



МОДИФИКАЦИЯ МЕТОДА ФУНКЦИОНАЛЬНЫХ ТОЧЕК ДЛЯ УЧЕТА ИЗМЕНЧИВОСТИ ТРЕБОВАНИЙ К ПРОГРАММНОМУ ПРОДУКТУ

*Васильцова Н.В., Панферова И.Ю.*

*Харьковский национальный университет радиоэлектроники*

Классический метод функциональных точек в настоящее время остается одним из наиболее часто применяемых методов количественного оценивания затрат на выполнение ИТ-проектов создания программных продуктов. Данный метод позволяет на основе информации о функциях и потоках данных разрабатываемого продукта определить количество функциональных баллов как условных единиц оценки объема работ по написанию программного кода данного продукта. В общем случае эти функциональные баллы рассчитываются по формуле [1]

$$\begin{aligned} DFP = & \left( \sum_{i=1}^{n_{ILF}} UFP_i + \sum_{j=1}^{n_{EIF}} UFP_j + \sum_{k=1}^{n_{EI}} UFP_k + \sum_{l=1}^{n_{IEO}} UFP_l + \sum_{m=1}^{n_{EO}} UFP_m + \right. \\ & \left. + CFP \right) \times \left( (0,01 \times \sum_{p=1}^{14} DI_p) + 0,65 \right), \end{aligned} \quad (1)$$

где  $DFP$  – количество скорректированных функциональных точек, оценивающих объем продукта (программного продукта или информационной системы), создаваемого в результате выполнения оцениваемого ИТ-проекта, с учетом влияния общесистемных характеристик этого продукта;  $\sum_{i=1}^{n_{ILF}} UFP_i$  – сумма нескорректированных функциональных точек, характеризующая сложность структур данных, которые используются создаваемым продуктом в ходе выполнения своих функций (ILF);  $n_{ILF}$  – количество ILF;  $\sum_{j=1}^{n_{EIF}} UFP_j$  – сумма нескорректированных функциональных точек, характеризующая сложность структур данных, которые поступают из внешних по отношению к создаваемому продукту файлов (например, из базы данных) в ходе выполнения создаваемым продуктом своих функций (EIF);  $n_{EIF}$  – количество EIF;  $\sum_{k=1}^{n_{EI}} UFP_k$  – сумма нескорректированных функциональных точек, характеризующая сложность внешних интерфейсов для ввода данных, которые используются создаваемым продуктом для ввода данных от пользователя или устройства сбора информации в ходе выполнения своих функций (EI);  $n_{EI}$  – количество EI;  $\sum_{l=1}^{n_{IEO}} UFP_l$  – сумма нескорректированных функциональных точек, характеризующая сложность внешних интерфейсов для вывода данных, которые используются создаваемым продуктом для вывода результатов выполнения своих функций (EO);  $n_{EO}$  – количество EO;  $\sum_{m=1}^{n_{EQ}} UFP_m$  – сумма нескорректированных функциональных точек, характеризующая сложность внешних запросов, которые используются создаваемым продуктом для предоставления информации пользователю или



## Секция 2. Управление проектами и программами

другой системе в ответ на заданные условия поиска (EQ);  $n_{EQ}$  – количество EQ; CFP – количество дополнительных функциональных точек, описывающих функции, необходимые в ходе внедрения создаваемого продукта;  $DI_p$  – системная характеристика создаваемого продукта.

Однако данный способ не позволяет учитывать изменения требований к разрабатываемому программному продукту, которые возникают в ходе выполнения соответствующего ИТ-проекта. В то же время в модели COCOMO II подобная проблема решается путем ввода специального коэффициента BRAK, который представляет собой процент кода, выбрасываемого из ИТ-проекта в результате изменения требований [2].

Предлагается модифицировать способ расчета количества функциональных баллов программного продукта с использованием аналогичного подхода путем осуществления приближенного оценивания по формуле:

$$MDFP = DFP * \left( 1 + \frac{MREQ}{100} \right), \quad (2)$$

где MDFP – количество скорректированных функциональных точек, оценивающих объем продукта с учетом модификации и отбрасывания требований к нему в ходе выполнения ИТ-проекта; MREQ – процент модифицированных или отброшенных требований к разрабатываемому программному продукту.

При этом расчет значения показателя MREQ следует осуществлять по формуле:

$$MREQ = \left( \frac{\sum_{s=1}^{n_{req}} Req_{mod}}{n_{req}} + \frac{\sum_{s=1}^{n_{req}} Req_{del}}{n_{req}} \right) * 100\%, \quad (3)$$

где  $\sum_{s=1}^{n_{req}} Req_{mod}$  – количество требований к программному продукту, которые были модифицированы в ходе выполнения ИТ-проекта;  $n_{req}$  – количество требований к программному продукту, которые были сформулированы в ходе выполнения ИТ-проекта;  $\sum_{s=1}^{n_{req}} Req_{del}$  – количество требований к программному продукту, которые были выброшены из ИТ-проекта.

1. Functional Point Counting Practices Manual. Release 4.1.1 [Text]. – Troy: IFPLUG, 2001. – 370 p. 2. COCOMO II Model Definition Manual [Электронный ресурс] // Сайт «Center for Systems and Software Engineering». – Режим доступа: [ftp://ftp.usc.edu/pub/soft\\_engineering/COCOMOII/cocomo99.0/modelman.pdf](ftp://ftp.usc.edu/pub/soft_engineering/COCOMOII/cocomo99.0/modelman.pdf). – Заголовок с экрана.