

ODHADY NÁKLADŮ VÝVOJE WORKFLOW SYSTÉMU

Jaroslav Ráček^a

Jan Ministr^b

a) Fakulta informatiky MU v Brně, racek@fi.muni.cz

b) Ekonomická fakulta VŠB-TU Ostrava, jan.ministr@vsb.cz

Abstrakt

Príspevek popisuje spôsoby odhadu nákladnosti vytváření workflow systému. K tomu lze využít měření velikosti zdrojového kódu (COCOMO – Constructive Cost Model), funkčně orientované metriky vycházejících z principů analýzy funkčních bodů (FPA – Function Point Analysis) nebo metriky založené na postupném ohodnocování jednotlivých činností uvnitř procesu (ABC – Activity Based Costing). V případě workflow systémů založených na zasílání zpráv nebo sdílení dokumentů lze vyjít z velikosti kódu nebo z funkčních bodů. V případě procesně orientovaných workflow systémů je navíc třeba odhadnout náklady vytvoření definicí procesů, k čemuž lze použít principy ABC.

Abstract

Papers deals with cost estimation of workflow system development. A measurement of source code lines (COCOMO – Constructive Cost Model), function oriented metrics (FPA – Function Point Analysis) or metrics based on measurement of individual process activities (ABC – Activity Based Costing) can be used for this purpose. If the workflow system is mail based or document based then should be used COCOMO or Function Point Analysis. In the case of process based system the cost of process analysis and cost of process definition development can be estimated by using principles of Activity Based Costing.

Klíčová slova

Odhady nákladů softwaru, Workflow systém, COCOMO, Funkční body, Activity Based Costing.

Key words

Software Cost Estimation, Workflow system, COCOMO, Function Point Analysis, Activity Based Costing.

1. ÚVOD

Príspevek se snaží pohlízet na zavádění nového workflow systému očima budoucího uživatele, tedy firmy, která je buď v pozici vývojáře, který pro sebe vyvíjí nový systém, nebo je v pozici zákazníka, který kupuje a implementuje nabízené softwarové řešení.

Zavádění nového informačního systému je složitý proces, který s sebou nese nemalá finanční rizika a vyžaduje důkladné plánování a velkou míru zodpovědnosti při rozhodování. Ať se již jedná o vývoj zcela nového softwaru nebo implementaci některého z komerčně nabízených produktů, je třeba již na startu projektu stanovit (pokud možno s co největší přesností) náklady, které každý takový projekt s sebou přináší. Při odhadování finančních i časových nákladů projektu, na jejichž základě se pak vytváří rozpočet a plán prací, je zapotřebí vycházet mimo jiné i z údajů o softwarovém produktu, který je vytvářen. Jinými slovy, je třeba změřit velikost výsledného produktu, a to často v okamžiku, kdy ještě jeho vývoj není zdaleka dokončen, což má samozřejmě vliv na přesnost odhadu.

2. TYPY WORKFLOW SYSTÉMŮ

V případě informačních systémů, které v sobě mají zakomponovanu podporu firemního workflow, může dojít k několika typovým situacím, podle nichž se následně odvíjí způsob stanovení (odhadu) velikosti zaváděného systému.

Jedním z faktorů ovlivňujících způsob odhadu je typ systému. Zde je třeba rozlišovat zejména mezi systémy pracujícími na principech sdílení dokumentů a dat (tzv. *Document Based Workflow*) a systémy založenými na procesech (tzv. *Process Based Workflow*). Dokumentově orientované systémy pracují na principu *document flow*, mají schopnost komunikace s externími aplikacemi, jsou vhodné především pro administrativní workflow (podpůrné a zároveň opakovatelné procesy) a zpravidla bývají navrhovány jako třívrstvá architektura obsahující prezentační, aplikační a datovou vrstvu. Data těchto systémů mohou být sdílena s dalšími informačními systémy firmy. Z pohledu návrhu a implementace se toto pojetí WfMS (Workflow Management System) nijak významně neliší od většiny informačních systémů, které firmy používají.

Oproti tomu procesně orientované systémy obsahují definice firemních procesů, podle nichž se zřizují a řídí jednotlivé instance procesů. Procesně orientovaný WfMS si lze představit jako „interpret procesů“, který má širokou škálu rozhraní a iterací s ostatními aplikacemi, ovšem využívá vlastní komunikační kanály a databáze, které nejsou ostatním systémům ve firmě přímo přístupné. Procesně orientované systémy jsou vhodné zejména pro produkční workflow (rozhodující a zároveň opakovatelné procesy). Architektura takovýchto systémů často vychází ze standardů WfMC (Workflow Management Coalition), kde základ tvoří obslužný WfMS, který pracuje s importovanými definicemi procesů navrženými pro potřeby konkrétní firmy [Hol95, Hol99].

