

**SEMESTRÁLNÍ PRÁCE KE KURZU
4IT421 ZLEPŠOVÁNÍ PROCESŮ BUDOVÁNÍ IS**

Zimní semestr 2014/2015

Odhady při vývoji SW

Autoři : Bc. Lucie Čecháková, xcecl10
Bc. Jiří Snítil, xsnij00

11. 12. 2014

Titulní list Semestrální práce ke kurzu 4IT421 Zlepšování procesů budování IS	
Semestr	Zimní semestr 2014/2015
Autoři	Bc. Lucie Čecháková, xcecl10 Bc. Jiří Snítal, xsnij00
Téma	Odhady při vývoji SW
Datum odevzdání	25. 10. 2014

Klíčová slova

Odhadování pracnosti, metody odhadování, budoucnost odhadování.

Obsah

1	Úvod...	4
1.1	Definice některých pojmů	4
2	Důležitost odhadování	5
3	Metody odhadování	6
3.1	Metody odhadu pracnosti	6
3.1.1	Expertní odhady	6
3.1.2	Analogie	7
3.1.3	Metoda funkčních bodů – Function Point Analysis (FPA).....	8
3.1.4	Odhady založené na objektech – Use Case points (UCP)	9
3.2	Podpůrné metody odhadování	9
3.2.1	Diagram rozkladu prací – Work Breakdown structure (WBS)	9
3.2.2	Metoda kritické cesty – Critical Path Method (CPM).....	10
4	Budoucnost odhadování	12
4.1	Vlastnosti odhadování, ve kterých lze spatřit budoucnost rozvoje	12
4.1.1	Neexistence „nejlepšího“ modelu nebo metody pro odhadování pracnosti	12
4.1.2	Zákaznické zaměření na nízkou cenu je hlavním důvodem pro překročení odhadované pracnosti	13
4.2	Vlastnosti odhadování, které nevedou k rozvoji odhadování	13
4.2.1	Přesné odhadování pracnosti ve velmi velkých a komplikovaných projektech	14
4.2.2	Měření velikosti softwaru a jeho složitosti pro přesný odhad pracnosti	14
5	Závěr.....	15
6	Zdroje.....	16

1 Úvod

Semestrální práce se zaměřuje na metody odhadování softwarových projektů. Nejprve nastíníme důležité pojmy pojící se s odhadováním a dále, v kapitole „Důležitost odhadování“ podpoříme několika argumenty, proč je nutno se touto oblastí zabývat.

Dále v kapitole „Metody odhadování“ nastíníme princip těch nejdůležitějších metod a v poslední kapitole „Budoucnost odhadování“ se věnujeme článku od Magne Jørgensena, kde jednotlivé vlastnosti odhadování podporujeme nebo vyvracíme dalšími zdroji.

Hlavním cílem naší semestrální práce je shrnout základní poznatky z oblasti odhadování při vývoji softwaru a objasnit základní principy několika metod odhadování. Budeme čerpat z mnoha zdrojů, které se tématy jednotlivých metod zabývají.

1.1 Definice některých pojmů

Na začátku je třeba si ujasnit některé pojmy, které souvisejí s oblastí odhadování. Nejčastěji jsou skloňována tato slova:

Odhad = Odhad je vypočítané nebo jen ze zkušenosti předpovězené více či méně přibližné určení výsledku nebo informace, které je využitelné, i když jsou vstupní data nekompletní, nejistá, nebo zašumělá. (Wikipedia, 2013)

Nadhodnocení odhadu = nesprávný odhad, který převyšuje reálnou, původně odhadovanou hodnotu. (Zdroj: autorka)

Podhodnocení odhadu = nesprávný odhad, který je nižší než reálná, původně odhadovaná hodnota. (Zdroj: autorka)

Pracnost = Množství vynaložené práce na zpracování daného projektu, které se obvykle uvádí v člověkodnech. (Čecháková, 2014)

2 Důležitost odhadování

Odhadování jednotlivých aktivit u softwarových projektů, tedy správné stanovení rozpočtu, potřebných zdrojů a délky trvání prací, je velice důležité pro úspěšný průběh každého projektu. Na první pohled se zdá nemožné odhady poskytnout, zejména při přípravě či v raných fázích projektu. Je však velmi podstatné se tomuto tématu věnovat, neboť jen po porovnání plánovaného přínosu projektu a odhadů jeho nákladů lze stanovit výhodnost celého projektu a určit, zda má smysl jej v zamýšleném rozsahu realizovat¹. (McConnell, 2006)

