The object of this study is the process of estimation and control of an IT project.

During the study, the problem of increasing the accuracy of estimating the efforts of an IT project to build an information system (IS) was solved. Research in this area is aimed at simplifying the original assessment methods or at applying elements of artificial intelligence to these methods. The task of the impact of the features of IS development methodology on effort assessment remains unexplored.

During the study, the method of functional point analysis was modified. The proposed modification makes it possible to increase the accuracy of estimating efforts to build an IS under the conditions of re-using its individual functions. These conditions allow the construction of new system functions by reusing a previously developed function of the same system.

The developed method was tested during the evaluation of efforts of an IT project to form the functional task "Forming and executing an individual plan for a research and teaching staff member of the department." The option of re-using one of the functions of a task during the construction of two other functions of the same task is considered. For this option, the estimate was 72 function points (the estimate using the standard method was 144 function points).

The use of the results allows us to increase the accuracy in assessing the efforts of IT projects to construct IS under the conditions of applying the methodology of reuse of previously developed system elements. This, in turn, makes it possible to improve the accuracy of estimating time costs, personnel requirements, and financial costs for the implementation of IT projects for constructing IS.

The results obtained are used to solve the task of estimating efforts during the planning of IT projects for developing information systems and their software

Keywords: IT project, effort, parametric estimation, functional point, information system, re-use

UDC 004.02

DOI: 10.15587/1729-4061.2024.301227

ESTIMATION OF IT-PROJECT EFFORTS FOR INFORMATION SYSTEM CREATION IN THE CONDITIONS OF RE-USE OF ITS FUNCTIONS

Maksym Ievlanov
Corresponding author
Doctor of Technical Sciences, Professor*
E-mail: maksym.ievlanov@nure.ua

Viktor Levykin
Doctor of Technical Sciences, Professor*

Olga Neumyvakina
PhD, Leading Engineer
Educational and Research Department**

Ihor Levykin
Doctor of Technical Sciences, Associate Professor
Department of Media Systems and Technologies**

Artem Nakonechnyi*
*Department of Information Control System**
**Kharkiv National University of Radio Electronics
Nauky ave., 14, Kharkiv, Ukraine, 61166

Received date 20.01.2024
Accepted date 27.03.2024
Published date 30.04.2024

***How to Cite**: Levykin, V., Ievlanov, M., Neumyvakina, O., Levykin, I., Nakonechny, A. (2024). Estimation of IT-project efforts for information system creation in the conditions of re-use of its functions. Eastern-European Journal of Enterprise Technologies, 2 (2 (128)), 6–19. https://doi.org/10.15587/1729-4061.2024.301227*

1. Introduction

The task to improve the accuracy of estimating the values of characteristics of IT projects (efforts, time costs, resource costs, quality) remains one of the most difficult problems in this area. The main groups of methods for assessing these characteristics were identified back in the early 2000s and have not changed fundamentally since then [1, 2]. The accuracy ranges for estimating these characteristics remain almost unchanged [3].

One of the most important characteristics of an IT project is effort. This term replaces the term "Labor costs" and is defined as the number of work units required to complete a schedule activity or component of a work breakdown structure [1, 2]. To estimate effort, man-hours, various time units (hours, days, weeks) or conventional units designated as "points" (functional points, object points, property points, story points) can be used. The use of conditional units for estimating efforts requires their additional conversion into units of labor costs or (most often) time. This conversion pays off:

a) by the use of conditional estimation methods, which are simpler compared to methods of parametric estimation of direct labor costs or time costs;

b) by a better understanding of the evaluation results by those participants in the IT project who are targeted by the methods of conditional effort evaluation.

Features of the use of models and methods for assessing the efforts of an IT project depend quite strongly on the type of result of this project. It should be recognized that the vast majority of such models and methods are focused on assessing the efforts spent on developing IT software products. The design and construction of such IT products mainly involves work on software development. However, the use of such models and methods for assessing the parameters of IT projects for constructing information systems (IS) for managing enterprises and organizations raises certain difficulties.

One of the methods most suitable for estimating the efforts of an IT project to build IS is the function point analysis (FPA) method. This method allows for assessment based on the analysis of representations of individual IS functions in the form of sets of structural units for various purposes.

Such information makes it possible to quantify both the amount of work required to implement a specific IS function and the amount of work required to implement the interfaces of this function with other IS functions, the system database, and users [4]. Although FPA was devised quite a long time ago, it is still the basis for research in the field of Software Effort Estimation [4].

However, this method is not without drawbacks. One of these shortcomings is its poor adaptation to modern approaches to the design and development of IT products. This drawback can be explained by the underlying representation of the function point method of individual IS functions as unique elements of the system being built. According to this view, the assessment of each specific IS function should be carried out separately, even in cases where this function is formed on the basis of previously implemented functions. Application of the FPA method to estimate an IT project for constructing an IS under the conditions of reuse of individual system functions will give greatly inflated results. Therefore, conducting research that allows adapting the FPA method to the features of modern approaches to constructing IS is relevant from theoretical and applied points of view.

2. Literature review and problem statement

The modern representation of the FPA method was reported in 1990. The development and standardization of this method is carried out by the International Function Point Users Group (IFPLUG). This organization published version 4.3.1 of the description of this method in 2010, which is currently valid [5].

In 2015, the authors of [6] reviewed and analyzed improvements to increase the accuracy of the FPA method that had been proposed over the previous 13 years. The search for relevant papers returned 1600 results, from which 454 primary studies were selected for analysis. Among these studies, improving the accuracy of FPA was addressed in only 18 papers. Based on the results reported in [6], it was concluded that, in general, the FPA method is quite advanced. However, in [6] a large number of problems that arise during the calculation of the functional size were noted. Based on this, the authors [6] concluded that the FPA method needs to be revised to cover possible improvements proposed by researchers.

One of the areas of this research should be the study of the features of using the FPA method in the early stages of developing IT products. It should be noted that the process of applying the standard FPA method (hereinafter referred to as FPA IFPLUG) is often too time-consuming, too expensive, or requires more knowledge than is available at the time of compiling the effort estimate [7]. The proposed early estimation methods are typically simplified compared to the standard FPA IFPLUG by eliminating one or more steps of the original method. Therefore, paper [7] reports results regarding the relative amount of effort required during the steps of the FPA IFPLUG method. The results are proposed to be used to estimate the expected savings that early assessment methods provide. However, in [7], the features of using the FPA IFPLUG method for various IT product development methodologies are not taken into account.

The second area of research in the field of improving the FPA IFPLUG method is to adapt it to the characteristics of specific stages of the life cycle of an IT product. Thus, in [8], the FPA-SDP model is proposed, which adapts the application of the FPA IFPLUG method to the features of the software development stage. Similar to the FPA-SDP model, a SCEE model was constructed in [9], based on a combination of the change impact analysis method and the FPA IFPLUG method. This model adapts the application of the above methods to the characteristics of software maintenance stage. The results make it possible to agree on descriptions of the states of software artifacts and estimate the actual size of a change request, taking into account its level of complexity. However, the application of these results in the process of describing the IS architecture and in the process of determining the functional structure of the IS is difficult.

The third area of research is to improve the FPA IFPLUG method through the use of artificial intelligence tools. The earliest works in this area are based on the use of fuzzy logic. Thus, in [10] it was proposed to increase the accuracy of the FPA IFPLUG method by using fuzzy logic to categorize individual system functions in accordance with their relative functional complexity. The results make it possible to assert that the accuracy of estimates of costs and execution time of an IT project has increased as a result of using a modified method for analyzing fuzzy function points. In [11], a named entity recognition (NER) model based on deep learning is proposed. The tests have confirmed a significant increase in the accuracy and efficiency of the modified FPA IFPLUG method.

However, the application of the results in the practice of solving the task of assessing the efforts of an IT project to construct an IS is currently extremely difficult. The main factor behind this difficulty is the lack of a widespread IT product that allows one to automate the solution to the estimation problem based on the function point method. Despite the existence of a standard for automating the execution of the FPA IFPLUG method, the development of such products is rather a scientific and applied task. An example that confirms this statement is the software described in [12].

