

# Metode določevanja obsega funkcionalnosti

Tomaž Kralj  
Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko  
tomaz.kralj-ptuj@gov.si

## Izvleček

Merjenje je v programskem svetu vse bolj prisotno, saj omogoča analize, ki se uporabljajo za prikazovanje stanja posameznega projekta. Funkcijske točke so se izkazale kot primerna metrika določevanja obsega informacijskih rešitev. Z njimi merimo funkcionalnost in tako dosegamo neodvisnost metrike od tehnologije in metodologije implementacije. Obstaja več metod, ki ustrezajo standardu ISO 14143, kjer se ne opredelijo metode, temveč pogoji, ki jih mora zagotavljati metoda, da se lahko označi kot veljavna za merjenje obsega funkcionalnosti. V prispevku so predstavljene štiri metode določevanja obsega funkcionalnosti in kratka primerjava med njimi.

## Abstract

### Methods for Measuring Functionality

Measurement in software world is more common nowadays, because it ensures us the base for analyses to track projects. Function point analysis is the most appropriate metric for the size of software application. We measure functionality – this is the reason, why a method is independent of technology and methodology of implementation. There are several methods, which correspond to the ISO 14143 standard. The ISO standard determines conditions which a method must comply to become a legal function point method. In this article we describe and compare four function point methods.

## 1 Uvod

**V programskem svetu se z merjenjem srečujemo kratek čas. Lahko rečemo, da se nahajamo v fazi razvoja metrik merjenja informacijskih rešitev. V primerjavi z drugimi industrijami, kjer je bila potreba po merjenju prisotna že od vsega začetka, je merjenje v programskem svetu šele v povojih.**

Rezultati merjenja omogočajo primerjavo izdelkov in posledično njihovo izboljšavo. Za to potrebujemo pravo enoto, postopke merjenja in merilo, s pomočjo katerih se nedvoumno predstavi velikost izdelka uporabniku ali pa so pridobljeni podatki osnova za analize oz. poročila.

Iz enakih razlogov merimo v svetu informacijskih rešitev. Želja po izboljšanju procesa izdelave informacijskih rešitev je vodilo pri zagotavljanju ustrezne metrike, s pomočjo katere ugotovimo prednosti oz. slabosti vnešenih sprememb. Trenutna dinamika zahteva čim krajši čas od zajemanja zahtev do končnega testiranja in predaje informacijske rešitve stranki, pri čemer krajši čas ne sme biti v škodo natančnosti, funkcionalnosti in majhnemu številu odpovedi informacijske rešitve, ki jo uporabnik prevzame [4].

## 2 Funkcijske točke

Merjenje obsega PO s pomočjo funkcijskih točk je trenutno vodilna metrika programskega sveta [7]. Ena izmed metrik določanja obsega informacijskih rešitev je tudi določevanje obsega funkcionalnosti, ki nam prikaže rezultat v funkcijskih točkah. Obseg določimo s funkcionalnimi specifikacijami, kar pomeni, da se obseg informacijske rešitve lahko določi že v začetni fazi, ko so dokončno podane uporabniške funkcionalne zahteve. Takšen način štetja omogoča neodvisnost merjenja obsega od metodologije in tehnologije implementacije.

Funkcijske točke niso nastale šele pred kratkim. Imajo podobno zgodovino kot objektno orientirana (OO) tehnologija, ki je bila definirana in predstavljena veliko prej, preden je postala paradigma programskega sveta. Idejni oče funkcijskih točk je Alan Albrecht – funkcijske točke so rezultat njegove študije v sedemdesetih letih 20. stoletja. V današnjem času dosegajo razmah, ki pomeni novo obdobje merjenja informacijske rešitve oz. pomoč pri definiranju novih postopkov.

Funkcijske točke nudijo širok spekter uporabe. Ena izmed prednosti je možnost uporabe funkcijskih točk kot normalizacijske metrike, ki omogoča primerjavo med informacijskimi rešitvami, implementiranimi v različnih programskih jezikih [3]. Nekaj primerov uporabe:

- določevanje obsega opazovane informacijske rešitve;
- ocenjevanje potrebnih virov projekta, ko so izdelane funkcionalne specifikacije zahtev [6];
- podlaga za primerjanje med konkurenčnimi organizacijami, saj omogočajo izvajanje primerjalnih testov v svetu PO ne glede na metodologijo načrtovanja in tehnologijo implementacije;
- omogočajo možnost izdelave resnih ekonomskih analiz [2].

Znanih je že nekaj metod, s pomočjo katerih merimo obseg funkcionalnosti informacijskih rešitev. Metode morajo ustrezati standardu ISO 14143, kjer se ne opredelijo metode, temveč pogoji, ki jih mora zagotavljati metoda, da se lahko označi kot veljavna za merjenje obsega funkcionalnosti.

Sledi kratka predstavitev štirih najpogostejših metod določevanja obsega funkcionalnosti, njihove glavne značilnosti in kratka primerjava med njimi.

### 3 Standardna metoda FPA (Function Point Analysis)

Avtor metode FPA, ki jo imenujemo tudi osnovna oz. standardna metoda FPA, je Alan Albrecht. Za standardno metodo FPA je od leta 1984 odgovorna neprofitna organizacija IFPUG (*International Function Point Users Group*), ki skrbi za njen nadaljnji razvoj.

