

Univerzita Pardubice
Fakulta ekonomicko-správní

Požadavky zákazníků jako východisko pro určení
rozsahu softwarového projektu

Aleš Hudec

Bakalářská práce

2010

Univerzita Pardubice
Fakulta ekonomicko-správní
Ústav systémového inženýrství a informatiky
Akademický rok: 2009/2010

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: Aleš HUDEC

Studijní program: B6209 Systémové inženýrství a informatika

Studijní obor: Informatika ve veřejné správě

Název tématu: Požadavky zákazníků jako východisko pro určení rozsahu softwarového projektu

Z á s a d y p r o v y p r a c o v á n í :

Cílem práce je stanovení vhodného způsobu ohodnocení rozsahu softwarového projektu na základě definovaných zákaznických požadavků.

Rozsah grafických prací:

Rozsah pracovní zprávy:

cca 40 stran

Forma zpracování bakalářské práce:

tištěná/elektronická

Seznam odborné literatury:

COCKBURN, A. *Writing effective use cases*. [s.l.]: Addison-Wesley Professional, 2000. 270 s. ISBN 0201702258.

WIEGERS, K. E. *Požadavky na software*. Tomáš Znamenáček. 1. vyd. Brno: Computer Press, 2008. 448 s. Přeloženo z angličtiny. ISBN 978-80-251-1877-1.

ROBERTSON, S, ROBERTSON, J. C. *Mastering the Requirements Process*. 2nd edition. [s.l.]: Addison-Wesley Professional, March 27, 2006. 592 s. ISBN 0321419499.

PALETA, P. *Co programátory ve škole neučí, aneb, Softwarové inženýrství v reálné praxi*. 1. vyd. Brno: Computer Press, 2003. 337 s. Obsahuje rejstřík. ISBN 80-251-0073-1.

MCCONNELL, S. *Odhadování softwarových projektů: Jak správně určit rozpočet, termíny a zdroje*. Jiří Fadrný. 1. vyd. Brno: Computer Press, 2006. 320 s. Přeloženo z angličtiny. ISBN 80-251-1240-3.

Šimonová

Vedoucí bakalářské práce:

Ing. Stanislava Šimonová, Ph.D.

Ústav systémového inženýrství a informatiky

Datum zadání bakalářské práce:

5. října 2009

Termín odevzdání bakalářské práce:

30. dubna 2010

Renáta

doc. Ing. Renáta Myšková, Ph.D.

děkanka

L.S.

Jiří Křupka

doc. Ing. Jiří Křupka, Ph.D.

vedoucí ústavu

V Pardubicích dne 5. října 2009

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Praze dne 26. června 2010

Aleš Hudec

ANOTACE

Práce se zabývá postupem návrhu metody pro tvorbu odhadu rozsahu (pracnosti) informačních systémů ve fázi zahájení projektu.

KLÍČOVÁ SLOVA

softwarový projekt, požadavky, odhad rozsahu, expertní odhady, Use Case Point, Functional Point

TITLE

Customer requirements are the method by which the scope of a software project can be determined

ANOTATION

In the present work deals with the process design methods for constructing estimates of the scope of information systems in the inception phase of the project.

KEYWORDS

software project, requirements, estimation of scope, expert's estimation, Use Case Point, Functional Point

Obsah

1. Úvod	17
2. Proces řízení softwarových projektů	19
2.1. Úspěšnost softwarových projektů	19
2.2. Metoda The Rational Unified Process	20
2.3. Požadavky zákazníků	21
2.3.1. Analýza požadavků	22
2.3.2. Správa požadavků	22
2.4. Rozsah softwarového projektu	23
2.4.1. Okamžik vytvoření odhadu rozsahu softwarového projektu	23
3. Metody odhadování rozsahu softwarových projektů	25
3.1. Charakteristiky metod	25
3.2. Metody založené na úsudku	26
3.2.1. Expertní odhady	26
3.2.2. Analogie	26
3.3. Metody založené na výpočtu	27
3.3.1. Use Case Point	27
3.3.2. Funkční body	30
3.4. Ostatní metody	33
4. Návrh metody odhadu rozsahu softwarového projektu	35
4.1. Charakteristika prostředí	35
4.2. Zpracování požadavků aplikačních změn	36
4.2.1. Projektový záměr	36
4.2.2. Seznam zákaznických požadavků	37
4.3. Analýza zákaznických požadavků	37
4.4. Popis problému	37
4.4.1. Zdroje historických dat	38
4.4.2. Analýza historických dat	38
4.5. Výběr metody tvorby odhadů	40
4.5.1. Vyhodnocení metod	41
4.6. Rozšíření metody expertních odhadů	41
4.6.1. Způsob výpočtu	42
4.6.2. Ověření metody	43
4.7. Úprava procesu vývoje software	43

4.8. Softwarová podpora	44
5. Analýza výsledků	45
5.1. Analýza historických dat a výstupů aplikace	45
5.2. Historické projekty	46
5.2.1. Porovnání parametrů expertních odhadů	46
5.2.2. Porovnání odhadů rozsahu projektů	46
5.2.3. Porovnání odchylek rozsahu projektů	47
5.3. Realizované projekty	47
5.3.1. Porovnání parametrů expertních odhadů	48
5.3.2. Porovnání odhadů rozsahu projektů	48
5.3.3. Porovnání odchylek rozsahu projektů	49
5.4. Kvalita odhadů	49
5.4.1. Ukazatel kvality odhadu	49
5.4.2. Ukazatel relativní chyby	50
5.5. Vyhodnocení výsledků	50
6. Závěr	53
Literatura	56
Přílohy	I
A. Správa a evidence požadavků	III
B. Návrh aplikace pro tvorbu odhadů	VII

Seznam tabulek

2.1. Informace evidované u seznamu požadavků. Zdroj: [14, 15], vlastní	23
3.1. Kategorie pro případy užití. Zdroj: [3], vlastní	27
3.2. Příklad výpočtu <i>UUCW</i> . Zdroj: [3], vlastní	28
3.3. Kategorie složitosti aktérů pro metodu Use Case Point. Zdroj: [3], vlastní	28
3.5. Příklad výpočtu <i>UAW</i> . Zdroj: [3], vlastní	28
3.6. Vlastnosti a hodnota technického faktoru <i>TF</i> . Zdroj:[3], vlastní	29
3.7. Vlastnosti a hodnota faktoru prostředí <i>EF</i> . Zdroj:[3], vlastní	29
3.8. Počty funkcí systému. Zdroj: [7]	32
3.9. Váhy složitosti jednotlivých funkcí. Zdroj: [16]	32
3.10. Obecné charakteristiky prostředí. Zdroj: vlastní, podklady [7]	33
4.1. Odchytky odhadu pracnosti projektů 1. poloviny roku 2009 Zdroj: vlastní	39
4.2. Vyhodnocení metod pro odhadování rozsahu software . Zdroj: vlastní	40
4.3. Příklad výpočtu odhadu rozsahu pracnosti činnosti jednotlivých prvků aplikace. Zdroj: vlastní	43
5.1. Odhady a skutečný rozsah historických projektů. Zdroj: vlastní	46
5.2. Hodnoty parametrů expertních odhadů pracnosti pro historické projekty. Zdroj: vlastní	46
5.3. Hodnoty odhadů rozsahu historických projektů. Zdroj: vlastní	47
5.4. Tabulka odchylek hodnot odhadů a skutečného rozsahu projektu. Zdroj: vlastní	47
5.5. Odhady a skutečný rozsah realizovaných projektů. Zdroj: vlastní	48
5.6. Hodnoty parametrů expertních odhadů pracnosti pro realizované projekty. Zdroj: vlastní	48
5.7. Hodnoty odhadů rozsahu realizovaných projektů. Zdroj: vlastní	49
5.8. Tabulka odchylek hodnot odhadů a skutečného rozsahu projektu. Zdroj: vlastní	49
5.9. Kvalita odhadu rozdělená dle fází projektu Zdroj: vlastní	50
5.10. Relativní chyba odhadu rozdělená dle fází projektu Zdroj: vlastní	50

Seznam obrázků

2.1. Porovnání úspěšnosti softwarových projektů od roku 1994 do 2009 podle The Standish Group's Chaos report. Zdroj dat: [6], zpracování vlastní	19
2.2. Fáze a etapy metody RUP Zdroj:[1]	20
2.3. Rozložení chyb v softwarových projektech. Zdroj: [2]	22
2.4. Zpřesňování odhadu rozsahu v průběhu projektu Zdroj: [12], vlastní	24
3.1. Schéma výpočtu metody FP. Zdroj: [16]	30
4.1. Vazby mezi objekty. Zdroj: vlastní	37
4.2. Procentuální odchylka odhadu rozsahu vybraných projektů. Zdroj: vlastní	39
B.1. Model případu užití. Zdroj: vlastní	VII
B.2. Model tříd aplikace pro tvorbu odhadů rozsahu projektů Zdroj: vlastní	X
B.3. Návrh ovládání obrazovek aplikace . Zdroj: vlastní	X
B.4. Přehled projektů Zdroj: vlastní	XI
B.5. Profil projektu aplikace. Zdroj: vlastní	XI
B.6. Položka odhadu pracnosti prvku Zdroj: vlastní	XII
B.7. Přehled prvků jednotlivých požadavků Zdroj: vlastní	XII

Seznam zkratek

- COCOMO The Constructive Cost Model, metoda tvorby odhadu
- FP Functional Points, metoda tvorby odhadu
- IS Information System, Informační systém
- MD Manday, člověkoděn
- PERT Program Evaluation and Review Technique, metoda odhadu času
- RUP The Rational Unified Process, Jednotný modelovací proces
- UCP Use Case Point, metoda tvorby odhadu pracovní
- UML Unified Modeling Language, Unifikovaný modelovací jazyk
- WBS Work breakdown structure, Struktura rozpisu práce

1. Úvod

Současný technologický i ekonomický vývoj představuje potřebu začlenit do firemních procesů velké množství změn (technické, ekonomické, personální či právní). Změny je nutné promítnout do informačních systémů (dále jen IS), které jsou podpůrnou součástí firemních procesů. Zákazníci, v pozici uživatelů IS, definují požadavky na vytvoření nových nebo úpravu stávajících aplikací IS. Zpracování požadavků je dále řešeno v rámci softwarových projektů. Pro realizaci projektu mohou zákazníci zvolit jako dodavatele vlastní interní oddělení nebo oslovit externí společnost.

Prvním úkolem dodavatele je stanovení rozsahu softwarového projektu. To znamená mít informaci o velikosti, práci, vlastnosti, rozvrhu nebo ceně projektu. Požadavky zákazníků jsou v úvodní etapě projektu obecné a postupně dochází k jejich zpřesňování. Pro stanovení rozsahu projektu se v různých etapách používají odhady.

Úkolem odhadů je získat informaci o rozsahu projektu co nejdříve, tak aby bylo možné vyhodnotit přínosy projektu. Nejlépe v době, kdy nebylo zahájeno programování, ale jsou známy požadavky na vlastnosti a funkce systému. Pro tento případ lze požadavky zákazníků použít jako východisko pro stanovení rozsahu softwarového projektu.

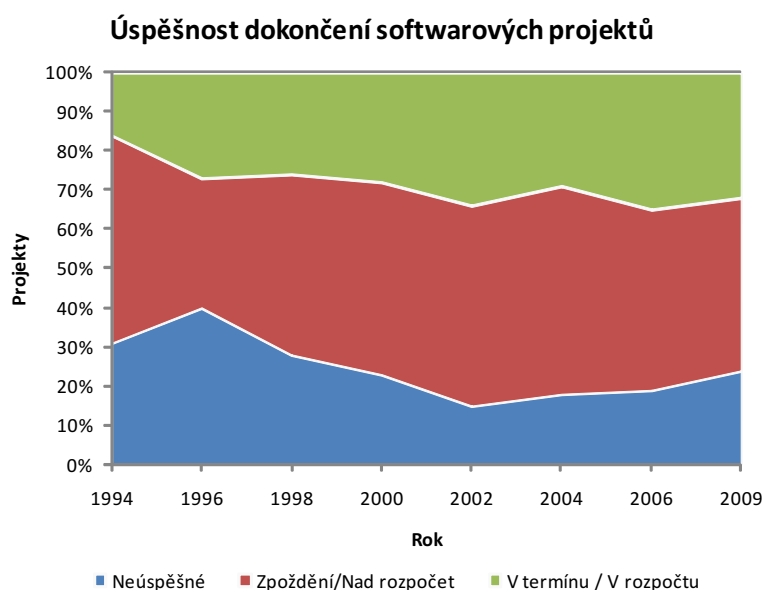
Cílem práce je stanovení vhodného způsobu ohodnocení rozsahu softwarového projektu na základě definovaných zákaznických požadavků. Navržená metoda tvorby odhadů rozsahu softwarových projektů bude ověřena v reálném prostředí.

2. Proces řízení softwarových projektů

Pro řízení softwarových projektů lze zvolit různé metody podle typu projektu. Většina metod rozděluje projekt na jednotlivé etapy a fáze. Rozdělení je důležité pro určení časových intervalů a milníků projektu. Na základě vymezení časového intervalu je možné stanovit okamžik pro vytvoření odhadu rozsahu projektu. Jednou z klíčových částí metod řízení softwarových projektů je příprava podkladů pro tvorbu odhadu rozsahu projektu. Úkolem je získat představu o velikosti projektu, pracovních a ekonomických nákladech.