Therefore, individual studies are focused on ways to improve the accuracy of the FPA IFPLUG method that do not require additional complex calculations and algorithms for processing large amounts of data. An example of such research is the method of analyzing function points based on a tree of function points described in [13]. This method was devised according to the IFPUG FPA steps. In [13], a description of a prototype tool is also presented, which make it possible to automate the proposed modification of the method.

Another example of such studies is work [14]. It reports an analysis of the influence of individual factors (approach to counting function points, field of activity, industry sector, relative size) on the assessment of effort using the FPA IFPUG method. Also, in [14], the factors influencing performance and the possibility of its estimation using the FPA IFPUG method are studied.

However, all the studies reviewed have a significant drawback, which is the recognition of unchanged all the premises and assumptions on which the FPA IFPUG method is based. One of these assumptions is the assumption of the uniqueness of each specific function within the framework of the IS being built. At the same time, modern methodologies for developing IS and other software systems involve the re-use of previously devised components and, in particular, system functions.

The concept of re-use in the IT field arose quite a long time ago. However, its transformation into a standard paradigm, which is practiced by most leading software providers (IBM, HP, Motorola, etc.) was completed only at the beginning of the 21st century. Software reuse is now viewed as a process in which organizations describe a set of systematic activities to create, organize, and discover reusable components for future development [15]. It is recognized that the application of this paradigm makes it possible to reduce production, purchasing, and logistics costs, as well as cover a larger number of market segments [16].

To implement the re-use paradigm, two main techniques are proposed [15]: development with re-use and development for re-use. Development with re-use involves solving the problem of classifying the component being developed and searching for re-used software components based on the results of solving this problem. Developing for reuse involves designing and constructing individual components that can be reused in the future. Development for re-use is mandatory for further development with re-use of components [15]. It is customary to distinguish the following levels of scale of software re-use: system re-use; re-use of the application; re-use of components; re-use of software objects and functions.

However, the application of the re-use paradigm in relation to such components of an IS as its individual functions or functional tasks contradicts the assumption of the uniqueness of each function within the IS. This assumption is observed in the case of developing a separate IS function for its further re-use but is violated in the case of development with re-use of previously developed functions. The consequence of this violation is excessively inflated estimates of effort and, accordingly, excessively inflated costs for completing an IT project. Therefore, there is a need to adapt the FPA IFPUG method to the peculiarities of re-using individual system elements during the design and construction of an IS.

3. The aim and objectives of the study

The purpose of our study is to modify the FPA IFPUG method to estimate the efforts of an IT project to construct an IS under the conditions of re-using its elements. As a result of this modification, the accuracy of estimating the efforts spent on completing those work packages of an IT IS project whose descriptions coincide with each other should increase. This coincidence makes it possible to design and develop such elements with repeated use of the same solutions.

To achieve the goal, the following tasks were set:

– to improve the rules for calculating the values of parameters for estimating the number of functional points of a separate IS function under conditions of re-use of its elements;

– to modify the FPA IFPUG method to estimate the efforts of an IT project to construct an IS, during which it is planned to carry out work on the re-use of individual functions of this system;

– to check the possibility of estimating the efforts spent on constructing individual IS functions, the descriptions of which coincide, using the modified method.

4. The study materials and methods

The object of our study is the process of assessing and monitoring the implementation of a project. This process is one of the typical processes of technical management of a system construction project [17]. The subject of the study is the FPA IFPUG method, which is one of the methods for parametrically estimating the efforts spent on IT projects.

The main hypothesis of this study assumes the possibility of using parametric estimation methods to estimate the efforts spent on modifying reusable elements of the system. The validity of this hypothesis is partially confirmed by the results of studies on modification of COCOMO II parametric labor cost estimation models for cases of reuse and modification of source program code.

The study is based on the FPA IFPUG method (release 4.3.1). This method involves performing the following steps [5]:

a) stage 1 – collection of available documentation;

b) stage 2 – determining the scope and boundaries of the system being estimated and determining the functional requirements of users;

c) stage 3 – determining the complexity of the estimated function by data;

d) stage 4 – determining the complexity of the estimated function based on transactions;

e) stage 5 – calculation of the number of function points;

f) stage 6 – compilation and publication of the results of calculating the number of functional points.

A diagram of the sequence of stages of the FPA IFPUG method is shown in Fig. 1 [5].

A detailed description of the FPA IFPUG method is given in [5].

In the above release, the FPA IFPUG method provides the following options for calculating function points [5]:

a) development project function point count;

b) enhancement project function point count;

c) application function point count.

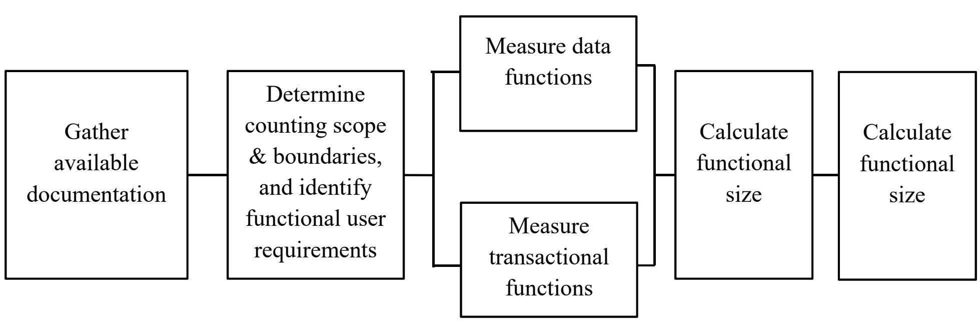

Fig. 1. Scheme of the sequence of stages of the classical function point method

The first of these options allows us to estimate the functional size of an IT project for the development and delivery of the first release of IS application software to users. The second of these options make it possible to estimate the functional size of an IT project to improve the operated (maintained) IS application software. The third of these options make it possible to assess the functional size of an IT project for the development of an IS at the early stages of its life cycle. The term “Application” in [5] refers to a holistic set of automated procedures and data that support the achievement of the business goal of developing an IS. Such a set may consist of one or more components, modules, or subsystems of the IS being built.

The FPA IFPUG method is based on the representation of the implemented system as a set of structural elements that belong to two main subgroups: data and transactions. Data can be divided into two classes: internal logical files (ILF) and external interface files (EIF). Transactions are divided into the following classes: external inputs (EI), external outputs (EO), and external queries (EQ). An example of highlighting these elements for a HR application is shown in Fig. 2 [5].

To record a quantitative assessment of the functional size of the estimated IS function, it is proposed in [5] to use Tables 1, 2.

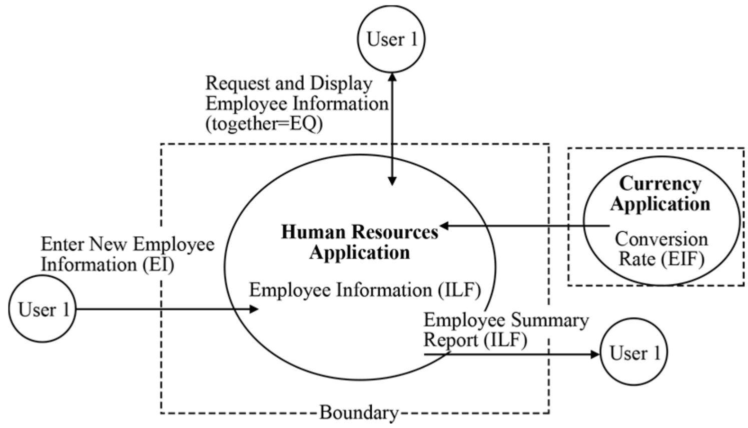

Fig. 2. Diagram of elements underlying calculations of the number of functional points of an individual information system function: EIF — External Interface File; ILF — Internal Logical File; EI — External Input; EO — External Output; EQ — External Inquiry

