internými aj externými faktory. Interné faktory môžeme posudzovať pomocou niektorých foriem hodnotenia práce. Hodnotenie externých faktorov je založené na sledovaní sadziieb na trhu práce. Alternatívne môžu byť úrovne finančných odmien dohodnuté prostredníctvom kolektívneho vyjednávania s odborovými zväzmi alebo na základe individuálnych dohôd. [2]

Zásluhová odmena

Ďalšie peňažné odmeny, ktoré je možno poskytovať, sú odmeny vzťahujúce sa k pracovnému výkonu, schopnostiam, prínosu a skúsenostiam, sú to tzv. „zásluhové finančné odmeny“. Je možné ich zlúčiť so základným platom. Ak takéto finančné odmeny nie sú zlúčené so základným platom (tzn. sú platené ako **bonusy**), potom označujú ako „premenlivé finančné odmeny“ [2]

Zamestnanecké výhody

Tvorí dôchodok, nemocenské dávky, úhrada poistného, služobné autá a mnoho ďalších výhod. [2]

Definovanie individuálnej zásluhovej odmeny

Systém individuálneho zásluhového odmeňovania – s výnimkou systému odmeňovania podľa dĺžky zamestnania – sú založené na procese prepojovania peňažnej odmeny s výkonom, schopnosťami, prínosom alebo vedomosťami. Poskytuje odpoveď na 2 základné otázky odmeňovania: 1) čo oceňujeme? a 2) čo sme ochotný platiť?

Použité ukazovatele môžu byť vyjadrené ako bodové hodnotenie, ktoré sa pomocou určitého vzorca premenia na výplaty. Alternatívne nemusí existovať žiadne formálne hodnotenie. Potom je rozhodnutie o odmene založené skôr na rozhodnutí ako na určitom vzorci. [2]

Výskyt zásluhového odmeňovania

Britské šetrenie e-reward Survey of Contingency Pay z roku 2004 zistilo, že 100 respondentov používalo 189 systémov zásluhového odmeňovania, a to v nasledujúcom vyjadrení [2]:

- Odmena podľa výkonu – 65%
- Odmena podľa prínosu – 33%
- Odmena podľa dĺžky zamestnania – 15%
- Odmena podľa schopností – 8%

Zásluhová odmena ako motivátor

Mnoho ľudí považuje zásluhovú odmenu za najlepší spôsob motivácie zamestnancov. Ale je zjednodušením predpokladať, že to, čo vytvára dlhodobú motiváciu, sú iba externé motivátory v podobe peňazí. Interné motivátory, ktoré môžu vychádzať z práce samotnej a z pracovného prostredia, môžu mať hlbší a dlhodobejší účinok.

Výskumy zisťovali vzťah medzi finančnými, externými formami motivovania a internými odmenami, kde sa zistilo, že v prostredí úzkostlivej manažerskej kontroly a obmedzenej komunikácie boli tieto dve formy motivované navzájom v nepriamej úmere, tzn. čím väčší bol dôraz kladený na finančné odmeny, ako je odmena podľa výkonu, tým nižšia bola interná motivácia k práci. Na druhej strane v spoločnosti s vysokou mierou komunikácie rástla interná aj externá motivácia súčasne. Z toho sa dá usúdiť, že rolu hrajú podmienky v ktorých sa pracuje. [2]

Rozdiel medzi stimulmi a odmenami

Finančné stimuly – sú určené k priamej motivácii. Hovoria zamestnancom, koľko dostanú v budúcnosti peňazí, ak budú dobre pracovať. Príkladom finančných stimulov sú systémy odmeňovania manuálnych pracovníkov podľa výsledkov a systém provízneho odmeňovania obchodných zástupcov.

Finančné odmeny – hrajú úlohu nepriamych motivátorov, pretože sú hmatateľnými nástrojmi uznávania a oceňovania úspechov v prípadoch, kedy zamestnanci očakávajú, že to čo urobia im v budúcnosti prinesie niečo, čo stojí za to. Odmeny môžu byť retrospektívne orientované, orientované na minulosť. Ale môžu byť aj orientované na budúcnosť. [2]

Argumenty pre a proti zásluhovému odmeňovaniu

Argumenty pre:

- Uznanie odmeňovania lepšieho výkonu
- Získavanie a udržovanie si vysoko kvalitných ľudí
- Zlepšovanie výkonu organizácie
- Zameranie pozornosti na hlavné výsledky a hodnoty
- Vysielanie posolstva o význame výkonu
- Motivovanie ľudí
- Ovplyvňovanie chovania
- Podpora zmeny kultúry

Argumenty proti:

- Kritéria úspešnosti sú náročné a ťažko dosiahnuteľné
- Peniaze sami o sebe neprivedú k trvalej motivácii
- Ľudia reagujú na akúkoľvek podobu motivovania najrôznejšími spôsobmi – nie všetkých motivujú peniaze rovnako
- Finančné odmeny možno motivujú tých, ktorí ju dostanú, ale môžu demotivovať tých, ktorí ich nedostanú.
- Systém zásluhového odmeňovania závisí na existencii presných a spoľahlivých metód merania, či posudzovania výkonu, schopností a prínosu. [2]

Kritéria úspešnosti v 5 kritériách

1. Jednotliví pracovníci by mali vidieť súvislosť medzi tým, čo robia, a tým čo za to dostávajú. Tento model je ilustrovaný na obrázku č. 2. Odmena by mala byť jasne a tesne spojená so splnením niečoho či s vynaložením úsilia – zamestnanci vedia čo dostanú, ak dosiahnu definovaných a dohodnutých cieľov alebo štandardov a môžu sledovať a posudzovať svoj výkon.

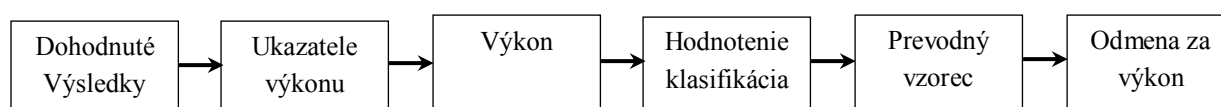
2. Odmeny stoja za to.
3. Pre účely merania alebo posudzovania výkonu, schopností, prínosu alebo sú k dispozícii spravodlivé, slušné, logické a dôsledné nástroje.
4. Ľudia musia byť schopný ovplyvňovať svoj výkon tým, že zmenia svoje chovanie a rozvinú svoje schopnosti a zručnosti.
5. Odmena by mala nasledovať čo najskôr po splnení úloh, z ktorých je odvodená. [2]



Obrázok 2: Model jasnej súvislosti [2]

Odmena podľa výkonu

Metódy uplatňovania podľa výkonu sa výrazne odlišujú, ale hlavné rysy vystihuje obrázok.



Obrázok 3: Odmena podľa výkonu[2]

Základy systému

Prírastky finančnej odmeny sa vzťahujú k dosiahnutiu dohodnutých výsledkov definovaných ako úlohy alebo ciele. [2]

Zvýšenia zahrnuté do mzdy/platu

Existuje možnosť pre zvyšovanie odmeny v rámci základnej odmeny, teda v rámci mzdového rozsahu priradeného k stupňom alebo úrovniam v štruktúre stupňov alebo v štruktúre skupín (druhov) kariéry, alebo v rámci zón širokopásmovej štruktúry. Tieto zvýšenia sú stále – je ojedinelé, ak sú odňaté. [2]

Finančné bonusy (premenlivá odmena)

Alternatívne alebo dodatočne môže byť vysoká úroveň výkonu odmenená finančnými bonusmi, ktoré nie sú súčasťou základnej mzdy a ľudia si ju musia opakovane získavať. Jednotlivec môže mať na tento bonus nárok, ak dosiahne hornej hranice rozpätia svojho stupňa, alebo ak je hodnotený ako plne schopný jedinec. [2]

Zvyšovanie finančnej odmeny

Miera zvyšovania odmeny v rámci rozpätia v štruktúre stupňa a mzdovej štruktúre sú obvykle, ale nie nevyhnutne, určené hodnotením a bodovaním pracovného výkonu. Často sa vykonáva s tým, keď dochádza v rámci riadenia pracovného výkonu ku skúmaniu pracovného výkonu. Toto hodnotenie je možné vykonať aj oddelene ako zvláštne hodnotenie pre účely odmeňovania. Niektoré organizácie sa skôr spoliehajú na celkové posudzovanie toho, ako sa mala mzda jednotlivých pracovníkov zvýšiť v závislosti od výkonu a úrovne platov ich kolegov.

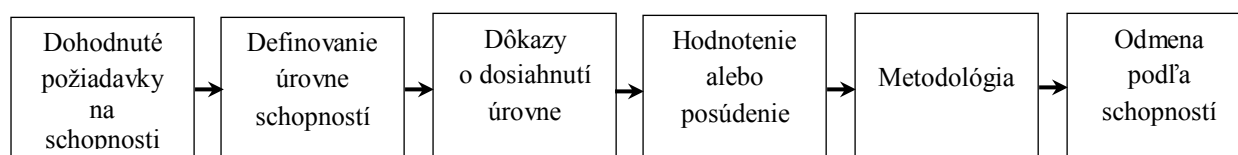
Pri rozhodovaní o veľkosti zvýšenia sa často používa nejaký vzorec v podobe mzdovej matice vid'. tabuľka. Tá ukazuje percentuálne zvýšenie vyplácané na základe rôzneho bodového ohodnotenia výkonu a podľa postavenia mzdy daného jednotlivca v mzdovom rozpätí. Tiež sa to dá nazvať „porovnávací pomer“ a vyjadruje to veľkosť mzdy v podobe percenta vzťahujúce sa ku stredu rozpätia. Porovnávací pomer vo výške 100% znamená, že mzda odpovedá stredu rozpätia. [2]

Hodnotenie výkonu	Percentuálne zvýšenie mzdy v závislosti na hodnotení výkonu a na postavení v mzdovom platovom rozpätí			
	Postavenie v mzdovom rozpätí			
	80 – 90 %	91 – 100 %	101 – 110 %	111 – 120 %
Vynikajúce	12 %	10 %	8 %	6 %
Veľmi efektívne	10 %	8 %	6 %	4 %
Efektívne	6 %	4 %	3 %	0
Rozvojový	4 %	3 %	0	0
Nedostatočný	0	0	0	0

Tabuľka 1: Mzdová/platová matica pri odmeňovaní podľa výkonu [2]

Odmena podľa schopností

Hlavné rysy systému odmeňovania podľa schopností znázorňuje obrázok



Obrázok 4: Odmena podľa schopností [2]

Základy systému

Zamestnanci dostávajú finančné odmeny v podobe prírastkov ku svojej základnej mzde alebo platu podľa úrovne schopností, ktoré pri výkone svojich úloh preukážu. Je to metóda platenia ľudí za ich schopnosť pracovať v súčasnosti aj v budúcnosti.

Zvýšenie zahrnuté do mzdy

Podobne ako v prípade odmeny podľa výkonu existuje možnosť pre zvýšenie odmeny v rámci základnej mzdy, tzn. v rámci mzdového rozpätia priradeného k stupňom alebo úrovňam v štruktúre úzkych stupňov alebo v štruktúre skupín kritérií.

Zvyšovanie finančnej odmeny

Miera hranice zvyšovania odmeny v rámci rozpätia môže byť založená na hodnotení, klasifikácii schopností za použitia podobnej matice ako u odmien podľa výkonu, ale môžu sa riadiť všeobecnejším posúdením rozvoja schopností. [2]

2.1.2 Systém bonifikácie

Definícia

Systém bonifikácie poskytuje zamestnancom finančné výplaty, ktoré sa vzťahujú k výkonu ich spoločnosti, tímu, samému sebe, alebo prienikom medzi týmito možnosťami. O bonusoch sa často hovorí ako o „premenlivej finančnej odmene“.

Definujúcou charakteristikou určitého bonusu je to, že musí byť opakovane získavaný na rozdiel od prírastku mzdy zo systému individuálneho zásluhového odmeňovania, ako je odmeňovanie podľa výkonu, prínosu alebo dĺžky zamestnania, ktoré sa stávajú súčasťou mzdy.

Finančné odmeny môžu byť samostatnou metódou, ako poskytovať ľuďom odmeny navyše k ich základnej mzde.

Ciele systému bonifikácie

- Umožniť zamestnancom, aby sa podieľali na úspešnosti organizácie, a tým sa zvýšila ich oddanosť a lojalita k tejto organizácii.
- Poskytovať odmeny odvodené od výkonu podniku, aby sa zvýšila motivácia, lojalnosť a zánietenosť zamestnancov.

- Poskytnúť odmeny uznávajúce doterajšie výkony alebo úspechy stimulujúce jednotlivcov a tímy k tomu, aby dobre pracovali aj v budúcnosti.
- Zaisťovať aby úrovne, výška odmien boli konkurenčne schopné a priťahovali a udržali kvalitných zamestnancov.

Dôvody existencie bonusov

Základným dôvodom existencie systému bonifikácie vychádza z presvedčenia, že tento systém bude plniť jeden alebo viacero vyššie spomenutých cieľov a bude tak prispievať k vytvoreniu kultúry vysokého výkonu a mať kladný dopad na výsledky spoločnosti.

Ak o tomto dôvode uvažujeme, je nutné posúdiť argument pre existenciu systému bonifikácie buď, ako alternatívu konvenčného systému individuálneho zásluhového odmeňovania, alebo ako nadstavbu k zásluhovej odmene. Výhody a nevýhody nájdeme v tabuľke. [2]

Prístup	Výhody	Nevýhody
Iba bonus	<ul style="list-style-type: none"> ▪ Musí sa znovu získať (zaslúžiť). ▪ Môže byť spojená s podnikovým alebo tímovým výkonom aj s individuálnym výkonom, a môže teda zvyšovať oddanosť a zlepšovať tímovú prácu. ▪ Finančné čiastky a tiež ich absolútne či relatívne veľkosti môžu mať bezprostredný vplyv na motiváciu a angažovanosť. 	<ul style="list-style-type: none"> ▪ Môže byť vnímaný ako výraz určitej ľubovôle. ▪ Mnohí ľudia budú skôr preferovať príležitosť k zvýšeniu svojej základnej mzdy ako sa spoliehať na potenciálne nepravidelné udeľovanie bonusu. ▪ Pokiaľ nie sú súčasťou mzdy, nie sú tieto čiastky započítané do dôchodkového systému.
Bonus a zásluhová odmena	<ul style="list-style-type: none"> ▪ Ponúka to najlepšie s oboch možností: zvýšenie zahrnuté do mzdy a zvláštne finančné platby. 	<ul style="list-style-type: none"> ▪ Potenciálne zložité. ▪ Vplyv buď bonusu, alebo do mzdy zahrnutých čiastok sa môže rozptýliť a vytratiť, hlavne ak sa čaká, ktorá je k dispozícii, rozdelenie na dve časti.

Tabuľka 2: Výhody a nevýhody alternatívneho usporiadania bonifikácie [2]

Zavedenie systému bonifikácie

- Konzultovať s tými, ktorých sa to dotýka, o účele a rysoch systému.
- Definovať zodpovednosť za zavedenie, udržovanie a vyhodnocovanie systému.
- Overiť či je systém uskutočniteľný v jednom z útvarov resp. v reprezentatívnej vzorke zamestnancov, aby sa získali informácie ako daný vzorec funguje, ako primerané je meranie, aké sú náklady systému, aký je jeho dopad, aká je efektívnosť procesu rozhodovania o bonuse. A aké sú odozvy od zamestnancov.
- Ak je to nutné, tak na základe overenia vykonať úpravy.
- Pripraviť nejakú charakteristiku pre informovanie pracovníkov o účele systému.
- Zaistiť, aby sa systém osvojil do každodenného procesu, ktorý prebieha v organizácii.
- Vypracovať rozpočet nákladov na systém.
- Definovať procesy uplatňovania a kontroly, vrátane zodpovedností, plnenie cieľov správnosti, spravodlivosti, dôslednosti a kontroly nákladov.
- Pripraviť plán zavedenia systému, zahrňujúce ukazatele a ciele dohody o pracovnom výkone, metódy skúmania a posudzovania pracovného výkonu, proces rozhodovania o preplácaní bonusu. [2]

2.2 Informačný systém

Spoločnosť je odkázaná na použitie informačných technológií. Informačné systémy, informačné a komunikačné technológie sa stávajú súčasťou podnikania, bez ktorých by si podnikateľ nedokázal udržať krok s konkurenciou. Schopnosť využívať ICT sa stala súčasťou riadenia spoločností a organizácií. Nasadzovanie IT technológií vyvoláva vznik nových podnikateľských príležitostí, ktoré budú zohrávať neoddeliteľnú súčasť v ekonomike v normálnom živote.

Slovo informácie je možné chápať všeobecne ako správu, poznámku, respektíve ako jav. Z filozofického hľadiska je informácia nehmotná a predstavuje to, čo je vnímané človekom. Význam pochopíme iba v určitom kontexte. Dobré informácie sú často kritickým faktorom dobrého rozhodovania v podnikaní, ale aj v iných oblastiach. Musí však spĺňať určité náležitosti ako cielenosť, včasnosť, presnosť a musí byť zrozumiteľná. Kvalitu informácie významne ovplyvňuje jej cesta od zdroja k príjemcovi. Na tejto ceste môže byť informácia zmenená, čím môže byť pre príjemcu až dezinformáciou.

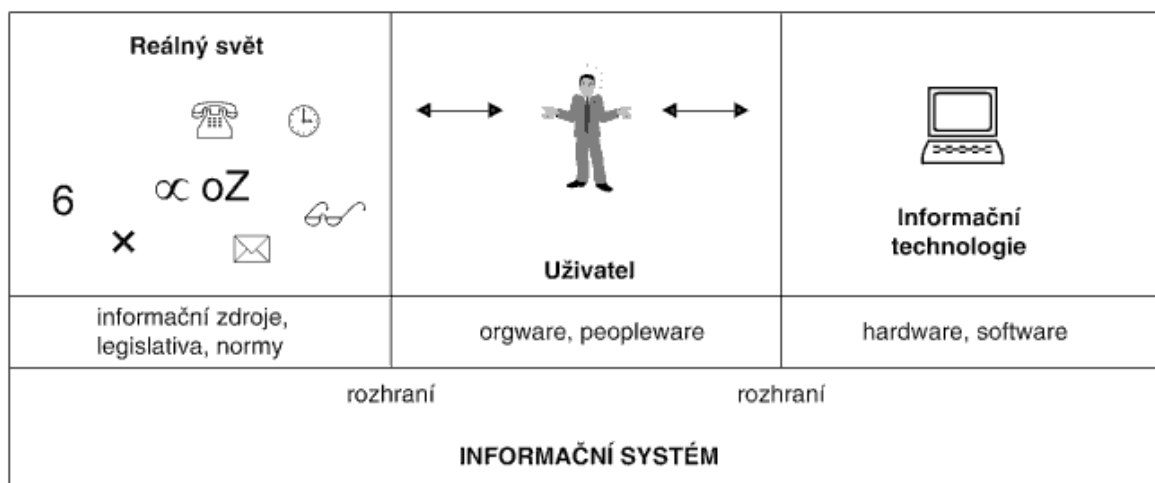
Ak vychádzame z teórie informácie, potom je informácia správa, ktorá nám upresňuje javy alebo objekty reálneho svete. V súčasnosti sa informácie stali výrobným zdrojom rovnako ako pracovná sila, suroviny, výrobné zariadenia či peniaze. Preto je dôležité vedieť ich získať, zapracovať a používať vo svoj prospech.