V standardni metodi FPA se srečamo s petimi temeljnimi komponentami, ki bodo podrobneje opisane v korakih algoritma na sliki 1 [7], [8], [9], [10]:

- notranje podatkovne strukture ILF (*Internal Logical File*);
- zunanje vmesniške podatkovne strukture EIF (*External Interface File*);
- zunanji vhodi EI (*External Input*);
- zunanji izhodi EO (*External Output*);
- zunanja povpraševanja EQ (*External inquiry*).

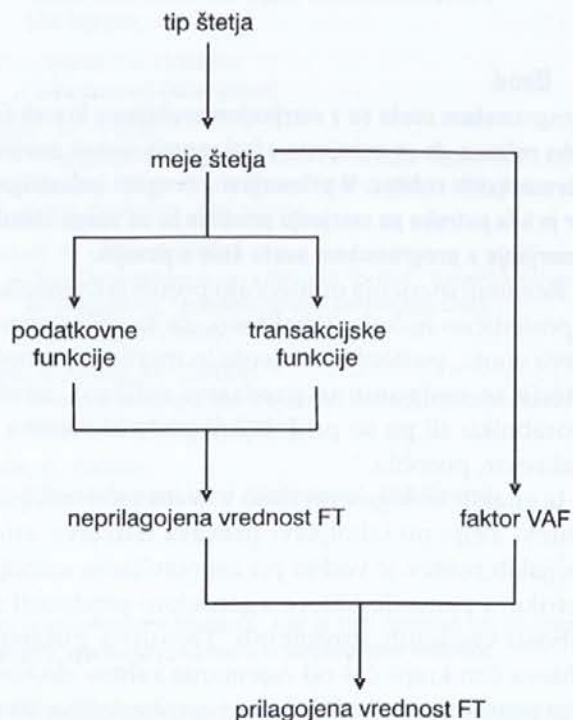
Standardna metoda FPA je predstavljena v obliki algoritma. Rezultat algoritma je obseg funkcionalnosti merjene informacijske rešitve. Algoritem vsebuje naslednje korake:

#### a) Določitev tipa štetja

- *Razvojni projekt* – želimo določiti obseg projekta, pri katerem je končana faza analize in načrtovanja. To pomeni, da so specifične funkcionalnosti, ki jih mora vsebovati informacijska rešitev. Med razvojem se večkrat zgodi, da se kakšna funkcionalnost doda (*scope creep*), zato je treba popraviti uporabniške funkcionalne zahteve, s tem pa se spremeni tudi obseg informacijske rešitve.
- *Nadgradnja programa* – merijo se modifikacije v obstoječi informacijski rešitvi, v kateri je bila dodana, spremenjena ali brisana funkcionalnost.
- *Program* – meri se obseg informacijske rešitve, ki je instalirana oz. v uporabi. Ta vrednost se imenuje obseg osnovne verzije (*baseline*), ki jo je treba ažurirati sproti (še posebej po nadgradnji informacijske rešitve).

#### b) Identifikacija meje štetja in mej programov

pomeni določitev konceptualne meje med informacijsko rešitvijo in uporabnikom. Deluje kot membrana, skozi katero prehajajo podatki v informacijsko rešitev in preko transakcij tudi iz nje.



Slika 1: Postopek štetja FT

Znotraj meje informacijske rešitve obstajajo logične strukture, ki jih vzdržuje opazovana informacijska rešitev. Od zunaj prihajajo podatkovne strukture, ki se uporabljajo preko vmesnikov, njihovo stanje pa vzdržuje informacijska rešitev zunaj meja opazovanega sklopa.

c) **Štetje podatkovnih funkcij** – podatkovne funkcije predstavljajo podatkovne zahteve, ki jih je definiral uporabnik. Predpišejo podatkovne strukture, katerih naloga je hranjenje podatkov.

Pri identificiranju podatkovnih funkcij se je treba distancirati od tehničnih rešitev in se skoncentrirati na uporabnikove zahteve. To pomeni, da fizična implementacija ne pomeni tudi logične definicije podatkovnih skupin, lahko pa sovpadata. Podatkovne funkcije morajo biti uporabljene vsaj v eni transakcijski funkciji, v nasprotnem primeru njihov obstoj ni upravičen.

Podatkovne funkcije se delijo na tiste, ki se vzdržujejo znotraj informacijske rešitve (ILF), in podatkovne funkcije, ki se le uporabljajo znotraj informacijske rešitve (EIF) in so drugje zastopane kot notranje podatkovne funkcije. Ko so identificirane, se določi kompleksnost in izračuna prispevek podatkovnih funkcij k skupni vrednosti funkcijskih točk s pomočjo obstoječih tabel.

č) **Štetje transakcijskih funkcij** – transakcijske funkcije predstavljajo funkcionalnost procesiranja podatkov v informacijski rešitvi. Predstavljajo manipulacijo s podatki, ki so shranjeni v podatkovnih funkcijah. Poznamo tri tipe transakcijskih funkcij (slika 2):

- EI (*External Input*) **zunani vhod** je elementarni proces, ki procesira podatke ali kontrolne informacije, ki pridejo od zunaj glede na mejo štetja. Osnovni namen je vzdrževanje ene ali več ILF znotraj meje štetja in/ali sprememba obnašanja sistema.

- EO (*External Output*) **zunani izhod** je elementarni proces, ki pošilja podatke na drugo stran meje informacijske rešitve. Osnovni namen je prikaz informacij uporabniku s pomočjo poslovne logike. Procesorska logika mora vsebovati vsaj en podatek, ki se pridobi s pomočjo formule ali kalkulacije. Vsebuje sestavljen podatek, vzdržuje eno ali več podatkovnih funkcij ILF in/ali spremeni stanje sistema.