Table 1

Determining the functional complexity of the function being estimated

Data functions		RETs	DETs	Functional complexity
No. of entry	Name			
Internal logic files				
ILF number	ILF name	RET quantity	DET quantity	Functional complexity assessment
External interface files				
EIF number	EIF name	RET quantity	DET quantity	Functional complexity assessment
Transactional functions		FTRs	DETs	Functional complexity
No. of entry	Name			
External inputs				
EI number	EI name	FTR quantity	DET quantity	Functional complexity assessment
External outputs				
EO number	EO name	FTR quantity	DET quantity	Functional complexity assessment
External queries				
EQ number	EQ name	FTR quantity	DET quantity	Functional complexity assessment

Table 2

Calculation of the functional size of the estimated function

Function type	Quantity	Functional complexity		Complexity of everything	Full function types
ILF	Number of low- complexity ILFs	Low	x 7	Quantifying complexity	Quantifying the complexity of all ILFs
	Number of ILFs of medium complexity	Average	x 10	Quantifying complexity	
	Number of high- complexity ILFs	High	x 15	Quantifying complexity	
EIF	Number of low- complexity EIFs	Low	x 5	Quantifying complexity	Quantifying the complexity of all EIFs
	Number of EIFs of medium complexity	Average	x 7	Quantifying complexity	
	Number of high- complexity EIFs	High	x 10	Quantifying complexity	
EI	Number of low- difficulty EIs	Low	x 3	Quantifying complexity	Quantifying the complexity of all EIs
	Number of EIs of medium difficulty	Average	x 4	Quantifying complexity	
	Number of high- difficulty EIs	High	x 6	Quantifying complexity	
EO	Number oflow- difficulty EOs	Low	x 3	Quantifying complexity	Quantifying the complexity of all EOs
	Number of EOs of medium difficulty	Average	x 4	Quantifying complexity	
	Number of high- difficulty EOs	High	x 6	Quantifying complexity	
EQ	Number of low- difficulty EQs	Low	x 3	Quantifying complexity	Quantifying the difficulty of all EQs
	Number of EQs of medium difficulty	Average	x 4	Quantifying complexity	
	Number of EQs of high complexity	High	x 6	Quantifying complexity	
Full functional size					Number of function points

In Tables 1, 2, the following notations are used [5]:

a) RET – record element type (Record Element Type, user recognizable sub-group of data element types within a data function);

b) DET – data element type (Data Element Type, unique, user recognizable, non-repeated attribute);

c) FTR – file type referenced (File Type Referenced, data function read and/or maintained by a transactional function).

In the corresponding columns of Table 1, the number of RETs and DETs for each individual ILF or EIF should be specified, as well as the number of FTRs and DETs for each individual EI, EO, or EQ of the function being estimated. Detailed rules for determining the number of RET, DET, and FTR are given in [5].

In the FPA IFPUG method, the facts of changes in elements during the design and development of IS are taken into account only in the option of calculating the function points of an improvement project (Enhancement project function point count). This option involves calculating the functional size of an IT project to improve the operated (maintained) IS application software using the following formula [5]:

$$aEFP=[(ADD+CHGA+CFP)*VAFA]+(DEL*VAFB), \quad (1)$$

where *aEFP* is the adjusted functional size of the IT project; *ADD* – the size of the functions that are planned to be added during the IT project; *CHGA* – the size of the functions that are planned to change during the IT project; *CFP* – the size of functions that should be converted during the IT project; *VAFA* is the IS size adjustment factor after completion of the IT project; *DEL* – size of functions that are planned to be removed during the IT project; *VAFB* is an adjustment factor for the size of functions planned to be removed before the start of an IT project.

However, for this option, the calculation of the sizes of functions that should be adjusted in accordance with (1) is performed according to the same sequence of calculations as the calculation of the sizes of functions in other variants of IT projects. In this case, each IS function is considered separately from other IS functions when calculating its size. This limitation seriously distorts the accuracy of estimating the effort spent on IT projects that involve re-using previously developed system elements. The term “system element” here should be understood as individual IS functions or individual operations on data, the use of which makes it possible to describe a separate IS function. The essence of this distortion is that all data structures that are involved in determining the values of the EIF, ILF, EI, EO, and EQ elements are considered to be built from scratch. Constructing these data structures involves doing a certain amount of work. At the same time, the re-use of existing data structures, the descriptions of which coincide as closely as possible with the descriptions of the data structures of the IS function being estimated, can significantly reduce the amount of this work.

Thus, the main idea of the proposed modification of the FPA IFPUG method should be to improve the rules for calculating the values of the elements EIF, ILF, EI, EO, and EQ under the conditions of re-using the corresponding data structures.

5. Results of modification of the FPA IFPUG method taking into account the features of re-using information system elements

5. 1. Improving the rules for calculating the values of parameters for estimating the number of functional points of an individual information system function

As noted, the FPA IFPUG method is based on the assumption of constructing each IS function separately. This does not take into account the similarity of the constructed IS function with previously designed and developed functions of the same system. This assumption can be formally described as a representation of descriptions of all IS functions in the form of an unordered set:

$$F_{IS}=\left(f_1,f_2,\ldots,f_i,\ldots,f_n\right), \quad (2)$$

where f_i is a description of the *i*-th function of the constructed IS in the form of a set of structured sets; *i* – IS function identifier, $i=1, \ldots, n$; *n* – number of functions in the constructed IS.

The use of a methodology for re-using individual functions during the planning and implementation of an IT project for constructing an IS requires an initial assessment and implementation of functions that will later be re-used during the construction of other functions of this IS. To fulfill this requirement, we introduce a partial order relation on set (2):

$$R^{re\_use}\left(F_{IS}\right)=\left(f_a \preceq f_b \mid f_a,f_b\in F_{IS}, f_a\rightarrow f_b\right). \quad (3)$$

This relation represents at the formal level the case of re-using the function f_a during the construction of a new function f_b as the implication of a multiple description of the function f_a into a multiple description of the function f_b.

The use of a methodology for re-using individual functions during the construction of an IS allows us to represent the entire set of functions of this IS in the form of a category of structured sets:

$$L_{F_{IS}}^{re\_use}=\left[Ob=F_{IS}^{re\_use};Mon=\left(R^{re\_use}\left(f_a,f_b\right)\right)\right], \quad (4)$$

where $L_{F_{IS}}^{re\_use}$ is the designation of the category of structured descriptions of the functions of the constructed IS; *Ob* – designation of a class of objects of a category $L_{F_{IS}}^{re\_use}$; $F_{IS}^{re\_use}$ – a set of descriptions of the functions of the constructed IS with introduced partial order relations; *Mon* is a class of monomorphisms of the category $L_{F_{IS}}^{re\_use}$ that can be considered a generalization of implications between the descriptions of the functions f_a and f_b.

Taking into account the introduced partial order relations ($R^{re\_use}(f_a,f_b)$), the set $F_{IS}^{re\_use}$ will take the following form:

$$F_{IS}^{re\_use}=\left(\begin{array}{c} <f_1,\left(f_{11},f_{12},\ldots,f_{1j},\ldots,f_{1m}\right)>,\\ <f_2,\left(f_{21},f_{22},\ldots,f_{2j},\ldots,f_{2m}\right)>,\ldots,\\ <f_i,\left(f_{i1},f_{i2},\ldots,f_{ij},\ldots,f_{im}\right)>,\ldots,\\ <f_k,\left(f_{k1},f_{k2},\ldots,f_{kj},\ldots,f_{km}\right)>,\\ f_{k+1},\ldots,f_n \end{array}\right), \quad (5)$$

where $(f_{i1}, f_{i2}, \ldots, f_{ij}, \ldots, f_{im})$ is a subset of functions that are built as a result of re-using the function f_i, $i=1, \ldots, k$; *k* – the number of functions that are planned to be reused during the design and development of other IS functions; m – the maximum number of functions that are built as a result of re-using the function f_i, i=1, ..., k; f_{k+1}, ..., f_n – functions that cannot be built by re-using previously developed IS functions.