2.1. Úspěšnost softwarových projektů

Vývoj software je technickým a zároveň tvůrčím procesem. Do vývoje vstupuje mnoho faktorů, které mají přímý nebo nepřímý vliv na úspěšnost softwarového projektu.



Obrázek 2.1.: Porovnání úspěšnosti softwarových projektů od roku 1994 do 2009 podle The Standish Group's Chaos report. Zdroj dat: [6], zpracování vlastní

Podle studie The Standish Group's Chaos report[6], znázorněno na obrázku 2.1, je více než

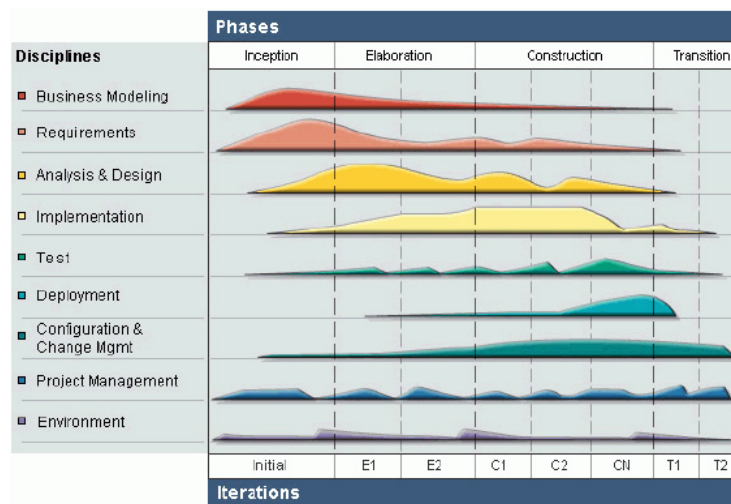
2. Proces řízení softwarových projektů

3/4 softwarových projektů dodáno se zpožděním nebo nejsou dodány vůbec. Metoda řízení software může výrazným způsobem ovlivnit úspěšnost projektu.

2.2. Metoda The Rational Unified Process

Pro potřebu definování fází projektu jsem zvolil metodu řízení vývoje software pomocí RUP (Rational Unified Process) [17]. Důvodem volby metody je univerzální použití oproti alternativním metodám. Přehled dalších metod řízení softwarových projektů popisuje ve své diplomové práci můj bratr [8].

Vznik metody Rational Unified Process se datuje od osmdesátých let minulého století. V současné době je RUP vyvíjen pod hlavičkou IBM. RUP je obecnou a velmi rozsáhlou metodikou. Metoda vznikla na základě zkušeností s vývojem software a vychází z vodopádového modelu životního cyklu projektu. Metodu lze přizpůsobit podle typu i rozsahu projektu. Velký důraz je kladen na iterační vývoj, vizuální modelování, objektový přístup a komunikaci. Metoda využívá standardů objektově orientovaného programování a jako modelovací platforma je prosazován standard UML [8].



Obrázek 2.2.: Fáze a etapy metody RUP Zdroj:[1]

Na obrázku 2.2 je znázorněno rozdělení projektu na *disciplíny* (Disciplines), *fáze* (Phases), které se dělí na *iterace* (Iterations) podle potřeby projektu. Fáze slouží jako projektové milníky.

- Zahájení (Inception)

Fáze zahájení obsahuje etapy modelování podnikových procesů (Business Modeling) a analýzu požadavků (Requirements). Cílem této fáze je definování hranic projektu a identi-

fikace klíčových požadavků, vlastností a funkcí na výsledný produkt. Nedílnou součástí první fáze je i analýza rizik. Požadavky následně slouží jako podklad tvorby analýzy a návrhu.

- Rozpracování (Elaboration)

V průběhu fáze probíhá upřesňování požadavků pro potřeby vytvoření analýzy a modelů UML. Vytváří se scénáře chování, navrhuje architektura, plánuje implementace a testování. Zpravidla vzniká i prototyp k ověření zvolené architektury.

- Konstrukce (Construction)

Fáze konstrukce představuje vlastní programování. Postupně vzniká zdrojový kód, databázové struktury, kompilují se knihovny a vytváří se uživatelské rozhraní. Dochází k testování jednotlivých komponent i funkčních celků. Jedná se o časově i zdrojově nejnáročnější část procesu. Z intelektuální náročnosti, typické pro předchozí fáze, přechází těžiště na méně náročnou práci. Ta však díky svému množství činností vyžaduje vysoké úsilí v oblasti projektového řízení. V této fázi se čím dál více prosazuje změnové řízení.

- Předávání (Transition)

Závěrečná fáze, kdy dochází k přenosu systému z vývojového prostředí do produkčního prostředí zákazníka. Systém prochází závěrečným testováním a instalací. Dochází k propojení s externími systémy. Paralelně probíhá školení uživatelů a správců systému. Součástí této fáze může být i následné provozování a údržba systému.

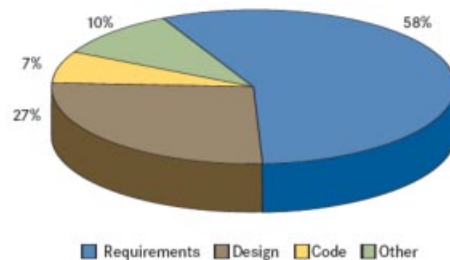
Fáze procházejí dle potřeby jednou nebo více iteracemi. Metoda klade důraz na analýzu a řízení rizik. Nedílnou částí je aktivní správa požadavků.

2.3. Požadavky zákazníků

Požadavek je konkrétní ověřitelná funkce chování systému, tak aby byl splněn cíl obchodní specifikace. Základní vlastnosti požadavku jsou proveditelnost, ověřitelnost a musí mít vztah k obchodnímu záměru. Analýza požadavků slouží k pochopení obchodní problematiky zákazníka a k vyřešení protichůdných požadavků. Klíčovou roli hrají požadavky při testování softwarového díla.

2. Proces řízení softwarových projektů

Důležitost analýzy a správy zákaznických požadavků potvrzuje graf rozdělení chyb softwarových projektů na obrázku 2.3. Studie Jamese Martina ukazuje, že 58 % všech chyb v softwarových projektech vzniká ve fázi požadavků.



Obrázek 2.3.: Rozložení chyb v softwarových projektech. Zdroj: [2]

Dále z ní vyplývá, že 50% chyb v požadavcích je způsobeno špatně napsanými, dvojnásobnými, nejasnými a nesprávnými požadavky. Druhá polovina chyb je způsobena nesystematickými úpravami a nekompletní specifikace, např. požadavky, které byly jednoduše vynechány[2].

2.3.1. Analýza požadavků

Analytik na základě dostupných informací definuje cíl projektu, *funkční* a *nefunkční* požadavky na systém a vytváří slovník obchodních pojmů zákazníka. Pro získání a specifikování požadavků jsou používány zejména rozhovory se zákazníkem, tvorba prototypů, modelování uživatelského rozhraní.

Výstupem fáze je dokumentace o rozsahu projektu, seznam požadavků, otevřené body a rizika projektu. Specifikované zákaznické požadavky slouží pro vytvoření představy o hranicích projektu, složitosti a jeho celkové pracnosti. Výstupy etapy analýzy požadavků se stávají součástí objednávky jako zadání. Následující etapa Analýza a design používá seznam požadavků pro tvorbu modelů systému.

2.3.2. Správa požadavků

Evidence požadavků slouží pro vyhledávání vazeb mezi požadavky, vazby na obchodní zadání a cíle projektu. Dále pak vazby na systémové modely, zejména případy užití, a návrhy uživatelského prostředí. Stanovení priorit požadavků umožňuje vložit do vývoje určitou volnost pro případy změn pracnosti projektu.

Požadavky lze rozdělit do dvou základních typů[14, 15]:

- Funkční požadavky

Představují základní předmět systému a jsou měřeny konkrétními prostředky, jako jsou například hodnoty dat, logika a algoritmy rozhodování. Funkční požadavky specifikují,

co má produkt dělat.

- Nefunkční požadavky

Představují vlastnosti v oblasti chování, které musí mít stanovené funkce, jako například výkonnost, použitelnost atd. Nefunkční požadavky specifikují, jaké vlastnosti má produkt mít.

Tabulka 2.1.: Informace evidované u seznamu požadavků. Zdroj: [14, 15], vlastní

Položka	Popis
Typ	Funkční požadavky / Nefunkční požadavky.
Kód požadavku	Jednoznačný identifikátor požadavku.
Název požadavku	Název požadavku.
Priorita	Významnost požadavku.
Stručný popis	Popis požadavku. V případě potřeby odkaz na další dokumenty.
Stav	Aktuální stav požadavku.
Stakeholder	Zadavatel požadavku.
Vazba na jiný požadavek	Vazba na související požadavky.
Use Cases	Seznam případů užití.
Akceptační kritéria	Popis postupu jakým bude ověřen požadavek.

Příklad informací evidovaných ke každému požadavku je uveden v tabulce 2.1. Za správu požadavků je zodpovědný analytik. V průběhu projektu se mohou požadavky měnit. Proces změnového řízení požadavku by měl být součástí správy požadavků.

2.4. Rozsah softwarového projektu

Rozsahem softwarového projektu rozumíme skupinu informací, které jsou využívány pro plánování, rozhodování a vyhodnocování projektu. Velikost projektu je možné reprezentovat pojmem pracnost. Jako jednotka pracnosti se používá člověkodén, v angličtině manday (MD) . Informace o pracnosti je využita pro tvorbu časového plánu, ceně a seznamu vlastností výsledného produktu.

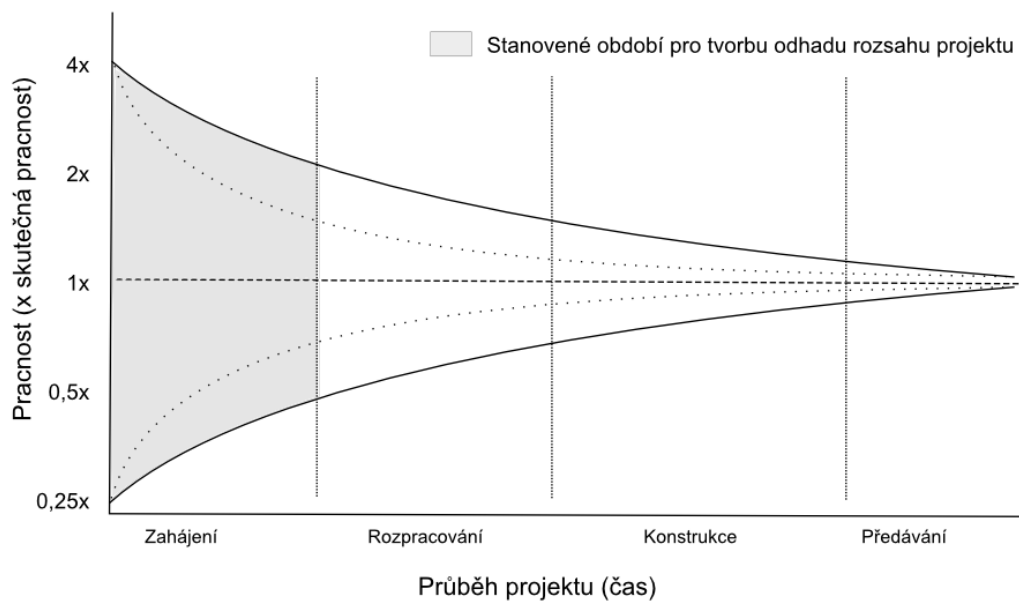
2.4.1. Okamžik vytvoření odhadu rozsahu softwarového projektu

V průběhu jednotlivých fází projektu dochází k postupnému zpřesňování požadavků zákazníka na softwarový produkt a postupnému upřesňování odhadu pracnosti, jak je znázorněno na obrázku 2.4. Kvalitní analýzou zákaznických požadavků lze zvýšit přesnost již ve fázi zahájení projektu, vyznačeno tečkovanou čarou.

Pro dodavatele je výhodné odhad rozsahu určit až v době předávání softwarového díla, kdy nejvíce odpovídá skutečnosti. Dolní i horní hranice přesnosti odhadu rozsahu jsou nejbližší sku-

2. Proces řízení softwarových projektů

tečnosti. V praxi by tato skutečnost znamenala potřebu neomezených finančních a časových zdrojů.



Obrázek 2.4.: Zpřesňování odhadu rozsahu v průběhu projektu Zdroj: [12], vlastní

Zákazník má potřebu obdržet odhady co možná nejdříve. Podle odhadu rozsahu je provedeno vyhodnocení přínosu projektu. Včasné rozhodnutí o realizaci dovoluje ušetřit čas a finanční prostředky, které by zákazník vydal na přípravu projektu.

V praxi se odhad zaměňuje za závazek nebo plán, jak dosáhnout cíle [10]. Takto vytvořený závazek negativně ovlivňuje další zpřesňování odhadu rozsahu projektu.