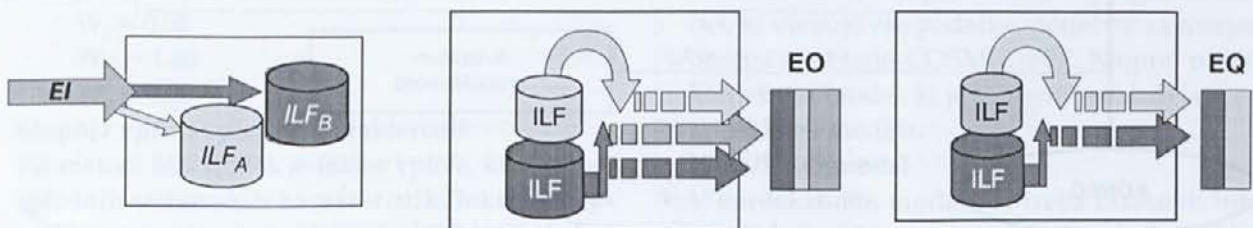
- EQ (*External Inquiry*) **zunanje poizvedovanje** je prav tako elementarni proces in je podobno zunanjemu izhodu, le da prikazuje podatke brez nadaljnje obdelave in zato ne sme vsebovati sestavljenih podatkov oz. podatkov, ki vsebujejo matematično formulo. Ne sme vzdrževati podatkovnih funkcij ILF in spreminjati stanja sistema.

Tudi pri transakcijskih funkcijah se določi kompleksnost in prispevek le-teh k skupni vrednosti funkcijskih točk merjene informacijske rešitve.

d) **Določevanje neprilagojene vrednosti funkcijskih točk** – gre za vsoto števila funkcijskih točk podatkovnih in transakcijskih funkcij.

e) **Določanje vrednosti faktorja prilagoditve VAF** (*Value Adjustment Factor*) – faktor VAF se izračuna na podlagi 14 splošnih sistemskih karakteristik GSC (*General System Characteristic*). Za vsako karakteristiko se posebej oceni vpliv v opazovani informacijski rešitvi z vrednostmi od 0 (ni vpliva) do 5 (močan vpliv).

f) **Izračun prilagojene vrednosti funkcijskih točk** – prilagojena vrednost pomeni, da so upoštevane dodatne zahteve informacijske rešitve, kot so stopnja varnosti, zahtevnost procesiranja ipd. Te karakteristike se zrcalijo v faktorju prilagoditve. Prilagojeni obseg merjene informacijske rešitve se izračuna tako, da se dobljeno število funkcijskih točk pomnoži s faktorjem prilagoditve (VAF).



Slika 2: Transakcijske funkcije

#### 4 Metoda MK II FPA

MK II FPA pomeni *Mark II Function Point Analysis*. Varianto MK II FPA je leta 1988 definirala Charles Symons. Trenutno skrbi za njen razvoj Metrics Practices Committee (MPC) UK Software Metrics Association [16].

Slika 3 prikazuje proces merjenja z metodo MK II FPA, kjer je metoda razdeljena na tri dele: merjenje obsega funkcionalnosti, produktivnosti in prilagojene vrednosti funkcijskih točk (opcijsko).

Algoritem vsebuje naslednje korake:

a) **Določitev namena in tipa štetja** – poznamo tri vidike:

- *projektni vidik* – t. i. delovni vidik;
- *aplikacijsko vodeni vidik* – obseg informacijskih rešitev za specifično poslovno področje;
- *poslovni vidik* – obseg informacijskih rešitev podjetja.