The use of set (5) during the planning of an IT project for constructing an IS requires performing operations to re-use the function f_i during the construction of each function from the subset (f_{i1}, f_{i2}, ..., f_{ij}, ..., f_{im}) immediately after constructing the function f_i. This requirement is due to the desire to maximize the use of the experience of the IT project performer, gained during the construction of the f_i function.

The use of a category-theoretic representation of IS (4) and a set of IS functions (5) during the planning of an IT project for constructing an IS allows us to revise the rule for calculating the functional size of individual functions of a given system.

For the original function f_i, the decision to re-use which was made during the formation of the set F_{IS}^{re-use}, the rule for calculating the number of uncorrected functional points as the value of the "Full functional size" cell of Table 2 is determined by the following expression [5, 18]:

$$FP_{f_i} = \sum_{a=1}^{n_{ILF}} FC_{ILF_{ia}} + \sum_{b=1}^{n_{EIF}} FC_{EIF_{ib}} + \sum_{c=1}^{n_{EI}} FC_{EI_{ic}} + \sum_{d=1}^{n_{EO}} FC_{EO_{id}} + \sum_{e=1}^{n_{EQ}} FC_{EQ_{ie}}, \quad (6)$$

where FP_{f_i} is the number of uncorrected functional points as an estimate of the value of the "Full functional size" cell for the function f_i; a – numeric identifier ILF_{ia} participating in the description of function f_i; n_{ILF} – number of ILF_{ia} participating in the description of function f_i; $FC_{ILF_{ia}}$ – quantitative assessment of the functional complexity of ILF_{ia} involved in the description of the function f_i; b – numeric identifier of EIF_{ib} participating in the description of function f_i; $FC_{EIF_{ib}}$ – number of EIF_{ib} participating in the description of function f_i; $FC_{EI_{ic}}$ – quantitative assessment of the functional complexity of EIF_{ib} involved in the description of the function f_i; c – numeric identifier of EI_{ic} participating in the description of function f_i; n_{EI} – number of EI_{ic} participating in the description of function f_i; $FC_{EI_{ic}}$ – quantitative assessment of the functional complexity EI_{ic} involved in the description of the function f_i; d – numeric identifier of the EO_{id} participating in the description of the function f_i; n_{EO} – number of EO_{id} participating in the description of function f_i; $FC_{EO_{id}}$ – quantitative assessment of the functional complexity EO_{id} involved in the description of the function f_i; e – numeric identifier of EQ_{ie} participating in the description of function f_i; n_{EQ} – number of EQ_{ie} participating in the description of function f_i; $FC_{EQ_{ie}}$ – quantitative assessment of the functional complexity of EQ_{ie} involved in the description of the function f_i.

The rules for calculating the $FC_{ILF_{ia}}$, $FC_{EIF_{ib}}$, $FC_{EI_{ic}}$, $FC_{EO_{id}}$ and $FC_{EQ_{ie}}$ values are described, according to Table 1, by the following expressions:

$$FC_{ILF_{ia}} = f\left(\left\langle n_{RET_{ia}}, n_{DET_{ia}} \right\rangle\right),\quad FC_{EIF_{ib}} = f\left(\left\langle n_{RET_{ib}}, n_{DET_{ib}} \right\rangle\right),$$
$$FC_{EI_{ic}} = f\left(\left\langle n_{FTR_{ic}}, n_{DET_{ic}} \right\rangle\right),\quad FC_{EO_{id}} = f\left(\left\langle n_{FTR_{id}}, n_{DET_{id}} \right\rangle\right),\quad (7)$$
$$FC_{EQ_{ie}} = f\left(\left\langle n_{FTR_{ie}}, n_{DET_{ie}} \right\rangle\right),$$

where $n_{RET_{ia}}$ is the number of RET elements present in the ILF_{ia} description; $n_{DET_{ia}}$ – the number of DET elements present in the LF_{ia} description; $n_{RET_{ib}}$ – the number of RET elements present in the EIF_{ib} description; $n_{DET_{ib}}$ – the number of DET elements present in the EIF_{ib} description; $n_{FTR_{ic}}$ – number of FTR elements present in the description of EI_{ic}; $n_{DET_{ic}}$ – the number of DET elements present in the EI_{ic} description; $n_{FTR_{id}}$ – the number of FTR elements present in the EO_{id} description; $n_{DET_{id}}$ – the number of DET elements present in the EO_{id} description; $n_{FTR_{ie}}$ – number of FTR elements present in the EQ_{ie} description; $n_{DET_{ie}}$ – the number of DET elements present in the EQ_{ie} description; $f(<\bullet>)$ is a function that matches a tuple of values of structural elements with an assessment of the functional complexity of the corresponding ILF_{ia}, EIF_{ib}, EI_{ic}, EO_{id} or EQ_{ie}.

Then for the function f_{ij}, which during the formation of the set F_{IS}^{re-use} is planned to be constructed by re-using the function f_i, the rule for calculating the number of uncorrected functional points as the value of the "Full functional size" cell of Table 2 is proposed to be described by the following expression:

$$FP_{f_{ij}} = \sum_{a=1}^{n_{ILF}} FC_{ILF_{ija}} + \sum_{b=1}^{n_{EIF}} FC_{EIF_{ijb}} + \sum_{c=1}^{n_{EI}} FC_{EI_{ijc}} + \sum_{d=1}^{n_{EQ}} FC_{EO_{ijd}} + \sum_{e=1}^{n_{EQ}} FC_{EQ_{ije}}. \quad (8)$$

The rules for calculating the values of $FC_{ILF_{ija}}$, $FC_{EIF_{ijb}}$, $FC_{EI_{ijc}}$, $FC_{EO_{ijd}}$ and $FC_{EQ_{ije}}$ are described, according to Table 1, by the following expressions:

$$FC_{ILF_{ija}} = f\left(\left\langle n_{RET_{ija}-RET_{ia}}, n_{DET_{ija}-DET_{ia}} \right\rangle\right),$$
$$FC_{EIF_{ijb}} = f\left(\left\langle n_{RET_{ijb}-RET_{ib}}, n_{DET_{ijb}-DET_{ib}} \right\rangle\right),$$
$$FC_{EI_{ijc}} = f\left(\left\langle n_{FTR_{ijc}-FTR_{ic}}, n_{DET_{ijc}-DET_{ic}} \right\rangle\right),\quad (9)$$
$$FC_{EO_{ijd}} = f\left(\left\langle n_{FTR_{ijd}-FTR_{id}}, n_{DET_{ijd}-DET_{id}} \right\rangle\right),$$
$$FC_{EQ_{ije}} = f\left(\left\langle n_{FTR_{ije}-FTR_{ie}}, n_{DET_{ije}-DET_{ie}} \right\rangle\right),$$

where $n_{RET_{ija}-RET_{ia}}$ is the number of RET elements of sets (RET_{ija}-RET_{ia}) difference from the ILF_{ija} description;

$n_{DET_{ija}-DET_{ia}}$– the number of elements DET of sets (DET_{ija}-DET_{ia}) difference from the description of ILF_{ija};

$n_{RET_{ijb}-RET_{ib}}$ – the number of RET elements of sets (RET_{ijb}-RET_{ib}) difference from the EIF_{ijb} description;

$n_{DET_{ijb}-DET_{ib}}$ – the number of elements DET of sets (DET_{ijb}-DET_{ib}) difference from the description of EIF_{ijb};

$n_{FTR_{ijc}-FTR_{ic}}$ – the number of elements FTR of sets (FTR_{ijd}-FTR_{id}) difference from the EI_{ijc} description;

$n_{DET_{ijc}-DET_{ic}}$ – the number of elements DET of sets (DET_{ijc}-DET_{ic}) difference from the EI_{ijc} description;

$n_{FTR_{ijd}-FTR_{id}}$ – the number of FTR elements of sets (FTR_{ijd}-FTR_{id}) difference from the EO_{ijd} description;

$n_{DET_{id}}$ – the number of elements DET of sets (DET_{ijb}-DET_{ib}) difference from the EO_{ijd} description;

$n_{FTR_{ijd}-FTR_{id}}$ – the number of FTR elements of sets (FTR_{ijd}-FTR_{id}) difference from the EQ_{ije} description;

$n_{DET_{ije}-DET_{ie}}$ – the number of elements DET of sets (DET_{ije}-DET_{ie}) difference from the description of EQ_{ije}; $f(<\bullet>)$ is a function that matches a tuple of values of structural elements with an assessment of the functional complexity of corresponding ILF_{ija}, EIF_{ijb}, EI_{ijc}, EO_{ijd} or EQ_{ije}.