Kompromisem okamžiku sdělení prvního odhadu rozsahu projektu je fáze zahájení, kdy dochází k popisování podnikových procesů a analýze požadavků zákazníka. Tedy v době, kdy známe klíčové požadavky, ale ještě nebyla zahájena systémová analýza a programování. Jedná se o milník projektu, kde lze rozhodnout o realizaci projektu a zabránit tím ekonomickým ztrátám na straně dodavatele i zákazníka.

3. Metody odhadování rozsahu softwarových projektů

Kapitola obsahuje popis metod využívaných v úvodních fázích softwarových projektů. U popisu metody je uveden postup tvorby odhadu, charakteristika použitelnosti metody a konkrétní příklad výpočtu nebo použití metody.

3.1. Charakteristiky metod

Metod pro odhadování rozsahu software je několik a lze je podle přístupu rozdělit na dvě základní skupiny. První skupinou jsou metody založené na zkušenostech nebo porovnávání, tj. metody založené na úsudku. Druhou skupinou jsou metody založené na historických datech a matematických modelech.

Použitelnost jednotlivých metod odhadu softwarového projektu lze posuzovat podle [10]:

- Odhadovaných vlastností.

Pro softwarové projekty lze odhadovat například tyto vlastnosti: velikost, rozsah práce, rozvrh, vlastnosti.

- Velikosti projektu.

Malé projekty, do pěti pracovníků na projekt a s dobou trvání týdny až měsíce.

Střední projekty, tým mezi 5 a 25 pracovníky s dobou trvání od 3 do 12 měsíců,

Velké projekty, tým okolo 25 pracovníků a doba trvání řádově 6 až 12 měsíců nebo i déle.

- Fáze vývoje.

Vycházejí z použité metody pro vývoj software.

- Stylu vývoje.

Postupný, vyžaduje ukončení jedné fáze a navázání fáze následující nebo opakující se, iterativní přístup, kdy se fáze může několikrát opakovat.

- Dosažitelné přesnosti

3. Metody odhadování rozsahu softwarových projektů

Vycházím z rozdělení na nízkou, střední a vysokou. Některé metody generují vysokou přesnost, ale za velkou cenu. Metodu je nutné volit podle všech uvedených charakteristik.

V dalším textu uvádím jednotlivé metody. U každé metody jsem uvedl vhodnosti použití podle velikosti projektu, fáze projektu a dosažitelné přesnosti.

3.2. Metody založené na úsudku

Metody založené na úsudku využívají zkušeností a znalostí pracovníka, který vytváří odhad. Prakticky se jedná o nevyužívanější metody (70-85 % [9, 10]). V případě znalosti daného prostředí a problematiky mohou být velmi přesné. Z důvodu kognitivních zkreslení (sebejistota a profesionální hrdost, úzký rozsah vnímání, podhodnocení chyb) se jedná o odhady rizikové [11].

3.2.1. Expertní odhady

Použitelnost metody

Projekty: M, S, V

Přesnost: vysoká

Metoda vychází z odhadu provedeného expertem, který využívá vlastních zkušeností. Expertní odhady jsou vhodné pro případy, kde neexistují historické údaje nebo pro drobné a malé projekty. Výhodou expertních odhadů je jejich jednoduchost provedení. Neexistuje přesvědčivá studie o výhodnosti formálních metod a tím k popření metody expertních odhadů. Nevýhodou metody je subjektivnost. Čím více expertů se na odhadu podílí, tím je metoda objektivnější a zároveň přesnější. Z psychologického hlediska je vhodné, aby expert neznal plány a cíle projektu.

3.2.2. Analogie

Použitelnost metody

Projekty: M, S, V

Přesnost: střední

Jedná se určitou modifikaci metody expertních odhadů o informace z projektů vytvořených v minulosti. Při projektech se objevuje požadavek ve formě: Tabulka A bude stejná jako tabulka Z, pouze bude rozšířena o 3 atributy. Uživatel se snaží porovnávat navrhovaný systém podle existujícího systému. Výhodou metody je představa o rozhraní v nové aplikaci. Nevýhodou jsou případná technologická omezení, která jsou před uživatelem skryta. Například rozšíření systému o tři atributy může znamenat zásah do datového modelu systému.

3.3. Metody založené na výpočtu

První podmínkou metod využívajících výpočtů je existence úvodní části analýzy (diagram případů užití, diagram tříd, ad.). Druhou podmínkou je existence (nebo odhad) faktorů prostředí. Hodnoty faktoru prostředí dostaneme z historických hodnot a do výpočtu zavádějí vliv prostředí (produktivitu práce, náročnost technologie, ap.).

3.3.1. Use Case Point

Použitelnost metody

Projekty: M, S, V

Přesnost: vysoká

Na základě rozvoje objektové orientovaného vývoje software vznikla metoda Use Case Point [3]. Základem pro výpočet bodů, je existující model případů užití a zařazení prvků do kategorií podle složitosti. Aplikace metody se sestává z výpočtu těchto hodnot:

- *UUCP* - nevyrovnaný počet bodů (Unadjusted Use Case Points)
 - *UUCW* - nevyrovnané váhy případů užití (Unadjusted Use Case Weight)
 - *UAW* - nevyrovnané váhy aktérů (Unadjusted Actor Weight)
- *TCF* - technický faktor (Technical Complexity Factor).
- *ECF* - faktor vlivu prostředí (Environment Complexity Factor).
- *PF* - faktor produktivity (Productivity Factor).

K výpočtu celkovému vyrovnanému počtu bodů *UCP* použijeme následující vzorec:

$$UCP = UUCP \cdot TCF \cdot ECF \quad (3.1)$$

Pro výpočet hodnoty *UUCP* nejprve zařadíme každý případ užití podle složitosti do kategorie. Seznam kategorií je uveden v tabulce 3.1.

Tabulka 3.1.: Kategorie pro případy užití. Zdroj: [3], vlastní

Kategorie	Definice	Počet transakcí (tříd)	Váha
Jednoduchý	Jednoduché uživatelské rozhraní s vazbou na jednu databázovou entitu.	< 4 (5)	5
Střední	Víceuživatelské rozhraní s vazbou na několik databázových entit.	4 - 7 (5-10)	10
Složitý	Složitě uživatelské rozhraní s vazbou na více jak tři databázové entity.	> 7 (>10)	15

3. Metody odhadování rozsahu softwarových projektů

Následně sečteme počty případů užití za danou kategorii a vynásobíme přidělenou vahou. Vypočítáme, tak nevyrovnanou váhu všech případů užití (Unadjusted Use Case Weight - *UUCW*). Příklad výpočtu je uveden v tabulce 3.2

Tabulka 3.2.: Příklad výpočtu *UUCW*. Zdroj: [3], vlastní

Kategorie	Počet transakcí (tříd)	Váha	Počet případů užití	Výpočet	Celkem
Jednoduchý	< 4 (5)	5	3	5 · 3	15
Střední	4 - 7 (5-10)	10	4	10 · 4	40
Složité	> 7 (>10)	15	2	15 · 2	30
Celkem					85

Podobný postup použijeme pro výpočet nevyrovnané váhy všech aktérů (Unadjusted Actor Weight - *UAW*). Aktéry zařadíme do kategorií podle tabulky 3.3.

Tabulka 3.3.: Kategorie složitosti aktérů pro metodu Use Case Point. Zdroj: [3], vlastní

Kategorie	Definice	Váha
Jednoduchý	Jiný systém komunikující přes aplikační programové rozhraní (API).	1
Střední	Uživatel se znakovým terminálem nebo jiný systém komunikující přes TCP/IP.	2
Složité	Osoba komunikující přes uživatelské rozhraní nebo webové stránky.	3

Všechny aktéry za danou kategorii sečteme a vynásobíme přidělenou vahou, jak je uvedeno v tabulce 3.5.

Tabulka 3.5.: Příklad výpočtu *UAW*. Zdroj: [3], vlastní

Kategorie	Váha	Počet aktérů	Výpočet	Celkem
Jednoduchý	1	0	1.0	0
Střední	2	1	2.1	2
Složité	3	2	3.2	6
Celkem				8

Nyní pro výpočet *UUCP* použijeme vzorec:

$$UUCP = UUCW + UAW \quad (3.2)$$

Pro zohlednění technických vlastností spočítáme hodnotu technického faktoru *TF*. Ty jsou tvořeny dílčími vlastnostmi, příklad uveden v tabulce 3.6, které se ohodnocují podle vlivu na

odhadovaný systém. Stupnice faktoru vlivu je 0 (nemá žádný vliv) až 5 (silný vliv).

Tabulka 3.6.: Vlastnosti a hodnota technického faktoru TF . Zdroj:[3], vlastní

Vlastnost	Váha	Faktor	Výpočet	Celkem
Distribuovaný systém	2	0	$2 \cdot 0$	0
Odezva systému/výkon	1	4	$1 \cdot 4$	4
Efektivita koncového uživatele	1	3	$1 \cdot 3$	3
Složitost procesu	1	4	$1 \cdot 4$	4
Znovupoužitelnost kódu	1	1	$1 \cdot 1$	1
Jednoduchost instalace	0,5	0	$0,5 \cdot 0$	0
Jednoduchost užití	0,5	2	$0,5 \cdot 2$	1
Přenositelnost	2	0	$2 \cdot 0$	0
Snadnost změny	1	1	$1 \cdot 1$	1
Souběžnost	1	3	$1 \cdot 3$	3
Bezpečnost	1	4	$1 \cdot 4$	4
Přístup třetí straně	1	0	$1 \cdot 0$	0
Požadavek speciálního školení	1	0	$1 \cdot 0$	0
Celkem				21

Vlastností prostředí zohledňuje faktor prostředí EF . Vlastnosti a příklad je uveden v tabulce 3.6.

Tabulka 3.7.: Vlastnosti a hodnota faktoru prostředí EF . Zdroj:[3], vlastní

Vlastnost	Váha	Faktor	Výpočet	Celkem
Znalost UML	1,5	1	$1,5 \cdot 1$	1,5
Zkušenosti s aplikacemi	0,5	2	$0,5 \cdot 2$	1
Zkušenosti s OOP	1	2	$1 \cdot 2$	2
Schopnosti vedoucího analytika	0,5	3	$0,5 \cdot 3$	1,5
Motivace	1	2	$1 \cdot 2$	2
Vyváženost požadavků	2	2	$2 \cdot 2$	4
Zaměstnanci na částečný úvazek	-1	0	$-1 \cdot 0$	0
Složitost programovacího jazyka	-1	0	$-1 \cdot 0$	0
Celkem				12

Vypočtené hodnoty dosadíme do vzorců 3.3 a 3.4,

$$TCF = 0,6 + 0,01 \cdot TF \quad (3.3)$$

kde TCF je hodnota faktoru technické složitosti,

$$ECF = 1,4 - 0,03 \cdot EF \quad (3.4)$$

a ECF je hodnota faktoru složitosti prostředí. Vypočtené hodnoty vložíme do vzorce 3.1:

3. Metody odhadování rozsahu softwarových projektů

$$UCP = 85 \cdot 8 \cdot 0,81 \cdot 1,04 \doteq 78,34 \quad (3.5)$$

Pro výpočet celkového rozsahu pomocí metody UCP vynásobíme počet bodů UCP faktorem produktivity PF podle vzorce 3.6.

$$Celkový\ Odhad = UCP \cdot PF \quad (3.6)$$

Hodnota bodu faktoru produktivity se pohybuje od 15 do 30 hodin za jeden bod v závislosti na zkušenostech a výkonu týmu. Obecně doporučenou hodnotou je 20 hodin [3].

$$Celkový\ Odhad = 78,34 \cdot 20 \doteq 1566$$

V uvedeném příkladu je hodnota $Celkový\ Odhad$ rovna 1566 hodin. Celkem to znamená 195 člověkodní při počtu 8 hod/den.

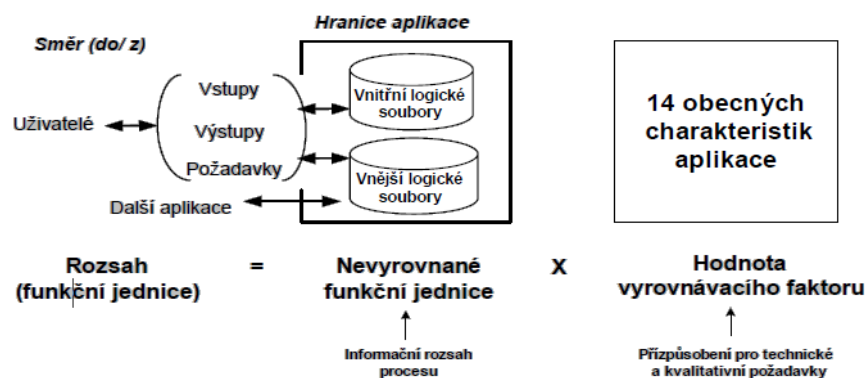
3.3.2. Funkční body

Použitelnost metody

Projekty: M, S, V

Přesnost: vysoká

Metoda funkční bodů (angl. Funtional Points) FP je standardizovaná metrika softwarového projektu, která měří rozsah softwarového projektu počtem aplikačních funkcí a dat. Technická oblast není měřena. Metodou lze odhadovat jak nové systémy, tak i pracnost systémů existujících. Příklad výpočtu převzatý po Marka Ashlouwa [7]. Obrázek 3.1 znázorňuje princip metody výpočtu.