3. METODY VÝPOČTŮ NÁKLADŮ

Dalším faktorem, který ovlivňuje způsob odhadu velikosti systému, je dostupnost informací o velikosti zdrojového kódu a dalších zdrojových dat, jako jsou konfigurační soubory či definice procesů. Pokud jsou informace o velikosti zdrojových souborů dostupné, což v praxi znamená, že řešitelé projektu jsou schopni tuto velikost odhadnout, lze vyjít z úvahy, že pracnost, cena a čas vytvoření nového systému jsou úměrné velikosti zdrojových souborů. Pak lze pro výpočet těchto veličin použít například metodu COCOMO (Constructive Cost Model), kterou publikoval poprvé v roce 1981 B. Boehm a v průběhu 80. a 90. let ji několikrát modifikoval, takže tato metodika odhadu pracnosti je použitelná i pro současné techniky vývoje softwaru [Boe81]. Výpočet pomocí COCOMO nebo jiného podobného modelu vycházejícího z velikosti zdrojových souborů, je vhodný pro nově vyvíjené workflow systémy postavené na třívrstvé architektuře.

Případem, kdy je model COCOMO nepoužitelný, je situace, kdy vývojáři nejsou schopni dostatečně přesně odhadnout velikost zdrojových souborů. To se stává zejména v případech, kdy tým nemá dostatek zkušeností z předešlých podobných projektů. Zvláště v počátečních fázích řešení bývá odhad velikosti zdrojových souborů značně obtížný a nepřesný. Tehdy se nabízí způsob odhadnutí velikosti softwaru pomocí požadované funkcionality, která by měla být známa již v úvodních fázích projektu (nejpozději na konci analýzy) tedy ještě před tím, než je zahájena vlastní tvorba systému. Pravděpodobně nejrozšířenější technikou tohoto typu odhadu je odhad pomocí funkčních bodů (Function Point Analysis), což je metrika popisující funkcionalitu softwaru z pohledu koncového uživatele, která zcela zakrývá pohled ze strany tvůrce systému. Princip techniky spočívá v přepočtu (ohodnocení) jednotlivých uživatelských funkcí, jež bude systém nabízet, na příslušné množství „výrobních jednotek“ (funkčních bodů) a následně stanovení pracnosti vytvoření jedné takové jednotky [Jon91]. Podobně jako COCOMO i byly funkční body

prvotně určeny pro ohodnocení nově vyvíjených systémů postavených na třívrstvé architektuře. Princip funkčních bodů však lze modifikovat i na ohodnocení workflow procesů.

Při odhadování pracnosti a nákladů zavádění procesně orientovaných systémů se zpravidla zvláště odhaduje zavedení WfMS a zvláště vytvoření a zavedení definic procesů. V případě zavádění WfMS se často jedná o implementaci již hotového komerčně nabízeného produktu. Pak je cena a doba trvání implementace dána podmínkami implementační smlouvy a techniky odhadování pracnosti nejsou na straně zákazníka nezbytné, neboť toto je úkolem dodavatele. Popis technik, které dodavatelé používají pro odhadování ceny licencí a implementace komerčního softwaru, přesahuje rozsah tohoto článku. Pokud se však firma z jakýchkoli důvodů rozhodne vyvinout vlastní systém řízení workflow, což zpravidla bývá poměrně nákladná záležitost, lze pracnost tohoto vývoje odhadnout pomocí COCOMO nebo funkčních bodů. Případně se může firma nejdříve pokusit odhadnout pracnost vývoje vlastního systému, porovnat jej s cenou komerčně nabízeného produktu a na základě toho se rozhodnout, zda daný systém zakoupí nebo vyvine sama.

Samostatnou činností v rámci plánování projektu bývá odhad pracnosti vytvoření a implementace definic procesů do již zavedeného WfMS. Zde rovněž existují dvě základní varianty postupu. První varianta je obdobou odhadu na základě velikosti zdrojového kódu, nicméně v tomto případě se zdrojovým kódem myslí velikost popisu procesů, respektive velikost zdrojových souborů definic procesů v příslušném datovém formátu, s nimiž bude WfMS pracovat. Aby však bylo možné takovýto způsob odhadu použít, musí být k dispozici údaje o množství a velikosti firemních procesů. Ty jsou ale zpravidla k dispozici až v okamžiku, kdy má firma zmapovány své procesy, což nemusí být na začátku projektu k dispozici. Pak lze pracnost zavedení jednotlivých procesů ohodnotit pomocí principů metody ABC (Activity Based Costing), tedy stanovit pracnost zavedení jednotlivých činností, z nichž se procesy skládají, a odtud odvodit pracnost zavedení celého systému.