However, the proposed technique for calculating the number of function points does not reflect the amount of effort to re-use function f_i caused by syntactic differences between the domains of functions f_i and f_{ij}. To estimate these efforts, it is proposed to modify expression (8) as follows:

$$FP_{f_{ij}} = \left(\begin{array}{l} \sum_{a=1}^{n_{ILF}} FC_{ILF_{ija}} + \sum_{b=1}^{n_{EIF}} FC_{EIF_{ijb}} + \sum_{c=1}^{n_{EI}} FC_{EI_{ijc}} \\ + \sum_{d=1}^{n_{EQ}} FC_{EO_{ijd}} + \sum_{e=1}^{n_{EQ}} FC_{EQ_{ije}} \end{array} \right) + \\ + FP_{f_i} \times \frac{\left| Name_{ij} - Name_i \right|}{\left| Name_i \right|}, \quad (10)$$

where $Name_{ij}$ is the set of elements present in the name of the function f_{ij}; $Name_i$ – set of elements present in the name of the function f_i.

It is proposed to form the sets of elements $Name_{ij}$ and $Name_i$ by stemming the names of the functions f_{ij} and f_i with subsequent removal of stop words.

5. 2. Modifying the FPA IFPLUG method

Our results make it possible to modify the FPA IFPUG method for assessing the efforts of an IT project to construct an IS in the context of reusing its elements. The modified method involves performing the following steps:

a) stage 1 – collection of available documentation;

b) stage 2 – determining the scope and boundaries of the system being estimated and determining the functional requirements of users;

c) stage 3 – making decisions on the advisability of re-using individual functions and forming a variety of descriptions of system functions, taking into account the re-use of individual functions;

d) stage 4 – determining the complexity of the estimated function by data;

e) stage 5 – determining the complexity of the estimated function based on transactions;

f) stage 6 – calculation of the number of function points;

g) stage 7 – compilation and publication of the results of calculating the number of functional points.

The added Stage 3 is proposed to be considered as a sequence of the following actions:

Step 3. 1. Formation of solution options on the possibility of re-using individual functions of the constructed IS.

Step 3. 2. Selection of functions f_i, whose re-use is considered appropriate within the framework of an IT project for constructing an IS.

Step 3. 3. For each function f_i selected at Step 3. 2, the formation of subsets of functions ($f_{i1}, f_{i2}, ..., f_{ij}, ..., f_{im}$), which are built as a result of re-using the function f_i.

Step 3. 4. Formation of a set of descriptions of system functions $F_{IS}^{re\_use}$, for which it is necessary to quantify the efforts to create them.

In addition to adding Stage 3 in Stages 4–6 of the modified method, it is proposed to use expressions (10) and (9) instead of expressions (6) and (7), respectively, for estimating functions from subsets ($f_{i1}, f_{i2}, ..., f_{ij}, ..., f_{im}$).

5. 3. Checking the possibility of assessing the efforts of an IT project under conditions of re-use of individual functions of the information system

To verify the results obtained, it is proposed to use the data published in [19] from the IT project for the development of the functional task "Forming and eexecuting an individual plan (IP) for a research and teaching employee of the department." The goal of this IT project was to construct a functional task that allows the user to automate work on the formation and subsequent changes of the document "IP of a scientific and pedagogical employee of the department." This IT project was carried out with the active participation of student from gr. UPGITm-20-1 Elizaveta Anatolyevna Kuzma.

During the implementation of Stages 1 and 2 of the modified method, the following was performed:

a) an operational description of the architecture of the functional task "Forming and executing IP for a scientific and pedagogical employee of the department" in the form of a data flow diagram;

b) a structural description of the architecture of the functional task "Forming and executing IP for a scientific and pedagogical employee of the department" in the form of an "essence – connection" diagram.

The operational description of the functional task architecture is given in Table 3. As numerical numbers, Table 1 lists the numbers of jobs, input and output flows that were generated by the AllFusion Process Modeler CASE tool during the construction of a data flow diagram [19].

A structural description of the architecture of the functional task is shown in Fig. 3 [19]. The set of descriptions of essences displayed in Fig. 3 is given in Table 4 [19].

Table 3

Operational description of the functional task "Forming and executing an individual plan for a scientific and pedagogical worker of the department" (based on a data flow diagram)

Task		Input flow		Output flow	
No.	Title	No.	Title	No.	Title
1	2	3	4	5	6
1	Conversion of the section «Educational work»	1	Lecturer's teaching load for the academic year	2	Information from the section of the individual plan (IP) "Academic work"
2	Formation of the «Scientific Work» section	2	Lecturer information	3	Information from the IP section «Scientific work»
		3	Information about work planned for execution		
		5	Information about recommended work		
		8	Information from the IP section «Scientific work»		
		12	Remaining hours		

Continuation of Table 3

1	2	3	4	5	6
3	Formation of the section «Methodological work»	2	Lecturer information	4	Information from the IP section «Methodological work»
		3	Information about work planned for execution		
		5	Information about recommended work		
		9	Information from the IP section «Methodological work»		
		12	Remaining hours		
4	Formation of the «Organizational work» section	2	Lecturer information	5	Information from the IP section «Organizational work»
		3	Information about work planned for execution		
		5	Information about recommended work		
		10	Information from the IP section «Organizational work»		
		12	Remaining hours		
5	Formation of a list of positions and long-term assignments	4	Information about positions and long-term assignments	6	Information from the IP section «List of positions and long-term assignments»
		11	Information from the IP section «List of positions and long-term assignments»		
6	Formation of a list of recommended works	5	Information about recommended work	1	Information about recommended work
7	Formation and maintenance of regulatory and reference information on key performance indicators (KPIs)	6	Information about key KPIs of the department	7	Information about key KPIs of the department
8	Formation of KPI of the lecturer and part of the KPI of the department	8	Information from the IP section «Scientific work»	9	Information about the KPI of the lecturer and parts of the KPI of the department
9	Formation of a summary table for the academic year	9	Information from the IP section «Methodological work»	8	Information on the number of hours by IP sections
		7	Information from the IP section «Educational work»		
		8	Information from the IP section «Scientific work»		
		10	Information from the IP section «Organizational work»		
10	Formation of the output document «IP»	9	Information from the IP section «Methodological work»	10	IP
		7	Information from the IP section «Educational work»		
		8	Information from the IP section «Scientific work»		
		10	Information from the IP section «Organizational work»		
		11	Information from the IP section «List of positions and long-term assignments»		