Slovo systém sa používa v rôznych súvislostiach a jeho význam závisí na historickom vývoji poznatkov. Pôvodný význam z gréckej filozofie znamenal zoskupenie, zjednotenie alebo celok. Dnes je ale vnímaný ako definovaná množina prvkov a väzieb medzi nimi. Tiež sa používa ako označenie určitá časť reálneho s charakteristickými vlastnosťami. Takéto systémy sa delia na systémy prirodzené, kedy hlavné časti systému nie sú vytvorené človekom a existuje nezávisle na ňom, systémy umelé, vytvorené človekom.

Informačný systém je z tohto pohľadu systémom umelým a človek môže výrazne ovplyvňovať jeho kvality. Informačný systém môžeme definovať ako súbor ľudí, metód a technických prostriedkov, ktoré zaisťujú zber, prenos, uchovanie, zapracovanie a prezentáciu dát s cieľom poskytovať informácie podľa potrieb príjemcu. [4]

2.2.1 Štruktúra informačného systému

- Technické prostriedky (hardware) – informačné systémy rôzneho druhu a veľkosti, doplnené o potrebné periférne jednotky, ktoré sú prepojené prostredníctvom počítačovej siete a napojené na databázový systém pre prácu s veľkým objemom dát.
- Programové prostriedky (software) – tvorené systémovými programy, riadiace sa chodom počítača, efektívnou prácou s dátami a komunikáciou počítačového systému s reálnym svetom. Aplikačné programy riešia úlohy užívateľom.
- Organizačné prostriedky (orgware) – tvorené súborom nariadení a pravidiel, definujúcich využívanie informačného systému a IT.
- Ľudská zložka (peopleware) – riešenie otázky adaptácie a účinného fungovania človeka v počítačovom prostredí, do ktorého je zaradený.
- Reálny svet (informačné zdroje, legislatíva, normy) – kontext informačného systému [4]



Obrázok 5: Prvky informačného systému [4]

2.2.2 Vzťah informačného systému k systému riadenia

V súčasnosti pri nestálych externých podmienkach (ekonomické príčiny, zmeny v sociálnych a politických prostrediach, vedecko-technický rozvoj a iné) za hlavný zmysel riadenia sociálne ekonomických systémov považujeme upevňovanie ich vnútornej stability. Takto stabilný systém je predpokladom pre rýchlu a účinnú adaptáciu na vonkajšie zmeny.

Pokiaľ určitá firma alebo inštitúcia nie je schopná prispôbovať sa rýchlo sa meniacim podmienkam, môže nastať stav nerovnováhy. Systém riadenia nie je schopný vyrovnať sa so svojim pôsobením a dochádza k ohrozeniu existencie firmy. Spätná väzba sociálne ekonomických systémov je obvykle uvedená v činnosť porovnania riadiacich a ostatných, s problémom súvislých informácií. Na základe tohto porovnania je potom možné prijať rôzne opatrenia a korigovať riadiace informácie pre ďalšie obdobie. Nové firmy a inštitúcie, sú preto založené skôr na informácie, na rozdiel od tradičných, ktoré sú založené skôr na príkazy a kontrolu.

Získavanie informácií má obrovský význam pre zaistenie pružného systému riadenia. Je treba včas spoznať, aké informácie a v akej miere je nutné zhromažďovať, a správne a efektívne ich spracovávať, distribuovať a využívať. To umožní iba efektívny informačný systém.

Informačný systém môžeme klasifikovať podľa rôznych hľadísk. Je jasné, že s vyššou úrovňou riadenia rastie neurčitosť v požiadavkách na informačný systém a súčasne sa znižuje objem prijatých informácií v dôsledku ich selekcie a agregácie. Zároveň rastie potreba externých informácií z podstatného okolia firmy. Vzhľadom k úrovniam riadenia môžeme informačné systémy členiť na určité časti, z ktorých každá plní svoju funkciu. [4]

Transakčné systémy (Transaction Processing Systems - TPS) sú nasledovníkmi klasických dávkových systémov pre mechanizáciu agendových úloh, ako sú mzdy, fakturácia, inventarizácia, apod. Slúžia pre operatívnu úroveň riadenia a zaisťujú základné procesy v organizácii. Ťažisko ich poslania spočíva v zhromažďovaní základných dát firmy či inštitúcie. Ukážkou využitia on-line TPS súčasnej doby je napríklad spracovanie transakcie objednávky tovaru, vykonávané pracovníkom obchodného oddelenia. Požiadavky riadiacich štruktúr firmy zohľadňuje v prevádzkovom systéme iba funkcie operatívneho výkazníctva, ktorá zaisťuje podporu základných riadiacich procesov. [4]

Informačné systémy pre taktické a operatívne riadenie (Management Information Systems - MIS) vychádzajú z účtovných a ekonomických systémov a užívatelia si v nich vyhľadávajú tie informácie, ktoré ich zaujímajú. Pre ich potreby sa často vytvárajú periodické výstupy z transakčných systémov, robia sa sumarizácie, modelové agregácie a výbery informácií (reporty). [4]

Systémy pre podporu rozhodovania (Decision Support Systems -DSS) majú schopnosť robiť zložité analýzy dát bez potreby zložitého ovládania. Sú určené predovšetkým pre podporu stredných zložiek manažmentu. Jedná sa o počítačovú podporu metód rozhodovacej analýzy a operačnú systémovú analýzu. [4]

Manažérske aplikácie bussines inteligencie (EIS – Executive Information Systems) Zabezpečujú vrchol riadiacej pyramídy. Slúžia predovšetkým vrcholovému vedeniu organizácie, ktorá sa viac zaujíma o informácie z okolia organizácie (technické inovácie, trh, banka, politická situácia, konkurencia apod.) Sú navrhnuté tak, aby jednotky umožňovali prístup k externým dátam a boli napojené na informačný systém firmy. Zo základných dát operatívneho charakteru vytvárajú prísne štruktúrované a vysoko agregované dáta s vysokou vypovedacou hodnotou. Ich ďalšou typickou vlastnosťou je multidimenzionalita, ktorá umožňuje rýchlo a jednoducho vytvárať nový pohľad na dáta, ktorých radenie do nových súvislostí, vyhľadávanie zákonitostí (trendových charakteristík), indikácii odchýliek kľúčových ukazovateľov od plánovaných hodnôt, prácou s históriou a anticipáciou budúceho vývoja. [4]

2.3 Softwarové inžinierstvo

Definícia podľa Fritz Bauer

„Softwarové inžinierstvo je zavedenie a používanie riadnych inžinierskych princípov tak, aby sme dosiahli ekonomickej tvorby softwaru, ktorý je spoľahlivý a pracuje účinne na dostupných výpočtových prostriedkoch“

Táto definícia popisuje všetko podstatné, čo obsahuje softwarové inžinierstvo, čím sa zaoberá a o čo sa snaží. Úlohou programátora respektíve softwarovej firmy nie je iba napísať program, ale práca by mala zahŕňať mnoho ďalších nemenej dôležitých aspektov. Hlavne tieto aspekty popisuje softwarové inžinierstvo.

Softwarové inžinierstvo sa nestará len o implementovanie inžinierskych princípov, ale pokúša sa nabádať i k ich dodržiavaniu. V praxi je medzi týmito dvoma pojmami diametrálne veľký rozdiel. Implementovať môžeme hocičo, dokonca sa to môže javiť, že sa to používa, ale pokiaľ to nie je pravda, výsledky nebudú dokonalé. Dobrý príklad „predstieranie“ môžem nájsť v oblasti manažmentu kvality pri implementácii štandardu ISO 9000. Tento certifikát je možné získať po splnení mnoho podmienok, ktoré určite prinesú firme určitý rast a zlepšenie

voči konkurentom. Nemôžeme si myslieť, že získaním takéhoto certifikátu sa firma stane efektívnejšia, úspešnejšia a konkurencieschopnejšia. Je to omyl. Pokiaľ sa firma nebude skutočne starať o kvalitu, nepomôžu jej žiadne certifikáty.

Podmienky úspešnej, ekonomickej tvorby softwaru

- Je nutné vhodne zostaviť vývojový tím. Tím by mal mať primeraný počet členov v závislosti od problému. Mal by pokrývať všetky potrebné role, mal by rešpektovať zavedené metodiku, mal by mať vhodný *truck faktor* (závislosť celého tímu na znalostiach jeho jednotlivých členov, aby sa nestalo, že po odchode kľúčových členov nerozumie danej problematike nikto iní a môže nastať totálny krach projektu). Mal by byť zárukou efektívnej práce ľudí.
- Dôležitá je voľba správneho vývojového nástroja. Obvykle je voľba systému súčasťou zadania, ale v určitých prípadoch je nutné, aby firma navrhla zákazníkovi ďalšiu alternatívu. Správna voľba vývojového nástroja a prostredia podľa vonkajších podmienok, vybavenia a znalostí tímu môže ušetriť mnoho prostriedkov.
- Úvaha „vyvinúť/kúpiť“. Nie vždy sa oplatí vyvíjať systém, kde je veľká pravdepodobnosť- že už ich niekto robil. V takom prípade je rozumné sa poohliadnuť po knihovne, ktoré môžu projektu ušetriť nemalé prostriedky. U rozsiahlych projektoch hlavne pri práci s komponentovými technológiami je k dispozícii veľké množstvo hotových riešení, z ktorých si môžeme poskladať nový systém.
- Dôležitá je spoločná reč so zadávateľom. Pokiaľ si firma so zadávateľom nerozumie a nedokážu sa dohodnúť, môže to viesť buď k nezískaniu kontraktu, k sporom pri vývoji, alebo dokonca k odmietnutiu prevzatia.
- Je nutné uvažovať o budúcej údržbe/rozširovaní. [3]

2.3.1 Metodiky, životné cykly a procesy vývoja softwaru

Metodiky vývoja aplikácií sú jedným z najdôležitejších produktov softwarového inžinierstva. Rovnako ako má svoju históriu celý tento inžiniersky odbor, prechádzali určitým vývojom aj metodiky. Platí pritom, že metodiky sa vždy pokúšali odrážať aktuálnu situáciu

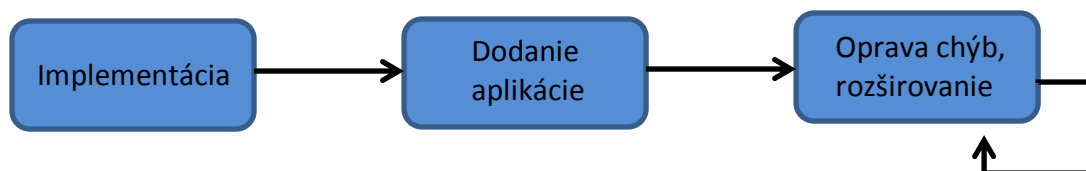
v čase svojho vzniku. To znamená, že sa pokúšali prispôbiť vývoj softwaru konkrétnym požiadavkám kladeným na software v danej dobe. [3]

Cieľom metodík bolo zároveň odstrániť nedostatky vývoja aplikácií, ktoré sa v príslušnom období najviac prejavovali.

Metodiky

Model napíš a oprav

Od úplných začiatkov vývoja programov (v 50. rokoch minulého storočia) sa používal tzv. *model napíš a oprav*. Tento model je možné veľmi jednoducho popísať: spočíval v písaní aplikácie, v jej následnej implementácii a oprave chýb. V prípade tohto prístupu sa nedá hovoriť o nejakej metodike, model vznikol spontánne a aj jeho pomenovanie prišlo až neskôr.



Obrázok 6: Metoda napíš oprav [3]

O efektívite tohto modelu a o aplikáciách, ktoré boli na jeho základe postavené, sa nedá povedať nič moc pozitívne. Na druhej strane je nutné uznať, že v dobe kedy bol používaný, vznikali nie moc rozsiahle programové systémy. [3]

Striktná postupnosť fáz

Časom sa prišlo k tomu, že vyvíjať software týmto spôsobom nie je dlhodobo možné. Preto bol už v roku 1957 definovaný tzv. *stagewise model* životný cyklus (model založený na striktnej postupnosti fáz), ktorý oddeľuje vývoj softwaru do niekoľkých fáz (definícia problému, špecifikácia požiadaviek, architektúra a návrh, implementácia, integrácia, prevádzka). [3]

Základným problémom tohto modelu bola úplná absencia akejkoľvek spätnej väzby, tzn. že po dokončení jednej fázy sa pokračovalo ďalej bez snahy overiť jej výsledky. Podľa tohto modelu sa po skončení fázy nevykonávali žiadne revízie, nehodnotili sa doterajšie výsledky, nerevidovali sa požiadavky a nehládali sa žiadne rizika. Vývoj postupoval iba

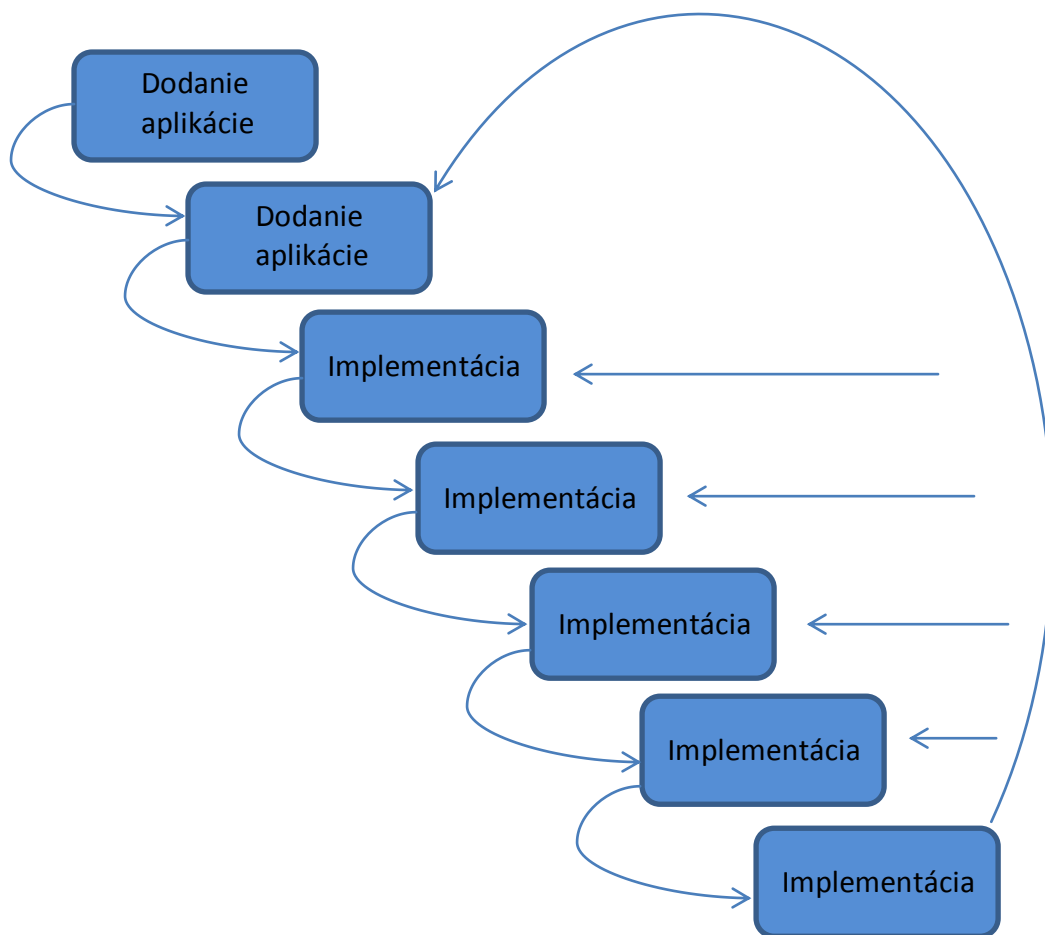
dopredu, nebolo možné návrat do predchádzajúcej fázy za účelom spresnenia požiadaviek alebo dokončeniu analýzy. Jediná možnosť na návrat bola až v samotnom závere – po dodaní aplikácie mohla nasledovať fáza tzv. revalidácie, v ktorej náplňou je rozhodnúť o navrátení späť a o opätovnom vstupe do procesu. [3]

Vodopádový model

V 70. rokoch vznikla disciplína softwarového inžinierstva. Čím sa zaužíval aj väčší dôraz kladený na spôsob vývoja softwaru. Konkrétnym výsledkom je definícia **vodopádového modelu**, ktorý vytvoril Dr. Winston Royce. Vodopádový model sa od svojho predchodcu (modelu stagewise) líši predovšetkým snahou o akúsi spätnú väzbu: na konci fázy dôjde k jeho vyhodnoteniu a prípadnému prepracovaniu „oprave“. Vodopádový model sa skladá z rovnakých fáz ako model stagewise, ale v jeho schéme sú spätné väzby medzi jednotlivými fázami.

Znamená to, že po dokončení jednotlivých fáz bolo možné vykonať akúsi revidenciu dosiahnutého stavu. Pokiaľ tento stav nezodpovedal požiadavkám alebo očakávaniam, bolo možné sa vrátiť k jadrú problému (napr. upraviť špecifikáciu, doladiť návrh či vybrať iné podmienky testov apod.) Vodopádový model znamenal vo svojej dobe malú revolúciu a mnoho autorov ho považuje za skvelý štartový bod pre vývoj softwaru. Medzi jeho najväčšie výhody patrí jasné oddelenie jednotlivých fáz (umožňuje dobre kontrolovať dosiahnutý pokrok), možnosť vrátiť sa pri vývoji o krok späť a tiež jeho jednoduchosť.

Aj napriek tomu postupne začali vystupovať na povrch niektoré jeho nevýhody, ako napríklad dodanie formou veľkého tresku (tzn. že zákazník dlhodobo nevidí žiadnu aktivitu a nakoniec dostane komplexný, hotový produkt), určitá nepružnosť (spôsobujú dlhú dodaciu lehotu) apod. Vodopádový model sa ďalej rozvíja a rôzni autori ho vylepšujú a upravujú. [3,14]



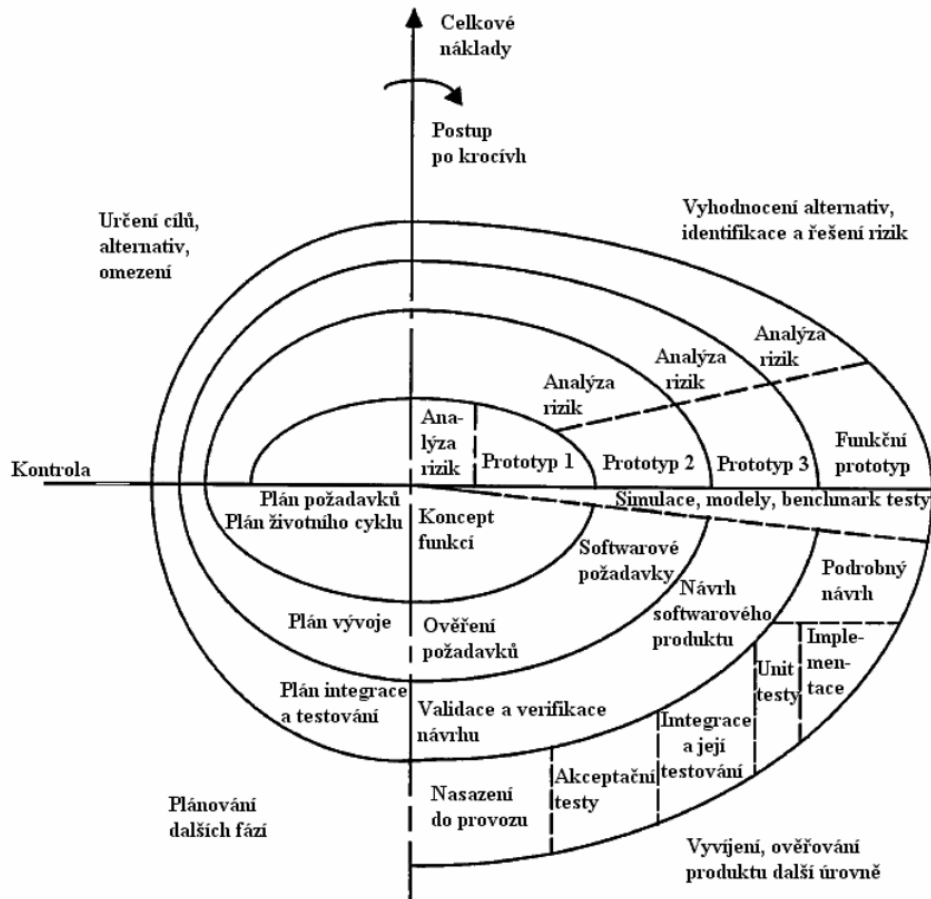
Obrázok 7: Vodopádový model [3]

Špirálový model

Ďalším dôležitým zlomom vo vývoji metodík a životných cyklov bol tzv. *špirálový model*. Tento model môžeme bez problémov vyhlásiť za skutočný míľnik, ktorý sa na dlhú dobu uchytil, a ktorý vlastne dodnes verne odráža životný cyklus širokého spektra systémov. Tento model bol navrhnutý známou postavou softwarového inžinierstva Barry Boehmem v roku 1985.