V situaci, kdy firma nemá zmapovány své procesy, lze použít pro zavedení procesů odhady založené na principu funkčních bodů (tzv. *funkční body procesů*), přičemž funkční body mohou být použity nejen pro odhad pracnosti implementace definic procesů, ale i pro odhad pracnosti zmapování firemních procesů.

4. ODHAD METODOU COCOMO

Odhad pracnosti a nákladnosti vývoje softwaru metodou COCOMO vychází z idey, že úsilí E , tj. počet vynaložených osoboměsíků, a doba T , tj. čas trvání projektu, jsou úměrné velikosti zdrojového kódu. Odtud je dále možné stanovit i finanční náklady projektu.

Slabým místem tohoto přístupu je zejména přesnost počátečního odhadu velikosti produktu, která se uvádí v *KSLOC*, což jsou tisíce řádků zdrojového kódu. Odhad velikosti se může od skutečnosti značně lišit (Boehm uvádí až čtyřikrát), a to oběma směry. Bez předešlých zkušeností nebo jiného zdroje empirických dat, není zpravidla reálné dostatečně přesný odhad budoucí velikosti kódu učinit.

Dále je před vlastním výpočtem třeba nastavit další parametry modelu, jimiž jsou *úroveň detailu* výpočetního modelu a použitý *vývojový mód*. Úroveň detailu výpočtu říká, zda a v jaké míře mají být do výpočtu zahrnuty i jiné faktory, než je vlastní velikost kódu. Existují tři úrovně, kterými jsou *základní model*, *střední model* a *pokročilý model*. Základní model žádné jiné faktory, než je velikost kódu, nepoužívá. Střední model zohledňuje další aspekty projektu. Pokročilý model používá stejné atributy jako model střední a dále ještě přihlíží k fázi, ve které se projekt nachází.

Zatímco úroveň detailu zohledňuje podmínky, v nichž software vzniká, vlastní složitost vyvíjeného produktu se projeví ve zvoleném vývojovém módu. I zde původní COCOMO rozlišuje tři úrovně, kterými jsou *organický mód*, *bezprostřední mód* a *vázaný mód*. Organický mód se volí v případě jednodušších, dobře řešitelných projektů, kde jsou

minimální rizika a velikost produktu je do 50 KSLOC. Bezprostřední mód se používá u středně rozsáhlých projektů, zpravidla do 300 KSLOC, kde je poměrně dobré pochopení požadavků a nehrozí velká rizika. V případě rozsáhlých (nad 300 KSLOC) nebo jinak rizikových projektů se volí vázaný mód. Základní vztahy pro výpočet úsilí a času mají následující podobu.

$$E = a (KSLOC)^b \quad (1)$$

$$T = c E^d \quad (2)$$

Symboly a , b , c a d představují empirické hodnoty parametrů, které jsou volené podle úrovně modelu a použitého vývojového módu z předdefinovaných tabulek. Konkrétně pro základní model $a \in [2,4 ; 3,6]$ a pro střední a pokročilý model $a \in [2,8 F_c ; 3,2 F_c]$. Ostatní hodnoty se pohybují v následujících mezích: $b \in [1,05 ; 1;20]$, $c = 2,5$, $d \in [0,32 ; 0,38]$. Zohlednění dalších faktorů projektu, než je pouze velikost kódu, je ve středním a pokročilém modelu skryto v korekčním faktoru F_c , který je součinem patnácti dalších hodnot popisujících softwarové atributy produktu, hardwarové atributy, atributy vývojového týmu a atributy projektu. Za normálních podmínek je každá z těchto patnácti hodnot rovna jedné. Pokud daný atribut má vliv na zvýšení ceny, vzrůstá i jeho hodnota. Jedná-li se o atribut, ke kterému není třeba příliš přihlížet, tak jeho hodnota naopak klesá. COCOMO nabízí tabulky obsahující konkrétní hodnoty atributů a definuje způsob jejich určení.

Model COCOMO prošel od doby svého vzniku velkým vývojem. V současnosti existuje nejen původní (výše popsaná) verze, ale i verze pro inkrementální vývojový cyklus, verze zohledňující etapu vývoje nebo verze používající statistický odhad *KSLOC* a kvantitativní přístup k nejistotě. COCOMO lze použít i pro odhad nákladů při modifikaci existujících aplikací.