Z odhadů pracnosti celého projektu musí projektoví manažeři odvodit jeho cenu a čas, za který jej mohou realizovat. Nesprávnost těchto odhadů v raných fázích projektu může způsobit jistý neúspěch projektu, který ještě ani nezačal. Jak v článku *What We Do and Don't Know about Software Development Effort Estimation*² píše Magne Jørgensen:

Drtivé množství důkazů dokumentují tendenci k překračování nákladů a pracnosti u softwarových projektů. V průměru je toto překročení okolo 30%. (Jørgensen, 2014)

Z tohoto důvodu se neustále mění a přizpůsobují metody odhadování na prostředí konkrétní firmy, aby byly výsledky odhadů reálné a k žádnému takovému překročení nákladů a pracnosti na projektu nedocházelo. V následující kapitole se blíže věnujeme několika metodám na odhadování pracnosti softwarových projektů.

¹ Známo pod anglickým pojmem business case.

² Překlad autorky: Co víme a co nevíme o odhadech pracnosti vývoje softwaru

3 Metody odhadování

V této kapitole vysvětlíme princip těch nejdůležitějších metod na vytváření odhadů. Rozdělujeme je na samotné metody pro vytváření odhadů (v podkapitole *Metody odhadu pracnosti*) a poté na další metody a nástroje, které ulehčí práci s předešlými metodami (v podkapitole *Podpůrné metody odhadování*).

3.1 Metody odhadu pracnosti

V následujících podkapitolách se zabýváme několika metodami pro stanovení složitosti softwaru a s tím souvisejícím odhadováním pracnosti.

3.1.1 Expertní odhady

Tato technika je založená na odhadu provedeném expertem, který poskytuje odhad na základě svých předchozích zkušeností. Provádí se nejčastěji za situace, kdy není dostatek informací o projektu, aby se mohly použít některé ze sofistikovanějších metod odhadování. (Čecháková, 2014)

Výhoda této metody spočívá v tom, že ji lze použít v kterékoli fázi projektu, a to i při úvodních etapách, kdy je k dispozici minimum informací o projektu. Na druhou stranu je založena na zkušenostech jedince, který nutně nemusí mít znalosti specifických podmínek daného projektu. Podle Vahalíka:

„Nevýhodou je, že tato metoda je jen tak dobrá, jak dobrý je expert. Čím více expertů provádí odhad, tím jsou výsledky odhadu objektivnější.“ (Vahalík, 2004, str. 6)

Při použití této metody je nutné do odhadů nepromítat jakákoli známá omezení projektu (např. očekávání zákazníka), neboť tyto informace mohou ovlivnit expertův odhad. Výsledkem expertního odhadování je interval, který určuje rozsah od odhadované minimální pracnosti do odhadované maximální pracnosti.

Metoda expertního odhadu **Wideband Delphi** je založena na následujících pravidlech (Lacko, 2010):

1. Počet expertů, kteří odhadují pomocí této metody, by neměl být nižší než 5 a zároveň vyšší než 15. Optimální počet je 10.
2. Experti pracují nezávisle, všechny jejich odpovědi jsou zaznamenávány anonymně a jsou následně hromadně zpracovány.

3. Získané výstupy jsou poté jednotlivým expertům předkládány s cílem získat další zpřesněné odhady a pomocí této zpětné vazby experti své odhady v několika iteracích upravují.
4. Výsledky jsou zpracovávány pomocí statistických metod, což umožní jejich objektivizaci.
5. Otázky k odhadům jsou pokládány tak, že odpovědi lze vyjádřit pouze číselnými hodnotami.
6. Pro zodpovězení otázek jsou poskytnuty expertům dostatečné podklady.
7. Expert je schopen dodat zdůvodnění ke svým odpovědím.