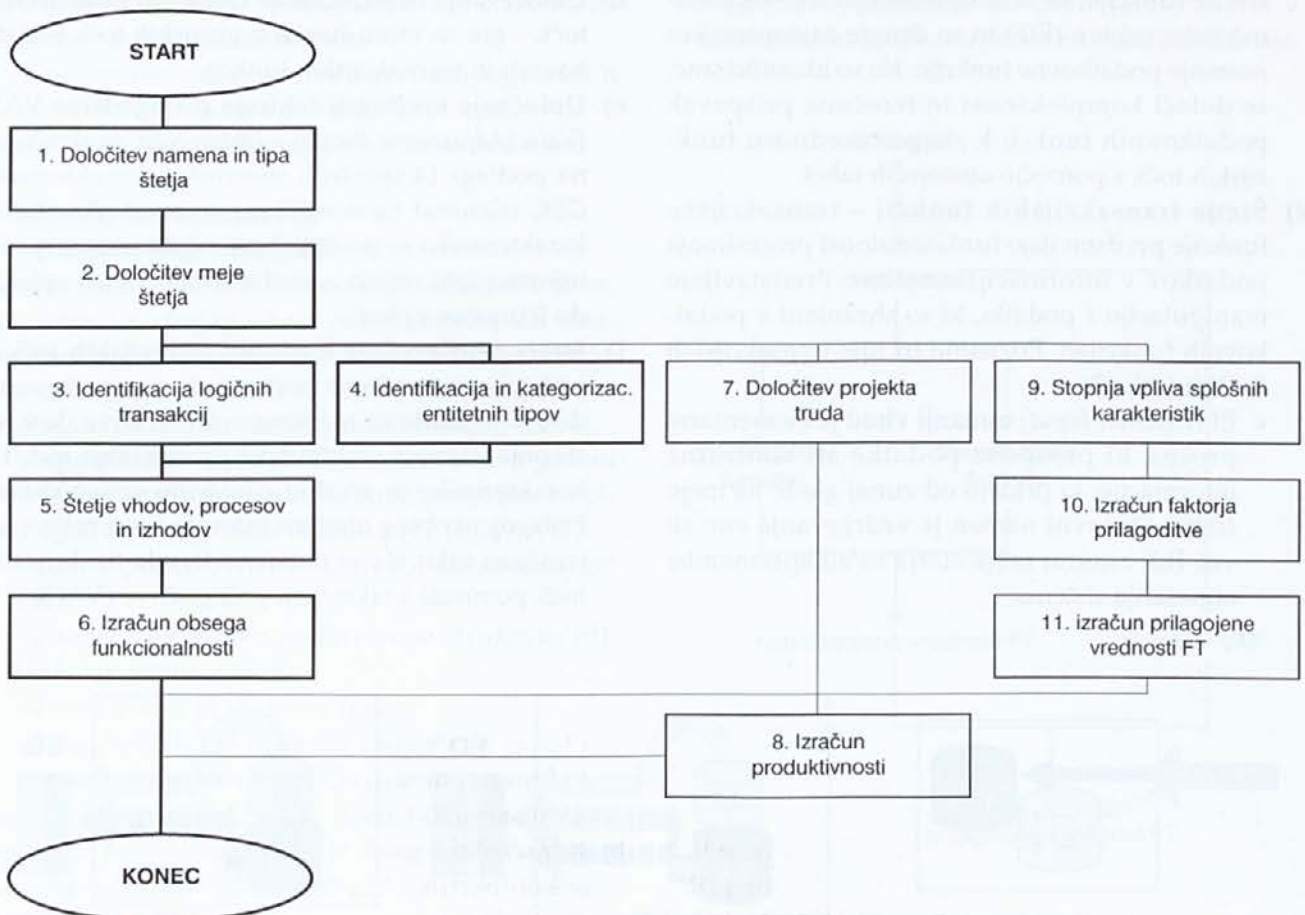
b) **Določitev meje štetja** – konceptualna meja med informacijsko rešitvijo in uporabnikom.

c) **Identifikacija logičnih transakcij** – najprej je treba definirati pojem logična transakcija: gre za poslovni proces na najnižjem nivoju, ki ga zagotavlja merjena informacijska rešitev. Vsebuje tri osnovne elemente:

- vhod podatkov preko programske meje;
- procesiranje podatkov;
- izhod podatkov preko programske meje.

Vsaka logična transakcija mora biti prožena z unikatnim dogodkom, ki je iniciran zunaj programske meje. Zahtevati jo mora uporabnik, pri tem pa mora transakcija pustiti sistem po končanem procesiranju v konsistentnem stanju.

č) **Identifikacija in kategorizacija entitetnih tipov** Poslovni uporabnik ima predstavo o svojem poslovnem svetu, zato lahko točno definira



Slika 3: Proces metode MK II FPA

podatke, ki jih potrebuje. Logične skupine takih podatkov imenujemo entitete. Primer take entitete je *stranka*, ki vsebuje več podatkovnih elementov DET (*Data Element Type*): ime in priimek, bivališče, davčna številka itd.

Pri metodi MK II FPA gre pri merjenju obsega logične transakcije za preštevanje entitetnih tipov. Poznamo dve vrsti le-teh:

- *primarni entitetni tip* so stvari v realnem svetu;
- *neprimarni entitetni tip* – vse neprimarne entitete združimo v eno entiteto, ki jo imenujemo *sistem-ska entiteta* (šifranti).

Določena entiteta je lahko primarna za eno informacijsko rešitev in neprimarna za drugo. Takšna klasifikacija prikazuje podobnost s podatkovnimi funkcijskimi tipi standardne metode FPA:

- primarna entiteta – ILF,
- neprimarna entiteta – EIF.

#### d) Štetje vhodov, procesov in izhodov

Ko so identificirane logične transakcije in entitetni tipi, se začne preštevanje vhodnih in izhodnih podatkovnih elementov. Pri vhodu in izhodu se prešteje število unikatnih podatkovnih elementov, pri procesiranju pa število referenciranih entitet. Ponavljanje posameznega elementa znotraj logične transakcije se upošteva kot en element.

#### e) Izračun obsega funkcionalnosti

Obseg funkcionalnosti je obtežena vsota komponent vhoda, procesiranja in izhoda. Izračuna se tako, da se pomnoži število vhodnih elementov DET ( $N_i$ ), število entitet, ki so bile referencirane v procesiranju ( $N_e$ ), in število izhodnih elementov DET ( $N_o$ ) z industrijskimi utežmi, ki so empirično določene. Prva formula vrne kot rezultat nepriprilagojeno vrednost funkcijskih točk, ki se pri metodi MK II FPA označuje z oznako FPI (*Function Point Indeks*).

$$FPI = W_i * \sum N_i + W_e * \sum N_e + W_o * \sum N_o \quad (1)$$

Industrijske uteži so določene kot konstante:

$$W_i = 0.58$$

$$W_e = 1.66$$

$$W_o = 0.26$$

#### f) Stopnja vpliva splošnih karakteristik

Pri metodi MK II FPA je faktor vpliva, ki je odraz splošnih sistemskih karakteristik, faktor prilagoditve tehnične kompleksnosti TCA (*Technical Complexity Adjustment*). Tudi tukaj se ocenjuje posamezna karakteristika z vrednostmi od 0 (brez

vpliva) do 5 (močan vpliv). Razlika je v številu sistemskih karakteristik, saj se jih pri metodi MK II FPA ocenjuje 19 (pri VAF le 14).