Model zareagoval predovšetkým na niektoré nedostatky vodopádového modelu, ktoré postupne začali byť viac a viac značné. Do procesu vývoja zavádza predovšetkým dva prelomové koncepty: Iteratívni prístup a opakovanú dôslednú analýzu rizík. Zatiaľ čo vodopádový model je postavený na priamočiarom postupe od zadania úlohy cez analýzu, špecifikácie, návrh, implementácie, otestovanie až k záverečnému predaniu. Špirálový model predpisuje inú postupnosť týchto fáz a predovšetkým iné poradie. Vzhľadom k tomu, že v úvode projektu je veľmi náročné dopodrobna špecifikovať všetky požiadavky a vymenovať

všetky jednotlivé funkcie, stanoví sa v úvode iba všeobecný, vonkajší rámec. Následne sa vyvinie časť aplikácie, vykoná konzultácia so zákazníkom a podľa konkrétnej situácie sa pokračuje analogicky ďalším krokom tzn. že vývoj aplikácie prebieha v opakovaných krokoch, v tzv. iteráciách. Táto koncepcia, v ktorej sa jednotlivé kroky opakujú, a v ktorej spresňujeme špecifikáciu a dotvárame produkt podľa konkrétnej situácie. [3]

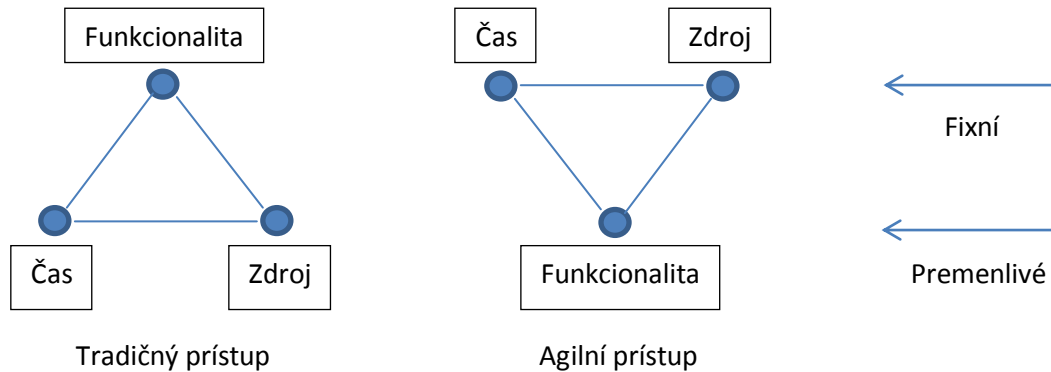


Obrázok 8: Špirálový model [10]

2.3.2 Agilné metodiky

Základné princípy

Rozdiel medzi tradičným prístupom a agilnými metodikami vystihuje obrázok, ktorý porovnáva videnie základných premenných pri vývoji softwaru.



Obrázok 9: Rozdiel medzi tradičným a agilným programovacím prístupom[3]

Tradičné metodiky vychádzajú z nutnosti naplniť za každú cenu dokument špecifickými požiadavkami. Požiadavky – teda funkcionality – sú fixné, zatiaľ čo v roli premenných vystupuje čas a potrebné zdroje. Pri takýchto prioritách vývoja softwaru je jasné čo bude program vedieť, avšak je náročné odhadnúť, koľko to bude stáť a ako dlho to bude trvať. [3,12]

Agilné prístupy oproti tomu považujú čas a zdroje za fixné a mení sa funkcionality. To znamená že vývojový tím sa snaží počas projektu prehodnocovať priority so zákazníkom, aby so zmestili do vyčleneného času a zdrojov.

Princípy:

- Iteratívny a inkrementálny vývoj s veľmi krátkymi iteráciami: plán je zostavený tak, že nové funkcie sa dodávajú často. Zákazník je priebežne konfrontovaný s novými konfiguráciami. Výhodou tohto prístupu je skutočnosť, že zákazník stále vidí, v akom stave je vývoj a na akých funkcionalitách tím práve pracuje. Nemôže sa tak stať, že by na konci dostal niečo iné, ako očakáva.
- Dôraz na priamu osobnú komunikáciu v tíme: rovnako ako v iných sociálnych skupinách, aj v pracovných tímoch je pôvodcom najzávažnejších problémov nefungujúca komunikácia. Agilné prístupy sa pokúšajú integrovať komunikáciu priamo do procesu ako jeho súčasť (časté schôdze, párové programovanie a pod.).

Výhodou je rýchlejšie odhalenie problému, a to ako v tíme, tak aj v samotnom projekte.

- Nepretržitá komunikácia so zadávateľom, resp. užívateľom: zákazník by ideálne mal byť členom vývojového tímu, mal by komunikovať s vývojovým tímom, podieľať sa na návrhu a spolupracovať pri testoch.
- Rigorózne, opakované, priebežné automatizované testovanie: vzhľadom k častým zmenám systému je nutné priebežne overovať jeho správnosť. Testy by mali byť automatizované a mali by byť napísané dokonca pred implementáciou testovaných častí. Vzhľadom ku zrýchleným fázam analýzy a návrhu je testovanie nevyhnutné a jediným istým prostriedkom zachovania kvality aplikácie. [3]

Agilné metodiky vyžadujú omnoho menší objem formálnych a byrokratických artefaktov. Vychádzajú pritom z presvedčenia, že základným, jednoznačným, exaktným, nespochybniteľným a jediným spoľahlivým nositeľom informácie je **zdrojový kód**. Preto na tvorbu zdrojového kódu kladú agilné prístupy najväčší dôraz.

Agilný prístup k vývoju je najvhodnejšie pre projekty s nejasným, neistým alebo časovo sa meniacim zadáním. Pri projektoch, pri ktorých je zadanie jasné, definitívne, nemenné, stabilné, jasne formulované a overiteľné, nie je nutné pristupovať k vývoju agilne, aspoň nie vo všetkých bodoch. [3,15]

Metodiky

Zoznam najvýznamnejších zástupcov agilných metodík :

- Adaptívny vývoj softwaru
- Vlastne riadený vývoj
- Extrémne programovanie
- Lean Development
- SCRUM Development proces
- Crystal
- Dynamic System Development Method
- Testy riadený vývoj [3]

2.4 UML

Jazyk UML (Unified Modeling Language), unifikovaný modelovací jazyk je univerzálny jazyk pre vizuálne modelovanie systémov. Napriek tomu, že sa spája s modelovaním objektovo orientovaných softwarových systémov, má omnoho širšie využitie, čo vyplýva z jeho zabudovaných rozširovacích mechanizmov.

Jazyk UML bol navrhnutý preto, aby spojil najlepšie existujúce postupy modelovacích techník a softwarového inžinierstva. Je navrhnutý takým spôsobom, aby ho mohli implementovať všetky nástroje CASE. Zmienená koncepcia pochádza z pochopenia skutočnosti, že sa rozsiahle softwarové systémy, obvykle bez podpory nástrojov CASE neobídu. Diagramy vytvorené v jazyku UML sa dajú pochopiť aj laikmi, ale navyše ich môžu ľahko interpretovať aj programy CASE.

Je dôležité si uvedomiť, že jazyk UML neponúka žiadny druh metodiky modelovania. Jazyk UML poskytuje iba vizuálnu syntax, ktorú môžeme využiť pri zostavovaní vlastných modelov. Metodikou pre UML je Unified Process. Táto metodiky nám napovedá akých pracovníkov máme použiť, aké činnosti vykonať a aké produkty vyrobiť, aby sa nám podarilo zostaviť model funkčného softwarového systému.

Jazyk UML nie je viazaný na žiadnu špecifickú metodiku alebo životný cyklus. Môžeme ho použiť spoločne so všetkými existujúcimi metódami. Unified Process využíva jazyk UML ako vlastnú syntaxi pre vizuálne modelovanie. Z tohto pohľadu môžeme považovať metodiku Unified Process za uprednostňovanú metodiku na využitie jazyka UML, lebo je pre tento jazyk najlepšie adaptovaná. [1]

Prečo „unifikovaný“ ?

Jazyk UML sa snaží o unifikáciu rôznych domén.

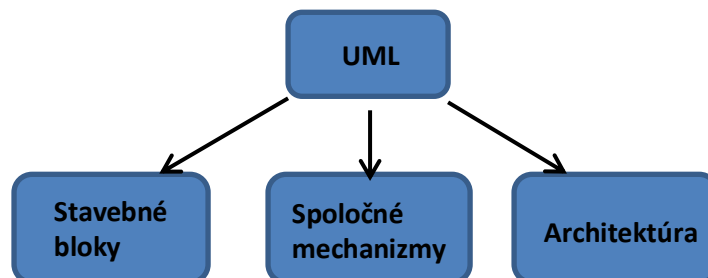
- Vývojový cyklus. Jazyk UML ponúka vizuálnu syntax pre modelovanie behom celého vývojového cyklu softwarového projektu – požiadavkami na analýzu počínajúc a implementáciou končiac.
- Aplikačné domény. Jazyk UML bol vytvorený ako nástroj pre modelovanie čohokoľvek od systémov reálneho času až po podporný systém rozhodovania.
- Implementačný jazyk a platformy. Jazyk UML je nezávislý na akomkoľvek programovacom jazyku a na akejkolvek platforme. Prirodzene, že má znamenitú

podporu objektovo orientovaných programovacích jazykov, ako sú Smalltalk, Java, C#, apod. Rovnako je efektívny pre hybridné objektovo orientované jazyky, ako je C++, aj pre jazyky založené na objektoch, napríklad Visual Basic. Tiež sa používa pre modelovanie projektov vedených v neobjektových jazykoch, napríklad v jazyku C.

- Vývojové procesy. Napriek tomu že je metodika UP spolu s jeho variantmi pravdepodobne najviac uprednostňovaná metodikou vývoja objektovo orientovaných systémov, môže jazyk UML podporovať mnoho ďalších osnov procesu tvorby softwarového vybavenia.
- Vlastné interné pojmy. Jazyk UML sa o vnútornú jednotku a konzistenciu snaží prostredníctvom malej množiny interných pojmov. Je pravda, že v tejto oblasti nie je vždy úspešný. Dá sa však povedať, že každý ďalší pokus je omnoho dokonalejší ako je predchodca. [1]

Štruktúra jazyka UML

Štruktúra jazyka UML obsahuje tieto súčasti vid' obrázok:



Obrázok 10: Štruktúra jazyka UML [1]

2.4.1 Stavebné bloky

Jazyk UML je zostavený z troch stavebných blokov a to z predmetov, relácii, a diagramov.

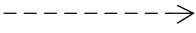


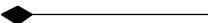


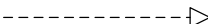
1. Predmety, ktoré delíme na:

- Štruktúrálna abstrakcia, čo sú vlastne podstatné mená modelu, ako triedy, rozhranie, spolupráca, prípady užitia, triedy aktivít, komponenty, uzle.
- Chovanie, čo sú vlastne slovesá modelu, napr. interakcia, stav.

- Zoskupovanie, to sú napríklad balíčky používané k zoskupeniu sémanticky súvisiacich prvkov modelu do súdržných jednotiek.
- Poznámky, anotácie ktoré je možné k modelu pripojiť s úmyslom zachytiť informáciu vytvorenú iba k tomuto účelu. [1]

2. Relácie (relationships)

Relácie umožňujú ukázať na modele, aký je vzťah medzi dvoma predmetmi. Znameníť analógiu roly, ktorú relácie hrajú v modeloch UML, je rodina a vzťahy medzi jednotlivými členmi. Relácia umožňuje zachytiť významový vzťah medzi predmetmi. Vzťahujú sa na štrukturálnu abstrakciu a zoskupovanie. Veľmi dôležitou súčasťou modelovania jazyka UML je pochopenie presnej sémantike rôznych typov vzťahov. [1]

Typ relácie	Syntax UML		Stručný popis
	Zdroj	Ciel	
Závislosť			Zmena v určitom predmete ovplyvňuje význam závislého predmetu
Asociácia			Popis množiny spojení medzi objektmi
Agregácia			Cieľový prvok je súčasťou zdrojového prvku
Kompozícia			Silnejšia forma agregácia
Ochranná nádoba			Zdrojový prvok obsahuje cieľový prvok
Zovšeobecnenie			Jeden prvok je špecializáciou iného prvku a je možné ho nahradiť všeobecnejším prvkom
Realizácia			Asociácia medzi klasifikátormy, kde jeden klasifikátor určuje dohodu, ktorej uskutočnenie zabezpečuje druhý klasifikátor

Tabuľka 3: Relácie [1]

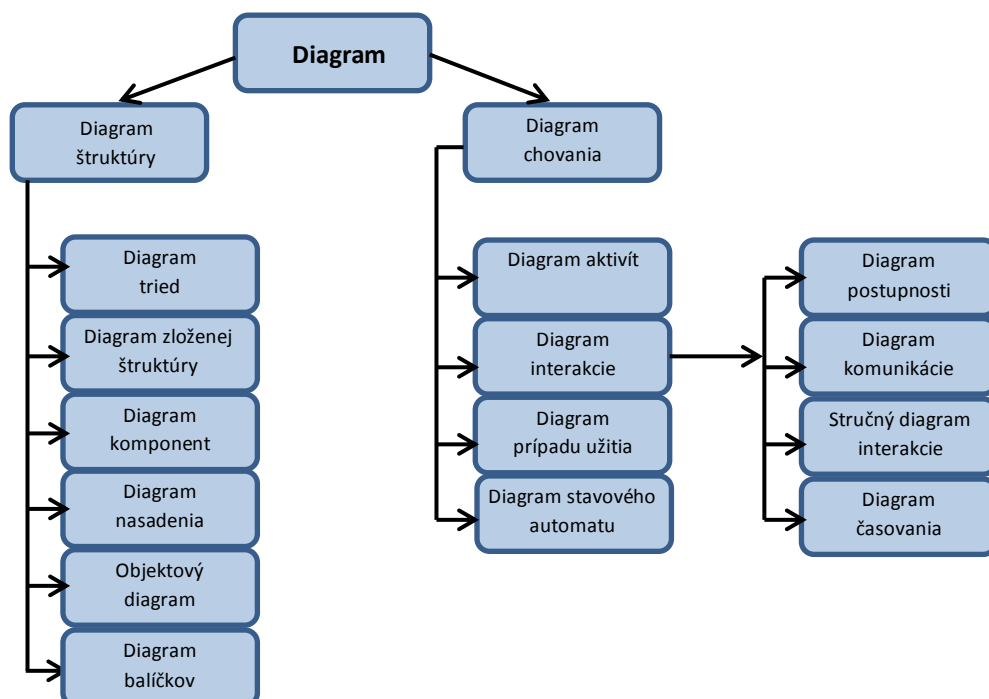
3. Diagramy

Vo všetkých nástrojoch CASE založených na jazyku UML je každý novo vytvorený predmet alebo relácia automaticky pridané do vznikajúceho modelu. Model je repozitárom všetkých predmetov a relácii vytvorených k tomu, aby požadované chovanie softwarového

systemu, ktorý sa pokúšame navrhnuť. Diagramy sú okná alebo pohľady na model. Diagram nie je model! V tom je veľký rozdiel. Predmety a relácie môžeme z diagramu odstrániť, je možné ich odstrániť vo všetkých diagramoch – ale v modele môžu stále existovať. V skutočnosti v ňom ostane až do doby, pokiaľ nebude explicitne vymazaný z modelu. Veľmi častou chybou začínajúcich analytikov a návrhárov v jazyku UML je odstránenie predmetu z diagramov, ale jeho ponechanie v modeli.

Celkom existuje 13 rôznych typov diagramov UML, 3 z nich boli rozšírené v modeli UML2. Môžeme ich rozdeliť na tie, ktoré modelujú statickú štruktúru systému (statický model) a na tie, ktoré modelujú dynamickú štruktúru systému (dynamický model). Statický model zachycuje predmety a štruktúrne asociácie medzi predmetmi. Dynamický model oproti tomu zachycuje spôsob, ktorým jednotlivé predmety navzájom pôsobia, aby sa dosiahlo požadovaného chovania systému.

Neexistuje žiadne pevne stanovené poradie, v ktorom by ste mali svoje UML diagramy vytvárať. Aj napriek tomu sa zvyčajne začína diagramom prípadu použitia, ktorý definuje rozsah platnosti navrhovaného systému. V skutočnosti často pracujete súbežne s niekoľkými rôznymi diagramami zároveň. Každý z nich upravujeme postupne, keď je odhalený ďalší detail navrhovaného softwarového systému. Tak sú diagramy nie len pohľadom na model, ale rovnako prvoradým a základným mechanizmom pre zadávanie nových informácií do existujúceho modelu. [1]



Obrázok 11: Diagramy UML [1]

2.4.2 Všeobecná mechanika jazyka UML

Jazyk UML obsahuje štyri spoločné mechanizmy používané v celom jazyku konzistentne. Popisujú štyri cesty k modelovaniu objektu, ktoré sú opakovane používané v rôznych kontextoch v celom jazyku UML.

1. Špecifikácie

Modely UML majú aspoň dva rozmery – grafické, ktoré umožňujú vizualizovať model prostredníctvom diagramov a symbolov (ikon), a textový, ktorý sa skladá zo špecifikácií rôznych prvkov modelu. Špecifikácie sú textovým popisom sémantiky jednotlivých prvkov.

2. Ornamenty

Veľmi príjemná vlastnosť jazyka UML je skutočnosť, že každý prvok modelu je vyjadrený veľmi jednoduchým symbolom, ktorý môžeme obohatiť rôznymi ornamentmi, ak je treba prostredníctvom diagramov zobrazit' viac informácií. Tvorbu veľmi zložitého modelu tak môžeme začať s niekoľkými základnými symbolmi a niekoľkými ornamentmi. Neskôr však môžeme model vylepšiť ďalšími ornamentmi, až kým príslušný prvok neobsahuje dostatočný počet podrobností.