V současnosti nejrozšířenější verzí je model COCOMO II z roku 1995, který používá tři modely. Jsou to *APM* (*Application Composition Model*) pro projekty s použitím moderních nástrojů a GUI, *EDM* (*Early Design Model*) pro hrubé odhady v úvodních etapách, kdy se architektura vyvíjí a *PAM* (*Post Architecture Model*) pro odhady poté, co byla specifikována architektura. COCOMO II také používá nové atributy projektu, které většinou vychází z kombinace dříve používaných atributů.

5. ODHAD POMOCÍ FUNKČNÍCH BODŮ

Pokud není k dispozici odhad velikosti budoucího kódu, je systém COCOMO nepoužitelný. V takovém případě lze rozsah aplikace a následně i pracnost jejího vytvoření odhadnout pomocí tzv. *funkčních bodů*. Funkční body jsou normalizovaná metrika softwarového projektu, která měří aplikační oblast (aplikační funkce a data) a nezkoumá technickou oblast (neměří kód). S trochou nadsázky lze říct, že funkční body představují jednotku výroby při procesu tvorby softwaru. Tímto způsobem lze ohodnocovat pracnost vývoje nejen nových softwarových produktů, ale i pracnost modifikace již existujících systémů.

Nejdříve je třeba spočítat tzv. *neupravené funkční body*. Ty jsou buď vztažené k transakčním funkcím, sem patří *externí vstupy* (*EI - External Inputs*), *externí výstupy* (*EO - External Outputs*) a *externí dotazy* (*EQ - External Enquiry*), nebo vztažené k datovým funkcím, což jsou *vnitřní logické soubory* (*ILF - Internal Logical Files*) a *soubory vnějšího rozhraní* (*EIF - External Interface Files*). Všechny nalezené EI, EO, EQ, ILF a EIF aplikace se roztrídí do skupin podle typu a podle své složitosti. Počet prvků každé skupiny se vynásobí příslušnou vahou a následně se vše sečte. Výsledný součet dává počet neupravených funkčních bodů (*NFP*). Postup výpočtu funkčních bodů přesně stanoví koeficienty jednotlivých vah i postup, jak nalezené funkční body rozdělit do jednotlivých skupin.

Dále se zjišťuje 14 faktorů obecných charakteristik systému, jako je například míra požadované spolehlivosti zálohování a zotavení nebo míra požadovaných on-line vstupů dat. Každý faktor se hodnotí celým číslem do 0 do 5 podle stupně vlivu na aplikaci. Výsledný počet *upravených funkčních bodů (FP)* se získá podle následujícího vztahu.

$$FP = 0,65 \cdot 0,01 \sum F_i \cdot NFP \quad (3)$$

Symbol $\sum F_i$ představuje součet hodnocení čtrnácti charakteristik systému a zkratka *NFP* představuje již zmiňované neupravené funkční body. Následně je třeba stanovit pracnost a cenu jednoho funkčního bodu a odtud stanovit celkové náklady softwarového projektu.

Upravené funkční body lze rovněž použít jako vstupní hodnoty pro výpočet metodou COCOMO. Capers Jones uvádí koeficienty podle nichž lze funkční body přibližně přepočítat na počet příkazů ve vybraném programovacím jazyce [Jon91]. Jednomu funkčnímu bodu odpovídá zhruba 13 příkazů v jazyce SQL, 64 příkazů v C++, 128 příkazů v C nebo 320 příkazů v assembleru.

Funkční body lze využít i pro další odhady. Podle statistik hodnota $FP^{1,15}$ předpovídá přibližný počet stran papírové dokumentace projektu, $FP^{1,2}$ přibližný počet vytvořených testovacích případů, $FP^{1,25}$ přibližný chybový potenciál u nových SW projektů, $FP^{0,4}$ přibližný plán vývoje v kalendářních měsících, $FP/150$ přibližný počet pracovníků potřebných pro řešení aplikace a $FP/750$ předpovídá přibližný počet pracovníků údržby, kteří budou udržovat aplikaci v aktuálně požadovaném stavu.

Značné rozdíly lze vyzorovat i v produktivitě jednotlivých vývojářských týmů, a to jednak v závislosti na zkušenostech, ale i na vybavení, kterým týmy disponují. Nezkušený tým, používající nestrukturované metody, běžné nástroje a jazyky na nízké úrovni má podle Jonese produktivitu 2,5 FP na osoboměsíc. Oproti tomu produktivita zkušeného týmu, používajícího strukturované metody, nástroje CASE a jazyky na vysoké úrovni je přibližně 40 FP za osoboměsíc.