- g) **Izračun prilagojene vrednosti funkcijskih točk** – tudi tukaj se izračuna prilagojena vrednost merjene informacijske rešitve kot produkt števila funkcijskih točk s faktorjem prilagoditve.

## 5 Metoda COSMIC-FFP

COSMIC (*COmmon Software Measurement International Consortium*) je najmlajša med naštetimi. Avtorji te metode trdijo, da vsebuje najboljše lastnosti naslednjih metod: standardne metode FPA, NESMA, MK II FPA in Full Function Point.

Metoda COSMIC-FFP je nastala zaradi pomanjkljivosti standardne metode FPA. Po mnenju avtorjev metode COSMIC-FFP so pomanjkljivosti standardne metode FPA naslednje [1]:

- FPA ima omejeno uporabnost zunaj MIS (*Management Information System*) sveta – primer takega sistema so sistemi v realnem času [5],[12];
- FPA tehnike niso vedno kompatibilne z modernimi metodami načrtovanja PO;
- rezultati metode FPA so subjektivni;
- vprašljiva je natančnost in konsistentnost rezultatov FPA metod;
- nezadovoljivi rezultati pri določanju obsega funkcionalnosti tehničnih sistemov.

Osnova za določanje obsega funkcionalnosti so, enako kot pri drugih metodah, uporabniške funkcionalne specifikacije. Na ta način dosejajo metode določanja obsega funkcionalnosti neodvisnost od tehnologije in metodologije implementacije. Na podlagi funkcionalnih zahtev se zgradi programski model, v katerem so združene vse informacije, ki so potrebne za merjenje (slika 4).

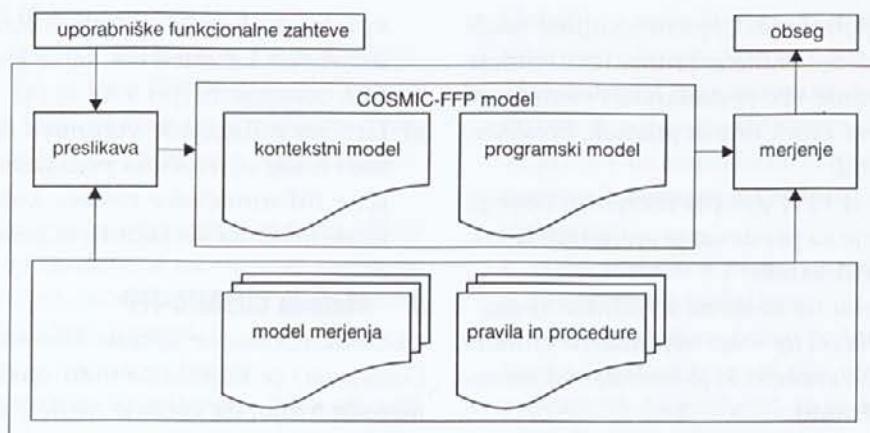
Faze procesnega modela COSMIC-FFP:

#### a) Faza preslikave

Vhod v fazo preslikave so uporabniške funkcionalne zahteve. V tej fazi nastane programski model, ki vsebuje vse podatke, potrebne za merjenje obsega z metodo COSMIC-FFP. Najprej nastane kontekstni model, ki je v pomoč pri izdelavi programskega modela.

#### b) Kontekstni model

V kontekstnem modelu je treba razmejiti informacijsko rešitev in operacijsko okolje. Slika 5 prikazuje splošni primer kontekstnega modela skupaj s pretokom podatkov.



Slika 4: COSMIC-FFP procesni model merjenja

### c) Programski model

Programski model se zgradi na osnovi kontekstnega modela, ki ga prikazuje slika 5. Uporabniške funkcionalne zahteve so implementirane s funkcionalnimi procesi, ki so sestavljeni iz podprocesov. Le-ti so sestavljeni iz premikanja podatkov in manipulacij z njimi. V programskem modelu poznamo štiri tipe premikanja podatkov: *vhod*, *izhod*, *branje* in *pisanje*. Podatki vstopijo v proces in izstopijo k uporabniku, branje in pisanje se izvrši preko pomnilniških enot.

### č) Faza merjenja

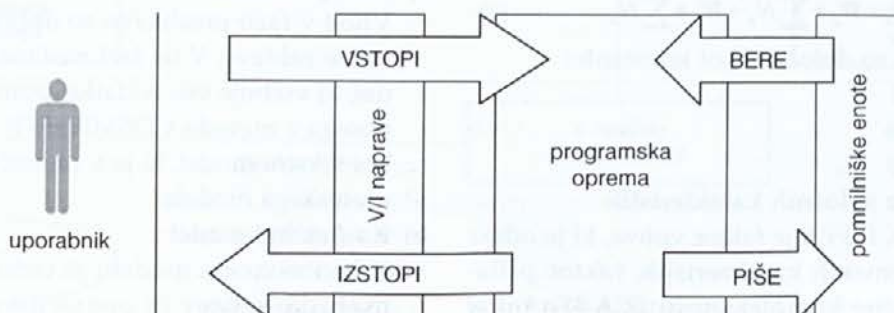
Merjenje obsega funkcionalnosti z metodo COSMIC-FFP pomeni štetje podprocesov premikanja podatkov v programskem modelu. Tipi premikanja so osnovne komponente metode COSMIC-FFP. Osnovne značilnosti merjenja so:

- enota merjenja je Cfsu (*Cosmic Functional Size Unit*), ki pomeni en premik podatkov na nivoju podprocesov;

- določiti je mogoče tudi natančnejše enote (analogija meter – centimeter);
- obseg funkcionalnosti posameznega nivoja je enak vsoti vrednosti obsega funkcionalnosti posameznih funkcionalnih procesov, celotne merjene informacijske rešitve pa vsoti vrednosti obsega funkcionalnosti posameznih nivojev;
- najmanjši obseg dela informacijske rešitve je 2 Cfsu, saj mora imeti najmanjši funkcionalni proces vsaj en vstop in en izhod oz. vpis. Zgornje meje ni.