Obrázek 3.1.: Schéma výpočtu metody FP. Zdroj: [16]

Cílem je pomocí vzorce 3.7 vypočítat počet upravených funkčních bodů.

$$FP = 0,65 \cdot 0,01 \sum F_i \cdot UFP \quad (3.7)$$

kde $\sum F_i$ je součet hodnocení obecných charakteristik systému a UFP je počet neupravených funkčních bodů.

Nejdříve spočítáme neupravené funkční body. Tuto část lze provést na základě detailního návrhu software. Transakčních funkcí se týkají,

- externí vstupy (EI - External Inputs)
- externí výstupy (EO - External Outputs)
- externí dotazy (EQ - External Enquiry)

datových funkcí se týkají,

- vnitřní logické soubory (ILF - Internal Logical Files)
- soubory vnějšího rozhraní (EIF - External Interface Files)

Příklad určení počtu transakčních funkcí a funkcí datových:

1. Externích vstupů = 7

Vyhledání zákazníka; Založení nového zákazníka; Informace o objednávce; Dodací lhůty; Platební historie zákazníka; Formulář pro výběr

2. Externí výstupy = 6

Úvěrové informace; Fakturace; Dodací list; Fakturační údaje; Stav objednávky; Stav skladu

3. Vnější dotazy = 3

Poptávka zákazníka; Objednávka zákazníka; Stav skladu

4. Externích souborů = 0

bez položek

5. Vnitřní soubory = 4

Skladové transakce; Seznam zákazníků; Seznam zboží; Transakce zákazníka

3. Metody odhadování rozsahu softwarových projektů

Nalezené nevyrovnané funkční body prvků se roztrídí do skupin podle složitosti a typu.

Tabulka 3.8.: Počty funkcí systému. Zdroj: [7]

Externí vstup	jednoduchý	3
Externí výstup	jednoduchý	4
Externí dotazy	jednoduchý	3
Externí záznamy	žádný	0
Interní záznamy	jednoduchý	7

Prvky každé skupiny se vynásobí příslušnou váhou, vázanou na typ a složitost.

Tabulka 3.9.: Váhy složitosti jednotlivých funkcí. Zdroj: [16]

Funkce	Složitost funkcí		
	Jednoduchý	Střední	Složitý
Externí vstup	3	4	6
Externí výstup	4	5	7
Externí dotazy	4	5	7
Externí záznamy	7	10	15
Interní záznamy	5	7	10

Jednotlivé skupiny vynásobíme nastavenou váhou a výsledky sečteme. Výsledkem je hodnota neupravených funkčních bodů *UFP* (Unadjusted Function Point):

$$UFP = 3 \cdot 7 + 4 \cdot 6 + 3 \cdot 3 + 7 \cdot 0 + 5 \cdot 4 = 65 \quad (3.8)$$

Pomocí faktorů prostředí se zohlední ve výpočtu obecné charakteristiky systému (spolehlivost, přístupnost, odezva, ad.). Hodnota každé vlastnosti je zvolena od 0 do 5 podle významnosti pro daný systém.

Tabulka 3.10.: Obecné charakteristiky prostředí. Zdroj: vlastní, podklady [7]

1	Systém vyžaduje spolehlivé zálohování a obnovení?	3
2	Jsou požadovány datové přenosy?	0
3	Obsahuje funkce distribuovaného zpracování?	0
4	Je požadován kritický výkon?	4
5	Bude systém pracovat za silného provozu?	3
6	Požaduje systém přímý (on-line) vstup dat?	4
7	Jsou vstupy, výstupy, soubory nebo dotazy složité za pomoci více obrazovek nebo operací?	3
8	Jsou hlavní soubory aktualizovány přímo?	5
9	Jsou vstupy, výstupy, soubory a dotazy složité?	0
10	Je složité vnitřní zpracování?	2
11	Je kód navržen pro opakované použití?	4
12	Jsou v návrhu zahrnuty konverze a instalace?	0
13	Je systém navržen pro vícenásobné instalace na různých místech?	0
14	Je systém navržen pro modifikace a uživatelsky jednoduchý systém?	3
	Celkem	31

Počet upravených funkčních bodů vypočteme vztahu 3.7.

$$FP = (0,65 + 0,01 \cdot 31) \cdot 65 = 62,4 \quad (3.9)$$

Poté je nutné stanovit cenu a pracnost jednoho funkčního bodu a vynásobením získat celkovou cenu a pracnost projektu. Pokud výkon programátora bude v průměru 18 funkčních bodů za měsíc, pak celkový odhad pracnosti bude:

$$\text{Celkový Odhad} = 18/30 \cdot FP = 18/30 \cdot 62,4 = 37,44 \text{ člověkodní}$$

3.4. Ostatní metody

Existují další metody, které jsou založeny především na velikosti zdrojového kódu. Příkladem může být COCOMO [4] (ang. The Constructive Cost Model). Popis metody je mimo rozsah práce, jelikož metoda vyžaduje detailní analýzu včetně datových modelů. Analýza a datové modelování je zařazeno do fáze rozpracování.

4. Návrh metody odhadu rozsahu softwarového projektu

V kapitole je uveden postup návrhu metody pro tvorbu odhadu rozsahu softwarových projektů na základě zákaznických požadavků v konkrétním reálném prostředí. Nejprve je uvedena charakteristika prostředí a aktuální stav. Následuje analýza přípustných řešení a volba metody s návrhem způsobu použití metody, postup realizace a způsob ověření metody.

Pro aplikování metody tvorby odhadů bylo zvoleno reálné prostředí, ve kterém jsem působil na pozici analytika informačních systémů. Prostor umožňuje zobecnění metody a využití v podobných podnicích. Reálné prostředí odráží skutečné potřeby a možnosti realizace metody pro určení rozsahu softwarového projektu.

4.1. Charakteristika prostředí

Zákazník vlastní a provozuje *informační systém* (dále jen IS). Rozšiřování a úpravy IS provádí vývojový tým. Vývojový tým byl původně součástí organizační struktury zákazníka. Po zvážení všech přínosů a rizik se zákazník rozhodl tým prodat a následně tento tým využívat v rámci outsourcingových služeb. Dodavatel začlenil tým do své struktury a nastavil nové procesy pro spolupráci se zákazníkem.

Vývojový tým má vysokou znalost informačního systému. Na druhé straně dokumentace IS je neaktuální a neúplná. Z tohoto důvodu byl zaveden proces tvorby aplikací podle přizpůsobené metodiky vycházející z RUP. Počet členů vývojového týmu se pohybuje okolo 35 členů (projektoví manažeři, analytici, programátoři, testéři, dokumentarista).

Spolupráce mezi zákazníkem a dodavatelem je vedena smlouvou o zajištění služeb rozvoje a údržby IS. Tato smlouva zajišťuje pevný rozsah prací pro vývojový tým. Druhou částí využití týmu jsou projekty vývoje nových softwarových aplikací nebo úprava stávajících aplikací v rámci informačního systému. Prioritizace a schvalování projektů probíhá pravidelně jednou za půl roku.

4.2. Zpracování požadavků aplikačních změn

Požadavek na aplikační změny předává zákazník formou *Obchodní specifikace*. Obchodní specifikace je základním dokumentem pro stanovení cíle projektu, rizik a analýzu zákaznických požadavků. Výsledkem zpracování požadavků aplikačních změn je dokument *Projektový záměr* a *Seznam zákaznických požadavků*.

4.2.1. Projektový záměr

Výsledky zpracování požadavků se zaznamenávají do dokumentu *Projektový záměr*. Dokument obsahuje následující části:

1. Cíl projektu.

Definovaný stručný cíl projektu.

2. Návrh řešení.

Obsahuje popis řešení. Může být uvedeno i více variant. Většinou neobsahuje žádné modely.

3. Otevřené body.

Seznamem oblastí, které nebylo možné v dané etapě analyzovat.

4. Mimo rozsah projektu

Omezuje projekt pouze na vybrané části systému, které jsou předmětem úpravy.

5. Seznam rizik.

Obsahuje body, které mohou mít negativní dopad na realizaci nebo provoz systému.

6. Nabídka.

a) Rozsah projektu je definován jako počet člověkodní za zvolenou variantu návrhu řešení.

b) Předpokládaný termín nasazení je stanoven podle požadavku zadavatele s ohledem na plánování kapacity vývojového týmu.

Část 6. Nabídka, zpracovává projektový manažer až po analýze požadavků a na základě pokladů analytika.

4.2.2. Seznam zákaznických požadavků

Samostatný dokument obsahující definované zákaznické požadavky na aplikační změny. Jedná se upřesnění zadání, stanovení hranic projektu a definování cíle projektu. Pro evidenci a správu zákaznických požadavků se využívá tabulkového editoru.

Požadavky jsou rozděleny na funkční a nefunkční, obsahují informaci o způsobu ověření správné funkce a požadovaných vlastností. Požadavky obsahují vazbu na zadavatele a v době vytváření modelů a analýzy obsahují vazbu na jednotlivé komponenty aplikace (obrazovky, případy užití, procedury). Ukázky formulářů a přehledů pro správu požadavků jsou uvedeny v příloze A.

4.3. Analýza zákaznických požadavků

Po odsouhlasení projektového záměru a seznamu požadavků zákazníkem, vytvoří analytik konceptuální návrh systému a provede rozpad požadavků na jednotlivé části (moduly, prvky). Rozpad požadavku lze přirovnat k metodě WBS (angl. Work breakdown structure). Znázornění závislosti mezi jednotlivými objekty je uvedena na obrázku 4.1.



Obrázek 4.1.: Vazby mezi objekty. Zdroj: vlastní

Projektový záměr vychází z obchodní specifikace. Projektový záměr obsahuje zpravidla několik požadavků. Požadavky jsou realizovány konkrétním prvkem aplikace. Jeden prvek může být řešením několika požadavků současně.

Pro jednotlivé prvky je pomocí konzultace s programátorem vytvořen odhad pracnosti k vybranému prvku. Analytik provede korekturu odhadu pracnosti jednotlivých prvků a zohlední, tak složitost prostředí. Po ukončení tvorby odhadů pracnosti, předá analytik podklady projektovému manažerovi, který vytvoří nabídku pro zákazníka, obsahující celkový odhad rozsahu projektu. Na základě rozsahu projektu a plánování lidských zdrojů je navržen harmonogram realizace projektu.

4.4. Popis problému

Jak uvádím v teoretické části práce, expertní odhady jsou závislé na zkušenostech pracovníka a jsou zatíženy subjektivním vnímáním. Nevýhoda se projevuje i v případě obměny týmu. Další

4. Návrh metody odhadu rozsahu softwarového projektu

nevýhodou je neexistence využívání historických informací pro odhadování nových projektů. Tyto důvody vedou k úvaze o zavedení metody odhadování rozsahu projektu na základě vyhodnocování historických hodnot. Na základě rozboru historických dat, provedu zkoumání o úspěšnosti používaných expertních odhadů.

4.4.1. Zdroje historických dat

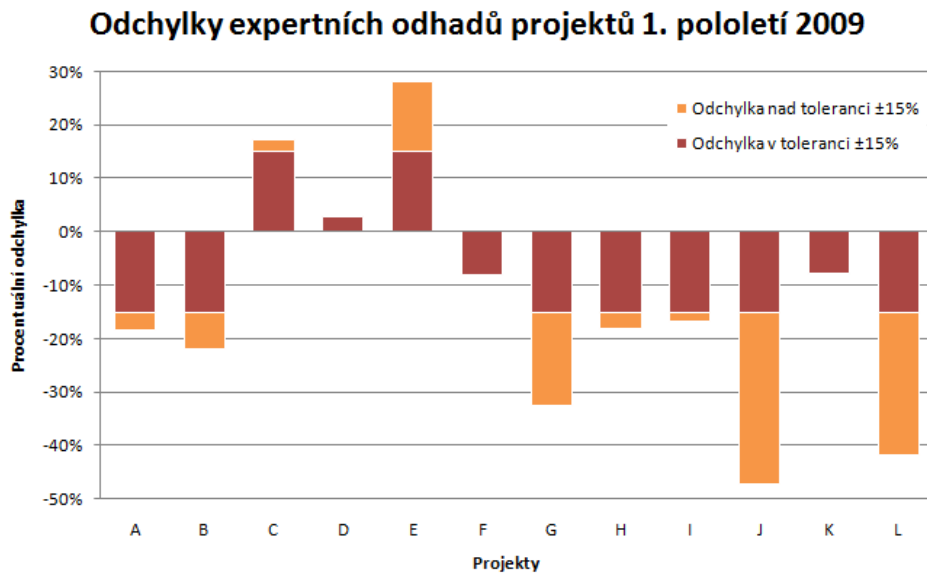
Projektový záměr, spolu s dokumenty výkazu práce, je základním zdrojem historických dat. Odhady pracnosti jednotlivých prvků aplikace jsou zaznamenávány u rozpadu jednotlivých činností. Dokumenty obsahují tuto strukturu informací:

- Projektový záměr - odhady rozsahu pracnosti podle jednotlivých kategorií:
 - analýza
 - programování a datové modelování
 - testování
 - projektové řízení
 - dokumentace
 - uživatelské akceptační testy
- Hodnocení projektů
 - informace o skutečném rozsahu projektu
- Rozpad projektu na prvky - každý prvek je zařazen do dvou skupin:
 - modul: je informace o konečné části systému umístění prvku (aplikace, core, web,...)
 - prvek: informace o prvku (formulář, procedura, webservices,...)