6. ODHAD NÁKLADŮ VYTVOŘENÍ DEFINICÍ PROCESŮ

COCOMO i FPA jsou metody, které lze s úspěchem použít při odhadu pracnosti a nákladů tvorby informačních systémů postavených na třívrstvé architektuře. Z pohledu členění workflow systémů se jedná zejména o tzv. *document based* a *mail based* workflow systémy, které provádějí *administrativní*, *kolaborativní* a *ad hoc* workflow [Car03].

Klíčové firemní procesy však zpravidla spadají do oblasti *produkčního* workflow a jsou stále častěji podporovány *process based* workflow systémy. Tyto systémy však vycházejí z architektury definované workflow referenčním modelem [Hol95]. Nákladnost vybudování takového systému pak není dána pouze nákladností pořízení programů, které podporují vykonávání instancí procesů, ale do nákladů vývoje systému je třeba započítat i náklady vytvoření všech *definicí procesů*, se kterými systém bude pracovat. Jednotlivé definice procesů je dle Davida Hollingsworta dále vhodné sdružovat do větších rozsáhlejších *workflow modelů*, které definují celé workflow firmy [Hol99, Rac02].

Workflow model je popis workflow firmy ve formě, která umožňuje jeho automatickou manipulaci, jakou je například modelování nebo řízení vykonávání prostřednictvím systému workflow. Workflow model obsahuje zejména popis činností, z nichž se skládají jednotlivé procesy, dále popis přechodů mezi těmito činnostmi, popis účastníků, aplikací a dat.

Při odhadu velikosti definicí procesů lze vyjít z principů metody Activity Based Costing. Pokud se podaří identifikovat všechny základní entity, které tvoří workflow model, a odhadnout velikost kódu potřebného pro jejich popis (např. ve formátu XPDL), je možné stanovit celkovou velikost kódu workflow modelu *WfSLOC*.

$$WfSLOC = A_a N_a + A_t N_t + A_u N_u + A_p N_p + A_d N_d \quad (4)$$

Symboly A_a , A_t , A_u , A_p a A_d představují průměrné počty atributů, které se ve workflow modelu použijí pro popis činností, přechodů, účastníků, aplikací a dat. Symboly N_a , N_t , N_u , N_p a N_d pak udávají počty jednotlivých entit v modelu.

S takto získaným odhadem velikosti kódu workflow modelu může být dále pracováno obdobně, jako s odhadem KSLOC používaným v modelu COCOMO, tedy dosadit jej do vztahů (1) a (2). Hodnoty a , b , c a d však budou jiné, nicméně v současnosti není k dispozici dostatečné množství empirických dat pro jejich přesné určení. Rovněž další atributy projektu, jako je například existence či neexistence procesních map, jejichž vytvoření předcházela procesní analýza, by měly být v následném odhadu úsilí E a času T zohledněny ve výpočtu korekčního faktoru F_c .

7. ZÁVĚR

Pojem workflow systém je velmi široký. Při rozhodování se, jaký způsob odhadu nákladů bude pro konkrétní workflow systému použit, je zapotřebí určit, o jaký typ systému se jedná. V případě systémů založených na sdílení dokumentů nebo výměně zpráv de facto jde o klasické informační systémy, kde je podpora procesů skryta ve funkcionalitě systému. V takových případech lze náklady odhadnout pomocí modelu COCOMO nebo, pokud z jakéhokoli důvodu nejsou k dispozici dostatečně spolehlivé odhady velikosti kódu, pomocí funkčních bodů.

V případě procesně orientovaných systémů budovaných zpravidla na principech workflow referenčního modelu se workflow systém skládá ze dvou částí. První část představuje vlastní software, který řídí vykonávání procesů. Zde je opět možné použít COCOMO či funkční body. Druhou část systému tvoří workflow model sdružující všechny definice firemních procesů. Odhad velikosti workflow modelu lze provést technikami využívajícími principy ABC a poté spočítat potřebné úsilí a čas obdobně jako v COCOMO.

LITERATURA

- [Boe81] Boehm, B.: Software Engineering Economics. Prentice-Hall, 1981.
- [Car03] Carda, A., Kunstová R.: Workflow – Nástroj manažera pro řízení podnikových procesů. Grada Publishing, 2003.
- [Hol95] Hollingsworth, D.: The Workflow Reference Model. Issue 3.0, Workflow Management Coalition, 1995.
- [Hol99] Hollingsworth, D.: Terminology & Glossary. Workflow Management Coalition, 1999.
- [Jon91] Jones, C.: Applied Software Measurement: Assuring Productivity and Quality. McGraw-Hill, 1991.
- [Rac02] Ráček, J.: Workflow správních procesů v oblasti životního prostředí. Disertační práce, Masarykova univerzita, 2002.