Treba si uvedomiť, že všetky diagramy UML sú iba pohľadom na daný model. Mali by sme v nich preto ornamenty používať iba v prípade, kedy zvyšujú zrozumiteľnosť a čitateľnosť diagramu, poprípade, ak zdôrazňujem určitú dôležitú funkciu modelu. V diagrame obvykle nie je treba zobrazit' všetky podrobnosti. Omnoho dôležitejšie, je aby bol diagram zrozumiteľný a čitateľný, a aby znázorňoval tie ciele, ktoré chceme dosiahnuť. [1]

3. Podskupiny

Podskupiny popisujú rôzne spôsoby videnia sveta. V jazyku UML rozlišujeme dve podskupiny: skupinu klasifikátorov a inštancii a skupinu rozhraní a implementácií. [1]

4. Mechanizmy rozšíriteľnosti

Autori jazyka UML si uvedomili, že návrh úplne univerzálneho modelovacieho jazyka, ktorý by uspokojil potreby všetkých užívateľov v súčasnosti aj v budúcnosti, je proste nemožné. Z toho dôvodu zahrnuli do jazyka tri veľmi jednoduché mechanizmy, umožňujúce jeho rozšíriteľnosť: [1]

Mechanizmy rozšíriteľnosti	Popis
Obmedzenie	Obmedzujúce podmienky rozširujú sémantiku prvkov tým, že umožňujú pridať k nemu nové pravidlá
Stereotypy	Stereotypy umožňujú definovať nový prvok modelu UML, ktorý je založený na existujúcom prvku – definujeme sémantiku stereotypu Stereotypy pridávajú do metamodelu jazyka UML nové prvky.
Označené hodnoty	Označené hodnoty umožňujú rozšíriť špecifikáciu prvkov tým, že k takémuto prvku prípadne informácii zostavý iba k tomuto účelu.

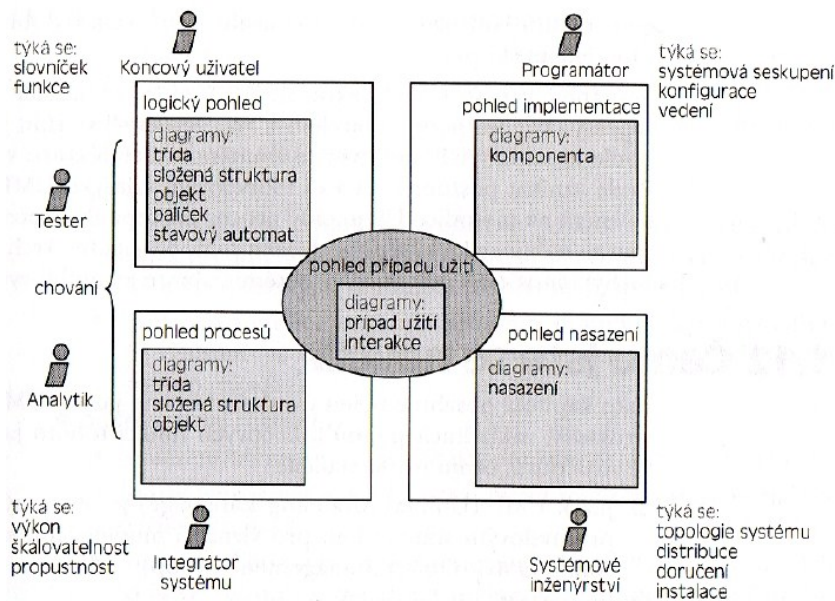
Tabuľka 4: Mechanizmy rozšíriteľnosti [1]

2.4.3 *Architektúra*

Je zachytením strategických aspektov vyššej štruktúry systému. Na architektúru môžeme pozeráť rôznymi spôsobmi, ale najčastejšie sa na ňu pozeráme pohľadom „4+1“. Aby sme boli schopný zachytiť všetky podstatné aspekty architektúry daného systému, definuje jazyk UML štyri rôzne pohľady na systém: logický pohľad, pohľad procesov, pohľad implementácie a pohľad nasadenia. Všetky zmienené pohľady sú integrované do piateho pohľadu prípadu užitia.

- Logický pohľad. Zachycuje slovník oblasti problémov ako množinu tried a objektov. Dôraz je pritom kladený predovšetkým na zobrazenie spôsobov, akým objekty a triedy tvoria základ systému a implementujú jeho chovanie.
- Pohľad procesov. Modeluje spustiteľné vlákna a procesy ako aktívne triedy. Je to procesne orientovaný variant logického pohľadu, ktorý obsahuje rovnaké artefakty.
- Pohľad implementácie. Modeluje súbory a komponenty, ktoré vytvárajú hotový kód systému. Slúži k znázorneniu závislostí medzi jednotlivými komponentmi. Umožňuje definíciu verzie systému.

- Pohľad nasadenia. Modeluje fyzické nasadenie komponentov na množinu fyzických výpočtových uzlov. Umožňuje modelovanie distribúcie komponent na príslušné uzly distribuovaného systému.
- Pohľad prípadu užitia. Všetky ostatné pohľady sú odvodené od pohľadu prípadu užitia. Tento pohľad zachycuje základné požiadavky kladené na príslušný systém ako na množinu prípadov užitia a vytvára tak základ tvorby všetkých ďalších pohľadov. [1]



Obrázok 12: Architektúra UML [1]

2.5 Microsoft Access 2003

Microsoft Access uľahčuje návrh a konštrukciu databázových aplikácií bez toho, aby od Vás vyžadoval znalosť niektorého z programovacích jazykov. Aj keď v aplikácii Access začínate definovaním relačných tabuliek a polí v týchto tabuľkách, ktoré budú obsahovať dáta, rýchlo prejdete k definovaniu akcie nad týmito dátami prostredníctvom formulárov, zostav, makier a aplikácie Microsoft Visual Basic.

Prostredníctvom formulárov a zostav môžete definovať spôsob zobrazenia dát, a to aké dodatočné výpočty si prajete vykonať – podobne ako u tabuliek kalkulátoru. V takom prípade sú formátovacie a výpočtové oddelené od dát, takže je možné využívať dáta rôznymi spôsobmi bez toho, aby boli ovplyvnené. Jednoducho nadefinujete ďalší formulár alebo zostavu s využitím rovnakých dát.

Ak potrebujete automatizovať niektoré akcie v jednoduchej aplikácii, Access ponúka prostriedky pre definovanie makier, ktoré uľahčujú reakciu na udalosti. Ak potrebujete niečo zložitejšie pre funkcie formulárov a zostáv môžete použiť pokročilejšie techniky využívajúce aplikácie Visual Basic a objekty modulov.

Access ponúka pokročilé prostriedky pre vývoj databázových aplikácií, ktoré spracovávajú nie len dáta vo vlastných databázových štruktúrach, ale tiež informácie uložené v mnohých iných bežných databázových formátoch. Pravdepodobne najväčšou silou aplikácie Access je schopnosť pracovať s dátami z tabuliek, textových súborov, súborov dBASE, ľubovoľných databáz SQL, ktoré podporujú štandard ODBC. To znamená, že Access môžeme použiť pre vytvorenie aplikácie pod Windows, ktorá môže spracovávať dáta zo sieťového serveru SQL alebo z databázy SQL uloženej na mainframe. [5]

3 ANALÝZA SÚČASNÉHO STAVU

Útvar bol založený v rámci veľkého projektu, ktorého cieľom bolo centralizovať administratívnu činnosť z pobočkovej siete do zázemia back-office. Takéto usporiadanie činností malo za následok uvoľnenie bankových poradcov na pobočkách od činností, ktoré neboli hlavnou podstatou ich práce, čo bolo hlavne predaj produktov a obsluha klientov. Poradcovia navyše nemali dostatočné znalosti systému a metodiky, takže pri spracovaní administratívnych požiadavkách mali väčšiu chybovosť. Pri centralizácii procesov vzniká priestor pre lepšie presadzovanie zmien v procesoch a metodike a pre zníženie pomeru ručných procesov v prospech automatizovaných procesoch.

Proces centralizácie v útvare nebol zďaleka ukončený. Nadalej je mapovaná časová snímka dňa a hľadajú sa ďalšie bloky činností, ktoré bude možné v budúcnosti z pobočky predať na centrálné pracovisko, bez toho, aby to spôsobilo zhoršenie servisu klientom alebo zvýšenie pracovnosti poradcov.

Medzi hlavné procesy stiahnuté z pobočiek na centrálné pracovisko patrí:

- Schvaľovanie a čerpanie úverových produktov.
- Zakladanie produktov v systéme (bežné účty, úvery, sporiace účty).
- Zakladanie služieb na existujúcich účtoch (rôzne druhy poistenia, zmena služieb na bežnom účte, zmena charakteru účtu).
- Rušenie produktov v systémoch.
- Vypovedanie produktov klientom, ktorí porušili zmluvné podmienky.
- Oprava chýb vzniknutých pri platobnom styku chybou klienta alebo poradcu na pobočke.
- Riešenie klientskych sťažností, odpúšťanie poplatkov, nastavovanie individuálnych zliav.
- Opravy pokladničných rozdielov, ktoré vznikli na pobočkách.
- Servis pre VIP klientov.
- Telefonický helpdesk pre pobočkovú sieť, na ktoré sa môžu obracať v prípade problému.

Väčšina činností prebieha tzv. ručným spracovaním alebo v systémoch, ktoré nedokážu reportovať na úrovni zázemia back-office. To znamená, že u väčšiny procesov nie je možné získať prehľad spracovania položiek za dané časové obdobie.

Na centrálnom pracovisku pracuje v túto chvíľu 105 zamestnancov a predpokladá sa, že ich počet bude s rastúcou centralizáciou naďalej rásť.

Zamestnanci sú rozdelení do ôsmich tímov, každý z tímov má svojho vedúceho tímu, ktorý priamo reportuje riaditeľovi útvaru. Manažérsku zložku tak tvorí 8 vedúcich tímov a riaditeľ. Okrem funkcie operátorov sú traja pracovníci zamestnaní na pozícií dátových a procesných špecialistov a majú na starosti reporting, procesné riadenie, riadenie zmien a účasť na projektoch, ktoré sa útvaru týkajú.

Každý tím má svoje spektrum činností, ktoré má na starosti, a ktoré sa medzi tímami väčšinou neprenášajú.

Riadenie činností prebieha pridelovacím spôsobom, tj. v rámci procesov spracovaných v jednom tíme dostane každý pracovník svoj podiel, ktorý musí spracovať, väčšinou vo forme zostavy, či už papierovej alebo elektronickej. Pokiaľ to spracovať nestihne môže požiadať kolegov o výpomoc.

Odmeňovanie pracovníkov prebieha raz ročne, kedy má každý možnosť získať až 30% (v závislosti na plnení strategických cieľov nielen útvarom, ale aj celou bankou) bonusu z vyplatenej ročnej hrubej mzdy. Hodnotenie pracovníka a teda cieľovú čiastku, ktorú dostane v rámci ročných bonusov vyplatenú, je v kompetencii jeho priameho nadriadeného, vedúceho tímu. Ten môže svoj názor konzultovať s riaditeľom útvaru, a ide predovšetkým o subjektívne hodnotenie v rámci jedného tímu, pretože vedúci tímu nie je schopný porovnávať vyťaženosť pracovníkov s inými tímami, vzhľadom k rôznorodosti činností medzi tímami. Okrem jednoznačne subjektívneho pohľadu na množstvo odvedenej práce je pracovník hodnotený na základe svojich „soft-skills“, ako je tímová spolupráca, schopnosť komunikovať, ochota pomáhať kolegom, proaktivita, stupeň chybovosti, a iné. Vedúci tímu vykonáva hodnotenie každý štvrťrok na schôdzi s pracovníkom a vzájomne si odsúhlasujú svoj pohľad na jeho výkon. Súhrnom týchto štvrťročných hodnotení je potom nárok na ročný bonus.

Výhody

Výhody tohto spôsobu oceňovania sú predovšetkým v nízkych nákladoch spojených s údržbou tohto systému. V podstate ide len o vytvorenie názoru vedúceho tímu na jeho pracovníkov a získanie spätnej väzby, ktorá by bola podstatou schôdzok aj v prípade, že by sa bonus nevyplatil. Nie je treba žiadneho systému a navyše je bonusu priamo naviazaný na splnenie cieľov, ktoré si banka pre daný rok stanovila. Pokiaľ nedôjde k plneniu percentuálneho tržného podielu, dôjde ku zníženiu bonusu u všetkých pracovníkov alebo dokonca nebude vyplatený vôbec.

Nevýhod je v porovnaní s výhodami celá rada.

- Pracovníci nie sú ročným odmeňovaním dostatočne motivovaný, lebo ročný interval je dlhé obdobie na to, že pracujú s každodenným prídelením úloh.
- Existuje neistota, či bude za daný rok bonus vôbec vyplatený.
- Pracovníci sú zvyknutý, že nejaký bonus dostanú. Je to hlavne tým, že majú dobré pracovné vzťahy s vedúcimi tímov a pokiaľ vážne neporušia pracovnú morálku, tak bonus dostanú.
- Nie je možné porovnávať výkon medzi tímami. Zatiaľ čo v jednom tíme môžu byť pracovníci o úroveň výkonnejší a pracovitejší, v konečnom dôsledku dostanú rovnaký podiel ako v inom menej výkonnom tíme.

S príchodom nového riaditeľa útvaru preto došlo k totálnej reštrukturalizácii systému bonusového odmeňovania.

Ciele:

- Stanoviť ciele krátkodobé, teda ideálne mesačne a odmeňovať pracovníkov v týchto intervaloch.
- Mať prehľad o počte a druhu spracovaných položiek aj tam, kde to systém neumožňuje, a to v detaile na pracovníka a deň. Vďaka tomu bude možné sledovať vyťaženosť pracovníkov v priebehu týždňa aj mesiaca a vytipovať špičky, kedy dochádza k nárastu položiek. Treba s tým počítať pri plánovaní dovolení a prenosom kapacít medzi tímami.
- Začať sledovať výkonnosť pracovníkov, nie len medzi tímami ale aj v rámci celého útvaru.

- Presne sledovať nákladovosť vo forme priamych mzdových nákladov.

Po analýze všetkých potrieb a možností bol vytvorený nasledujúci model:

- Rozdelenie činností na také celky, ktoré dávajú zmysel samostatne merať.
- Počiatočné nameranie všetkých činností.
- Vytvorenie evidenčnej tabuľky na každý mesiac po dňoch pre každého pracovníka.
- Vytvoriť jednotného hodnotenia na základe číselníka bodov.
- Prevod výkonnej zložky na body, tak aby sa dal sčítať s mäkkou zložkou. 1 minúta výkonovej činnosti = 2,5 bodu.
- Vytvorenie systému výpočtu bonusu.
- Udržovanie bonusového systému a reportingu kapacitného modelu po tímoch – teda porovnanie fondu pracovnej doby. Pre vytvorenie bonusového systému bol v prvej fáze zvolený Excel, čo sa neskôr ukázalo ako nevyhovujúce kvôli dôvodom, ktoré sú uvedené nižšie.

Výhody nového bonusového systému

- Výrazne vyššia motivácia pracovníkov.
- Možnosť porovnania výkonu medzi pracovníkmi a tímami na základe objektívneho hodnotenia.
- Vytvorenie kapacitného modelu.
- Možnosť reportingu aj u položiek, ktoré nie je možné automaticky získať zo systému.

Nevýhody nového bonusového systému

- Navýšenie nevykonných činností u pracovníkov (zaznamenávanie položiek, cca 5 minút denne za každého pracovníka, pri 96 operátorov znamená 8 hodín denne, čo predstavuje mzdu ďalšieho pracovníka).
- Vysoká pracnosť pri prvom nameraní všetkých položiek (následne sa premeriava iba tam, kde výkon ukazuje nadhodnotené časové normy).
- Z hľadiska pracovníka sa jedná o výhody, z hľadiska banky sa jedná o nevýhodu – bonusy sú z 2/3 vyplatené, aj keď nedôjde k naplneniu ročných cieľov.

Výhody excelovského spracovania bonusového systému

- prehľadnosť a zrozumiteľnosť pre pracovníkov
- nie je potreba programátora na jeho vytvorenie
- flexibilita v úpravách

Nevýhody spracovania bonusového systému v Exceli

- riziko straty dát.
- veľmi vysoká prácnosť udržovania systému. Každý mesiac musí byť vytvorený záznamový súbor pre každého pracovníka, z počiatku ručne, neskôr pomocou makier. Každá nová položka, ktorá pribudne v priebehu mesiaca, sa musí ručne doplniť ku všetkým pracovníkom, ktorý ju budú používať. Vzhľadom k zmenám v štruktúre položiek musí byť kumulácia za každý mesiac vytváraná ručne.
- Existuje kapacitný model, ale nevidíme nárast kapacít útvaru v prípade, že dôjde k prečasovaniu položiek– tj. na kapacitnom modeli sa prečasovanie na reálne nižšie časy prejaví ako pokles výkonnosti.

4 NÁVRH RIEŠENIA A JEHO REALIZÁCIE

Táto kapitola sa skladá z dvoch hlavných častí ako to vyplýva z názvu. V prvej časti kapitoly sa budem zaoberať návrhom riešenia bonusového systému podľa požiadaviek nemenovaného bankového subjektu. Návrh aplikácie som konzultoval s jednotlivými užívateľmi, aby som efektívne spracoval ich požiadavky a mohol ich zachytiť do UML notácie v prostredí Microsoft Visio, ktorý dostatočne vyhovuje požiadavkám pri mojom návrhu.

V druhej časti kapitoly popisujem postup realizácie navrhnutého riešenia podľa zachytených požiadaviek užívateľa. V tejto časti sa zaoberám návrhom relačného modelu, grafického užívateľského rozhrania, užívateľskými rolami, tvorbou analytických reportov a implementáciou systému.

4.1 Návrh riešenia bonusového systému

Aby som mohol navrhnuť systém v UML notácii, musel som pochopiť problémovú oblasť systému a jeho ciele. V prvom rade som vytvoril slovnú úlohu, z ktorej mi vyšli požiadavky na systém. Treba počítať s tým, že v priebehu vývoja aplikácie sa môžu požiadavky meniť a môžu sa odlišovať od návrhu.

4.1.1 *Spracovanie užívateľských požiadaviek*

Popis problému:

Cieľom bonusového systému je evidovať dochádzku a výkon pracovníkov za každý pracovný deň. Administrátor a vedúci tímu sú povinný zadávať iba dochádzku. Túto evidenciu bude vykonávať každý pracovník samostatne a to s maximálne päť denným oneskorením. Pracovníci si nebudú môcť medzi sebou zasahovať do dochádzky a výkonu, ale budú mať možnosť sledovať dochádzku a výkon v tíme. Možnosť úpravy bude mať tím líder s maximálne tridsať dňovým oneskorením a administrátor s neobmedzeným časovým intervalom. V aplikácii bude vedený zoznam činností podľa ktorého budú pracovníci zadávať svoj výkon. Jednou z hlavných úloh aplikácie bude mesačne počítať bonus pre pracovníkov za ich výkon a mäkkú zložku, ktorú subjektívne pripisuje každému zamestnancovi jeho vedúci tímu. Aplikácia by mala jednoducho evidovať pracovníkov, tímy, sviatky, pracovnú

dobu zamestnancov podľa ktorej sa sleduje dochádzka. Hodnotenie priebežnej dochádzky a výkonnosti jednotlivca, tímu i celého odboru by malo byť zachytené v rôznych analytických reportoch, ktoré si užívateľ bude môcť sám upravovať a meniť podľa potreby.

Na základe tejto slovnej úlohy som si mohol vytvoriť tabuľku s požiadavkami užívateľa, kde som určil prioritu a funkčnosť.