Agregacijske rezultati se računajo tako, da se obseg delov merjene informacijske rešitve prišteje k skupni vrednosti obsega funkcionalnosti. Za določevanje obsega funkcionalnosti posameznega nivoja, ki je izražen v enotah Cfsu, je na voljo formula 2:

$$\text{Velikost}_{\text{cfsu}}(\text{nivo}_i) = \sum \text{velikost}(\text{vhodi}_i) + \sum \text{velikost}(\text{izhodi}_i) + \sum \text{velikost}(\text{branja}_i) + \sum \text{velikost}(\text{pisanja}_i) \quad (2)$$



Slika 5: Pretok podatkov – funkcionalni pogled

## 6 Metoda NESMA

NESMA (*N*etherlands *S*oftware *M*etrics *u*sers *A*ssociation) je ena izmed prvih uporabniških skupin FPA. Zasnovana je bila na standardni metodi FPA, definirani v priročniku IFPUG CP2 2.0. Standardna metoda FPA je bila takrat namenjena predvsem merjenju produktivnosti, zato je merila končane projekte. Metoda NESMA je želela zapolniti to vrzel in omogočiti načrtovanje stroškov za dokončanje projekta na podlagi uporabniških funkcionalnih specifikacij. Metodi sta se približali leta 1994, ko je IFPUG izdala priročnik CPM 4.0, v katerem je predstavila metodo štetja funkcijskih točk na podlagi funkcionalnih specifikacij.

NESMA je izdala trenutno aktualni priročnik *Function Point Manual (FPM) 2.0* [14], v katerem je istimenska metoda natančneje opisana.

Koraki določevanja obsega funkcionalnosti metode NESMA so skoraj identični kot pri standardni metodi FPA. Metoda NESMA je specifična glede treh tipov štetja:

- detajlno štetje funkcijskih točk;
- ocenjevanje funkcijskih točk;
- določevanje (*indicative*) funkcijskih točk.

Zakaj takšno ločevanje? Z metodo NESMA se poskuša približati funkcijske točke uporabnikom, saj metoda na ta način omogoča grob vpogled v razsežnosti informacijske rešitve kakor tudi natančen obseg funkcionalnosti. Koraki algoritma posameznega štetja so naslednji:

- a) **Detajlno štetje funkcijskih točk** – standardni postopek:
  - identifikacija vseh funkcijskih tipov (ILF, EIF, EO, EI, EQ);
  - določitev kompleksnosti posamezne funkcije;
  - izračun neprilagojene vrednosti funkcijskih točk (UPF).
- b) **Ocenjevanje funkcijskih točk** – ne določamo kompleksnosti posamezne funkcije:
  - identifikacija vseh funkcijskih tipov (ILF, EIF, EO, EI, EQ);
  - določitev kompleksnosti posamezne podatkovne funkcije (*nizka*) in vsake transakcijske funkcije (*srednja*);
  - izračun neprilagojene vrednosti funkcijskih točk.
- c) **Označevanje (*indicative*) funkcijskih točk**:
  - identifikacija podatkovnih funkcijskih tipov (ILF, EIF);

- izračun neprilagojene vrednosti funkcijskih točk po formuli 3:

$$UFP = 35 * \text{število\_ILF} + 15 * \text{število\_EIF} \quad (3)$$

Formula 3 temelji na naslednji predpostavki:

- za vsako podatkovno funkcijo ILF bodo obstajale:
  - tri transakcijske funkcije EI (dodajanje, brisanje in spreminjanje),
  - dve EO,
  - ena EQ.
- za vsako podatkovno funkcijo EIF bo obstajala:
  - ena EO,
  - ena EQ.

Predpostavke so rezultati analize večih projektov. Zadnja metoda je znana kot »*the Dutch Method*«. Nudi hiter postopek za približno oceno obsega informacijske rešitve.

## 7 Primerjava metod

Sledi primerjava med standardno metodo FPA in drugimi metodami določevanja obsega funkcionalnosti. Izpostavljene bodo prednosti oz. slabosti posamezne metode, ki so razvidne že iz samih definicij metod.

### 7.1 Standardna metoda FPA – metoda MK II FPA

Obe metodi imata veliko stičnih točk. Ob primerjanju rezultatov do 400 funkcijskih točk so rezultati primerljivi, pri informacijskih rešitvah obsega nad 400 FT postanejo razlike bolj očitne [15].

Kje prihaja do razlik oz. na kakšen način je mogoče primerjati rezultate merjenja obsega funkcionalnosti obeh metod?