Informace o odhadu rozsahu pracnosti daného prvku se zaznamenávají při tvorbě rozpadu na jednotlivé činnosti.

4.4.2. Analýza historických dat

Z výše uvedených informací jsem získal hodnoty pro porovnání odhadu pracnosti vybraných projektů a skutečným rozsahem. Graf na obrázku 4.2 znázorňuje procentuální odchylku odhadu rozsahu projektu od skutečného rozsahu pro projekty za 1. pololetí roku 2009.



Obrázek 4.2.: Procentuální odchylka odhadu rozsahu vybraných projektů. Zdroj: vlastní

Pro odhady softwarových projektů jsem využil toleranci odchylky odhadu $\pm 15\%$. Uvedená hodnota se započítává jako projektové riziko. Porovnáním zjistíme, že u projektů A, B, G, H, I, J, L byl odhad rozsahu podhodnocen a u projektů C, E nadhodnocen. Pro projekt J existuje objektivní důvod odchylky. V projektu nebyl zohledněn požadavek na zabezpečení komunikace webové služby.

Analýza historických dat za 1. polovinu roku 2009 ukázala, že průměrná odchylka odhadů od skutečnosti je -17% (4.1). Jedná se o podhodnocení odhadů. Absolutní odchylka odhadů za vybrané projekty je 23% .

Tabulka 4.1.: Odchyly odhadu pracnosti projektů 1. poloviny roku 2009 Zdroj: vlastní

	Odhad (dny)	Rozdíl od skutečnosti (dny)	Odchylka
Průměrná odchylka	1258	-209	-17 %
Absolutní odchylka	1258	287	23 %

Analýza potvrzuje předpoklad, že expertní odhady jsou zatíženy optimistickým očekáváním, které způsobuje ekonomické ztráty u dodavatele. Zpřesněním odhadů by bylo možné tyto ztráty snížit. Podložené odhady upevňují pozici při vyjednávání se zákazníkem.

4.5. Výběr metody tvorby odhadů

Na základě mých konzultací s vývojovým týmem byly stanoveny na metodu odhadování rozsahu projektů následující požadavky:

1. *Velikost projektu* - metoda bude aplikována na malé až střední projekty.
2. *Okamžik vytvoření odhadu* - nejdéle po ukončení fáze zahájení podle RUP, tedy v době nadefinování zákaznických požadavků.
3. *Historická data* - metoda bude uchovávat historické údaje.
4. *Začlenění do procesu* - začlenění metody do stávajících procesů vývoje software.

Dle uvedených kritérií jsem provedl analýzu a porovnání metod uvedených v teoretické části práce. Výsledek analýzy zobrazuje tabulka 4.2 s hodnocením jednotlivých metod.

Tabulka 4.2.: Vyhodnocení metod pro odhadování rozsahu software . Zdroj: vlastní

	Expertní odhady	Analogie	UCP	FP
Okamžik vytvoření odhadu	vyhovuje	vyhovuje	nevyhovuje - zasahuje do fáze rozpracování	nevyhovuje - zasahuje do fáze rozpracování
Historická data	nevyužívá	nevyužívá	využívá - koeficienty prostředí	využívá - koeficienty prostředí
Začlenění do procesu	jednoduché	jednoduché	správa koeficientů	správa koeficientů
Výhody	používána, jednoduché zavedení	používána, jednoduché zavedení	výpočet na základě modelu, zohledňuje historické hodnoty	výpočet na základě funkčních částí, zohledňuje historické hodnoty
Nevýhody	nutná znalost prostředí, kognitivní zkreslení	nutná znalost prostředí, kognitivní zkreslení	problémy s úrovní abstrakce modelů	nutné zpracování detailní analýzy

Komentáře k jednotlivým metodám

- Analogie

Metoda je obdobou metody expertních odhadů. Historické údaje se vyskytují ve formě porovnání. V některých případech se metoda využívá podle aktuální situace.

- UCP

Preferovaná metoda. Metoda využívá model případů užití. Model je základem pro výpočet počtu bodů. Metoda nebyla přijata z důvodu nejednotné úrovně abstrakce modelů případů užití. Tvorba modelu také zasahuje do fáze projektování, etapa analýza.

- FP

Metoda využívající podrobnou analýzu. Nevyhovuje tedy podmínce okamžiku vytvoření odhadu.

4.5.1. Vyhodnocení metod

Žádná z uvedených metod nevyhověla zvoleným podmínkám. Nejbližším řešením bylo využití metody UCP. Na základě následných konzultací a zkušeností jsem došel k návrhu rozšíření metody expertních odhadů o výpočet podle metody PERT[13], která je využívána v projektovém řízení.

4.6. Rozšíření metody expertních odhadů

Expertní odhady rozšířím o parametry, které zohledňují vlivy prostředí na vytváření odhad. Prvky jednotlivých požadavků se opakují a je možné využít průměrnou historickou hodnotu odhadu pracnosti prvku. Veškeré hodnoty, pokud nebude uvedeno jinak, jsou udávány jako rozsah pracnosti v člověkodnech.

O_{avg} je průměrná hodnota skutečné pracnosti prvku.

Další parametry jak ovlivnit subjektivní složku expertního odhadu je zavedení intervalu nejlepšího odhadu pracnosti prvku, nejhoršího odhadu pracnosti prvku a očekávaného rozsahu pracnosti prvku:

O_{min} je hodnota nejllepšího případu pracnosti prvku.

O_{max} je hodnota nejhoršího případu pracnosti prvku.

O_{exp} je hodnota očekávané pracnosti prvku.

4. Návrh metody odhadu rozsahu softwarového projektu

Hodnoty O_{min} a O_{max} jsou zavedeny jako nové hodnoty při tvorbě odhadů. Hodnota O_{exp} je hodnotou expertního odhadu (očekávaná pracnost). Obdobně jako faktory vlivu prostředí u metod UCP a FP očekávám, že intervalem odhadu ovlivním výslednou hodnotu odhadu rozsahu pracnosti. Hodnoty pro interval a průměrná hodnota zabezpečí dvě podmínky pro výběr metody:

1. Omezení nevýhody subjektivního odhadu.
2. Využití historických hodnot.

Ostatním podmínkám uvedeným v kapitole 4.5 metoda vyhovuje.

4.6.1. Způsob výpočtu

Programátor vytvářející expertní odhad očekávané pracnosti prvku O_{exp} , vytvoří odhady i pro parametry O_{min} a O_{max} . Pro výpočet očekávané pracnosti použiji vzorec z analýzy odhadování trvání úkolu PERT:

$$ORP_O = \{O_{min} + (4 \cdot O_{exp}) + O_{max}/6\} \quad (4.1)$$

kde ORP_O je obecný odhad pracnosti prvku v člověkodnech. Pro započítání existujících průměrných hodnot skutečného rozsahu pracnosti jsem vzorec 4.1 upravil následovně:

$$ORP_A = \{O_{min} + (3 \cdot O_{exp}) + O_{avrg} + O_{max}\} / 6 \quad (4.2)$$

kde O_{avrg} je průměrná historická hodnota skutečného rozsahu pracnosti prvku v člověkodnech. Pokud nebude prvek mít svoji historickou hodnotu, pak bude využita hodnota O_{max} . Výpočet bude probíhat na základě následujícího vzorce:

$$ORP_B = \{O_{min} + (2 \cdot O_{exp}) + (2 \cdot O_{max})\} / 6 \quad (4.3)$$

V tabulce 4.3 je proveden ukázkový výpočet hodnot jednotlivých parametrů odhadu pracnosti jednotlivých prvků aplikace. Podklady pro výpočet jsem obdržel od programátora na základě jednoduchého zadání. Hodnoty O_{avrg} jsou vypočteny jako průměr historických hodnot pro daný prvek.

Tabulka 4.3.: Příklad výpočtu odhadu rozsahu pracnosti činnosti jednotlivých prvků aplikace.

Zdroj: vlastní

Prvek	O_{min}	O_{exp}	O_{max}	O_{avrg}	ORP_O	$ORP_{A(B)}$
1	2	2,5	3,5	4	2,6	2,8
2	7	8	12	14	8,5	9,5
3	4	5,25	7,5	-	5,5	5,9
4	12	14	20	15	14,7	14,8
5	0,5	1	2,5	-	1,2	1,4
6	4	3	8	6	4	4,5
Celkem	29,5	33,75	53,5		36,5	38,9
Odchylka	-13%	0%	59%		8%	15%

Celkový rozsah odhadů pracnosti prvků se v intervalu od 29,5 člověkodní (nejlepší případ) do 53,5 člověkodní (nejhorší případ). Expertní odhad programátora činí 33,75 člověkodne. Hodnoty ORP_O , ORP_A a ORP_B jsou o mírně vyšší. Odchylka uvedená v posledním řádku je procentní hodnotou, kde základem je expertní odhad O_{exp} .

Porovná-li výsledky odchylek ORP_O , ORP_A a ORP_B s výsledkem tabulky 4.1, pak zavedením výpočtu může dojít k zprávnění odhadů rozsahu softwarových projektů.

4.6.2. Ověření metody

Metodu pro stanovení rozsahu ověřím na realizovaných projektech z 1. poloviny roky 2009, tak pro vyhodnocení použiji projekty 2. poloviny roku 2009. U projektů již realizovaných jsem požádal jednotlivé pracovníky o zpětné vytvoření odhadu podle nové metody. Vzhledem k časovému posunu mezi odhady projektu 1. poloviny roku 2009, přibližně 8 měsíců, a odhady pomocí nové metody nepředpokládám, že by výsledek zpětného odhadu byl výrazně ovlivněn. Odhady projektů z 2. poloviny roku 2009 jsou již částečně ovlivněny prací programátora na daném projektu.

Pro ověření použiji porovnání těchto hodnot:

- Expertní odhad - původní metoda odhadování.
- Nová metoda - výsledky hodnot nové metody.
- Skutečnost - vyhodnocení pracnosti na daný projekt.

4.7. Úprava procesu vývoje software

Aby bylo možné metodu ověřit v praxi a získat tak reálné výsledky, bude upraven proces vývoje software. Rozšíření procesu tvorby software ovlivňuje procesy analýza požadavků a vyhodnocování projektů následujícím způsobem:

4. Návrh metody odhadu rozsahu softwarového projektu

- Analýza požadavků.

Výstup procesu rozpadu zákaznických požadavků a expertních odhadů bude rozšířen o interval nejlepšího a nejhoršího případu odhadu. Výsledky budou uchovány v podobě elektronického záznamu. Zodpovědnou osobou za proces je analytik.

- Hodnocení projektu.

Proces bude rozšířen o záznam skutečného rozsahu pracnosti činnosti na daný prvek. Zodpovědnou osobou za tento proces je projektový manažer.

Z výsledků skutečných rozsahů pracnosti činnosti na jednotlivých prvcích bude vypočtena průměrná hodnota rozsahu pracnosti pro daný prvek.

4.8. Softwarová podpora

Pomocným nástrojem pro zavedení metody odhadování rozsahu projektu poslouží softwarový nástroj. Jako vývojové prostředí aplikace jsem zvolil prostředí Lotus Notes. Důvodem výběru prostředí je programátorská znalost a také používání prostředí vývojovým týmem. Záznamy v prostředí Lotus Notes lze jednoduchým způsobem provazovat pomocí hypertextových odkazů a využívat informace z jiných aplikací. Znalost prostředí Lotus Notes vývojovým týmem usnadní zavedení metody do procesu vývoje softwarových produktů.

Aplikace bude sloužit k vytváření odhadů k jednotlivým projektům a uchovávání historických informací získaných na realizovaných projektech. Na aplikaci jsou kladeny tyto požadavky:

1. Informace o projektu.

Uživatel bude moci zadat základní informace o projektu (název, zodpovědné osoby, období realizace ad.).

2. Vytvoření odhadu.

Ke každému projektu bude vazba na jeden nebo více odhadů rozsahu daného projektu. Počet variant je neomezený. Varianty mohou být jak podle návrhu řešení, tak i podle aktuálnosti dané varianty.

3. Přehledy.

Aplikace bude obsahovat přehledy podle projektů i podle odhadů.

4. Parametrizace.

Správa číselníků a nastavení bude uživatelsky přístupné.

Návrh aplikace a modely jsou uvedeny v příloze B.

5. Analýza výsledků

Kapitola obsahuje analýzu výsledků z nasazení rozšířené metody expertních odhadů pro určení rozsahu softwarového projektu. Výstupy jsou rozdělené na dvě základní části. V první části je analýza výsledků na již realizovaných projektech. Druhá část obsahuje analýzu projektů, u kterých byl odhad tvořen původní používanou metodou, tak metodou rozšířenou.

5.1. Analýza historických dat a výstupů aplikace

Pro analýzu jsem zvolil porovnání historických hodnot, hodnot vytvářených metodou expertních odhadů založených na jednom čísle a hodnot vypočtených navrženou metodou využívající intervalu. Původním plánem bylo využití odhadování pracnosti pouze pro část programování (implementace) prvků aplikace. Doplněním parametrů do aplikace bylo možné zaznamenávat i položky původně nezařazené (analýza, testování, projektové řízení, akceptační testy). Změna nastavení umožní ověřit metodu i na dalších disciplínách projektu.