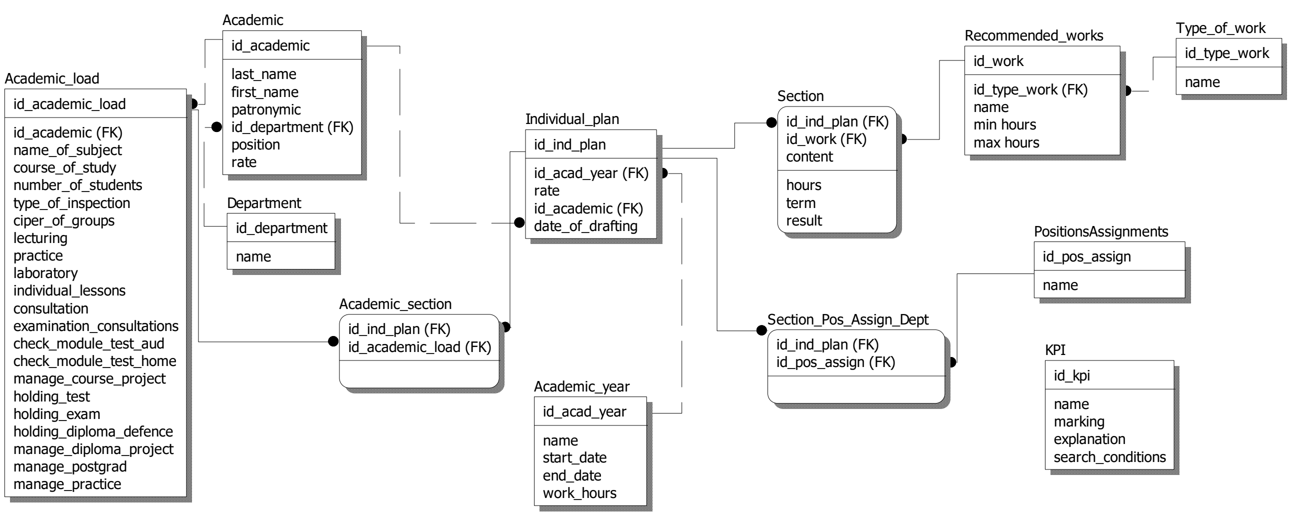

Fig. 3. Structural description of the functional task "Forming and executing an individual plan for a scientific and pedagogical worker of the department" in the form of an "essence — connection" diagram

Table 4

Set of descriptions of functional task essences

Numeric identifier	Title
1	Academic_load
2	Academic
3	Department
4	Individual_plan
5	Academic_section
6	Academic_year
7	Section
8	Recommended_works
9	Type_of_work
10	Section_Pos_Assign_Dept
11	PositionsAssignments
12	KPI

Table 4 gives as numeric identifiers the numbers of essences generated using the AllFusion ERwin Data Modeler CASE tool during the construction of essence-relationship diagram shown in Fig. 3 and imported into the AllFusion Process Modeler CASE tool to link the essence-relationship diagram with the task data flow diagram [19].

The results of determining the parameters of individual task functions, as well as the identifiers of essences used to describe the work, input, and output data flows, are given in Table 5.

In Table 3, the subset identifier takes the following values: 1 – subset of the data flow diagram operation description; 2 – subset of the description of the input flows of the data flow diagram; 3 – a subset of the description of the output flows of the data flow diagram. The contents of the "Element number" cells in Table 3 corresponds to task numbers (for subset 1), numbers of input streams (for subset 2), and numbers of output streams (for subset 3), taken from Table 1. Their numbers from table 2 were used as essence identifiers [19].

During Stage 3 of the modified method, clusters of functions were identified whose structural descriptions were at least partially similar to each other. The result of this stage is a dendrogram of functions as configuration elements of the task, shown in Fig. 4 [19].

Table 5

Initial information for the application of the FPA IFPLUG method

Function No.	Subset ID	Item No./Method parameter designation	Set of essence identifiers that describe an item
1	2	3	4
1	1	1/ILF	{1, 2, 3, 4, 5, 6}
	2	1/EIF	{1, 2, 3}
	3	2/EIF	{1, 2, 4, 5, 6}
2	1	2/ILF	{1, 2, 3, 4, 5, 6, 7, 8, 9}
	2	2/EIF	{2, 3}
		3/EI	{2, 4, 6, 7, 8, 9}
		5/EIF	{8, 9}
		8/EIF	{1, 2, 4, 6, 7, 8, 9}
		12/EIF	{1, 2, 4, 5, 6, 7}
	3	3/EIF	{2, 4, 6, 7, 8, 9}
3	1	3/ILF	{1, 2, 3, 4, 5, 6, 7, 8, 9}
	2	2/EIF	{2, 3}
		3/EI	{2, 4, 6, 7, 8, 9}
		5/EIF	{8, 9}
		9/EIF	{2, 4, 6, 7, 8, 9}
		12/EIF	{1, 2, 4, 5, 6, 7}
	3	4/EIF	{2, 4, 6, 7, 8, 9}
4	1	4/ILF	{1, 2, 3, 4, 5, 6, 7, 8, 9}
	2	2/EIF	{2, 3}
		3/EI	{2, 4, 6, 7, 8, 9}
		5/EIF	{8, 9}
		10/EIF	{2, 4, 6, 7, 8, 9}
		12/EIF	{1, 2, 4, 5, 6, 7}
	3	5/EIF	{2, 4, 6, 7, 8, 9}
5	1	5/ILF	{2, 4, 6, 7, 10, 11}
	2	4/EI	{2, 4, 6, 7, 10, 11}
		11/EIF	{2, 4, 6, 7, 10, 11}
	3	6/EIF	{2, 4, 6, 7, 10, 11}
6	1	6/ILF	{8, 9}
	2	5/EI	{8, 9}
	3	1/EIF	{8, 9}
7	1	7/ILF	{12}
	2	6/EI	{12}
	3	7/EIF	{12}

Continuation of Table 5

1	2	3	4
8	1	8/ILF	{1, 2, 4, 5, 6, 7, 8, 9, 12}
	2	8/EIF	{1, 2, 4, 6, 7, 8, 9}
	3	9/EO	{1, 2, 4, 5, 6, 7, 12}
9	1	9/ILF	{1, 2, 4, 5, 6, 7, 8, 9}
	2	7/EIF	{1, 2, 4, 5, 6}
		8/ EIF	{1, 2, 4, 6, 7, 8, 9}
		9/EIF	{2, 4, 6, 7, 8, 9}
		10/EIF	{2, 4, 6, 7, 8, 9}
	3	8/EO	{1, 2, 4, 5, 6, 7}
10	1	10/ILF	{1, 2, 4, 5, 6, 7, 8, 9, 10, 11}
	2	7/EIF	{1, 2, 4, 5, 6}
		8/EIF	{1, 2, 4, 6, 7, 8, 9}
		9/EIF	{1, 4, 6, 7, 8, 9}
		10/EIF	{2, 4, 6, 7, 8, 9}
		11/EIF	{2, 4, 6, 7, 10, 11}
	3	10/EO	{1, 2, 4, 5, 6, 7, 8, 9, 10, 11}

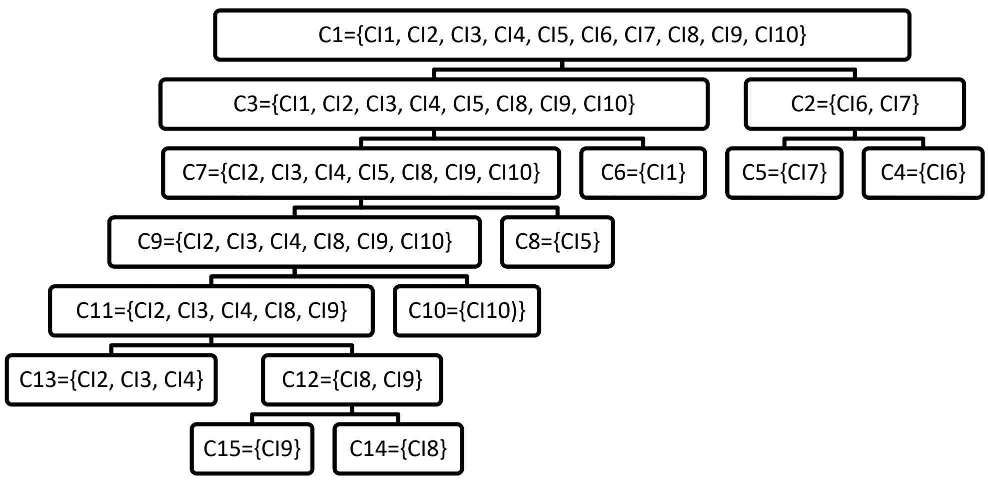

Fig. 4. Dendrogram of clusters of configuration elements of the task "Forming and executing an individual plan for a scientific and pedagogical employee of the department"

The numbers of configuration items (CI) shown in Fig. 4 coincide with the task numbers in Table 3 and function numbers in Table 5. A detailed description of the method for generating this dendrogram is reported in [19].