Metoda Wideband Delphi je označována za jednu z nejlepších a nejpropracovanějších metod k získávání expertních odhadů. Její výhoda spočívá v tom, že poskytuje relativně spolehlivé výsledky. Za nevýhodu lze považovat vysoké náklady na použití této metody – získání potřebného počtu expertů a náročnost zpracování. (Čecháková, 2014)

3.1.2 Analogie

Tato metoda je založena na hledání a odhalování rozdílů a podobností zkoumaného projektu s obdobnými – referenčními – projekty, ke kterým je k dispozici potřebné množství dat, např. pracnost celého projektu, pracnost testování. K nalezeným rozdílům se dále přiřazují odhady vlivu těchto rozdílů na pracnost zkoumaného projektu. Je nesmírnou výhodou, pokud existují data již realizovaných projektů z firem, kde se odhady provádí. (Vahalík IN Čecháková, 2014)

Postup této metody je podle (McConnell, 2006) následující:

1. Získání detailních informací o referenčním projektu, se kterým se srovnáváme. Je třeba znát jeho velikost, práci, cenu, přičemž je nezbytné zvolit vhodnou dekompozici (např. podle struktury kategorií práce).
2. Srovnání velikosti referenčního a odhadovaného projektu podle jednotlivých částí dle dekompozice.
3. Stanovení odhadu velikosti projektu jako procentuální vyjádření poměru k velikosti referenčního projektu.
4. Výpočet odhadu práce, založený na stanoveném poměru (viz bod 3).
5. Kontrola souladu předpokladů obou projektů.

Metodu analogie lze aplikovat pro odhady pracnosti testování obdobně jako na celé projekty. Pro stanovení objemu práce na testování využijeme stejným způsobem procentuální poměr velikosti referenčního a odhadovaného projektu.

3.1.3 Metoda funkčních bodů – Function Point Analysis (FPA)

Metoda funkčních bodů je jednou z nejstarších metod založených na modelu. Funkční bod je zde jakási měrná jednotka, která slouží k určení velikosti funkčnosti. Dala by se připodobnit k litru nebo metru, ale na rozdíl od fyzikálních jednotek není tak exaktní.

Metodou lze měřit velikost softwarového projektu a potažmo jeho cenu (náklady na realizaci) už v počáteční fázi projektu – nahlíží se totiž na projekt z pohledu uživatelů, kteří si stanovují požadavky na funkcionalitu systému.

Postup ohodnocení složitosti systému pomocí funkčních bodů není zcela jednoduchý. Skládá se z několika na sebe navazujících výpočtů, z nichž každý má svá specifika.

Základní kroky nutné pro výpočet počtu funkčních bodů a jejich výstupy jsou stanoveny následujícím způsobem (IFPUG, 2000):

- 1) identifikace hranic systému,
 - výstup: jasně stanovené hranice
- 2) identifikace funkcí systému,
 - výstup: jednotlivé funkce zařazeny do jedné z pěti komponent systému
- 3) stanovení složitosti funkcí systému,
 - výstup: zařazení do kategorie nízký, průměrný, vysoký
- 4) stanovení hodnot pro datové funkce,
 - výstup: počet nekorigovaných funkčních bodů
- 5) zhodnocení charakteristik vyvíjeného softwaru,
 - výstup: body přidělené jednotlivým charakteristikám systému
- 6) stanovení hodnoty korekčního faktoru,
 - výstup: korekční faktor
- 7) výpočet korigovaných funkčních bodů.
 - výstup: celkový počet korigovaných funkčních bodů

Popíši zde hlavně druhý krok, *Identifikace funkcí systému*. Tato metoda měří velikost softwaru na základě vyhodnocení dobře definovaných funkčních vlastností systému. To představuje identifikovat pět hlavních komponent systému, které se dále dělí do dvou skupin – transakční a datové funkce. Mezi transakční funkce patří **vstupy do aplikace** (EI – External Inputs), **výstupy z aplikace** (EO – External Outputs) a **uživatelské dotazy** (EQ – External Inquiry). Pod datové funkce spadají **vnitřní logické soubory** (ILF – Internal Logical Files) a **vnější soubory rozhraní** (EIF – External Interface Files). Rozdělení je uvedeno v přehledné tabulce 1.

