

# BODY PRÍPADOV POUŽITIA ALEBO AKO MERAŤ SOFTVÉR

*Pre efektívne riadenie celého projektu je potrebné  
merať jeho veľkosť*

Ondrej Jurčák

Slovenská technická univerzita  
Fakulta informatiky a informačných technológií  
Ilkovičova 3, 842 16 Bratislava  
jurcakondrej@gmail.com

**Abstrakt.** Monitorovanie a kontrolovanie priebehu vývoja v každej etape procesu je kľúčovou úlohou manažmentu. Nato aby sme dokázali dobre viesť projekt, naplánovať a kontrolovať jeho priebeh, zabezpečiť dostatočné množstvo zdrojov, je potrebné určiť jeho veľkosť hneď v prvých fázach vývoja. Keďže vývoj softvéru nie je homogénny proces a ovplyvňuje ho množstvo faktorov a charakteristík projektu, odhadovanie je zložitý problém. Najväčším problémom je že na začiatku vidíme len vrchol celého ladarca, čím je vývoj softvéru, a našou úlohou je určiť z relatívne tak mála informácií, ktoré máme na začiatku, aká veľká časť je ponorená pod vodou.

Pri vývoji softvéru hrajú významnú rolu prípady použitia. Vychádzajú z požiadaviek na systém a opisujú jeho funkcionality. Obodovaním prípadov použitia je možné odhadnúť veľkosť softvéru a úsilie potrebného na jeho vývoj čo dáva základ dostatočné množstvo informácií na vytvorenie hlavnej línie projektu. Body prípadov použitia vychádzajú z metódy funkčných bodov a boli prezentované Gustavom Karnerom v roku 1993. Keďže je táto metóda už relatívne dlho známa, jej použitie na rôzne projekty ukazuje na výhody jej použitia pri odhadoch. Ale tak ako má svoje pre má aj svoje proti. V eseji opisujem výhody a nevýhody využitia metódy prípadov použitia v rôznych projektoch.

**Kľúčové slová:** body prípadov použitia, analýza vyrobenej hodnoty, EVA, USECASE Points.

## Úvod

Manažment projektu je v súčasnosti doslova umenie. Prepojenie využitia technológií, ekonomiky a ľudskej spolupráce v kontexte softvérového projektu je ťažká úloha. Úspešnosť softvérového projektu závisí od spokojnosti ľudí stojacich za jeho vznikom, tak ako aj ľudí, ktorí softvérový systém používajú a spravujú.

Používatelia vyžadujú robustný, používateľsky prívetivý systém. Zadávatel' chce, aby bol produkt spoľahlivo dodaný v stanovenom čase a aby nepresiahol rozpočet. Šéfovia majú ambiciózne ciele, vyžadujú zisk a nechcú nepríjemné prekvapenia. Podporný tím vyžaduje produkt dobre dokumentovaný, ľahko modifikovateľný a bez chýb. Členovia vývojového tímu (často nadaní niekedy nezvládnuteľní) vyžadujú zaujímavé technické výzvy.

Vývoj softvérového systému je zložitý proces, ktorý je od začiatku po jeho koniec potrebné riadiť. Každý projekt má určité ciele, ktorých výsledkom v konečnom dôsledku je konkrétny systém. V softvérovom projekte k dosiahnutiu cieľa vedie množstvo ciest, ktoré môžu viesť priamo alebo môžu ísť okľukou a tým presiahnuť stanovený plán a rozpočet. Vybranie konkrétnej cesty a udržanie celého projektu na nej je dôležité pre úspešné ukončenie celého projektu. Je potrebné stále sledovanie priebehu celej cesty aby pri odklonení projektu od stanoveného kurzu bol opäť nasmerovaný na cieľ.

Aby mohol byť priebeh celého vývojového procesu efektívne kontrolovaný musí, je potrebné presne odhadnúť jeho veľkosť. Mnohé metódy používajú rôzne metriky ako sú napríklad riadky textov programu, funkčné body, počty modulov, atď.. Jedno riešenie ponúka metóda Bodov prípadov použitia.