Dendrogram analysis made it possible to identify a subset of functions f_2 "Formation of the "Scientific Work" section", f_3 "Formation of the "Methodological Work" section" and f_4 "Formation of the "Organizational Work" section" for making a decision on re-use. The structural descriptions and names of these functions coincide with each other to the maximum extent. This option is proposed to be described by the tuple $<f_2,(f_3, f_4)>$. According to this description, function f_2 will be re-used to construct functions f_3 and f_4.

First, the number of function points was calculated, which estimates the effort to construct function f_2. For this purpose, the tabular forms for representing the progress and results of calculations recommended in [5] were used (Tables 6, 7).

Then the number of function points was calculated, which estimates the effort to construct function f_3. First, the characteristics of the ILF, EIF, and EI parameters were determined for the "Formation of the "Scientific Work" section" function. The results of this operation are given in Table 8.

Table 6

Determining the characteristics of ILF, EIF, and EI parameters for the function "Formation of the "Scientific Work" section"

Data functions		RETs	DETs	Functional complexity
No. of entry	Title			
Internal logic files				
1	Formation of the «Scientific Work» section	7	60	High
External interface files				
1	Lecturer information	2	9	Low
2	Information about recommended work	2	7	Low
3	Information from the IP section «Scientific work»	6	51	High
4	Remaining hours	5	48	Average
Transactional functions		FTRs	DETs	Functional complexity
No. of entry	Title			
External inputs				
1	Information about work planned for execution	2	111	High

Table 7

Calculation of the functional size of the project for developing the function "Formation of the "Scientific Work" section"

Function type	Quantity	Functional complexity		Complexity of everything	Full function types
ILF	0	Low	x 7	0	15
	0	Average	x 10	0	
	1	High	x 15	15	
EIF	2	Low	x 5	10	27
	1	Average	x 7	7	
	1	High	x 10	10	
EI	0	Low	x 3	0	6
	0	Average	x 4	0	
	1	High	x 6	6	
Full functional size					48

Table 8

Determining the characteristics of the ILF, EIF, and EI parameters for the function "Formation of the "Methodological work" section"

Data functions		RETs	DETs	Functional complexity
No. of entry	Title			
Internal logic files				
1	Formation of the section «Methodological work»	7	60	High
External interface files				
1	Lecturer information	2	9	Low
2	Information about recommended work	2	7	Low
3	Information from the IP section «Methodological work»	6	51	High
4	Remaining hours	5	48	Average
Transactional functions		FTRs	DETs	Functional complexity
No. of entry	Title			
External inputs				
1	Information about work planned for execution	2	111	High

As follows from expression (9), it is necessary to first determine the number of elements in the differences of sets ($RET_{ija}-RET_{ia}$), ($DET_{ija}-DET_{ia}$), ($RET_{ijb}-RET_{ib}$), ($DET_{ijb}-DET_{ib}$), ($FTR_{ijc}-FTR_{ic}$) and ($DET_{ijc}-DET_{ic}$). However, the number of elements in the differences of sets ($RET_{ija}-RET_{ia}$), ($RET_{ijb}-RET_{ib}$) and ($FTR_{ijc}-FTR_{ic}$) for the functions $f_{ij}=f_3$ and $f_i=f_2$ is equal to 0, which means that the values of $FC_{ILF_{ija}}$, $FC_{EIF_{ijb}}$ and $FC_{EI_{ijc}}$ for the function f_3 are equal to zero. Consequently, the first part of expression (10) is also equal to 0. This means that the function f_3 at the level of its structural description by highlighted in Fig. 3 essences completely coincides with a similar structural description of the function f_2.

To calculate a value of the second part of expression (10), it is necessary to perform stemming of the names of functions f_3 and f_2 with subsequent removal of stop words. The results of these operations using Porter's stemmer are given in Table 9.

Table 9

Results of preprocessing of function names f_3 and f_2

Pre-processing stage	Title of function f_3	Title of function f_2
Original frame name	Formation of the section «Methodical work»	Formation of the section «Scientific work»
Results of using Porter's stemmer	Format of the section «Method work»	Format of the section «Scientif work»
Stopword removal results	Format section «Method work»	Format section «Scientif work»

The results given in Table 9 make it possible to determine the following values of the elements of the second part of expression (10):

a) $Name_3=(Format,section,Method,work)$;

b) $Name_2=(Format,section,Scientif,work)$;

c) $|Name_2|=4$;

d) $|Name_3-Name_2|=1$.

Thus, the final estimate of the effort to construct function f_3, taking into account the re-use of function f_2, is:

$$FP_{f_3}=0+48\times\frac{1}{4}=12 \text{ (function points)}. \quad (11)$$

It should be noted that if the original FPA IFPLUG method were applied, the value of the effort estimate for constructing function f_3 would be defined as $FP_{f_3}=48$ function points. This estimate for the case of re-using function f_2 during the construction of function f_2 is too high.

Then the number of function points was calculated, which estimates the effort to construct function f_4. First, the characteristics of the ILF, EIF, and EI parameters were determined for the "Formation of the "Organizational Work" section" function. The results of this operation are given in Table 10.

Table 10

Determining the characteristics of ILF, EIF, and EI parameters for the function "Formation of the "Organizational work" section

Data functions		RETs	DETs	Functional complexity
No. of entry	Title			
Internal logic files				
1	Formation of the section «Organizational work»	7	60	High
External interface files				
1	Lecturer information	2	9	Low
2	Information about recommended work	2	7	Low
3	Information from the IP section «Organizational work»	6	51	High
4	Remaining hours	5	48	Average
Transactional functions		FTRs	DETs	Functional complexity
No. of entry	Title			
External inputs				
1	Information about work planned for execution	2	111	High

By analogy with the calculations for function f_3, the final estimate of the effort to construct function f_4, taking into account the re-use of function f_2, is $FP_{f_4}=12$ function points. If the original FPA IFPLUG method were applied, the effort estimate value for constructing function f_4 would be defined as $FP_{f_4}=48$ function points.

Thus, the total estimate of the effort to construct functions f_2, f_3 and f_4 is equal to:

a) in the case of using the modified method for the tuple $<f_2,(f_3, f_4)>$ 48+12+12=72 function points;

b) in the case of using the original FPA IFPLUG method 48+48+48=144 function points.

It can be argued that the use of the modified method in this case increased the accuracy of estimating the efforts to construct functions f_2, f_3 and f_4 by a factor of two.

6. Discussion of results of adapting the FPA IFPUG method to the peculiarities of re-using information system elements

Existing features of the re-use of individual IS functions have determined the direction of improving the rules for calculating the values of parameters for estimating the number of functional points of a separate IS function under the conditions of re-using its elements. The improved rules are represented by expressions (9), (10). The results obtained are explained by taking into account during the assessment the coincidences and differences in the structural descriptions and names of the function f_{ij} and the re-used function f_i.

Our results made it possible to modify the FPA IFPUG method to estimate the efforts of an IT project to construct IS under the conditions of re-using its elements. The essence of the proposed modification is as follows:

a) a new stage has been added (Stage 3), within which a selection and decision is made on the advisability of re-using a separate IS function;

b) the sets of rules for calculating the functional complexity of the estimated function in Stages 4 and 5 of the method have been expanded by adding rules described by expression (9);

c) the set of rules for calculating the full functional size of the estimated function in Stage 6 of the method has been expanded by adding the rule described by expression (10).

As a result of this modification, it became possible to increase the accuracy of quantifying the efforts of IT projects to construct IS. This increase in accuracy is explained by the timely selection of re-used functions and taking into account the results of their re-use during the construction of other IS functions.

The modified method was tested in the course of assessing the efforts to complete an IT project for constructing the functional task "Forming and executing an individual plan for a scientific and pedagogical employee of the department." During the estimation, it was proposed to re-use the "Formation of the "Scientific Work" section" function in the course of constructing the following functions:

a) "Formation of the section "Methodological work";

b) "Formation of the "Organizational work" section."

Taking into account the proposed re-use solution, estimates of the effort to construct the corresponding functions were calculated using the modified method and the original FPA IFPLUG method. It is shown that the use of the original FPA IFPLUG method for estimating efforts leads to a significant overestimation. For the case considered, the estimates, compared with the estimates obtained as a result of applying the modified method, turned out to be twice as high. The result is explained by the complete coincidence of structural descriptions and minimal differences in the names of the estimated functions.