Označenie (Rx)	Slovná formulácia požiadaviek	Priorita	Druh
R ₁	Zadávanie dochádzky	***	funkčná
R ₂	Zadávanie výkonu	***	funkčná
R ₃	Kontrola dochádzky	***	funkčná
R ₄	Zadávanie mäkkej zložky pracovníkom	***	funkčná
R ₅	Výpočet bonusu	***	funkčná
R ₆	Evidencia zamestnancov	***	funkčná
R ₇	Evidencia činností	***	funkčná
R ₈	Evidencia tímov	**	funkčná
R ₉	Analýza údajov	***	funkčná

Tabuľka 5: Užívateľské požiadavky

R1 – Zadávanie dochádzky – každý zamestnanec si zadáva dochádzku v pracovných dňoch a to nezávisle či v práci bol, alebo bol neprítomný z dôvodu choroby, služobnej cesty, školenia alebo iného dôvodu. Musí evidovať prerušenie pracovnej činnosti z dôvodu opustenia pracoviska pre obed, lekára poprípadne iného dôvodu. Dochádzku si zadáva každý pracovník nezávisle na jeho postavenie v spoločnosti.

R2 - Zadávanie výkonu – výkon zadávajú iba pracovníci, ktorí vykazujú výkon. Výkon sa zadáva k pracovnému dňu, ktorý bol zamestnanec v práci a vykazoval výkon. Zadávať bude iba určené činnosti, ktoré budú časovo a bodovo nastavené. Každému zamestnancovi budú

pridelené činnosti, ktoré si bude môcť pridať do svojho výkonu podľa tímu, v ktorom pracuje. Výkon si môžu pracovníci medzi sebou kontrolovať.

R3 - Kontrola dochádzky – každému zamestnancovi sa bude kontrolovať dochádzka podľa pracovnej doby, ktorú má nastavenú. Budú sa sledovať rozdiely medzi zadanou dochádzkou a pracovnou dobou, ktorú má nastavenú.

R4 - Zadávanie mäkkej zložky – zamestnancom bude na konci mesiaca pridelená tím lídrom. Pracovný výkon v mäkkej zložke, ktorý nemá matematické podklady a je vecou subjektívneho názoru tím lídra. Táto zložka bude ovplyvňovať výpočet a získanie bonusu zamestnanca.

R5 - Výpočet bonusu – bonus sa bude počítat' na konci mesiaca podľa dosiahnutých bodov za výkon a zadanej mäkkej zložky. Na výpočet bonusu je vytvorený výpočet, ktorý určuje komu bude bonus pridelený a komu nebude. Tento bonus môžu získať iba pracovníci, ktorí zadávajú výkon. Zamestnanci, ktorí nedosiahnu minimálne 100% plnenie plánu, respektíve toto plnenie nedosiahne celý tím bonus nedostanú. Ďalšou možnosťou nepridelenia bonusu môže byť niektorá z K.O. (napr. pracovník je v skúšobnej dobe).

R6 - Evidencia zamestnancov – spoločnosť si udržiava prehľad o svojich zamestnancoch.

R7 - Evidencia činností – spoločnosť si eviduje činnosti, ktoré pracovníci vykonávajú. Ďalej sa evidujú zmeny na činnostiach pri zadaní nového bodového ohodnotenia činnosti.

R8 - Evidencia tímov – spoločnosť eviduje tímy, do ktorých sú pridelení pracovníci. Každý tím ma tím lídra, ktorý zodpovedá za svoj tím, pridáva, spravuje dochádzku a výkon svojím pracovníkom v tíme.

R9 - Analýza údajov – analýza agregovaných dát pre hodnotenie jednotlivca, tímu, celej pobočky, z hľadiska výkonnosti, finančnej náročnosti a efektívnosti.

Po analyzovaní požiadaviek a ich funkčnosti som vytvoril diagram prípadu použitia pre bonusový systém, ktorý pozostáva z jedného hlavného (viď. Obrázok 13) a jedného rozširujúceho diagramu pre evidenciu. (viď. Obrázok 14)

V hlavnom diagrame prípadu užitia je graficky naznačený vzťah actorov k jednotlivým prípadom použitia. Actor pracovník komunikuje s prípadmi užitia zadávania dochádzky, zadávania výkonu a výpočtom bonusu. Prípady užitia, zadávanie dochádzky

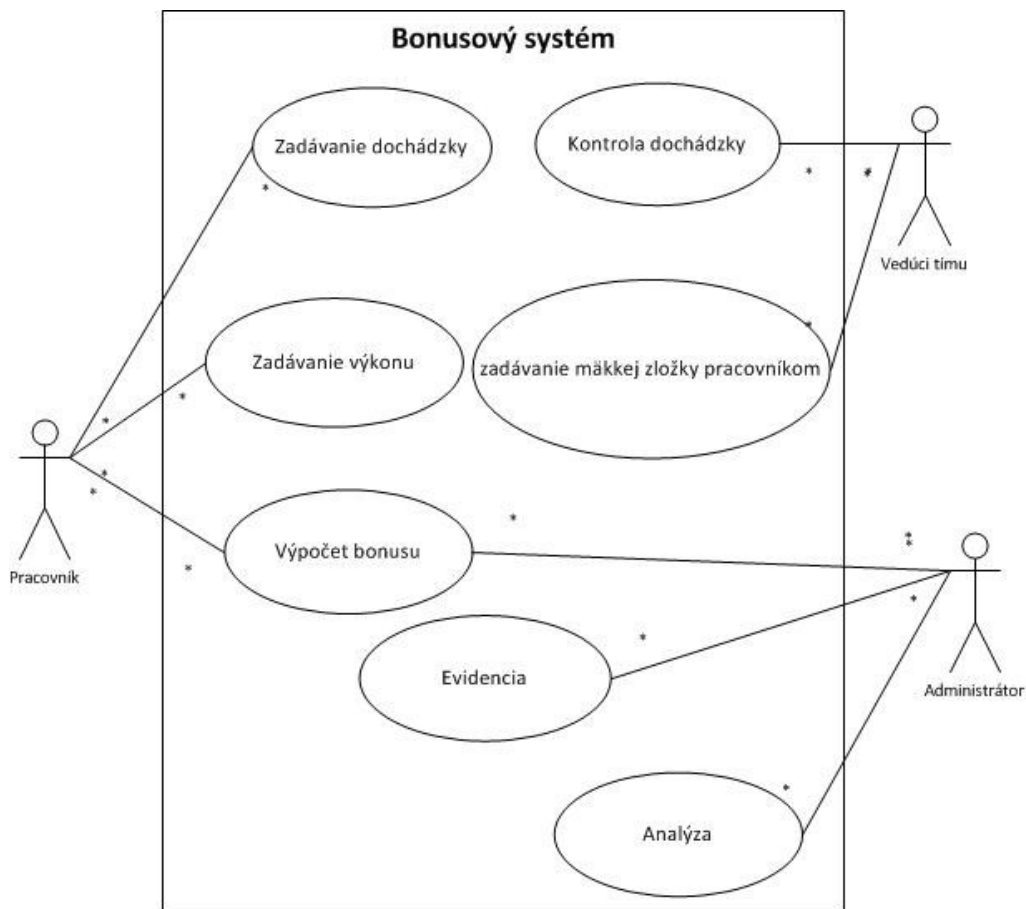
a zadávanie výkonu sú najdôležitejšie v systéme. Actor pracovník bude musieť denne plniť dáta, aby mu mohol byť pridelený bonus na konci mesiaca za jeho odpracovaný výkon.

Actor vedúci tímu komunikuje s dvoma prípadmi užitia a to kontrolou dochádzky v tíme a zadávanie mäkkej zložky pracovníkom. V priebehu mesiaca actor vedúci tímu kontroluje priebežne stav výkonu jednotlivých pracovníkov v sledovanom mesiaci a opravuje prípadne chyby a nedostatky. Na konci sledovaného obdobia pridá každému zamestnancovi ohodnotenie v podobe mäkkej zložky, ktorá hodnotí pracovníka na základe jeho schopností, empatie k zákazníkom, prácu v tíme a ďalších kritérií nezávislých na výkonnú zložku.

Actor administrátor vstupuje do prípadu užitia analýza a evidencia a tiež do výpočtu bonusu. Jeho úlohou je udržiavať systém v aktuálnom stave v závislosti na aktuálnej situácii v pobočke. Čiže aby zoznam zamestnancov a činností prislúchal realite. A tak mohol vyhodnocovať skutočnú situáciu pri výpočte bonusu a pri analýze dát.

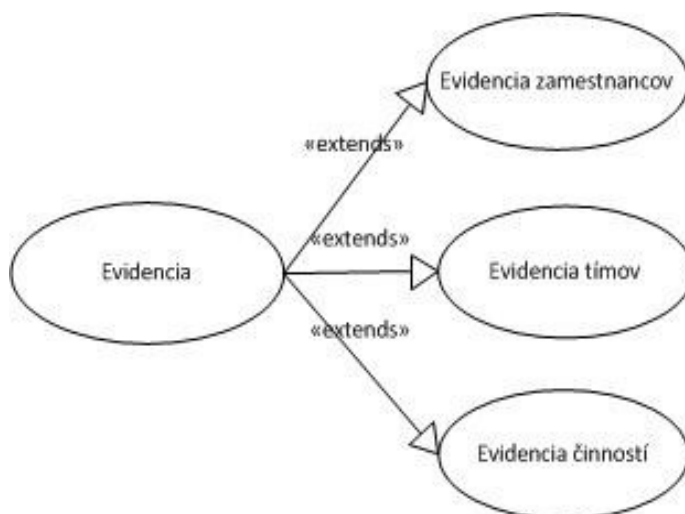
4.1.2 Diagram prípadu užitia Bonusového systému

Diagram prípadu užitia Bonusového systému (obrázok 13) zachycuje vzťah aktérov pracovník, vedúci tímu a Administrátor k jednotlivým prípadom užitia v navrhovanom systéme.



Obrázok 13: Hlavný diagram prípadu užitia bonusového systému

Rozširujúci diagram prípadu užitia evidencie zahŕňa evidenciu zamestnancov, evidenciu tímov a evidenciu činností. Celú evidenciu má na starosti iba jeden actor a to administrátor.



Obrázok 14: Rozširujúci diagram prípadu užitia evidencie

Identifikácia: UC1

Názov: Zadávanie dochádzky

Primárny aktér: Pracovník

Rozsah: IS

Úroveň: užívateľský cieľ

Účastníci a záujmy

Pracovník – mať kvalitnú aplikáciu na zadávanie dochádzky pre každý pracovný deň s možnosťou korekcie päť dní späť.

IS - prevádzky schopný

Vstupné podmienky

Pracovník bude priebežne každý pracovný deň zadávať svoju dochádzku a neprítomnosť. Po zadaní dochádzky si bude môcť overiť či pracoval načas alebo pracoval menej ako je jeho aktuálna pracovná doba.

Minimálne záruky

Pobočka si vedie dochádzku, aby mohla sledovať výšku nadčasov. Pracovník si môže nechať nadčas preplatiť alebo vybrať v náhradnom voľne.

Záruky úspechu

Pobočka bude schopná analyzovať dochádzku pracovníkov na sledovanie plánu celej divízie.

Spúšťač: Nový pracovný deň

Hlavný úspešný scenár:

1. Aplikácia identifikuje pracovníka podľa jeho ID.
2. 2.1: Pracovník zadá dochádzku za aktuálny pracovný deň.
2.2: Pracovník zadá neprítomnosť do dochádzky.
3. 3.1: Pracovník vyplní príchod a odchod.
3.2: Pracovník zadá dôvod neprítomnosti do dochádzky.
4. Pracovník uloží dochádzku.

Rozšírenie:

1a: Aplikácia neidentifikovala pracovníka

1a1: Pracovník nie je evidovaný v aplikácii.

1a2: Pracovník má zadané nesprávne ID.

1a3: Pracovník má zadanú nesprávnu pracovnú rolu.

2.1a: Pracovník nie je schopný zadať dochádzku za pracovný deň

2.1a1: Pracovnú dobu zadáva za nepracovný deň napríklad sviatok alebo víkend.

2.1.a1: Pracovník zadáva pracovnú dobu do dňa, v ktorom je už evidovaná jeho pracovná doba alebo neprítomnosť.

2.2.a: Pracovník nie je schopný zadať neprítomnosť do dochádzky

2.2.a1: Neprítomnosť zadáva do dňa, v ktorom je už zadaná iná neprítomnosť alebo pracovná doba.

2.2.a2: Neprítomnosť zadáva v nepracovný deň, napríklad sviatkov alebo víkend.

3.1.a: Pracovník nevyplní príchod a odchod

3.1.a1: Formát času a príchodu je nesprávny.

3.2.a: Pracovník nezadá dôvod neprítomnosti

3.2.a1: Pracovník nemá na výber požadovaný dôvod neprítomnosti.

4. Pracovník neuloží dochádzku

4.a Pracovník nevyberie dátum pre dochádzku.

4.b Pracovník stornuje zadávanie.

Identifikácia: UC2

Názov: Zadávanie výkonu

Primárny aktér: Pracovník

Rozsah: IS

Úroveň: užívateľský cieľ

Účastníci a záujmy

Pracovník – mať kvalitnú aplikáciu na zadávanie výkonu pre každý pracovný deň s možnosťou korekcie päť dní späť.

IS - prevádzky schopný

Vstupné podmienky

Pracovník bude priebežne každý pracovný deň, pokiaľ bol v práci, zadávať reálny výkon, ktorý odpracoval. Priebežne si bude môcť sledovať koľko položiek vykonal, koľko bodov získal, a koľko reálneho času strávil výkonom.

Minimálne záruky

Pobočka si vedie výkon, aby mohla sledovať výkonnosť pracovníkov, tímu a celej pobočky. Pracovníci si môžu navzájom kontrolovať výkon, aby boli výsledky pracovníkov transparentné.

Záruky úspechu

Pracovníkom bude možné na konci mesiaca vyhodnotiť ich pracovný výkon a prideliť im bonus. Pobočka bude schopná analyzovať výkon pracovníkov na sledovanie plánu celej divízie.

Spúšťač: Nový pracovný deň

Hlavný úspešný scenár:

1. Aplikácia identifikuje pracovníka podľa jeho ID.
2. Pracovník vyberie z ponuky položku, ktorú v daný deň vykonal.
3. Pracovník zadá počet položiek.
4. Uloží výkon.

Rozšírenie:

1a: Aplikácia neidentifikovala pracovníka

1a1: Pracovník nie je evidovaný v aplikácii.

1a2: Pracovník ma zadané nesprávne ID.

1a3: Pracovník má zadanú nesprávnu pracovnú rolu.

2.a: Pracovník nie je schopný zadať výkon za pracovný deň

2.a1: Položky, ktoré chce zadať nie sú na výber.

2.a2: Pracovník je nesprávne evidovaný.

3.a: Pracovník nevyplní počet položiek

3.a1: Výkon v danej položke bude nulový.

4. Pracovník neuloží výkon

4.a: Pracovník nevyberie položku.

4.b: Pracovník stornuje zadávanie.

Identifikácia: UC3

Názov: Zadávanie mäkkej zložky pracovníkom

Primárny aktér: Vedúci tímu

Rozsah: IS

Úroveň: užívateľský cieľ

Účastníci a záujmy

Vedúci tímu – mať kvalitnú aplikáciu na zadávanie mäkkej zložky pre každého pracovníka za mesačné obdobie, pre výpočet bonusu.

IS - prevádzky schopný

Vstupné podmienky

Vedúci tímu na konci mesačného obdobia pridelí jednotlivým pracovníkom tzv. mäkkú zložku, pomocou ktorej subjektívne hodnotí ich prístup k práci, k zákazníkom a k tímovej spolupráci. Taktiež môže prideliť K.O. kritéria, ktoré môžu pracovníkovi odňať bonus. Získané body majú 20% podiel na bonuse jednotlivca.

Minimálne záruky

Vedúci tímu si eviduje mäkkú zložku, aby mohol lepšie poukázať na pracovníkov, ktorí sú výkonnejší ako jeho kolegovia, a tým motivoval zvyšok tímu ku kvalitnejšiemu výkonu.

Záruky úspechu

Vedúci tímu pridelí zložku za sledované obdobie jednotlivým pracovníkom.

Spúšťač: Koniec kalendárneho mesiaca

Hlavný úspešný scenár:

1. Aplikácia identifikuje vedúceho tímu podľa jeho ID.
2. Vedúci tímu pridelí jednotlivým členom tímu body za každé kritérium obsiahnuté v mäkkej zložke.
3. Pracovníkovi sa pripočíta mäkká zložka k výkonu, z ktorého sa vypočíta dosiahnutý bonus.
4. Pridelenie odmeny za mäkkú zložku.

Rozšírenie:

1a: Aplikácia neidentifikovala vedúceho tímu

1a1: Vedúci tímu nie je evidovaný v aplikácii.

1a2: Vedúci tímu má zadané nesprávne ID.

1a3: Vedúci tímu má zadanú nesprávnu pracovnú rolu.

2.a: Vedúcemu tímu sa nezobrazujú všetci pracovníci z tímu

2.a1: Pracovníci nie sú správne pridelení do tímu.

2.a2: Pracovníci nie sú evidovaní.

3.a: Zle prepočítaná mäkká zložka

3.a1: Pracovník nedostane adekvátny bonus za mäkkú zložku.

4. Vedúci tímu nepridelí mäkkú zložku

4.a1: Pracovník nedostane z podielu za mäkkú zložku žiadny bonus.

Identifikácia: UC4

Názov: Kontrola dochádzky

Primárny aktér: Vedúci tímu

Rozsah: IS

Úroveň: užívateľský cieľ

Účastníci a záujmy

Vedúci tímu – priebežne kontrolovať pracovníkom dochádzku v priebehu sledovaného obdobia.

IS - prevádzky schopný

Vstupné podmienky

Korektne zadaná dochádzka pracovníkov.

Minimálne záruky

Pracovník zadal dochádzku za sledované obdobie.

Záruky úspechu

Korektne zadaná dochádzka bude označená za skontrolovanú.

Spúšťač: Zadaná dochádzka pracovníkom.

Hlavný úspešný scenár:

1. Aplikácia identifikuje vedúceho tímu podľa jeho ID.
2. Vedúci tímu si vyhladá pracovníka, ktorému chce skontrolovať dochádzku.
3. Po kontrole označí dochádzku za skontrolovanú.

Rozšírenie:

1a: Aplikácia neidentifikovala administrátora

1a1: Vedúci tímu nie je evidovaný v aplikácii.

1a2: Vedúci tímu má zadané nesprávne ID.

1a3: Vedúci tímu má zadanú nesprávnu pracovnú rolu.

2.a: Vedúci tímu nemôže nájsť pracovníka v systéme

2.a1: Pracovník nie je evidovaný v systéme.

3.a: Dochádzka nie je správne zadaná

3.a1: Vedúci neoznačí dochádzku za skontrolovanú.

Identifikácia: UC5

Názov: Výpočet bonusu

Primárny aktér: Administrátor

Rozsah: IS

Úroveň: užívateľský cieľ

Účastníci a záujmy

Administrátor – mať kvalitnú aplikáciu na zadávanie parametrov pre výpočet bonusu.

Pracovník – výsledok bonusu bude pridelený pracovníkovi k mzde.

IS - prevádzky schopný

Vstupné podmienky

Korektne zadaný výkon a mäkkú zložku pre sčítanie dosiahnutých bodov za mesačné obdobie. Korektne zadané vstupné parametre pre výpočet bonusu.

Minimálne záruky

Administrátor pridal parametre pre výpočet bonusu za príslušné mesačné obdobie.

Záruky úspechu

Z korektne zadaných dát je možné vypočítať bonus pre pracovníkom v závislosti od ich celkového výkonu.