Tabulka 1: Rozdělení funkcí na transakční a datové (zdroj autorka)

Transakční funkce	Vstupy do aplikace	EI
	Výstupy z aplikace	EO
	Uživatelské dotazy	EQ
Datové funkce	Vnitřní logické soubory	ILF
	Vnější soubory rozhraní	EIF

3.1.4 Odhady založené na objektech – Use Case points (UCP)

Metoda odhadů založená na objektech, která byla vyvinutá Gustavem Karnerem, vychází z předchozí metody funkčních bodů a zároveň z již vytvořených případů užití (Use Case). Analogií k měrné jednotce funkčního bodu je zde „Use case point“. Případy užití dále ohodnocuje různými vahami podle jejich složitosti. Dále ohodnotí aktéry v systému, kteří tyto případy užití vykonávají. (Struska, 2006)

Aktéři se ohodnocují podle následující tabulky 2.

Tabulka 2: Ohodnocení aktérů v systému a případů užití (zdroj Dědek, 2014)

Typ aktéra	Příklad	Váha
Jednoduchý	Další systém komunikující přes API.	1
Průměrný	Další systém komunikující přes nějaký protokol (např. TCP/IP) nebo člověk komunikující přes textové rozhraní (například konzole).	2
Komplexní	Člověk komunikující přes GUI nebo webové rozhraní.	3

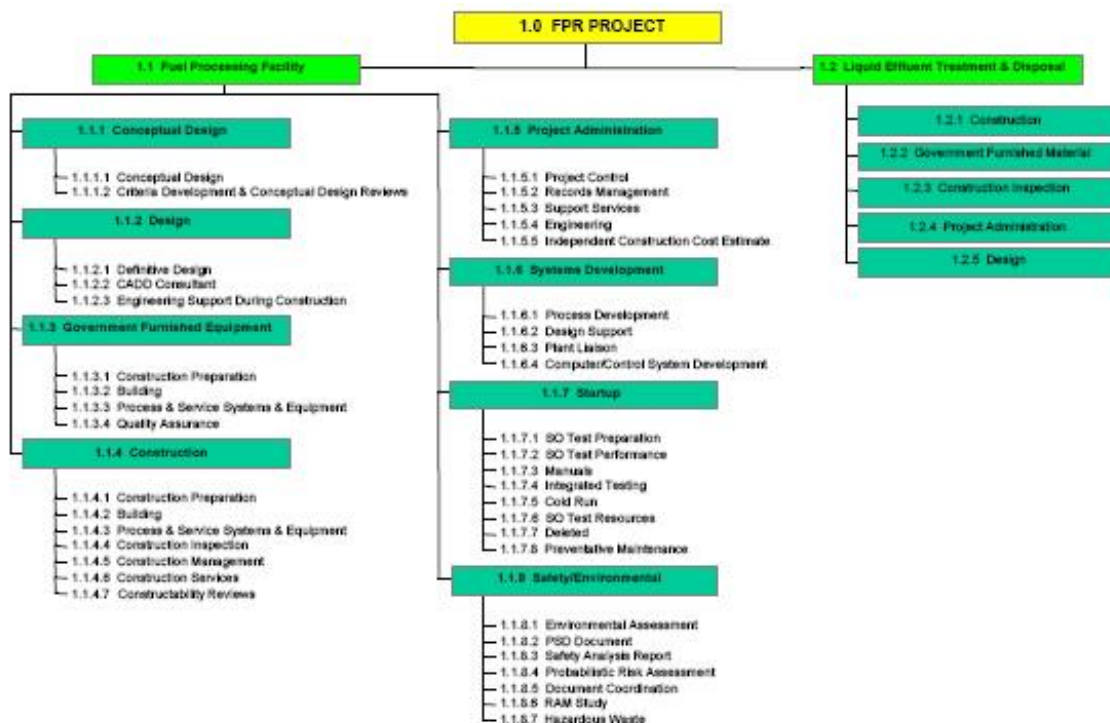
Následně se dále provádí výpočty podle složitosti případu užití. Odhad náročnosti projektu dále závisí na technické náročnosti vypracování. Proto se k celkovému odhadu počtu Use Case Points započítává ještě další faktor – Technical Complexity Factor (TCF), který v sobě vyjádří komplexitu zkoumaného systému.