Následující porovnání vychází z hodnot celkového rozsahu softwarového projektu. Pro porovnávání hodnot zavedeme parametr původního historického odhadu pracnosti O_{hist} a hodnotu odhadu rozsahu projektu ORP_{hist} , kde platí $O_{hist} = ORP_{hist}$. Veškeré hodnoty, pokud nebude uvedeno jinak, jsou uvedeny v člověkodnech (MD).

Analýza výstupů je rozdělena na dvě části:

1. Historické projekty - vybrané projekty realizované v 1. pololetí roku 2009 (leden-červen)
2. Realizované projekty - vybrané projekty realizované v 2. pololetí roku 2009 (červenec-listopad)

Analýza výstupů z jednotlivých částí je rozdělena takto:

1. Porovnání parametrů expertních odhadů - porovnání hodnot původních expertních odhadů a odhadů tvořených podle nové metody.
2. Porovnání odhadů rozsahu projektů - porovnání vypočtených odhadů rozsahů, skutečnému rozsahu projektu a původních expertních odhadů.
3. Porovnání odchylek rozsahu projektů - porovnání odchylek vypočtených hodnot proti skutečnému rozsahu projektu.

5.2. Historické projekty

Historické projekty, realizované projekty z 1. pololetí roku 2009, slouží k ověření metody a k základní analýze výstupů. Seznam projektů a jejich původní expertní odhady jsou uvedeny v tabulce 5.1 včetně vypočtených rozdílů a odchylek. Průměrná odchylka je ve stanovené toleranci $\pm 15\%$.

Tabulka 5.1.: Odhady a skutečný rozsah historických projektů. Zdroj: vlastní

Projekt	A	B	C	D	E	F	G	H	I	J	K	L	Součet	Průměr
Celkem	38	64	58	34	100	112	178	210	255	110	39	60	1258	105
- Analýza	5	8	10	4	28	16	35	30	42	15	8	5	206	17
- Programování	22	36	26	16	29	64	73	100	108	48	14	30	566	47
- Testování	6	6	8	5	20	14	30	35	40	12	7	17	200	17
- Akceptační testy	5	6	8	5	8	8	20	30	40	20	5	2	157	13
- Projektové řízení	0	8	6	4	15	10	20	15	25	15	5	6	129	11
Skutečnost	45	78	48	33	72	121	236	247	298	162	42	85	1467	122
Rozdíl	-7	-14	10	1	28	-9	-58	-37	-43	-52	-3	-25	-209	-17
Absolutní rozdíl	7	14	10	1	28	9	58	37	43	52	3	25	287	24
Odchylka [%]	-18	-22	17	3	28	-8	-33	-18	-17	-47	-8	-42	-	-14
Absolutní odch.[%]	18	22	17	3	28	8	33	18	17	47	8	42	-	22

5.2.1. Porovnání parametrů expertních odhadů

Srovnáním parametrů O_{hist} , O_{min} , O_{exp} a O_{max} , tabulka 5.2, lze vysledovat skutečnost, kde odhady parametru O_{exp} jsou u jednotlivých projektů rozdílné, proti odhadům tvořených na základě jednoho čísla O_{hist} . Podle teoretických základů by měla být hodnota O_{exp} shodná s hodnotou O_{hist} . Zavedením tvorby odhadů pomocí intervalu, ovlivnilo vnímání pro tvorbu očekávané pracnosti.

Tabulka 5.2.: Hodnoty parametrů expertních odhadů pracnosti pro historické projekty. Zdroj: vlastní

Projekt	A	B	C	D	E	F	G	H	I	J	K	L
O_{hist}	38	64	57	34	100	112	178	210	255	110	39	60
O_{min}	20,75	53	26	9	43,5	78,5	143	177	187,5	90	24,5	57,5
O_{exp}	38,5	69	41	32	69,5	102	196	220,5	237,5	121	35	79
O_{max}	65,5	90	63	42	104,5	159	273	304	345	176	59	118

5.2.2. Porovnání odhadů rozsahu projektů

Výpočet hodnot ORP_O , ORP_A a ORP_B vychází ze vzorců uvedených v kapitole 4.6.1. Tabulka 5.3 obsahuje také původní odhady rozsahu projektů a skutečnou hodnotu rozsahu projektu. Zvý-

rozněné hodnoty jsou nejbliže skutečnému rozsahu pracnosti projektu.

Srovnání odhadů rozsahu projektů ORP_{hist} , ORP_O , ORP_A , ORP_B a skutečného rozsahu projektu. Celkově nejpřesnější odhady poskytuje hodnota ORP_B , kde je váha parametru O_{max} dvojnásobná oproti použití ve vzorci pro výpočet parametru ORP_O .

Tabulka 5.3.: Hodnoty odhadů rozsahu historických projektů. Zdroj: vlastní

Projekt	A	B	C	D	E	F	G	H	I	J	K	L
ORP_{hist}	38,0	64,0	57,0	34,0	100,0	112,0	178,0	210,0	255,0	110,0	39,0	60,0
ORP_O	40,0	69,8	42,2	29,8	71,0	107,6	200,0	227,2	247,1	125,0	37,3	81,9
ORP_A	43,6	68,0	44,3	31,2	74,9	109,9	202,0	226,8	253,5	126,7	39,1	85,3
ORP_B	44,5	73,3	45,8	31,5	76,8	117,1	212,8	241,1	265,0	134,2	41,3	88,4
Skutečnost	45,0	78,0	48,0	33,0	72,0	121,0	236,0	247,0	298,0	162,0	42,0	85,0

5.2.3. Porovnání odchylek rozsahu projektů

Tabulka 5.4 obsahuje procentní odchylky jednotlivých hodnot odhadů projektů proti skutečnému rozsahu projektu. Navržené vzorce pro výpočet hodnot ORP_O , ORP_A a ORP_B přináší přesnější odhady rozsahu projektů. Nejnižší průměrnou odchylku a absolutní průměrnou odchylku má výpočet hodnoty ORP_B . Hodnoty -4,8 % a 6,5 % znamenají výrazné zpřesnění tvorby odhadů při stanovené toleranci $\pm 15\%$.

Tabulka 5.4.: Tabulka odchylek hodnot odhadů a skutečného rozsahu projektu. Zdroj: vlastní

Projekt	A	B	C	D	E	F	G	H	I	J	K	L	Průměr	Absolutní
$ORP_{hist}[\%]$	-18	-22	17	3	28	-8	-33	-18	-17	-47	-8	-42	-13,8	21,8
$ORP_O[\%]$	-12	-12	-14	-11	-1	-12	-18	-9	-21	-30	-13	-4	-13,1	13,1
$ORP_A[\%]$	-3	-15	-8	-6	4	-10	-17	-9	-18	-28	-7	0	-9,8	10,4
$ORP_B[\%]$	-1	-6	-5	-5	6	-3	-11	-2	-12	-21	-2	4	-4,8	6,5

5.3. Realizované projekty

Jedná se o projekty, kde byl odhad rozsahu projektu řešen primárně pomocí původní metody expertních odhadů a zároveň byl vytvořen odhad rozsahu pomocí metody výpočtu. Pro sběr odhadů bylo stanoveno, že hodnota expertního odhadu bude tvořena jako první a následně bude vytvořen odhad pomocí nové metody. Při obráceném postupu by mohl dojít k ovlivňování hodnoty expertního odhadu. Analýza hodnot historických projektů naznačuje, že při stanovování hodnot intervalu parametrů O_{min} a O_{max} , může dojít k ovlivnění tvorby expertního odhadu.

Tabulka 5.5 obsahuje seznam projektů a základní hodnoty. Průměrná odchylka projektů od skutečnosti činí -7 %. To znamená, že došlo ke zpřesnění tvorby odhadů proti historickým projektům.

5. Analýza výsledků

Tabulka 5.5.: Odhady a skutečný rozsah realizovaných projektů. Zdroj: vlastní

Projekt	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Součet	Průměr
Celkem	125	78	28	30	31	43	36	95	80	77	39	75	39	776	60
- Analýza	18	15	4	6	4	5	6	24	20	8	4	6	4	124	10
- Programování	64	38	12	15	19	25	14	48	38	52	25	50	23	423	33
- Testování	20	12	4	2	3	5	5	9	6	5	2	5	3	81	6
- Akceptační testy	8	5	6	4	3	5	7	6	10	4	5	6	6	75	6
- Projektové řízení	15	8	2	3	2	3	4	8	6	8	3	8	3	73	6
Skutečnost	138	90	33	36	28	46	36	102	84	86	44	81	34	838	64
Rozdíl	-13	-12	-5	-6	3	-3	0	-7	-4	-9	-5	-6	5	-62	-5
Absolutní rozdíl	13	12	5	6	3	3	0	7	4	9	5	6	5	78	6
Odchylka [%]	-10	-15	-18	-20	10	-7	0	-7	-5	-12	-13	-8	13	-	-7
Absolutní odchylka [%]	10	15	18	20	10	7	0	7	5	12	13	8	13	-	11

5.3.1. Porovnání parametrů expertních odhadů

Hodnoty expertního odhadu O_{hist} a hodnoty očekávaných pracností O_{exp} jsou si velmi podobné. Tato skutečnost může znamenat, že provedeným seznámením s novou metodou tvorby odhadů, došlo k ovlivnění pracovníků a využívají metodu podvědomě.

Tabulka 5.6.: Hodnoty parametrů expertních odhadů pracnosti pro realizované projekty. Zdroj: vlastní

Projekt	M	N	O	P	Q	R	S	T	U	V	W	X	Y
O_{hist}	125	78	28	30	31	43	36	95	80	77	39	75	39
O_{min}	87	59,75	18	15,25	15	31,25	25	66	65	56	28	56	24
O_{exp}	117	75,5	29	29	29	44	36	83	76	78,5	39,5	72	31
O_{max}	162	114	46	39	46	61	56	112	101	110	58,5	108	43

5.3.2. Porovnání odhadů rozsahu projektů

I přes skutečnost, že jsou hodnoty expertních odhadů ORP_{hist} dostatečně přesné, výpočty hodnot ORP_O , ORP_A a ORP_B obsahují zpřesnění odhadů. U realizovaných projektů se potvrdil jako nejbližší skutečnosti výpočet hodnoty ORP_B .

Tabulka 5.7.: Hodnoty odhadů rozsahu realizovaných projektů. Zdroj: vlastní

Projekt	M	N	O	P	Q	R	S	T	U	V	W	X	Y
ORP_{hist}	125	78	28	30	31	43	36	95	80	77	39	75	39
ORP_O	120	79	30	28	30	45	38	85	78	80	41	75	32
ORP_A	122	83	32	30	32	45	41	86	78	79	41	76	34
ORP_B	127	86	33	30	32	48	41	90	83	85	44	81	34
Skutečnost	129	90	33	27	27,5	46	36	82	84	86	44	81	34

5.3.3. Porovnání odchylek rozsahu projektů

Odchytky parametrů odhadu jednotlivých hodnot parametrů ORP_O , ORP_A a ORP_B proti skutečnému rozsahu projektu svoji přesností spadají do stanovené tolerance. Výpočet dále potvrzuje, že nejpřesnější odhady jsou vypočteny pomocí parametru ORP_B , kde absolutní odchylka činí 4,8 %. Průměrná odchylka parametru je -3,6 %. Odhady jsou mírně podhodnoceny.

Tabulka 5.8.: Tabulka odchylek hodnot odhadů a skutečného rozsahu projektu. Zdroj: vlastní

Projekt	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Průměr	Absolutní
$ORP_{hist}[\%]$	-10	-15	-18	-20	10	-7	0	-7	-5	-12	-13	-8	13	-7	10,6
$ORP_O[\%]$	7	12	9	-4	-9	2	-6	-4	7	7	7	7	6	3,2	6,7
$ORP_A[\%]$	5	8	3	-11	-16	2	-14	-5	7	8	7	6	0	0	7,1
$ORP_B[\%]$	2	4	0	-11	-16	-4	-14	-10	1	1	0	0	0	-3,6	4,8

5.4. Kvalita odhadů

Uchování historických údajů přináší možnosti sledování další ukazatele. Prvním je ukazatel kvality odhadu a druhý je ukazatel relativní chyby. Základem pro výpočet je existence historických hodnot odhadovaných projektů.

5.4.1. Ukazatel kvality odhadu

Jedná se o procentní vyjádření zda, skutečná hodnota pracnosti prvku byla v intervalu parametrů O_{min} a O_{max} . Jedním z výstupů je možnost vyhodnocení kvality odhadu za jednotlivé fáze projektu, jak je uvedeno v tabulce 5.9.

5. Analýza výsledků

Tabulka 5.9.: Kvalita odhadu rozdělená dle fází projektu Zdroj: vlastní

Fáze	Hodnota
Analýza	70 %
Programování	84 %
Projektové řízení	91 %
Testování	85 %
Akceptační testy	97 %

Přehled v tabulce ukazuje, že nejnižší kvalitu odhadu rozsahu pracností má fáze zpracování analýzy. Možností jak zvýšit kvalitu odhadu je rozšíření intervalu a zvýšení zejména horní hranice, parametr O_{max} . Obecně je odhadování pracnosti tvorby analýzy složité.