#### Faktor prilagoditve

Obe metodi uporabljata faktor, s katerim se prilagodi vrednost funkcijskih točk glede na uporabniške zahteve informacijske rešitve. Povprečne vrednosti, ki jih dosegajo faktorji posamezne metode, so naslednje [15]:

- standardna metoda FPA      1.0
- metoda MK II FPA            0.85

#### Logične transakcije

Metoda MK II FPA poskuša opisati uporabniške zahteve z logičnimi transakcijami. Vsaka logična transakcija vsebuje vhod, procesiranje in izhod. Logična transakcija se proži z dogodki zunaj meje informacijske rešitve, ki predstavljajo interese uporabnika.

Industrijske uteži za metodo MK II FPA (za vhod, procesiranje in izhod) so prilagojene tako, da so rezultati merjenj obeh metod primerljivi za projekte obsega 200 do 400 FT. Seštevek uteži da vrednost 2.5 (vhodi 0.58, procesiranje 1.66, izhod 0.26), zato ima najbolj preprosta logična transakcija obseg 2.5 FT.

### Zakaj razlika v obsegu?

Standardna metoda meri obseg transakcijske funkcije tako, da najprej oceni kompleksnost transakcijske funkcije in nato odčita število funkcijskih točk (odvisno od kompleksnosti) iz točno določenih tabel. Na ta način se za posamezno transakcijsko funkcijo določi obseg od 3 do 7 funkcijskih točk (odvisno od kompleksnosti in tipa transakcijske funkcije). To pomeni, da nobena transakcijska funkcija ne more biti ocenjena za manj kot 3 funkcijskimi točkami, pa naj je še tako preprosta. Enako velja za zgornjo mejo, saj je le-ta nastavljena na 7 funkcijskih točk ne glede na obsežnost oz. kompleksnost transakcijske funkcije.

Pri metodi MK II FPA tega problema ni, saj se z večanjem vhodnih oz. izhodnih elementov in referenciranih podatkovnih funkcij povečuje tudi izmerjeni obseg transakcijske funkcije.

## 7.2 Standardna metoda FPA – metoda COSMIC-FFP

V metodi COSMIC-FFP so združene dobre lastnosti drugih metod. Metoda je produkt obstoječih metod, še najbolj podobna metodi FFP (*Full Function Point*), zato tudi kratica FFP v njenem imenu. Principi metode FFP niso opisani posebej, ker so podobni metodi COSMIC-FFP.

V literaturi zasledimo, da standardna metoda FPA ne prikaže adekvatnih rezultatov pri merjenju aplikacij sistemov v realnem času, saj se le-te razlikujejo od poslovnih sistemov, za katere je bila razvita standardna metoda FPA. Pri aplikacijah sistemov v realnem času je veliko podprocesov, kar ne ustreza standardni metodi FPA. Metoda COSMIC-FFP upošteva podprocese znotraj unikatnega procesiranja tako, da identificira podatkovne skupine podatkov, ki jih podproces prejme, pošlje, bere ali zapiše. Posledica tega je večje število funkcijskih točk ob koncu merjenja, kar pomeni adekvatnejšo meritev obsega funkcionalnosti merjene informacijske rešitve.

Standardni metodi FPA se očita neprimernost faktorja prilagoditve, ki ne prikaže pravih rezultatov za sisteme v realnem času, zato so v teku pogajanja za modifikacijo faktorja prilagoditve oz. njegovo ukinitve.

### Vzrok za nastanek razlik

Razlogi za razlike so pri podatkovnih in transakcijskih funkcijah:

- **podatkovne funkcije:** v sistemih v realnem času je veliko podatkovnih skupin, ki vsebujejo le en zapis. Slednje je težje opisati s pomočjo podatkovnih funkcij ILF in EIF standardne metode FPA;
- **transakcijske funkcije:** sistemi v realnem času vsebujejo veliko podprocesov znotraj ene transakcije. Naslednji primer kaže vpliv velikega števila podprocesov na štetje funkcijskih točk: Glavni proces sistema v realnem času ima nalogo prikazati količino tekočine. Identificirani so trije podprocesi:
  - senzor odčita količino tekočine;
  - količina se primerja glede na dovoljene meje;
  - pošiljanje sporočila v primeru odstopanja od dovoljene meje.

Identificirani so trije podprocesi. Pri standardni metodi FPA bi bila v tem primeru identificirana le ena transakcijska funkcija, ker je treba upoštevati navodilo, da se identificira transakcijska funkcija takrat, ko gre za elementarni proces (vsi trije podprocesi tvorijo zaključeno celoto).

## 7.3 Standardna metoda FPA – metoda NESMA

Podobnost metod se kaže v naslednjih stičnih točkah:

- uporabljata pet funkcijskih tipov (EI, EO, EQ, ILF, EIF);
- uporabljata enake tabele za določevanje kompleksnosti;
- uporabljata 14 splošnih sistemskih karakteristik za določevanje faktorja prilagoditve;
- uporabljata in določata sledeče koncepte:
  - neprilagojena vrednost funkcijskih točk UFP;
  - faktor prilagoditve;
  - prilagojena vrednost funkcijskih točk AFP.

Kot je možno razbrati, metodi uporabljata enake koncepte in enako metodologijo merjenja obsega funkcionalnosti. Pomembneje sta se razlikovali na začetku nastanka metode NESMA, sedaj lahko glavne razlike strnemo le v nekaj točkah:

- možnost povpraševanja s spremenljivim izhodom, kar pomeni, da uporabnik na vходу določi, katere podatke bo pridobil s povpraševanjem. *Standardna FPA* prepozna tako transakcijsko funkcijo kot zunanje povpraševanje, saj ni dodanega nobenega sestavljenega polja oz. polja, pridobljenega s kakršno koli kalkulacijo.