## Body prípadov použitia

Prípady použitia sú artefakty dôležité pri vývoji každého, malého či veľkého systému. Prípady použitia zachytávajú požiadavky na systém a opisujú jeho funkcionality. Obodovanie prípadov použitia a aktérov, figurujúcich v modeli, je základ pre metódu nazývanú Body prípadov použitia. Algoritmus metódy je zachytený v tabuľke Tab. 1.

Tab.1. Metóda Bodov prípadov použitia[1].

Krok	Akcia	Výstup
1	Klasifikácia aktérov a) Jednoduchý, WF = 1 b) Priemerný, WF = 2 c) Zložitý, WF = 3	Neupravená váha hráčov (Unadjusted actor weights – UAW) $UAF = \sum (\text{počet aktérov} * WF)$
2	Klasifikácia prípadov použitia a) Jednoduchý - transakcie $\leq 3$ , WF = 1 b) Priemerný – 4 $\leq$ transakcie $\leq 7$ , WF = 2 c) Zložitý – transakcie $> 7$ , WF = 3	Neupravená váha prípadov použitia (Unadjusted Use Case weights – UUCW) $UUCF = \sum (\text{počet prípadov použitia} * WF)$
3	Výpočet neupravených Bodov prípadov použitia	$UUCP = UAW +$

		UUCW
4	Priradenie hodnôt pre technické faktory a faktory prostredia [0...5], vynásobenie ich váhami [-1..2], a výsledná suma(TFactor a EFactor). Výpočet TCF a EF.	Faktor technickej zložitosti(Technical Complexity Factor- TCF) = $0,6+(0,01*TFactor)$  Faktor prostredia(Enviromental factor - EF)= $1,4 + (-0,03 * EFactor)$
5	Výpočet Bodov prípadov použitia	$UCP = UUCP * TCF * EF$
6	Odhad úsilia v hodinách/osoba	$E = UCP * PH_{perUCP}^1)$

<sup>1)</sup>20-36 hodín/osoba na bod prípadu použitia, závisí od faktora prostredia.

## Výhody

Výhodou je možnosť plnej automatizácie odhadov na základe Bodov prípadov použitia. Pokiaľ je dokument s prípadmi použitia správne štruktúrovaný v rámci firmy je možné využiť automatický nástroj na odhady[3]. Je to výhodou hlavne veľkých projektov a veľkých firiem, v ktorých automatizácia tohto procesu môže zefektívniť prácu.

Ďalšou výhodou je, že v rámci jednej organizácie je možné vypočítať priemerný čas potrebný na implementovanie jedného bodu použitia. Toto môže byť v budúcnosti užitočné pri vytváraní rozvrhu. Nanešťastie ale musí byť každý prípad použitia vytvorený na rovnakej úrovni zložitosti, čo môže byť často problém, keď prípady použitia vytvára viacero autorov.[3]

## Nevýhody

Hlavnou nevýhodou prípadov použitia je, že odhady vytvorené na základe bodov prípadov použitia nie sú použiteľné na priebeh celého procesu. Body prípadov použitia analogicky môžeme použiť až v štádiu, keď ich máme vytvorené, čo môže byť až 10 – 20% celkového úsilia potrebného v softvérovom procese.[1][2]

Ďalšou nevýhodou je, že prípad použitia je príliš veľká jednotka na použitie v plánovacom systéme. Pri vytváraní hrubého odhadu celého projektu môžu byť Body prípadov použitia výhodné. Na vytváranie plánu pre jednotlivé iterácie nie sú body prípadov použitia veľmi dobre použiteľné.[3]

Ďalším relevantným problémom je, že neexistuje jednoznačné pravidlo na určenie, čo v sebe jednotka transakcia predstavuje. Počítanie jednotlivých krokov prípadov použitia je len odhad[3]. Každý krok nepredstavuje rovnakú zložitosť pri jeho implementovaní. Napríklad načítanie vstupu vyžaduje oveľa menšie úsilie ako vykreslenie 3D grafu. V celkovom výsledku to môže spôsobovať dosť veľké zmeny v úsilí potrebnom pri vývoji softvéru. Niektoré prípady použitia aj keď majú rovnaký počet transakcií môžu vyžadovať iné rozdielne úsilie pri ich realizácii.