Spúšťač: Koniec kalendárneho mesiaca

Hlavný úspešný scenár:

1. Aplikácia identifikuje administrátora podľa jeho ID.
2. Administrátor pridá parametre pre výpočet bonusu za príslušné obdobie pre jednotlivé tímy.
3. Administrátor spustí výpočet bonusu.
4. Administrátor pridelí bonusy jednotlivým pracovníkom.
5. Pracovník akceptuje výšku bonusu.

Rozšírenie:

1a: Aplikácia neidentifikovala administrátora

1a1: Administrátor nie je evidovaný v aplikácii.

1a2: Administrátor má zadané nesprávne ID.

1a3: Administrátor má zadanú nesprávnu pracovnú rolu.

2.a: Administrátor nezadal parametre pre výpočet bonusu na dané obdobie

2.a1: Nie je možné určiť výšku bonusu z dosiahnutých bodov.

3.a: Zle vypočítané bonusy

- 3.a1: Zle zadaný výkon, alebo mäkká zložka.
- 3.a2: Zle zadané parametre pre výpočet.
- 3.a3: Niektorí pracovníci nie sú evidovaní.
- 3.a4: Nesprávny výpočet bonusu.
- 4. Nepridelenie bonusu pracovníkom
 - 4.a1: Pracovník nedosiahol požadovaný počet bodov za obdobie.
 - 4.a2: Tím nedosiahol požadovaný počet bodov.
- 5. Pracovník neakceptuje výšku bonusu
 - 5.a1: Výška bonusu je neprimeraná výkonu.

Identifikácia: UC6

Názov: Evidencia

Primárny aktér: Administrátor

Rozsah: IS

Úroveň: užívateľský cieľ

Účastníci a záujmy

Administrátor – spravovať systém tak, aby jeho výpovedná hodnota bola totožná s realitou.

IS - prevádzky schopný

Vstupné podmienky

Korektne zadané dáta v systéme, aby neobmedzovali prácu so systémom.

Minimálne záruky

Administrátor evidoval všetky dáta do systému podľa aktuálneho stavu v pobočke.

Záruky úspechu

Pracovníci môžu korektne pracovať so zadanými dátami.

Spúšťač: Koniec kalendárneho mesiaca

Hlavný úspešný scenár:

1. Aplikácia identifikuje administrátora podľa jeho ID.
2. Administrátor eviduje dáta o pobočke v a činnostiach v systéme. Administrátor spustí výpočet bonusu.
3. Bezchybné využívanie dát.

Rozšírenie:

1a: Aplikácia neidentifikovala administrátora

- 1a1 Administrátor nie je evidovaný v aplikácii.
- 1a2 Administrátor ma zadané nesprávne ID.
- 1a3 Administrátor má zadanú nesprávnu pracovnú rolu.

2.a: Administrátor neeviduje nové dáta do systému

- 2.a1: Do systému sa zadávajú neplatné, alebo zastarané dáta.
- 2.a2: V systéme chýbajú dáta.

3.a: Chybné používanie dát

Identifikácia: UC7

Názov: Analýza

Primárny aktér: Administrátor

Rozsah: IS

Úroveň: užívateľský cieľ

Účastníci a záujmy

Administrátor – analyzovať dáta v systéme pre hodnotenie jednotlivcov, tímov, celej pobočky.

IS - prevádzky schopný

Vstupné podmienky

Evidované všetky dáta, zadaná dochádzka a výkon pracovníkov.

Minimálne záruky

Administrátor evidoval všetky dáta do systému podľa aktuálneho stavu v pobočke.

Záruky úspechu

Administrátor dokáže analyzovať situáciu v pobočke zo zadaných dát.

Spúšťač: Požiadavka na analýzu.

Hlavný úspešný scenár:

1. Aplikácia identifikuje administrátora podľa jeho ID.
2. Administrátor analyzuje dáta v aplikácii.
3. Administrátor prezentuje dáta v aplikácii.

Rozšírenie:

1a: Aplikácia neidentifikovala administrátora

- 1a1: Administrátor nie je evidovaný v aplikácii.
- 1a2: Administrátor ma zadané nesprávne ID.
- 1a3: Administrátor má zadanú nesprávnu pracovnú rolu.

2.a: Administrátor nedokáže analyzovať dáta

2.a1: Na analýzu je nedostatok dát v systéme.

2.a2: Vložená dáta sú v nesprávnom formáte a nedajú sa použiť.

3.a: Administrátor nedokáže prezentovať dáta v aplikácii

3.a1 Nástroj nepodporuje všetky požiadavky na analýzu.

4.1.3 Matica RTM (matica sledovanosti požiadaviek)

Je to nástroj, ktorý nám komplexne zhodnotí celistvosť projektu. Vyhodnocuje či existuje požiadavka, ktorá by neriešila žiaden z prípadov užitia v navrhnutom systéme a zároveň či existuje prípad užitia, ktorý nerieši žiadny z požiadaviek zadávateľa. Pokiaľ by sme našli nespracovanú požiadavku museli by sme návrh prepracovať. V našom prípade sú všetky požiadavky pokryté niektorým z prípadov užitia, čo by malo zaručiť plnenie predstáv zadávateľa.

		Prípady užitia						
		UC1	UC2	UC3	UC4	UC5	UC6	UC7
Požiadavky	R ₁	x						
	R ₂		x					
	R ₃			x				
	R ₄				x			
	R ₅					x		
	R ₆						x	
	R ₇						x	
	R ₈						x	
	R ₉							x

Tabuľka 6: Matica RTM

4.1.4 Analytický model tried

V ďalšej časti návrh bonusového systému chcem vytvoriť diagram tried, na ktorý využijem jednoduchú pomôcku pri analyzovaní tried, atribútov a metód. Vypracujeme si slovné zadanie, kde vyznačím všetky podstatné mená. Tie budú reprezentovať triedy a ich atribúty. Ďalej si vyznačíme všetky slovesá, ktoré by mali v danej triede reprezentovať metódy. Z následnej pomôcky si vytvoríme diagram tried.

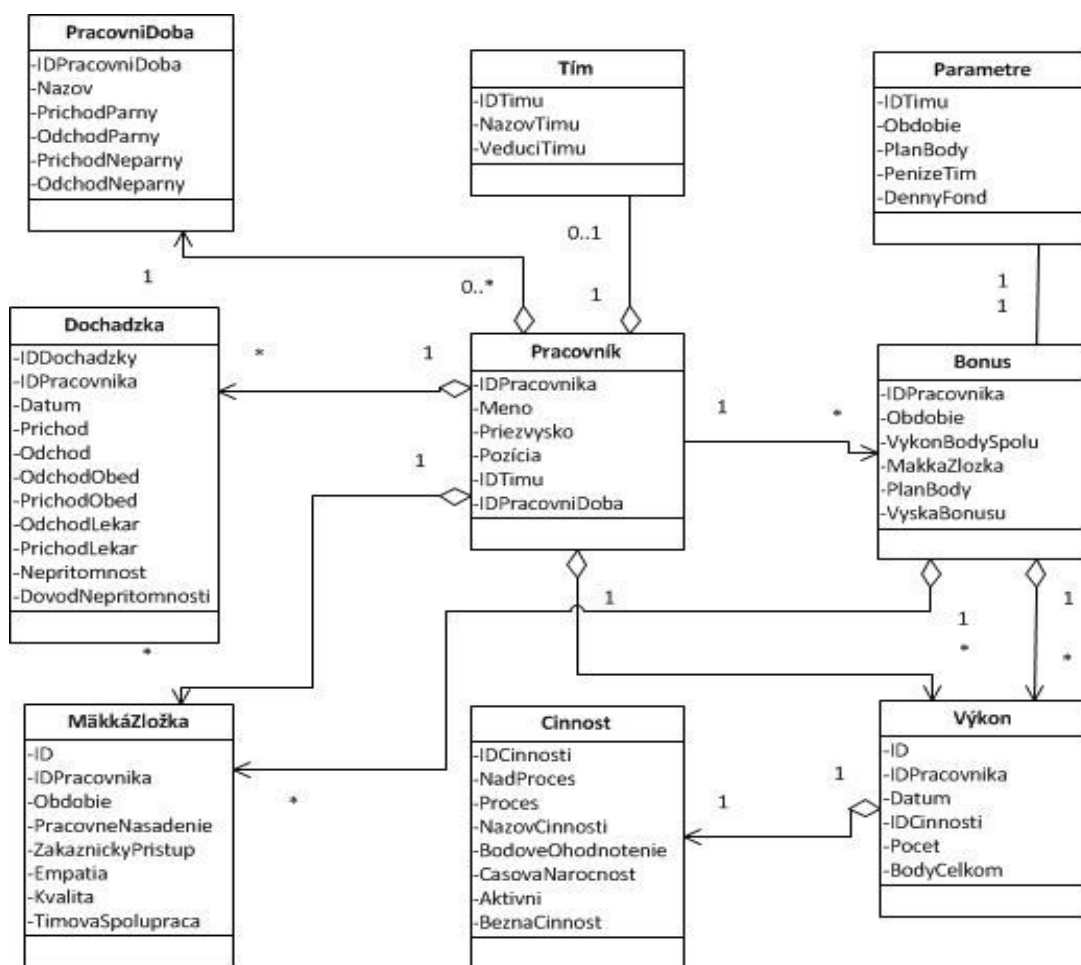
Analýza **podstatných mien** a *slovies*:

Pracovník si v nový pracovný **deň** *zadá* svoju **dochádzku** s **príchodom** a **odchodom** na pracovisko. Pokiaľ bol **pracovník** na obede tak *vyplní* **odchod** a **príchod** na **obed**. **Pracovník** má možnosť *vyplniť* **prerušenie** práce buď z dôvodu návštevy **lekára** alebo z iného **dôvodu**, *musí* toto **prerušenie** zadať do **dochádzky**. Pracovník si môže celodennú **neprítomnosť** *zadať* neobmedzene do budúcnosti. V pracovný **deň** si **pracovník** *zadáva*

odpracované **činností** a ich **počet**. Každá **činnosť** má určenú výšku **bodov** a časovú **náročnosť**.

Vedúci tímu na konci sledovaného obdobia **zadáva** **pracovníkom** z jeho **tímu** **mäkkú zložku**, kde *prideluje* výšku **bodov** za **pracovné nasadenie**, **prístup k zákazníkovi**, **empatiu**, **kvalitu**, **tímovú spoluprácu**. **Vedúci tímu** sa *stará* o transparentnosť **dochádzky** a **výkon pracovníkov**.

Administrátor *eviduje* zoznam **tímov**, kde *zadá* **názov tímu** a kto je **vedúci tímu**. Ďalej *eviduje* zoznam **pracovníkov**, kde je jeho **ID**, **meno**, **priezvisko**, **pracovná rola** (Pracovník, Vedúci tímu, Administrátor), v akom **tíme pracuje**, **názov pracovnej pozície**. Ďalej *eviduje* zoznam **činností** podľa, ktorej **Pracovník** *zadáva* svoj **výkon**. Na začiatku sledovaného **obdobia** *zadá* všetky **parametre**, **plán bodov** za obdobie, výška finančnej **odmeny** na tím, denný **časový fond** nutný k **výpočtu bonusu**. V systéme sa *evidujú* všetky **sviatky** a **voľná**, aby v tento deň pracovníci nemohli *zadávať* **dochádzku** a **výkon**. Ďalej sa *eviduje* **pracovné doby** pre **pracovníkov**, podľa ktorých sa *sleduje* či pracovník v daný deň *dosiahol* **nadčas** alebo pracoval menej.



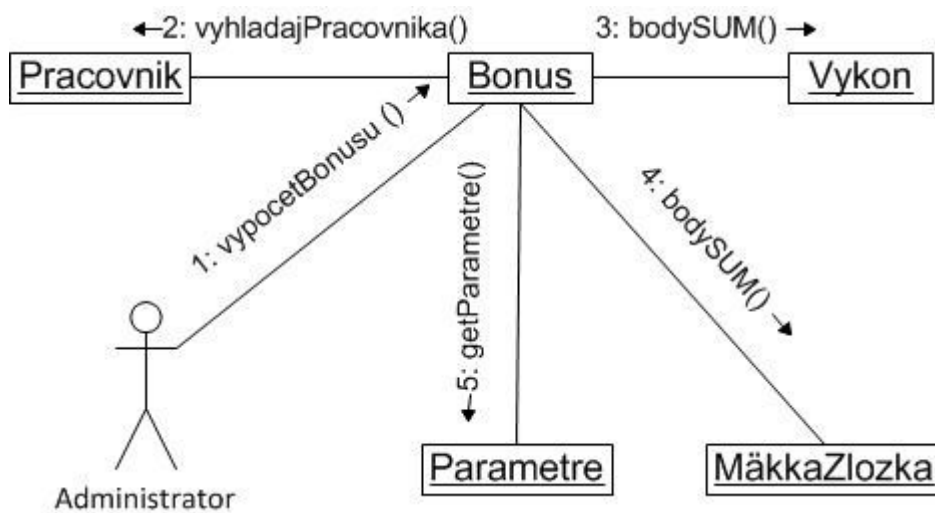
Obrázok 16: Diagram tried

4.1.5 Realizácia prípadu užitia

Po analýze tried je ďalšou kľúčovou časťou pri návrhu systému realizovať prípady užitia. V mojej diplomovej práci budem prezentovať návrh hlavného prípadu užitia UC5 – výpočet bonusu. Diagram názorne naznačuje spoluprácu skupín objektov, tak aby sme dosiahli chovania špecifikovaného v prípade užitia.

Komunikačný diagram

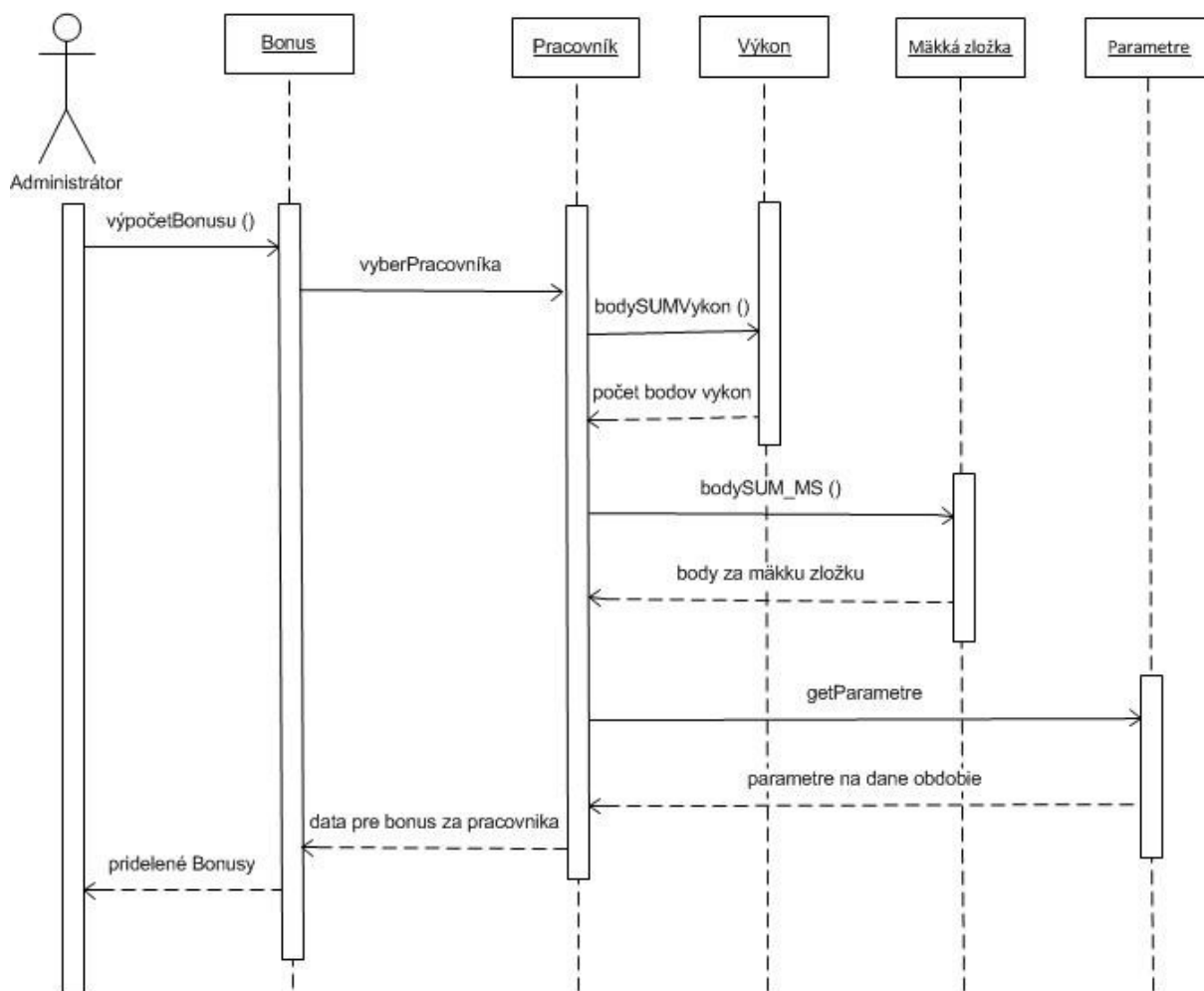
Komunikačné diagramy zdôrazňujú interakciu pomocou spojenia životných čiar. Číslovanie nám naznačuje postupnosť vykonávania jednotlivých činností. Na obrázku je jasné že celý proces výpočtu bonusu spúšťa Administrátor, ktorý má záujem dostať vypočítané bonusy pre pracovníkov.



Obrázok 16: Komunikačný diagram prípadu užitia UC5 – výpočet bonusu

Sekvenčný diagram

V tomto diagram graficky vyjadrujem sekvenčné vzťahy medzi činnosťami pri realizácii výpočtu bonusu. Postup vyjadruje, že pokiaľ nedostane všetky údaje za výkon, mäkkú zložku a parametre, nie je možné dosť výpočet za pracovníka.