NESMA določa takšen tip funkcije kot zunanji izhod, saj uporabnik pridobi podatke spremenljive velikosti.

- velikokrat je treba uporabniku prikazati podatke, preden jih izbriše oz. spremeni. Tako uporabnik vidi, kateri podatki se bodo spremenili. Tako prikazovanje podatkov se imenuje implicitno povpraševanje (*implicit inquiry*). Kako obravnavata tako funkcionalnost obe metodi?

*Standardna FPA* šteje tako povpraševanje kot zunanje povpraševanje, če slednje zahteva uporabnik.

*Metoda NESMA* šteje tako povpraševanje kot dodatni EQ le v primeru, če uporabnik zahteva prikazovanje posebne oblike. V nasprotnem primeru upošteva implicitno povpraševanje kot del algoritma za brisanje oz. shranjevanje spremenjenih podatkov.

## 8 Zaključek

Določanje obsega funkcionalnosti je metoda, ki pomaga pri upravljanju informacijske rešitve in stroškov znotraj organizacije. Izbira ustrezne metode je opravilo, ki se ga je treba lotiti preudarno. Pa vendarle je najbolj pomemben prvi korak, ko se odločimo za merjenje z eno od metod določevanja obsega funkcionalnosti. S prikazovanjem obsega informacijske rešitve s pomočjo enote funkcijskih točk bomo pridobili na več področjih znotraj organizacije:

- možnost natančnejšega planiranja potrebnih virov za dokončanje projekta;
- možnost izvajanja resnih analiz na nivoju upravljalca projektov;
- izdelava poročil na najvišjem nivoju vodstva, saj je razumljivejši pojem *obseg funkcionalnosti* kakor *število vrstic programske kode*, ki v poročilih privedejo tudi do nekonsistentnih rezultatov.

Primerjave metod prikažejo prednosti oz. slabosti posamezne metode. Ugotovili smo, da je standardna metoda FPA primerna za informacijske rešitve poslovnega sveta in kot taka ne daje zadovoljivih rezultatov pri aplikacijah sistemov v realnem času. Pri le-teh se bolj izkažeta metodi MK II FPA in metoda COSMIC-FFP.

Poplava različnih metod bi lahko odvrnila morebitne uporabnike merjenja s funkcijskimi točkami in

obratno: če bi obstajala močna baza, ki bi promovirala merjenje s pomočjo funkcijskih točk in skrbela za njihov razvoj, bi tovrstno merjenje pridobilo na popularnosti, kar bi posledično pomenilo izboljšanje metode in splošno zadovoljstvo uporabnikov. Velik korak je storjen že, če se zavedamo potrebe merjenja v svetu informacijskih rešitev, saj brez merjenja ni mogoče izvajati ustreznega nadzora.

## 9 Literatura

- [1] Abram Alain, Jean-Marc Desharnais, Serge Oligny, Denis ST-Pierre, Charles Symons: COSMIC-FFP Measurement Manual, version 2.1, Maj 2001
- [2] Arifoglu, A.: A Methodology for Software Cost Estimation. Software Engineering Notes, 1993
- [3] Garmus David, Herron David: Function Point Analysis, Addison-Wesley, 2000
- [4] Guidelines to Software Measurement – Release 1.1, 1996, IFPUG
- [5] Ho, V.T., Abran, A., Oligny, S.: Using COSMIC-FFP to Quantify Functional Reuse in Software Development, in Escom-Scope 2000, April 18-20, 2000
- [6] Hürten Robert: Function-Point Analysis - Theorie und Praxis, Expert Verlag, Renningen-Malmsheim, 1999
- [7] IFPUG Function Point Counting Practices Manual – Release 4.1, januar 1999, IFPUG
- [8] IFPUG Function Point Counting Practices: Case Study 1 – Release 1.0, 1994, IFPUG
- [9] IFPUG Function Point Counting Practices: Case Study 3 – Release 1.0, 1994, IFPUG
- [10] IFPUG Guidelines to Software Measurement - Release 1.1, 1996, IFPUG
- [11] Jones, Capers: Backfiring - Converting Lines of Code to Function Points, IEEE Computer 28, 11, November 1995: 87-8
- [12] Krishna C. M., Kang G. Shin: Real-Time Systems, McGraw-Hill, 1997, str: 1-39
- [13] Meli, R., Abran, A., Ho, V.T., Oligny, S.: On the Applicability of COSMIC-FFP for Measuring Software Throughout its Life Cycle, in Escom-Scope 2000, April 18-20
- [14] Netherlands Software Metrics Association (NESMA): Definitions and Counting Guidelines for the Application of Function Point Analysis, Version 2.0. Amsterdam, 1997
- [15] Symons Charles: Conversion between IFPUG 4.0 and MkII FPA Function Points -Version 3.0, Software Measurement Services Ltd, 1999
- [16] United Kingdom Software Metrics Association (UKSMA): MK II Function Point Analysis - Counting Practices Manual, version 1.3.1, UKSMA, september 1998

Tomaž Kralj je leta 1997 diplomiral na Fakulteti za elektrotehniko, računalništvo in informatiko Univerze v Mariboru, 2002 pa je na tej fakulteti končal podiplomski študij z zagovorom magistrskega dela. Kot svetovalec direktorja Davčne uprave Republike Slovenije je zaposlen na Davčnem uradu na Ptujju.