5.4.2. Ukazatel relativní chyby

Na historických informacích o odhadech lze aplikovat vzorec 5.1 pro výpočet relativní chyby $MRE[5, 10]$,

$$MRE = | (O_{real} - ORP_{(x)}) / O_{real} | \quad (5.1)$$

kde O_{real} je hodnota skutečného odhadu pracnosti prvku a ORP je hodnota rozsahu vypočteného na základě vzorce vycházejícího z PERT.

Tabulka 5.10.: Relativní chyba odhadu rozdělená dle fází projektu Zdroj: vlastní

Fáze	Hodnota
Analýza	18 %
Programování	21 %
Projektové řízení	18 %
Testování	16 %
Akceptační testy	7 %
Průměr	16 %

Výsledky v tabulce 5.10 uvádějí odchylku relativní chyby za jednotlivé fáze projektu. Kvalita odhadu analýzy je 70 %, ale hodnota relativní chyby vytvořeného odhadu je 18 % proti 21 % u fáze programování.

5.5. Vyhodnocení výsledků

Analýza výsledků hodnot historických projektů potvrdila záměr, že navržená metoda umožní vytvářet přesnější odhady rozsahu softwarových projektů. Pro ověření byla metoda použita zpětně

po realizaci jednotlivých projektů. Hodnoty mohou být ovlivněny skutečnou pracností vybraného prvku. K ověření nové metody odhadování rozsahu softwarových projektů bylo stanoveno používání metody souběžně se stávající metodou expertních odhadů na nově realizované projekty. Expertní odhad byl vytvářen jako první a následně byla zařazena nová metoda, aby se snížilo ovlivňování jednotlivých odhadů.

Vyhodnocení zavedením metody odhadování rozsahu softwarových projektů na realizovaných projektech potvrdilo, že došlo ke zpřesnění vytvořených odhadů. Skutečnost, že členové týmu byli seznámeni s metodou tvorby intervalu u pro odhad pracnosti jednotlivých prvků daného požadavku, podle mého názoru vedla ke zpřesnění využívané metody expertních odhadů.

Využitím výpočtu parametrů ORP_O , ORP_A a ORP_B lze docílit přesnějších výsledků. Volba konkrétního vzorce a nastavení jednotlivých vah závisí na výsledcích odhadů v minulosti s porovnáním se skutečným rozsahem softwarového projektů.

6. Závěr

Cílem práce bylo stanovení vhodného způsobu ohodnocení rozsahu softwarového projektu na základě definovaných zákaznických požadavků. Volba a ověření vhodné metody pro tvorbu odhadů rozsahu softwarových projektů je časově náročný úkol. Hlavní překážkou jsou vlastnosti prostředí, zejména vztah zákazníka a dodavatele nebo využívání metod řízení a vývoje softwarových projektů. Další překážkou jsou technologické i ekonomické změny ovlivňující obchodní procesy, které mají vliv na změny v informačních systémech.

Náplň bakalářské práce, tzn. zavedení způsobu ohodnocení rozsahu softwarového projektu, vychází z reálného prostředí, ve kterém jsem působil na pozici analytik informačních systémů. Důvodem pro volbu nové metody odhadování bylo zpřesnění odhadů. Porovnáním historických hodnot odhadů rozsahu projektů se skutečností, za období 1. pololetí roku 2009, bylo zjištěno podhodnocování odhadů rozsahu projektu. Mezi další důvody patří využívání historických hodnot pro tvorbu nových odhadů, užitečnost odhadu a rychlost stanovení odhadu. Praktické ověření metody bylo provedeno na projektech z období 2. pololetí roku 2009.

Postup zpracování bakalářské práce spočíval v následujících krocích:

1. Analýza zákaznických požadavků
2. Analýza využití historických dat
3. Srovnání použitelnosti existujících metod
4. Návrh vlastní metody včetně výpočtu
5. Ověření navrženého postupu

Přínosy bakalářské práce jsou:

- Návrh metody pro odhad softwarového projektu; tato metoda zpřesňuje hodnotu odhadu, což bylo ověřeno na reálných datech.
- Využití historických dat organizace z realizovaných softwarových projektů.

Navržený postup přinesl, kromě zpřesnění, nový pohled způsobu tvorby odhadů. Výhodou metody je jednoduchost zařazení do firemních procesů. Nad rámec práce byla vytvořena aplikace pro tvorbu odhadů a uchování historických hodnot. Výsledek práce lze rozšířit o kontrolní výpočty a dlouhodobé zkoumání hodnot odhadů metody.

Literatura

- [1] BERTRAND, Portier. *SOA terminology overview, Part 2: Development processes, models, and assets* [online]. [cit. 2009-12-26]. Dostupný z WWW: <http://www.ibm.com/developerworks/webservices/library/ws-soa-term2/>
- [2] *Borland ČR [online]. 2008* [online]. [cit. 2010-04-26]. Testování založené na požadavcích. Dostupné z WWW: http://www.borland.cz/resource/requirements_based_testing.html
- [3] CLEMMONS, Roy. *Project Estimation with Use Case Points*. [online]. [cit. 2009-12-26]. Dostupný z WWW: <http://www.codeproject.com/KB/architecture/usecasep.aspx>
- [4] *COCOMO II - Constructive Cost Model* [online]. [cit. 2009-12-26]. Dostupný z WWW: http://sunset.usc.edu/csse/research/COCOMOII/cocomo_main.html
- [5] CONTE, Samuel D. *Software engineering metrics and models*. Redwood City, CA, USA : Benjamin-Cummings Pub Co, 1986. 396 s. ISBN 0-8053-2162-4.
- [6] DOMINGUEZ, Jorge. *The Curious Case of the CHAOS Report 2009*. [online]. [cit. 2010-04-26]. Dostupný z WWW: <http://www.projectsmaart.co.uk/the-curious-case-of-the-chaos-report-2009.html>
- [7] ELSHAW, Mark. *Albrecht's Function Point Example* [online]. [cit. 2009-12-26]. Dostupný z WWW: http://www.his.sunderland.ac.uk/~cs0mel/Alb_Example.doc
- [8] HUDEC, Marek. *Moderní metodiky vývoje software a jejich aplikace v malých týmech : Přehled metodik vývoje software a jejich využití jednotlivci a malými vývojovými týmy*. Hradec Králové, 2008. 99 s. Diplomová práce. Univerzita Hradec Králové. Vedoucí práce Ing. Hana Šafránková.
- [9] KUKOL, Vladan. *Odhadování projektů*. In *Prezentace LaTes 2009* [online]. Praha : KOMIX s.r.o., 2009 [cit. 2010-05-6]. Dostupné z WWW: http://www.komix.cz/cs-cz/Tisk/Lates_2009/~media/Komix/Tisk/

Literatura

LaTe_2009/Prezentace/LaTes09_Kukol_Odhadovani_pracnosti_projektu.ashx>.

- [10] McCONNELL, Steve. *Odhadování softwarových projektů : Jak správně určit rozpočet, termíny a zdroje*. Vyd. 1. Brno : Computer Press, a.s., 2006. 317 s. ISBN 80-251-1240-3.
- [11] PALETA, Petr. *Co programátory ve škole neucí, aneb, Softwarové inženýrství v reálné praxi*. 1. vyd. Brno : Computer Press, 2003. 337 s. Obsahuje rejstřík. ISBN 80-251-0073-1.
- [12] PERGL, R. *Metody odhadu pracnosti založené na modelu*. In , 5. 6. 2006 Ostrava. ČSSI. Ostrava: ČSSI, 2006, s. 1-8.
- [13] *Program Evaluation and Review Technique*. [online]. [cit. 2010-04-26]. Wikipedia. Dostupný z WWW: http://en.wikipedia.org/wiki/Program_Evaluation_and_Review_Technique
- [14] ROBERTSON, Suzanne, ROBERTSON , James C. *Mastering the Requirements Process*. 2nd edition. [s.l.] : Addison-Wesley Professional, March 27, 2006. 592 s. ISBN 0321419499.
- [15] ROBERTSON, James; ROBERTSON, Suzanne. *The Volere Template* [online]. 9. vydání. London : Atlantic Systems Guild, 2003 [cit. 2010-04-28]. Dostupné z WWW: <http://www.volere.co.uk/pdf%20files/template_cz.pdf>.
- [16] STRUSKA, Z. *Srovnání metod COCOMO a analýzy Function Points*. [online]. [cit. 2010-04-26]. Dostupný z WWW: <http://www.agris.cz/etc/textforwarder.php?iType=2&iId=152840&PHPSESSID=3e>
- [17] *The Rational Unified Process* [online]. [cit. 2009-12-26]. Dostupný z WWW: <http://www-01.ibm.com/software/awdtools/rup/>

Přílohy

A. Správa a evidence požadavků

Příklad evidence pomocí aplikace

Formulář evidovaného zákaznického požadavku

Name: Statistics and client's info	
Module	ADDRESSBOOK
Feature Group	AGENTS
F004	
Coment: Agen can see set of statistics data , limited ONLY for his clients, and all details about his client	
Acceptation Criteria AGENT má přístup finančním informacím pouze klientů přidělených dánemu agentu.	
Remark:	
Priority:	<input checked="" type="radio"/> Must be <input type="radio"/> Ok <input type="radio"/> Nice to have
Type:	<input type="radio"/> Business Rule <input checked="" type="radio"/> Funkční <input type="radio"/> Nefunkční
Stakeholder:	Jan Valenta
Reference RQ:	F002 Client Assignments F025 Company Role F062 ACTIONS (TASKS)
Use Cases	UC002 Show Client's Statistics UC005 Manage Account Info UC020 Show Financial & Projects statistics
Status	<input type="radio"/> nový <input type="radio"/> odložený <input type="radio"/> akceptovaný <input checked="" type="radio"/> implementovaný <input type="radio"/> otestovaný <input type="radio"/> uzavřený <input type="radio"/> změněný <input type="radio"/> zrušený
Category	
SubCategory	

A. Správa a evidence požadavků

Přehled zákaznických požadavků podle Use Case

▶ UC005 Manage Account Info		
▼ UC006 Generate Statements		
F037	🔍 Statements creating	▼ For every SUPPLIER, TC keeps MONTHLY ACCOU closed are the STATEMENT is generated Předpokládám, že kategorie by se mohla jmenů kteří nejsou zahrnuti do systému Statement rep Lze Statement Report vystavit i dříve? Případně
▼ UC007 Charging Order		
F039	🔍 Statement/Invoice List + Money Orders	▶ TC displays list of suppliers invoices, organized by l systems.
F124	🔍 Supplier Invoice Payments	TC system should generate a payment list that can Reference with these payments should be the mon
▼ UC008 Claim Receive Invoice		
F039	🔍 Statement/Invoice List + Money Orders	▶ TC displays list of suppliers invoices, organized by l systems.
F124	🔍 Supplier Invoice Payments	TC system should generate a payment list that can Reference with these payments should be the mon
▶ UC009 Check Payment		
▶ UC010 Issue Invoice		
▶ UC011 Check Projects Status		
▶ UC013 Send Projekt (iPort)		
▶ UC014 Show Invoices Reports		
▶ UC015 Manage Avialibity		

Přehled zákaznických požadavků podle stavu

▶ IMPLEMENTOVANÝ	68%	89
▶ NOVÝ	7%	9
▼ ODLOŽENÝ	17%	22
▶ ADDRESSBOOK	2%	3
▼ BANKS	2%	2
▼ Invoice ISSUED to client	2%	2
F127	🔍 HOW To make new invoices	Selected jobs are used
F035	🔍 To displaied statistics	see statistics
▶ COORDINATION	1%	1
▶ EXTERNAL MODULES	7%	9
▶ FILE MANAGEMENT	1%	1
▶ STATISTICS	4%	5
▶ WEB INTERFACES	1%	1
▼ ZMĚNĚNÝ	2%	3
▼ ADDRESSBOOK	2%	2
▶ SUUPL. SCORRING	2%	2
▶ STATISTICS	1%	1
▶ ZRUŠENÝ	5%	7
	100%	130