3.2 Podpůrné metody odhadování

3.2.1 Diagram rozkladu prací – Work Breakdown structure (WBS)

Tato metoda je založena na hierarchickém uspořádání činností projektu s přiřazením jejich základních charakteristik. Na celý projekt pak není nahlíženo jako na celek, ale díky proce-

su dekompozice, jako na jednotlivé dílčí etapy, fáze a elementární činnosti. Tyto části získáme z hierarchického diagramu – viz obrázek č. 1. Výhodou této metody je, že menší celky činností a jejich detailní rozklad se odhadnou mnohem lépe, než celý projekt (popřípadě jeho velké části). K jednotlivým částem se dále přiřazují charakteristiky a vlastnosti.

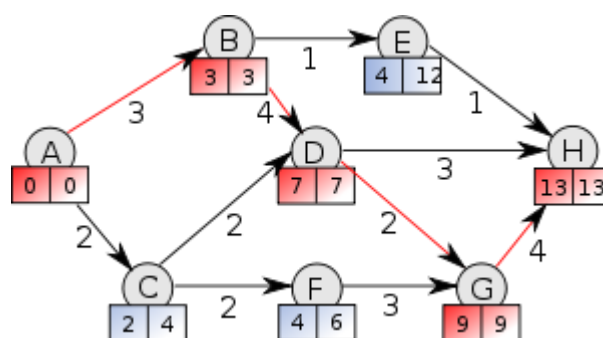


Obrázek 1: Diagram rozkladu prací (Zdroj: http://www.projectpro.co.za/e-Zine/Tips_for_WBS/body_tips_for_wbs.html)

Jednotlivé diagramy rozkladu prací se od sebe liší úrovní granularity, tj. do jakého detailu se při rozkladu činností zajde. Čím větší je granularita, tím jsou jednotlivé části větší a celkově je jich méně.

3.2.2 Metoda kritické cesty – Critical Path Method (CPM)

Metoda kritické cesty patří mezi deterministické metody síťové analýzy. Klade důraz na stanovení nákladů projektu a doby trvání jednotlivých činností na základě délky *kritické cesty*. Kritická cesta je definována jako (časově) nejdelší možná cesta z počátečního bodu grafu do koncového bodu grafu. Každý projekt má minimálně jednu kritickou cestu.



Obrázek 2: Kritická cesta (Wikipedia 2)

Jedním z hlavních důvodů vývoje metody byla reakce na slabiny Ganttových diagramů, které mají malou flexibilitu a účinnost v oblasti řízení nákladů. Mezi výhody, které využití metody kritické cesty poskytuje, lze zařadit: grafické zobrazení harmonogramu projektu, odhad času potřebný pro realizaci projektu, identifikace činností, které jsou umístěny na kritické cestě a nemají časové rezervy. (Král, 2012)

Na obrázku č. 2 je kritická cesta podbarvena červeně. Na odhady těchto činností je potřeba se více zaměřit, neboť určují délku celého projektu.

4 Budoucnost odhadování

Z porovnání přesnosti odhadování z roku 1980 s nedávnými hodnotami odhadů vyplývá, že se od té doby přesnost odhadů moc nezměnila. Stejně tak se ani rapidně nezměnily používané metody. I přes rozsáhlý výzkum o formálních modelech pro odhadování jsou stále nejčastější metodou pro odhadování expertní odhady. (Jørgensen, 2014)

Zřejmý nedostatek ve zlepšení přesnosti odhadů však neznamena to, že by se nevědělo o odhadech nyní více informací. V následujících kapitolách jsou tyto poznatky shrnuty a rozděleny do dvou částí – na ty, které mají potenciál zlepšit přesnost odhadu a na ty, které s největší pravděpodobností nepovedou ke zlepšení odhadů. Takto poznatky rozdělil Magne Jørgensen, z jehož článku následující podkapitoly vycházejí.

4.1 Vlastnosti odhadování, ve kterých lze spatřit budoucnost rozvoje

Podle výzkumů na odhadování pracnosti Magne Jørgensen rozpoznal sedm aspektů, o kterých se již ví hodně a které mají potenciál zlepšit odhady v budoucnosti. V následujících oddílech se zaměřuji na dva nejvýznamnější aspekty.