## Využitie v reálnych projektoch

Podľa článku štúdie[1], v ktorom bolo porovnané využitie Bodov prípadov použitia na štyroch projektoch s rovnakými požiadavkami na systém. Tímy v každého projektu mali podobnú kvalifikáciu a taktiež funkcionality vytvorených systémov bola takmer rovnaká. Tímy využívali rôzne procesy vývoja softvéru a taktiež kládli rôzne požiadavky na kvalitu kódu. Táto štúdia poukazuje na to, že vývojový proces a požiadavky na kvalitu kódu zvyšujú úsilie až o 100% a že metóda Bodov prípadov použitia nedokáže dostatočne zachytiť zvyšovanie úsilia súvisiaceho s výrobným procesom a kvalitou kódu.

V prípadovej štúdii [2] bolo vytvorené porovnanie na veľkom projekte v ktorom boli dostupné aj odhady pomocou expertov. Štúdia ukazuje, že prípady použitia sú dobrým ukazovateľom pri vývoji softvéru ale taktiež zdôrazňuje fakt, že musia byť prípady použitia už vytvorené. Taktiež bolo celá štúdia založená len na jednom projekte, takže výsledky dostatočne neodzrkadľuje zmeny v úsilí, ktoré súvisia s vývojovým procesom a kvalitou.

## Záver

Metóda Prípadov použitia je dobre známa metóda, ktorá je podobná metóde funkčných bodov. Vychádza z prípadov použitia v ktorých sú zachytené požiadavky na systém. Aj keď štúdie ukazujú, že táto metóda je použiteľná a používaná v reálnych projektoch, poukazujú na jeden dôležitý nedostatok a to je fakt, že prípady použitia už musia byť vytvorené. Na odstránenie tohto nedostatku by bola možná kombinácia prípadov použitia aj s metódami, ktoré do odhadov zahrňujú aj získavanie požiadaviek a vytvorenie prípadov použitia. Možný kandidát je napríklad Metóda funkčných bodov.

Taktiež Body prípadov použitia nezachytávajú v dostatočnej miere zmeny vo vývojovom procese.

## Použitá literatúra

1. Anda, B.; Benestad, H.C.; Hove, S.E.; , "A multiple-case study of software effort estimation based on use case points," Empirical Software Engineering, 2005. *2005 International Symposium on*,17-18 Nov. 2005
2. Anda, B. Comparing Effort Estimates Based on Use Case Points with Expert Estimates. In Empirical Assessment in Software Engineering (EASE 2002), Keele, UK, April 8-10, 2002
3. Koirala, S. (2004): How to Prepare Quotation Using Use Case Points, <http://www.codeproject.com/gen/design/usecasepoints.asp>,15 January 2007

## **Annotation**

### *UseCase points or how measure software*

*Monitoring and controlling of project in every phase of development process is important task of management. For the best project control, planning and control its performance, provide sufficiency of resources, is important estimate size and required effort of project in early phases of development. Software development is not homogenous process and depends on many factors and project characteristic, so effort estimation is a very difficult task. Biggest issue is that we see on beginning only top of the iceberg, what is software development, and our task is estimate, from relative small amount of information, which we have on begging, what size of iceberg body is under water.*

*Use cases play important role in software development. They depend from software requirements and describe its functionality. Making points of Use Cases we can estimate size and effort needed for software development, what is sufficient amount of information to create main line of project. Use Case Points outgoing from Function points and was first presented by Gustav Karner at 1993. However this method is known quite long time, it's using on different types of projects point on success usage of this method. So UseCase points have advantages and disadvantages. In the essay below is described disadvantages and advantages of using this method on different kind of projects.*