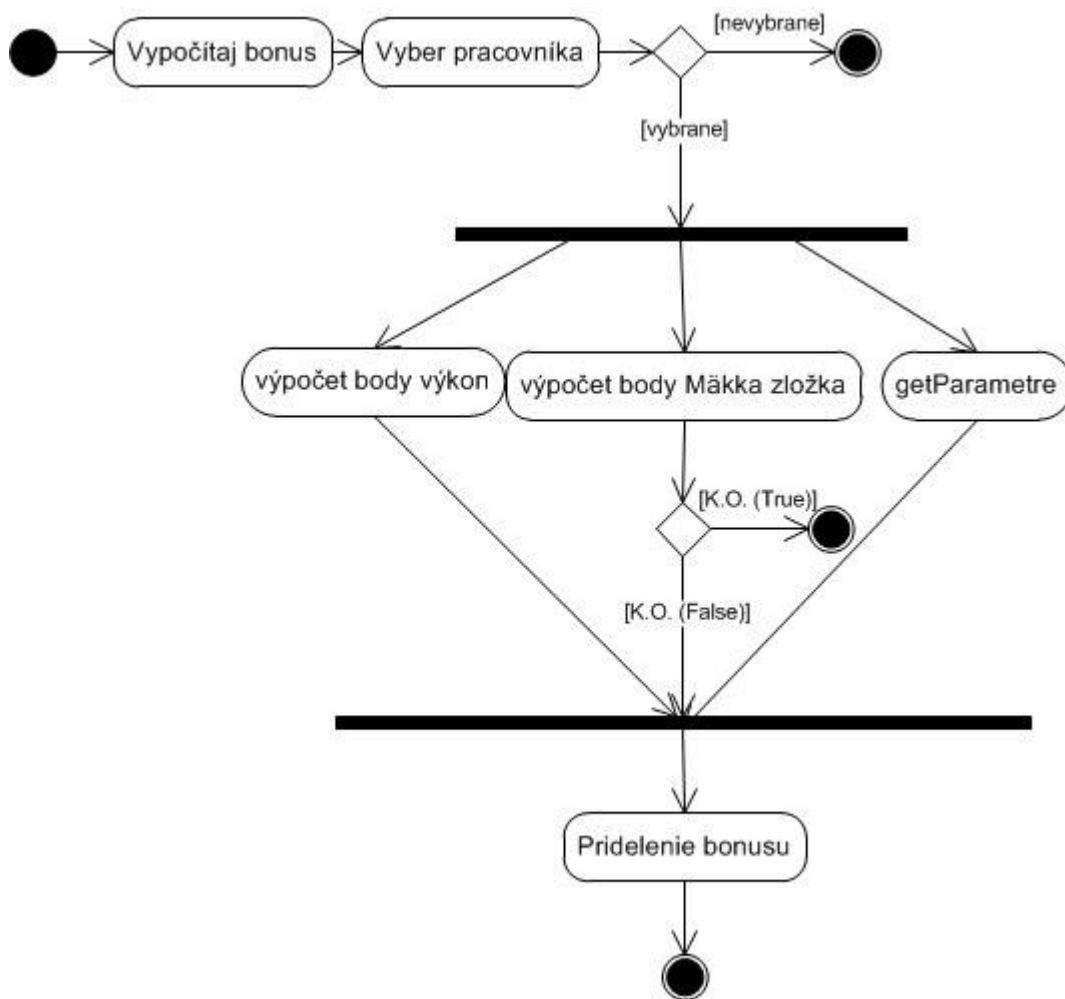


Obrázok 17: Sekvenčný diagram prípadu užitia UC5 – výpočet bonusu

Diagramy aktivít

Diagram aktivít nám graficky popisuje tok pracovný postup s obmedzujúcimi kritériami. V UC5 výpočet bonusu môže nastať viacero prípadov neúspešného výpočtu bonusu. Tento diagram nám zobrazuje logické vyhodnocovanie aktivít.

V diplomovej práci ilustrujem iba jeden diagram aktivít, ktorý nám graficky popisuje realizáciu prípadu užitia UC5 – výpočet bonusu.



Obrázok 18: Diagram aktivít prípadu užitia UC5 - výpočet bonusu

4.2 Realizácia bonusového systému v prostredí Microsoft Access

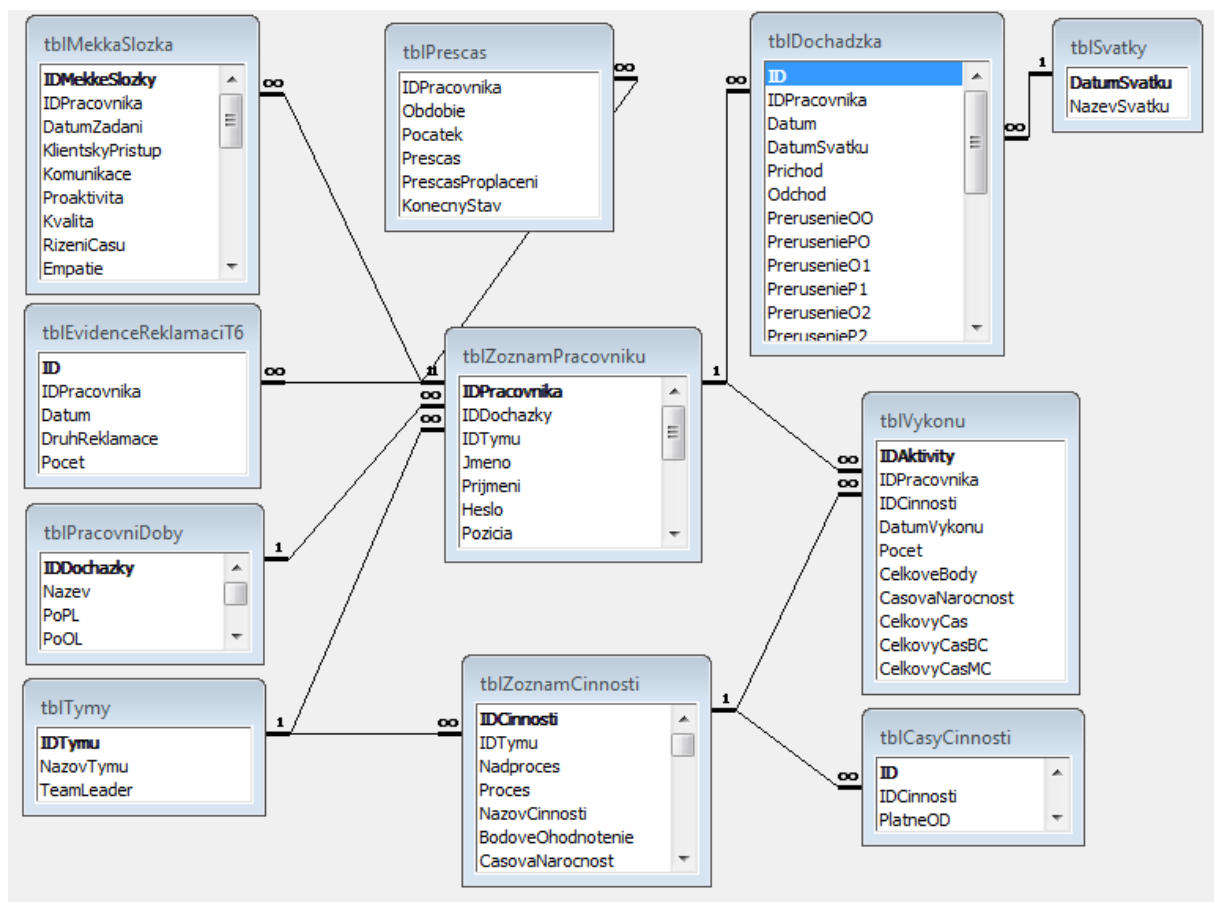
Po návrhu bonusového systému v prostredí Microsoft Visio sa v druhej časti tejto kapitoly budem venovať realizácii (programovaniu) databázovej aplikácie vo vývojovom prostredí Microsoft Access 2003.

Tento nástroj som si vybral z viacerých dôvodov. Jedným z najhlavnejších dôvodov je bezpečnostná politika. Aplikácie v Microsoft Access majú pridelenú výnimku pre používanie a vyvíjanie aplikácií vo všetkých pobočkách spomínanej banky. Ak by sa mal implementovať iný systém ako je schválený v smerniciach o bezpečnosti, musel by byť tento systém testovaný a schválený, čo by mohlo priniesť výrazné časové oneskorenie zavedenia aplikácie do prevádzky. Ďalším dôvodom pre výber prostredia je, že vyhovuje požiadavkám navrhnutého systému a každú z nich je Access schopný realizovať. Taktiež užívatelia majú

skúsenosti s aplikáciami v tomto prostredí, čím sa systém jednoduchšie infiltruje do prostredia.

4.2.1 Relačný model

Na začiatku realizácie databázového systému som vytvoril relačnú databázu, v ktorom som definoval všetky tabuľky, atribúty, obmedzenia a vzťahy medzi tabuľkami. Pri návrhu relačnej databázy som využil grafický nástroj na tvorbu modelu v prostredí Microsoft Access.



Obrázok 19: Relačný model bonusového systému

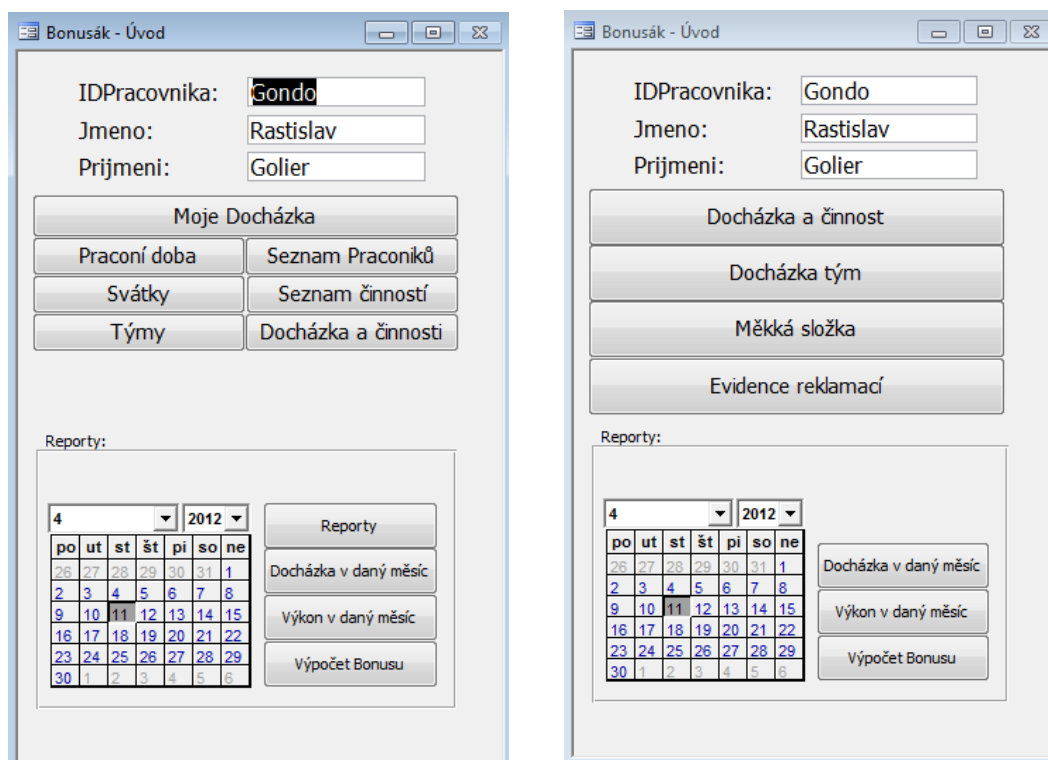
Po návrhu relačného modelu som databázovú aplikáciu rozdelil na dve časti. Na databázovú a aplikačnú. Databázová časť bude umiestnená na lokálnom servere pobočky, kde budú mať všetci pracovníci prístup. Druhá aplikačná časť bude u každého pracovníka na jeho lokálnej pracovnej stanici. Pri rozdeľovaní databázy nám Access vytvoril v aplikačnej časti odkazy na tabuľky. Pri každej zmene umiestnenia databázovej časti sa musí obnoviť cesta k tabuľkám. Takúto aplikáciu môžeme nazvať „Two – layer“, lebo sa skladá iba z dvoch vrstiev. V tomto prípade aplikačná vrstva zahŕňa ako GUI, tak celú bussines logiku.

4.2.2 Tvorba formulárov

Po návrhu databázovej časti aplikácie som začal tvorbu formulárov, pomocou ktorých zabezpečím rozhranie medzi databázou a užívateľom (tzv. GUI). V prostredí Microsoft Access má formulár viacero úloh. Pomocou formulára môžem riadiť tok aplikácie, prijať vstup, editovať a tlačiť informácie. Access nám ponúka viacero spôsobov tvorby formulára. Najjednoduchšou cestou je použiť tzv. sprievodcu, ktorý po pár kliknutiach Vám vytvorí funkčný formulár. Táto cesta je síce najjednoduchšia, ale nemusí spĺňať všetky požiadavky. Preto som všetky formuláre tvoril výlučne v návrhovom zobrazení. Tento spôsob je zdĺhavejší, ale o to efektívnejší pri dodržaní všetkých užívateľských požiadaviek. Pri tvorbe formulárov som využil programovací jazyk VBA. Tento nástroj, ktorý je vytvorený spoločnosťou Microsoft dodáva formulárom, ale aj iným komponentom množstvo funkcionalít, ktoré automatizujú business logiku. Tento programovací jazyk robí z Accessu kvalitný nástroj na tvorbu podnikových aplikácií.

Prvý formulár, ktorý užívateľ uvidí pri štarte aplikácie je formulár frmUvod. Tento formulár rozpozná podľa IDPracovnika, aká je jeho užívateľská rola. Podľa toho sa mu zobrazia všetky tlačidlá, ktoré je oprávnený používať. Významnú úlohu zohráva aj pri autorizácii užívateľov. Aplikácia nevyužíva klasické prihlasovanie pomocou ID a hesla.

Problém s autorizáciou užívateľov som prenechal na autorizáciu pri prihlasovaní do pracovnej stanice. Pobočka má systém užívateľských mien, pomocou ktorých sa užívatelia prihlasujú. Je teda viac menej pravdepodobné, že pri počítači sedí daný pracovník za predpokladu, že sa k pracovnej stanici prihlásil sám. Moja aplikácia pri štarte zistí názov prihláseného užívateľa od systému Windows pomocou jednoduchého príkazu „IDPracovnika2 = (Environ\$(\"Username\"))“, a tým autorizuje pracovníka a zistí jeho užívateľskú rolu. Od tohto momentu pracuje aplikácia s ID prihláseného pracovníka a nie je možné ho zmeniť. Aby mohol nový užívateľ pristupovať do aplikácie musí mu byť vytvorené konto s takým ID, ktorým sa prihlasuje na pracovnú stanicu.



Obrázok 20: Formulár úvod (vľavo administrátor v pravo pracovník)

Na úvodnom formulári môžeme nájsť prístup k rôznym formulárom. Ako vidíme vľavo na obrázku 19, administrátor má prístup k všetkým formulárom na evidenciu. V pravo je zobrazený úvod pre pracovníka, kde má možnosť pristupovať iba k formulárom a reportom, ktoré sa ho priamo týkajú.

Po autorizácii užívateľa vo formuláry, si každý užívateľ zadáva dochádzku. V aplikácii nájdeme dva druhy formulárov na zadávanie dochádzky. Prvý formulár je pre pracovníkov, ktorý k dochádzke zadávajú aj výkon. Druhý je pre vedúceho tímu a administrátora, ktorý sú povinný zadávať dochádzku, ale nezadávajú výkon. Každý pracovník si má zadávať dochádzku buď v daný deň, alebo s maximálnym päť denným oneskorením. Do budúcnosti si môže zadať iba neprítomnosť, čo môže byť (dovolenka, PN, náhradné voľno a iné.). Každá nová dochádzka je sledovaná podľa pracovnej doby, ktorú má pracovník nastavenú. Štandardná je 7:00 príchod a 15:30 odchod od pondelka do piatku. Niektorí pracovníci ale majú výnimku. Napríklad v niektoré dni má pracovnú dobu dlhšiu alebo kratšiu. Z tohto dôvodu bolo nutné vytvoriť tabuľku s pracovnými dobami, aby sa dala sledovať dochádzka a nadobudnutý nadčas prípadne neplnenie pracovnej doby. Zadávanie dochádzky som ošetril, ak by chcel pracovník zadať dochádzku na víkend alebo na štátny sviatok. Pracovník má možnosť vyplniť do dochádzky opustenie pracoviska a to z dôvodu návštevy lekára alebo iných dôvodov. Návšteva lekára sa neodčíta od dochádzky, ale eviduje sa, aby bol prehľad o pohybe pracovníka. V priebehu dňa si môže pracovník zadať poldennú

neprítomnosť. Táto neprítomnosť má dĺžku iba štyri hodiny. Po uložení dochádzky sa prepočíta dĺžka pracovnej doby, dĺžka neprítomnosti a iné parametre, potrebné pre sledovanie dochádzky zamestnancov.

K príslušnému dátumu v dochádzke si pracovníci zadávajú výkon za aktuálny deň. Vo formuláre frmDochádzka som vytvoril pod formulár sbfVýkon, ktorý je prepojený podľa IDPracovníka a dátumu s frmDochádzka. Vo frmDochádzka som vytvoril aj štatistiku výkonu. Pracovník môže sledovať počet bodov za deň, mesiac, percentuálne plnenie plánu (viď. Príloha 1). Zadávanie výkonu je dôležité pre výpočet bonusu, na ktorom pracovníkom záleží, preto bolo nutné venovať výpočtu bodov náležitú pozornosť.

Pre výpočet bonusu je dôležité správne zadať mäkkú zložku pracovníkom. Zadávanie mäkkej zložky majú na starosti vedúci tímov. Táto činnosť sa vykonáva na konci sledovaného obdobia, kedy sa hodnotí každý pracovník subjektívne. Jedným z hlavných požiadaviek na formulár mäkkej zložky bolo, aby mal vedúci tímu možnosť zadávať parametre všetkým pracovníkom z jedného tímu súčasne na jednom formulári (viď Príloha 2). Mäkká zložka má 20% podiel na výške bonusu. Tento podiel nie je malý a môže ovplyvniť výšku bonusu u ostatných pracovníkov. Aby si pracovníci z jedného tímu mohli sledovať výšku mäkkej zložky vytvoril som im formulár iba na sledovanie informácií.

Ďalším dôležitým formulárom je formulár činností. Správu činností má na starosti administrátor. Zoznam činností sa pravidelne aktualizuje, a treba aby bol vždy aktuálny pre nové obdobie výkonu. Problém nie je len v tom, že sa pridávajú nové činnosti, ale že sa staré činnosti menia.

Aby sme pochopili lepšie tento problém, musíme si vysvetliť ako činnosti a výkon fungujú. Každá činnosť, ktorú pracovník vykonáva má svoju časovú náročnosť, ktorú zistíme meraním časovej náročnosti na činnosť pri výkone. Z hodnoty časovej náročnosti sa potom počíta výška bodov pomocou jednoduchého vzorca „ $BodoveOhodnotenie = CasovaNarocnost / 420 * 1000$ “.

Takto prepočítané body sa odzrkadľujú vo výkone pracovníkov. Jednou z požiadaviek je, aby sa pri zmene časovania činností, ponechali predchádzajúce údaje. Aby som vyhovel tejto požiadavke, vytvoril som formulár na zadávanie nového časovania. Pri zadaní nového časovania sa zadáva aj dátum platnosti od akého je časovanie platné. Tento dátum určuje aké bodové ohodnotenie je aktuálne pri zadávaní výkonu. Tento systém vylučuje problém pri úprave výkonu v starom mesiaci, kedy bude výkon počítaný s predchádzajúcimi údajmi. (Prílohy 3,4)

4.2.3 Tvorba reportov

Pri návrhu bonusového systému bola jedna z požiadaviek zadávateľa vytvoriť analytické reporty. Na tvorbu reportov disponuje Access viacerými možnosťami. Najznámejšou prezentáciou dáta je vytvoriť zostavu, ktorá ponúka bohaté grafické nástroje. Ďalšou z možností je vytvoriť pivotnú tabuľku alebo graf. Jedno majú všetky reporty spoločné, že ich dátová základňa stojí na SQL dotaze.

Potom ako som dal testovať prvú verziu aplikácie do pobočky, som následne dostal zoznam reportov na analýzu a sledovanie dát. Hlavným reportom v bonusovom systéme je výpočet bonusu. Tento report som vytvoril ako prvý, aby sme mali možnosť otestovať jeho funkčnosť v porovnaní so starým systémom v Excely. Report som vytvoril v Access zostave. Prvým krokom bolo nutné vytvoriť dotaz, kde selektujem dáta za vybrané mesačné obdobie, pre všetkých pracovníkov v pobočke rozdelených do tímov. V dotaze je ďalej vygenerovaná výška výkonu a mäkkej zložky za daný mesiac, suma ktorá sa bude pre tím rozdeľovať v pomere 80:20 (Výkon : Mäkká zložka) a plán bodov za obdobie.