4.1.1 Neexistence „nejlepšího“ modelu nebo metody pro odhadování pracnosti

Přestože existuje mnoho studií pro porovnávání přesnosti jednotlivých metod pro odhadování, kde vždy některá z nich „vyhraje“, nejsou tyto výsledky podle Jørgensena vůbec relevantní. Tím hlavním důvodem pro vyjádření nedůvěry k těmto průzkumům je fakt, že se závislost mezi pracností vývoje a velikostí projektu, případ od případu, velmi liší. Kromě toho aspekty, které mají největší vliv na pracnost vývoje, a které tu pracnost mohou nejvíce ovlivnit, se také obrovsky různí. Z toho vyplývá, že jakákoli metoda nebo model pro odhadování pracnosti by měla být navržen tak, aby byla co nejvhodnější pro kontext projektu, při kterém se odhady provádějí.

„Nedostatek ustálenosti při odhadování základních celků projektu také vysvětluje fakt, že statisticky pokročilé metody obvykle nezvyšují přesnost odhadování, pokud vůbec, ve srovnání s jednoduššími modely.“ (Jørgensen, 2014)

U těchto statisticky pokročilejších metod je naopak pravděpodobnější, že budou vykazovat horší výsledky, jelikož v parametrech obsahují hodnoty, které jsou založeny na minulosti. Oproti tomu jednodušší modely, které se díky své jednoduchosti dokáží více přizpůsobit aktuálním podmínkám, na projektu zajišťují přesnější odhady. Z těchto výsledků plyne

jednoznačné doporučení pro softwarové firmy. A to takové, že by se měly snažit si vybudovat svůj vlastní model pro odhadování, který přesně zapasuje do jejich konkrétních podmínek. V žádném případě by neměly očekávat od obecných modelů odhadování, že budou přesné v jejich specifickém kontextu.

4.1.2 Zákaznické zaměření na nízkou cenu je hlavním důvodem pro překročení odhadované pracnosti

Tendence k podhodnocení odhadů je výsledkem dnešního trendu, tj. konkurování ostatním dodavatelům nízkou cenou. Dle ceny totiž většina dodavatelů rozhoduje na prvním místě a je také nejvíce porovnávána při nabídkových soutěžích. Trend je pochopitelný – kdyby se měl zákazník věnovat ostatním parametrům, jako jsou vlastní výstupy projektu, složitá porovnatelnost by jej stála mnoho dodatečných výdajů. A tak dodavatelé stále snižují cenu, aby se právě ten jejich projekt mohl uskutečnit.

Podhodnocování odhadů je zakotveno také v tzv. kognitivních chybách lidského úsudku. Daniel Kahneman ve své knize *Myšlení – rychlé a pomalé* popisuje tento jev jako „klam plánování“. Jedná se právě o toto podhodnocování za účelem získání zakázky, kde také hraje roli optimismus jedince, kteří odhady prováděli. (Kahneman, 2012, str. 269)

Tyto podhodnocené odhady obsahují však jedno velké úskalí – v momentě, kdy se začne přesahovat rozpočet, klient nakonec bude muset zaplatit vyšší částku, popřípadě skončí projekt neúspěchem, kde prvně vynaložené finance nebudou nijak zhodnoceny. Více nebezpečí plynoucích z podhodnocených odhadů popisuje bakalářská práce (Čecháková, 2014, str. 13).

Naproti tomu rozvoj softwaru interně ve firmě nebo kdekoli, kde cena nehraje takovou roli, jsou odhady mnohem realističtější a bez viditelného trendu na jejich podhodnocování. Tam cílová skupina věnuje také mnohem více pozornost na výsledný produkt a jeho vlastnosti než na samotnou cenu.

4.2 Vlastnosti odhadování, které nevedou k rozvoji odhadování

Dále Magne Jørgensen stanovil tři oblasti, o kterých na rozdíl od předchozí podkapitoly zatím nic nevíme. Popisuje je jako výzvy, které i přes velký objem výzkumů nemají zatím dobré řešení, poznání lidstva v těchto oblastech není vůbec uspokojivé. Vybrala jsem dvě z nich, abych je zde podrobněji popsala.