Příklad evidence požadavků v tabulkovém editoru

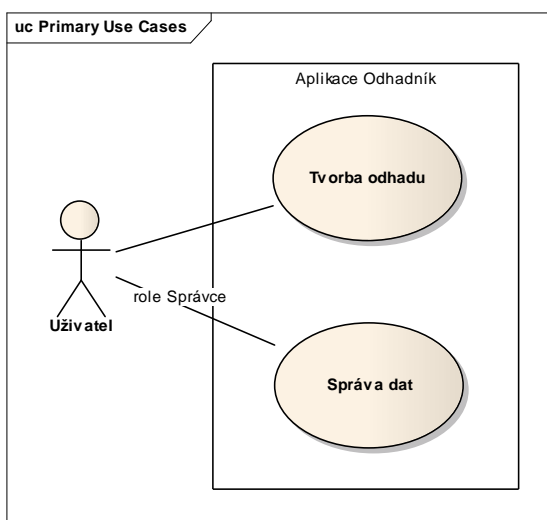
ID	Termin	Kategorie	Předmět	Otázky a úkoly	Řeší	Stav	Poznámka	Řešení	Riziko
005 09-01-27_T03 09-01-23_01	20.2.2009	Interface	Ošetření provozu pokladny při nedostupnosti AS400	Ošetření dalších typů pokladních operací - má existovat nějaká omezení? Které operace budou během výpadku dostupné a které ne?	Prouzová	Dohodnut 0	VP: Vklad: standardně dostupný bez omezení Výběr: dostupný bez možnosti kontroly zůstatku - VZDY vyžadat potvrzení SUPERVIZOREM	VP: Možnost debetu	
006 09-01-27_T03 09-01-23_01	20.2.2009	Interface	Ošetření provozu pokladny při nedostupnosti AS400	Případná aktualizace zůstatků u alternativních zdrojů? Bude potřeba informovat SPE, že kromě karet mohou zůstatek ovlivnit i výběry na pokladně?	Prouzová	Dohodnut 0	VP: NE	VP: Možnost debetu	
007 09-01-27_T03 09-01-23_01	20.2.2009	Interface	Ošetření provozu pokladny při nedostupnosti AS400	Případná aktualizace zůstatků pro internetové bankovníctví (UDEBS)?	Holub	Dohodnut 0	VP: NE DF: počítá s tím Bude zajištěna aktualizace zůstatku pro internetbanking v běžném provozu?	VP: Možnost debetu	
008 09-01-27_T03 09-01-23_01	20.2.2009	Interface	Ošetření provozu pokladny při nedostupnosti AS400	Je třeba získat vyjádření oddělení bezpečnosti banky na možnost výběrů z pokladny bez znalosti zůstatku účtu (se schválením supervizora)	Prouzová	Zadáno			
010 09-01-27_T05 09-01-23_012	20.2.2009	Účtování	Tranzitní účty TS - MID (a jejich propojení s VBCC)	Možnosti: - tříti sada účtů (současné sady: bezhotovostní platby, ZPS) - využití sady účtů pro bezhotovostní platby s dostatečným rozlišením hotovostních plateb pro reconciliaci	Holub	Dohodnut 0	Vznikne samostatná třetí sada účtů (aplikačních master) pro pokladní transakce		
017 09-01-23_G7	TBD	Recyklační pokladny	Interface	Přesný popis rozhraní TS a recyklační pokladny.	Fliger	Zadáno	DF: Proběhla schůzka s předávanými pokladny. Více v samostatném bodě 036.		
018 09-01-23_09	TBD	Recyklační pokladny	Návrh HW architektury	Zpracovat architekturu HW řešení a projednat s IT Security, IT Network a IT Application.	Holub	Zadáno	DF: Předpokládám, že se to týká recyklačních pokladen, pro rozhraní TS bude využít existující HW, ne?		
019 09-01-23_010	11.2.2009	Recyklační pokladny	Návrh HW architektury	Upřesnit komunikační protokol a centralizaci řešení, návrh: 1. centralizovaný server a z něj komunikace na pobočkové servery nebo 2. pobočkové servery	Holub	Dohodnut 0	DF: Vybrána varianta 2 - pobočkové servery (ne servery) komunikují přímo s ARP. S tím souvisí problém ošetření	ZMĚNA - bud	

B. Návrh aplikace pro tvorbu odhadů

Návrh

Případy užití

Základní činností, pro kterou je aplikace tvořena, je Tvorba odhadů. Vedlejší činností je správa dat. To znamená údržba a nastavení aplikace. Popis jednotlivých funkcí je popsán v následujících případech užití B.1.



Obrázek B.1.: Model případu užití. Zdroj: vlastní

Název Use Case Tvorba odhadu

Uživatel Registrovaný uživatel systému

Vstupní podmínky Aplikace je řádně nakonfigurována.

Hlavní scénář

1. Uživatel vytvoří Profil projektu a zaznamená potřebné informace:

B. Návrh aplikace pro tvorbu odhadů

a) Název projektu, popis, zodpovědné osoby, období realizace.

2. Uživatel vkládá do projektu jednotlivé položky prvků:

- a) Uživatel zvolí variantu řešení pro, kterou prvek vytváří.
- b) V dokumentu prvku zvolí kategorii, modul a prvek.
- c) Vloží informace o vytvořeném odhadu.
- d) Záznam prvku uloží.

3. Uživatel pokračuje uložením projektu.

4. Konec scénáře.

Alternativní scénáře

1. rozšíření bodu 2. úprava prvku.

- a) Uživatel otevře existující prvek a provede úpravy
- b) Uživatel pokračuje krokem 2.c Hlavního scénáře.

2. rozšíření bodu 2. vložení hodnot skutečného rozsahu.

- a) Uživatel pokračuje krokem 1. alternativního scénáře.

Výstupní podmínky Je založený záznam Projekt, který obsahuje záznamy Položka.

Obrazovky Profil projektu, Položka, seznam Projektů

Poznámka Aplikace využívá základní vlastnosti prostředí, které umožňují funkce Tisk a Odeslat.

Název Use Case Správa dat

Uživatel Registrovaný uživatel systému, role Správce

Vstupní podmínky Aplikace je nainstalována.

Hlavní scénář

1. Uživatel provede přepnutí aplikace do sekce *Konfigurace*.
2. Uživatel vloží potřebná nastavení pro zvolené prostředí:
 - a) Seznam kategorií.
 - b) Seznam stavů projektu.
3. Konec scénáře.

Alternativní scénáře

1. rozšíření bodu 2. úprava prvku číselníku.
 - a) Uživatel otevře příslušný prvek a zvolí *Editace*.
 - b) Uživatel pokračuje krokem 2. Hlavního scénáře.

Výstupní podmínky Jsou nastavené parametry pro fungování aplikace.

Obrazovky Profil projektu, Položka, seznam Projektů

Poznámka

Model tříd

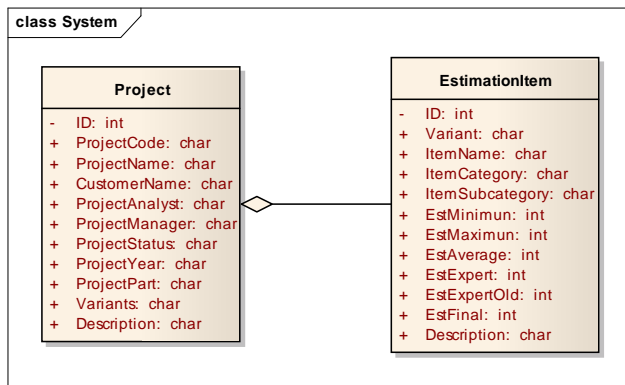
Model tříd vychází z reálných objektů používaných při tvorbě odhadu softwarového projektu. Základním objektem je *Projekt*, který obsahuje jednotlivé objekty typu *Prvek*.

Project objekt je obrazem reálného objektu *Profil projektu*. Objekt obsahuje informace o projektu: *Název, Kód, Zákazník, zodpovědné osoby, Popis, Stav projektu*.

Item je objektem *Prvku aplikace*. Objekt obsahuje atributy: *modul, prvek, hodnoty odhadů, ad.*

Model tříd je uveden na obrázku B.2.

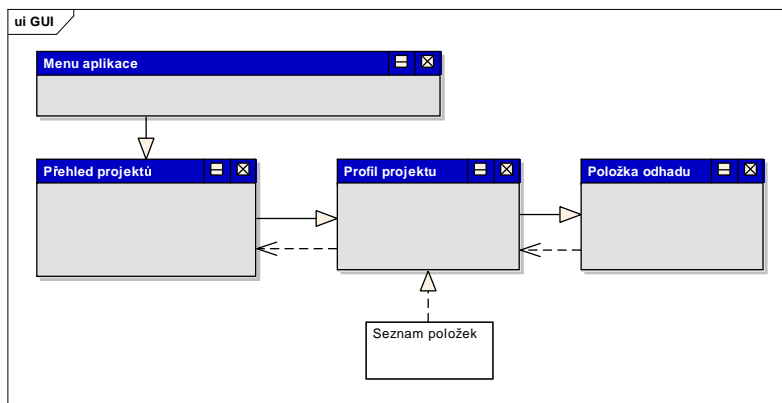
B. Návrh aplikace pro tvorbu odhadů



Obrázek B.2.: Model tříd aplikace pro tvorbu odhadů rozsahu projektů Zdroj: vlastní

Obrazovky

Obrazovky jsou navrženy, tak aby poskytovali uživateli přehled o aktuálních projektech a vytvořených odhadech. Obrázek B.3 obsahuje model závislosti jednotlivých obrazovek.



Obrázek B.3.: Návrh ovládání obrazovek aplikace . Zdroj: vlastní

Základní obrazovkou je *Přehled projektů*. Zobrazuje seznam projektů, včetně informací o stavu, zodpovědných osobách a další. Následují obrazovky *Profil projektu* a *Položka*. Ty slouží pro zadávání a editaci záznamů.

Ukázka uživatelského rozhraní aplikace

Přehled obrazovek aplikace.

- Přehled projektů - je základní a úvodní částí aplikace.

Rok	Číslo	Název projektu	Zadavatel	Manažer	Analytik
▼ 2009					
▶ 1. pololetí					
▼ 2. pololetí					
	50	<input checked="" type="checkbox"/> ECS I.	Skarpíšek	Brož	Kluz
	51	<input checked="" type="checkbox"/> ECS II.	Skarpíšek	Brož	Kluz
	54	<input checked="" type="checkbox"/> EU Měsíční generování výpisů	Rys	Brož	Šklíba
	55	<input checked="" type="checkbox"/> EU Identifikátor realizované transakce	Tafatová	Brož	Šklíba
	56	<input checked="" type="checkbox"/> EU Rozpis poplatků + šablony	Hamouzová	Brož	Kluz
	57	<input checked="" type="checkbox"/> EU Platební karty	Vakoč	Brož	Filip
	58	<input checked="" type="checkbox"/> EU Platby	Richter	Brož	Foldyna
	59	<input checked="" type="checkbox"/> EU Výpisy ve formátu PDF	Exnerová	Brož	Flieger
	60	<input checked="" type="checkbox"/> EU Účtování karet	Adensam	Brož	Filip
	61	<input checked="" type="checkbox"/> Poplatek za překročení úvěru	Voldřich	Šilar	Kluz
	62	<input checked="" type="checkbox"/> Specifické úročení provozních úvěrů	Vakoč	Šilar	Foldyna
	63	<input checked="" type="checkbox"/> Hypotéky W	Richter	Šilar	Filip
	64	<input checked="" type="checkbox"/> Micro kampaně	Zunová	Brož	Kluz
▼ 2010					
▶ 1. pololetí					

Obrázek B.4.: Přehled projektů Zdroj: vlastní

- Profil projektu - slouží pro vytváření záznamů o projektu.

Uložit a zavřít Uložit Zavřít Log He

Název projektu

Stručný popis

Zodpovědné osoby

Manažer:

Zákazník:

Analytik:

Ostatní

Kód projektu LN:

Rok:

Období:

Stav projektu:

Varianty:

Přílohy:

Obrázek B.5.: Profil projektu aplikace. Zdroj: vlastní

B. Návrh aplikace pro tvorbu odhadů

- Položka odhadu - informace o vytvořených odhadech. Formulář slouží také pro záznam skutečné hodnoty rozsahu pracnosti.

Položka odhadu						
Název položky:	odkazy					
Kategorie	<input type="radio"/> Analýza <input checked="" type="radio"/> Programování <input type="radio"/> Testování <input type="radio"/> UAT <input type="radio"/> Projektové řízení					
Požadavek:	RQ4; RQ1					
Modul:	web					
Prvek položky:	formulář					
Odhad položky:	Nový odhad					
Minimální O_{min}	Maximální O_{max}	Očekávaný O_{exp}	Průměrný O_{avg}	Původní (expertní)	Skutečný O_{real}	MRE
0,25	3	2	3	2	2	

Obrázek B.6.: Položka odhadu pracnosti prvku Zdroj: vlastní

- Přehled položek - přehled položek za zvolenou variantu a vybraný projekt.

Položky projektu		Vložit položku projektu...		Základní						
Prvek	O(hist)	O(min)	O(exp)	O(max)	Real	ORP(o)	ORP(a)	ORP(b)	Interv	
Celkem	34	9	32	42	33	29,8	31,2	31,5	100%	
Analýza	4	2	4	5	4	3,8	4,0	4,0	100%	
Analýza:	4	2	4	5	4	3,8	4,0	4,0		
Programování	16	3	15	21	17,5	14,0	14,7	15,0	100%	
Programování: core RQ15,RQ18	4	0,25	3	4	3	2,7	2,5	2,9		
Programování: interface RQ1	6	2	6	8	8	5,7	6,0	6,0		
Programování: web RQ3	3	0,25	3	4	2,5	2,7	2,7	2,9		
Programování: web RQ4,RQ1	2	0,25	2	3	2	1,9	2,0	2,0		
Programování: web RQ9,RQ10,RC	1	0,25	1	2	2	1,0	1,4	1,2		
Projektové řízení	4	1	3	4	3	2,8	3,0	3,0	100%	
Projektové řízení:	4	1	3	4	3	2,8	3,0	3,0		
Testování	5	2	5	6	4,5	4,7	4,8	4,8	100%	
Testování:	5	2	5	6	4,5	4,7	4,8	4,8		
UAT	5	1	5	6	4	4,5	4,7	4,7	100%	
UAT:	5	1	5	6	4	4,5	4,7	4,7		

Obrázek B.7.: Přehled prvků jednotlivých požadavků Zdroj: vlastní