Výpočet bonusu je nasledovný:

- Ak máme výšku výkonu pracovníka za obdobie, tak ho podelíme s celkovým výkonom tímu. Tento pomer nám určí aký nárok na finančnú odmenu bude mať pracovník z výkonnej zložky.
- Ak máme výšku mäkkej zložky za obdobie, tak ho podelíme celkovou výškou dosiahnutých bodov v tíme. Tento pomer nám určí výšku finančnej odmeny za mäkkú zložku pre pracovníka.
- Sčítaním finančnej odmeny za výkonovú a mäkkú zložku nám dáva výslednú finančnú odmenu pre pracovníka.
- Či finančnú odmenu pracovník dostane, záleží na tom, či dosiahol tím požadované plnenie. Plnenie tím dosiahne, ak je plán výkonu minimálne splnený na 100%. Taktiež pracovník nemusí dostať bonus, pokiaľ by jeho osobné plnenie nedosiahlo minimálne 100%. Poslednou možnosťou prečo by nemal dostať pracovník bonus je, ak by mal v mäkkej zložke vyplnené jedno z K.O. kritérií (napr. skúšobná doba, zaučenie sa, vysoká chybovosť, porušenie pracovného poriadku).
- Peniaze ktoré sa pracovníkom nepridelia, z niektorých dôvodov sa neprerozdeľujú. (viď. Príloha 5)

Na analyzovanie pobočky som vytvoril ďalšie reporty. Predstavíme si report, ktorý sa nazýva kumulácia. Jeho náplňou je sledovať vykonávané činnosti v pobočke v mesačných intervaloch. Sledujeme koľkokrát bola činnosť vykonaná a koľko reálneho času sa na nej strávilo (v hod). Report som vytvoril pomocou SQL dotazu, ktorý prezentujem ako pivot tabuľku (viď Obrázok 21)

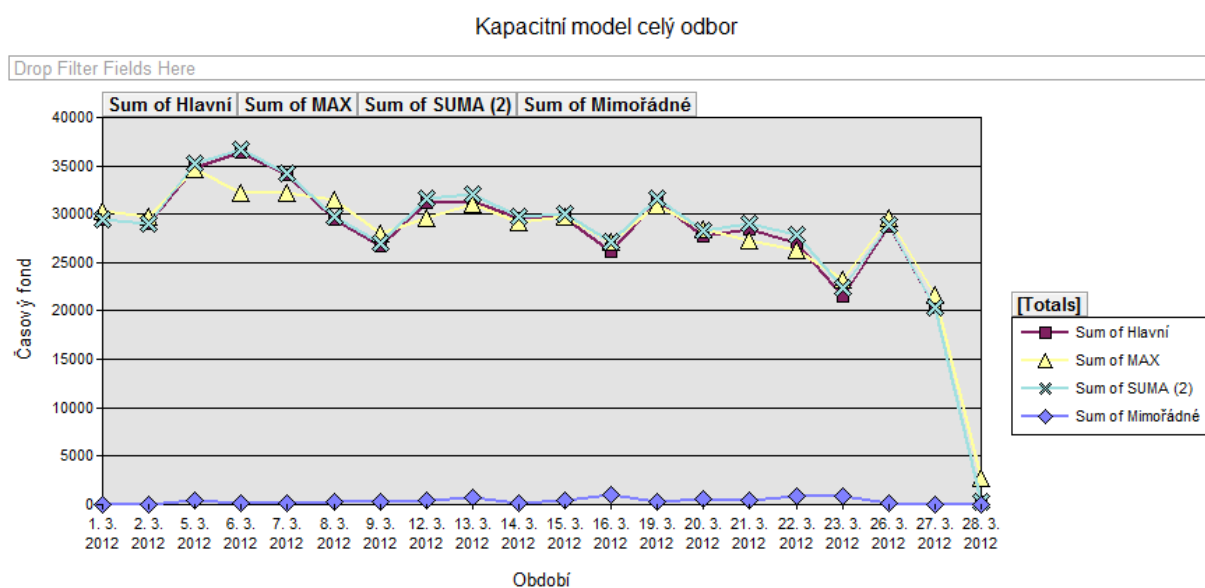
qryKumulaceExcel : Select Query

Drop Filter Fields Here

		Měsíc	Rok				
		2		3			
		2012		2012			
IDTymu	NazovCinnosti	Tmin	Položky	Túvazek	Tmin	Položky	Túvazek
	MO - kontrola v CRM	2	34	0,14	2	709	2,95
	MO - kontrola žádosti, ověření podpisů	2	48	0,20	2	1024	4,27
	MO - korespondence	3	39	0,24	3	902	5,64
	MO - nastavení benefitu a uzavř. podání	3	2	0,01	3	78	0,49
	MO - operativní evidence	5	39	0,41	5	905	9,43
	MO - průvodka	5	6	0,06	5	144	1,50
	MO - reklamáce				15	19	0,59
	MO - rušení karty (1pol./1účet)				1,5	17	0,05
	MO - rušení účtu s produkty				3	17	0,11
	MO - TP + SI zprac.	8	37	0,62	8	843	14,05
	MO - uzavření podání + dopl.evidence				1	335	0,70
	MO - zadání podání CRM + SCAN	3	47	0,29	3	983	6,14
	MO - změna inkasního účtu UFO	2	3	0,01	2	74	0,31

Obrázok 21: Report kumulácie

Ďalším dôležitým reportom pre sledovanie výkonnosti pracovníkov je kapacitný model pre celú pobočku. Tento report som vytvoril aj vo verzii pre každý tím. Report sleduje výkon z časového hľadiska za každý pracovný deň. V grafe môžeme vidieť aký podiel majú bežné a mimoriadne činnosti na výkone (viď Obrázok 22)



Obrázok 17: Kapacitný model

4.2.4 Implementácia systému

V mojom prípade implementácie som postupoval podľa agilných konvencií. Systém som nasadil už pri vytvorení prvej verzie bonusového systému. Hlavným dôvodom nasadenia neúplného systému bolo, aby si užívatelia navykli na nový systém a mohli tak systém testovať a hodnotiť. V priebehu testovania som vyvíjal ďalšie moduly, opravoval zistené chyby a prispôboval systém užívateľským požiadavkám.

Aplikácia je rozdelená na 2 časti. Prvá databázová časť je implementovaná na lokálnom serveri, ku ktorému majú prístup všetci pracovníci. Druhá aplikačná časť je rozmnožená po všetkých pracovných staniach, aby všetky výpočty boli prenesené na stranu užívateľa. Vo fáze vývoja bola aj aplikačná časť uložená na strane servera a užívatelia mali v pracovnej stanici vytvoreného zástupcu. Tento prístup distribúcie mal negatívny vplyv na výpočty v aplikácii, ale uľahčoval implementáciu nových verzii bez toho, aby sa na každej stanici musela nahráť nová verzia.

5 ZHODNOTENIE NAVRHOVANÉHO RIEŠENIA

Po implementácii nového bonusového systému sa výrazne znížila pracnosť na spracovanie výkonu a dochádzky. Nový systém centralizuje dáta na jednom mieste v pobočke a automatizuje výpočet bonusu pre každého pracovníka. Aplikácia priniesla nové možnosti pri analýze dát, a tým možnosť plánovať a zefektívniť výkon pracovníkov, tímov, respektíve celého útvaru.

Aplikácia sa nasadila do ostrej prevádzky a vymenila tak predchádzajúci bonusový systém. Je možné, že sa aplikácia bude rozširovať o nové funkcionality, aby sa čo najlepšie využil jej potenciál. Za hlavného kandidáta pri rozširovaní pokladám analýzu dát. Aplikácia bude obsahovať veľké množstvo dát, ktoré môžu priniesť pri správnej prezentácii veľké kapacitné aj finančné úspory v celej pobočke.

6 ZÁVER

V úvode mojej diplomovej práce som si definoval cieľ, ktorým bolo navrhnúť, vytvoriť a implementovať bonusový systém pre jednu z pobočiek nemenovanej banky. Následne by som chcel zhodnotiť, ako som tento cieľ dosiahol.

Skôr, ako som začal tvoriť aplikáciu, som analyzoval stav bonusového systému v pobočke. Analýza sa zaoberá dôvodom vzniku bonusového systému, jeho transformáciou v súčasný bonusový systém, až po jeho podobu v prostredí Microsoft Excel. Po analýze súčasného boli analyzované užívateľské požiadavky. Z požiadaviek bol navrhnutý Use Case diagram, ktorý graficky zachycuje hlavné prípady použitia v bonusovom systéme. Celý návrh aplikácie bol zachytený v UML notácii, aby bola uľahčená tvorba aplikácie.

Návrh v jazyku UML pripravil aplikáciu na realizáciu. Prvým krokom realizácie bola tvorba dátového modelu. Ďalším krokom bolo navrhnúť formuláre a ich funkčnosť podľa užívateľských požiadaviek z diagramu Use Case. Aby mala aplikácia výpovednú hodnotu bolo nutné vytvoriť množstvo reportov zvlášť pre pracovníkov, vedúcich tímov a administrátora, ktoré priebežne analyzovali situáciu v pobočke. Aplikácia bola vyvíjaná agilne. Už prvá verzia bola nasadená do prevádzky, aby bolo možné čo najefektívnejšie prispôbiť aplikáciu ďalším užívateľským požiadavkám.

V súčasnosti sa aplikácia používa v ostrej prevádzke a tak úspešne nahradila predchádzajúci bonusový systém. Táto skutočnosť mi iba potvrdzuje to, že som dosiahol moje ciele, ktoré som si stanovil v úvode diplomovej práce.

ZOZNAM POUŽITÉJ LITERATURY

Knihy

1. ARLOW, Jim a NEUSTADT, Ila. *UML 2 a unifikovaný proces vývoje aplikací: průvodce objektově orientovaného softwaru*. Brno: Computer Press, 2007. ISBN 978-80-251-1503-9.
2. ARMSTRONG Michael. *Odměňování pracovníků*. Praha: Grada Publishing a.s., 2009. ISBN 978-80-247-2890-2.
3. KADLEC, Václav. *Agilní programování : Metodiky efektivního vývoje softwaru*. Brno : Computer Press, 2004. ISBN 80-251-0342-0.
4. TVRDÍKOVÁ, Milena. *Aplikace moderních informačních technologií v řízení firmy*. Praha: Grada Publishing. a.s., 2008. . ISBN 978-80-247-2728-8.
5. VIASCAS, L. John. *Mistrovství v Microsoft Office Access 2003*. Brno: Computer Press, 2005. ISBN 978-80-251-0537-5
6. WROTHWELL, Aileen. *Microsoft Office Access 2003 Step by Step / Online Training Solutions, Inc*. Redmond, Washington: Microsoft Press, ISBN 0-7356-1517-9

Internetové zdroje

7. *How to count number of rows in a subform*. [online] 2007-05-11 [cit. 2012-03-02] Dostupné z WWW: <<http://bytes.com/topic/access/answers/645416-how-count-number-rows-subform>>
8. *How to get logged-in user's name in Access vba*. [online] 2009-03-24 [cit. 2012-02-23] Dostupné z WWW: <<http://stackoverflow.com/questions/677112/how-to-get-logged-in-users-name-in-access-vba>>
9. KNEŠL, Jiří. *Agilní vývoj: Úvod*. [online] 2009-12-2009 [cit. 2012-03-07]. Dostupné z WWW: <<http://www.zdrojak.cz/clanky/agilni-vyvoj-uvod/>>

10. *Spirálový model*. [online] 2009 [cit. 2012-04-02]. Dostupné z WWW: <<http://testovanisoftwaru.cz/manualni-testovani/modely-zivotniho-cyklu-software/spiralovy-model/>>
11. TIŠNOVSKÝ, Pavel. *Nástroje pro tvorbu UML diagramů* [online]. 1998 , 2009-04-08 [cit. 2012-03-06]. Dostupné z WWW: <<http://www.root.cz/clanky/nastroje-pro- tvorbu-uml-diagramu/>>
12. MYSLIVEČEK, Ondřej. *Agilní vývoj v praxi* [online]. 2011, [cit. 2012-03-22] Dostupné z WWW: <<http://www.systemonline.cz/sprava-it/agilni-vyvoj-v-praxi.htm>>
13. *Návrh aplikací v jazyce UML - Unified Modeling Language* [online]. 2003-11-05 [cit 2012-03-11]. Dostupné z WWW: <<http://interval.cz/clanky/navrh-aplikaci-v-jazyce-uml-unified-modeling-language/>>
14. *Understanding the pros and cons of the Waterfall Model of software development* [online]. 2006-09-22 [cit. 2012-03-02]. Dostupné z WWW: <<http://www.techrepublic.com/article/understanding-the-pros-and-cons-of-the-waterfall-model-of-software-development/6118423>>
15. COFFIN, Rod and Derek, LANE. *A Practical Guide to Seven Agile Methodologies, Part I* [online]. 2006-10-11 cit. [2012-02-24]. Dostupné z WWW: <<http://www.devx.com/architect/Article/32761/0/page/1>>
16. BELL, Donald. *UML basics: An introduction to the Unified Modeling Language* [online]. 2003-06-15 [cit 2012-02-22]. Dostupné z WWW: <<http://www.ibm.com/developerworks/rational/library/769.html>>

ZOZNAM SKRATIEK

CASE – Computer aided software enginery

DSS – Decision support system

EIS – Executive information system

ER – Entity-relationship

GUI – Graphical user interface

ICT/IT – informačné komunikačné technológie a informačné technológie

IS – informačný systém

IS/ICT – informačný systém a informačné komunikačné technológie

IT – informačné technológie

MIS – Management information systems

SW – softvér

TPS – Transaction processing system

UC – Use case

UML – Unified modeling language

PROHLÁŠENÍ O VYUŽITÍ VÝSLEDKŮ DIPLOMOVÉ PRÁCE

Prohlašuji, že

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

V Ostravě dne

.....

Bc. Rastislav Golier

ZOZNAM PRÍLOH

Príloha 1: Formulár dochádzky a výkonu pracovníka

Príloha 2: Formulár Mäkkej zložky tímu

Príloha 3: Formulár zoznam činností a subformulár časovanie činností

Príloha 4: Formulár úpravy časovania činností

Príloha 5: Zostava pre výpočet bonusu

Príloha 1: Formulár dochádzky a výkonu pracovníka

frmDochadzka : Form

Pracovník: Gondo Golier Rastislav

Dátum: 12. 4. 2012

Pracovní doba: Příchod: 7:00, Odchod: 16:00

Oběd: Odchod: 12:00, Příchod: 12:30

Přerušení: Odchod 1: 0:00, Příchod 1: 0:00, Odchod 2: 0:00, Příchod 2: 0:00

Přerušení lékař: Odchod 1: 0:00, Příchod 1: 0:00, Odchod 2: 0:00, Příchod 2: 0:00

Měsíční plnění: Kumulativní: 14,41%, Celkové: 5,77%

Poznámka:

Počet bodů za den: 553,5

Počet operací za den: 29

Počet bodů za měsíc: 553,5

Přesčas(v min): 30

Půldenní nepřítomnost:

Nová docházka, Upravit docházku, Odstranit docházku

Přidat nepřítomnost, Upravit nepřítomnost, Odstranit nepřítomnost

Proces	Činnost	Počet	Body celkom
Nevýkonová činnost	porada týmu, rozhovor s TL	4	285,68
Monitoring	Zpracování výzvy - dopis klientovi D4	5	83,3
Klient a výpisy	Klient a výpisy - založení klienta - kontrola	6	99,96
Duplicity	Duplicity - zpracování položky (2klienti = 1 řádek v reportu)	14	84,56

Přidat činnost, Upravit činnost, Odstranit činnost

Record: 1 of 17

Príloha 2: Formulár Mäkkej zložky tímu.

fmNUMekkaslozkaNova : Form

Datum Zadani: 1. marec 2012
 IDTymu: 6

Prijmeni Jmeno: Klientsky Pristup: Komunikace: Proaktivita: Kvalita: Rizeni Casu: Empatie: Tymova Spoluprace: BPMMS: KaizenAZV: CDS:

Cvaková Markéta	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Čháková Markéta	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Dědková Romana	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Golier Rastislav	50	100	100	100	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Klimková Luce	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Kopecká Ivana	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ortková Zuzana	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Osziová Kristýna	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Urbířová Bohdana	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Vaňková Aneta	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Vavráčová Nela	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Record: 4 of 11

Príloha 3: Formulár zoznam činností a subformulár časovanie činností

ID Činnosti:	M_T1_001
IDTymu:	1
Nadproces:	Ostatní správa (netýká se konkrétního produktu)
Proces:	Ostatní
Název činnosti:	ZV - volání na pobočky
Bodove ohodnocení:	11,9
Časová náročnosť:	5
Koeficient:	1
Aktivní:	<input checked="" type="checkbox"/>
KontrolaT6:	
CAS_REAL:	4
CAS_SLA:	4
L1:	Ostatní činnost
L2:	Ostatní činnost
L3:	
L4:	
L5:	

Kde hledat:

Zadej text:

Vyhledat

Zobrazit vše

Platné od	Časová náročnosť	Bodové ohodnocení
1. 1. 2012	4	9,52
1. 4. 2012	5	11,9

Časovanie

Príloha 4: Formulár úpravy časovania činností

Upravit časovou náročnosť pro činnost

Názov činnosti:	<input type="text" value="ZV - volání na pobočky"/>
ID Činnosti:	<input type="text" value="M_T1_001"/>
Platnosť od:	<input type="text" value="1. 1. 2012"/>
Časová náročnosť:	<input type="text" value="4"/>
Bodove Ohodnotenie:	<input type="text" value="9,52"/>

Príloha 5: Zostava pre výpočet bonusu

IDTymu	3	Rozpočet	50000										
<i>Príjmení</i> Jmeno	Výkon	Mětkka	Složka	Součet bodů	Plán bodů	Peníze výkon	Peníze mětkka	Peníze součet	Přídání peněz	% Plnění			
Glogarová Eva	18265,65	0	18265,65	17850	2 916	0	2 916	2 916	2 916	102,33%			
Gregorová Věra	18132,98	0	18132,98	15300	2 895	0	2 895	2 895	2 895	118,52%			
Kalužová Martina	20966,97	0	20966,97	17850	3 347	0	3 347	3 347	3 347	117,46%			
Kocáňová Iveta	19998,24	0	19998,24	18700	3 192	0	3 192	3 192	3 192	106,94%			
Macková Jarmila	22739,67	0	22739,67	18700	3 630	0	3 630	3 630	3 630	121,60%			
Mičko Vladimira	22917,82	0	22917,82	18700	3 658	0	3 658	3 658	3 658	122,56%			
Moravčíková Jana	20685,84	0	20685,84	18700	3 302	0	3 302	3 302	3 302	110,62%			
Mouchová Romana	5672,78	0	5672,78	18700	906	0	906	906	0	30,34%			
Petruželová Lenka	23120,3	0	23120,3	18700	3 691	0	3 691	3 691	3 691	123,64%			
Pospěchová Radomíra	19399,42	0	19399,42	17850	3 097	0	3 097	3 097	3 097	108,68%			
Šaňková Ludmila	15288,43	0	15288,43	14450	2 441	0	2 441	2 441	2 441	105,80%			
Witalová Hedvika	21610,23	0	21610,23	18700	3 450	0	3 450	3 450	3 450	115,56%			
Zelmannová Soňa	21776,63	0	21776,63	18700	3 476	0	3 476	3 476	3 476	116,45%			
Summary for 'IDTymu' = 3 (13 detail records)													
Sum	250575	0	250575	232900	40000	0	40000	40000	40000	Tým s BONUS M		% Plnění tým: 107,59%	