4.2.1 Přesné odhadování pracnosti ve velmi velkých a komplikovaných projektech

Velmi velké projekty zavádějí nutnost odhadovat ještě další aspekty, které odrazí právě jejich velikost a komplexnost. Nejen že mají další a i komplikovanější parametry ke zvážení, ale odhadování nám ztěžuje i neexistence relevantních historických dat, které by mohly odhady nasměrovat. Opravdu velké projekty navíc obsahují mnoho vazeb mezi zainteresovanými stranami, komplexní interakce mezi nimi i mezi softwarovými aplikacemi a dokonce i změny byznys procesů. Tyto další parametry jsou opravdu jen velmi těžko odhadnutelné.

4.2.2 Měření velikosti softwaru a jeho složitosti pro přesný odhad pracnosti

Magne Jørgensen se domnívá, že mezi odhadováním velikosti softwaru a odhadem na jeho vytvoření je veliký rozdíl. Ve svém článku píše:

„Léta praxe v odhadování velikosti softwaru a jeho složitosti však nezajistila úspěšnost při odhadování pracnosti na vývoj tohoto softwaru. Některé projekty sice vykazují opak, ale jsou jen velmi vzácné.“ (Jørgensen, 2014)

5 Závěr

Naše semestrální práce se zaměřovala na metody odhadování softwarových projektů, blíže jsme se věnovali čtyřem metodám odhadu pracnosti a dvěma podpůrným metodám. V kapitole „Budoucnost odhadování“ jsme se věnovali článku od Magne Jørgensena, kde jsme jednotlivé vlastnosti odhadování zhodnotili z hlediska budoucího rozvoje.

Hlavním cílem naší semestrální práce bylo shrnout základní poznatky z oblasti odhadování při vývoji softwaru a objasnit základní principy několika metod odhadování. V této práci byl náš cíl naplněn.

6 Zdroje

Tabulka zdrojů
BUCHALCEVOVÁ, Alena., STANOVSKÁ, Iva. <i>Příklady modelů analýzy a návrhu aplikace v UML</i> . Vysoká škola ekonomická v Praze, 2013. ISBN 978-80-245-1922-7.
ČECHÁKOVÁ, Lucie. <i>Metody odhadování pracnosti testování software</i> . Praha, 2014. Bakalářská práce. Vysoká škola ekonomická v Praze – Fakulta informatiky a statistiky.
DĚDEK, Vladimír. <i>Odhady práce při různých fázích projektu</i> [online]. 2010. Dostupné z: http://www.mitvsehotovo.cz/2010/03/odhady-prace-pri-ruznych-fazich-projektu/
Function Point Counting Practices Manual, release 4.1.1, International Function Point Users 2000, USA, ISBN 0-963-1742-7-4
JØRGENSEN, Magne. <i>What We Do and Don't Know about Software Development Effort Estimation</i> . IEEE Software magazine. 29. 8. 2014.
KAHNEMAN, Daniel. <i>Myšlení – rychlé a pomalé</i> . Jan Melvil Publishig, Brno 2012. ISBN 978-80-87270-42-4.
KRÁL, Miroslav. <i>Odhadování pracnosti IT projektů</i> . Katedra systémové analýzy, Fakulta informatiky a statistiky, Vysoká škola ekonomická v Praze. ISSN 1805-4951
LACKO, Bronislav. <i>Expertní odhady v projektech tvorby software</i> . In Sborník konference Tvorba softwaru 2010. Ostrava, 2010.
MCCONNELL, Steve. <i>Odhadování softwarových projektů – Jak správně určit rozpočet, termíny a zdroje</i> . Brno, 2006. ISBN: 80-251-1240-3
STRUSKA, Zdeněk., PERGL, Robert. <i>Metody odhadu prac-nosti založené na modelu</i> . In Tvorba softwaru. Ostrava: ČSSI, 2006. ISBN 80-248-1082-4. Dostupné z: formular-ekf.vsb.cz/formulare/F01/tsw/getfile.php?prispovekid=876
VAHALÍK, Tomáš. <i>Odhadujete pracnost projektu?</i> Komix, 2004. Dostupné z: http://plustek.cz/upload/noviny2004c.pdf
Wikipedia. <i>Odhad</i> [online]. 10. 3. 2013, [cit. 27. října 2014]. Dostupné z: http://cs.wikipedia.org/wiki/Odhad .
Wikipedia. <i>Kritická cesta</i> [online]. 16. 3. 2013, [cit. 1. prosince 2014]. Dostupné z: http://cs.wikipedia.org/wiki/Kritick%C3%A1_cesta