It should be noted that the ability to estimate the efforts to adapt previously developed software to implement an IS element is also inherent in other methods of parametric estimation of an IT project. This feature is most fully implemented in the COCOMO II set of parametric models. However, in COCOMO II models, to assess the efforts to adapt re-used software, it is necessary to use as initial values [3]:

a) the number of thousands of lines of source code that must be constructed from scratch;

b) the number of thousands of lines of source code that is subject to adaptation;

c) the percentage of source code subject to automatic translation or compilation;

d) the value of the integrated indicator, which estimates the percentage of modification of the structure and content, as well as the costs of integrating the reused code;

e) estimating the costs of understanding the re-used code;

f) the value of the estimation and assimilation indicator of the re-used code;

g) the value of the indicator of the level of disconnection of the project team.

Unlike COCMO II models, the proposed modified method allows the use of not only logical but also conceptual descriptions of the structural and transactional elements of the assessed IS functions for assessment. This approach greatly simplifies the solution of the problem of estimating the efforts to complete IT projects for constructing IS. At the same time, the accuracy of both the COCOMO II models and the original FPA IFPLUG method at the early stages of such IT projects is approximately the same due to the lack of accurate information about the parameters of the IT project. In a number of cases, at the early stages of an IT project, the accuracy of estimation using the FPA IFPLUG method and its proposed modification may be even higher than the accuracy of estimation based on COCOMO II models. The reason for this is the need to spend less effort on clarifying the descriptions of the structural and transactional elements of the estimated function than on estimating the size of the source program code.

The modification carried out is based on the assumption that it is possible to reuse the data structures of a single function during the construction of other functions, the descriptions of which fully or partially coincide with the descriptions of this function. The proposed solution does not depend on the type of IT project and can be considered as a behavioral pattern for assessing efforts to construct an IS under the conditions of re-using its elements.

The main limitations on the application of our study results in further applied work and theoretical research are:

– using data flow diagrams and "essence-relationship" diagrams as the main sources of information about the functions and data structures of the IS constructed;

– using the results of solving the problem of clustering configuration elements and, in particular, the method proposed in [19] for determining the distance between configuration elements to make a decision on the advisability of re-using the problem function;

– the need to construct, at least at a conceptual level, an "essence-relationship" diagram for the analyzed IT product.

The main shortcoming of our study is its dependence on the completeness of descriptions of functional requirements identified during the collection and analysis of IS requirements. The most accurate estimates using the modified FPA method will be achieved if there is a catalog of IS requirements with a detailed description of the attributes of each essence of the subject area and the system being constructed. This situation is possible mainly if a waterfall model of its life cycle is chosen for an IT project to construct an IS. However, the choice of such a model leads to a significant increase in the time spent on constructing an IS. If you choose a spiral or hybrid model of the life cycle of an IT project, the accuracy of the assessment of the proposed modification of the method will decrease due to the lack of detailed descriptions of the structural and transactional elements of the assessed IS functions.

Further development of this research can be carried out in several directions. Thus, the question remains open about the applicability of the modified method during the initiation stage and in the early stages of planning an IT project. The complexity of research into this issue is associated with the presentation of data about the constructed IS and its functions in the form of functional requirements. These requirements may be incomplete, contradict each other, and may also be represented in the form of weakly formalized texts in natural language. Possible attempts to overcome the noted difficulties may be associated with the use of artificial intelligence, which can lead to a significant increase in the costs of using the modified method and make such an application inappropriate from an economic point of view.

Another research area should be the possibility of using a modified method based not only on descriptions of the IS database elements but also on the basis of descriptions of the structural elements of the software of this IS. The possibility of using a modified method for estimating an IT project for refactoring an exploited IS (in particular, for cases of adding new, modifying, and developing existing functions) also remains unclear. The main difficulty that complicates research in these areas is the selection of a criterion or group of criteria used to decide on the advisability of re-using individual IS functions.

A separate direction for further research should be the possibility of assessing the efforts of an IT project to construct an IS based on the re-use of elements of previously completed IT projects. The proposed modification of the FPA IFPLUG method is based on the implicit assumption that the composition of the team of performers to construct a selected subset of IS functions remains unchanged. For the case of re-using elements of previously completed IT projects, this assumption cannot always be met.

7. Conclusions

1. The rules for calculating the values of parameters for estimating the number of functional points of an individual IS function have been improved under the conditions of re-using its elements. The essence of this improvement is to exclude from further consideration the descriptions of the structural elements of the re-used function that are present in the descriptions of the system function being estimated. In addition, a quantitative assessment of the efforts to adapt the re-used function to the characteristics of the subject area of the estimated function is proposed. This assessment is based on the results of an analysis of the names of the corresponding functions.

Our proposed improvements make it possible to increase the accuracy of estimating the efforts to construct individual functions within the framework of an IT project for constructing an IS when re-using individual functions of this system.

2. The FPA IFPLUG method (version 4.3.1) has been modified. The essence of this method is, first of all, adding a special stage of formation and decision-making on the advisability of re-using individual functions to reduce the effort required to construct a number of other IS functions. In addition, to increase the accuracy of estimating the efforts to construct individual IS functions, in subsequent stages of the method it is proposed to use improved rules for calculating the values of parameters for estimating the number of functional points. The resulting modification of the method makes it possible to increase the accuracy of estimating the efforts of an IT project to construct an IS under the conditions of re-using its individual functions.

3. An experimental verification of the developed modification of the method was carried out. This check was carried out during the planning of an IT project for the development of the functional task "Forming and executing an individual plan for a scientific and pedagogical employee of the department." To describe the functions of the task, visual models of the functional task were used, such as a data flow diagram and an "essence-relationship" diagram. A total estimate of the efforts to construct three functions was obtained, the descriptions of which almost completely coincide. This score is equal to 72 function points. A summary assessment of the same functions was also carried out using the original FPA IFPLUG method, which amounted to 144 function points. The test results suggest that the accuracy of estimating individual functions of a task that are planned to be constructed by reusing a previously developed function is doubled.

Conflicts of interest

The authors declare that they have no conflicts of interest in relation to the current study, including financial, personal, authorship, or any other, that could affect the study and the results reported in this paper.

Funding

The study was conducted without financial support.

Data availability

All data are available in the main text of the manuscript.

Use of artificial intelligence

The authors confirm that they did not use artificial intelligence technologies when creating the current work.

References

1. A guide to the project management body of knowledge (PMBOK guide) (2008). Project Management Institute, Inc., 459. Available at: https://catalog.loc.gov/vwebv/holdingsInfo?searchId=15282&recCount=25&recPointer=57&bibId=17154406
2. Настанова до зводу знань з управління проєктами. Настанова PMBOK (2021). Project Management Institute, Inc., 370. Available at: https://pmiukraine.org/pmbok7
3. COCOMO II Model Definition Manual. Available at: https://athena.ecs.csus.edu/~buckley/CSc231_files/Cocomo_II_Manual.pdf
4. Eduardo Carbonera, C., Farias, K., Bischoff, V. (2020). Software development effort estimation: a systematic mapping study. IET Software, 14 (4), 328–344. https://doi.org/10.1049/iet-sen.2018.5334
5. Function point counting practices manual. Release 4.3.1. IFPUG. Available at: https://ifpug.mclms.net/en/package/9832/course/18997/view
6. de Freitas Junior, M., Fantinato, M., Sun, V. (2015). Improvements to the Function Point Analysis Method: A Systematic Literature Review. IEEE Transactions on Engineering Management, 62 (4), 495–506. https://doi.org/10.1109/tem.2015.2453354
7. Lavazza, L. (2017). On the Effort Required by Function Point Measurement Phases. International Journal on Advances in Software, 10 (1&2), 108–120. Available at: https://web.archive.org/web/20180412232204/http://www.thinkmind.org/download.php?articleid=soft_v10_n12_2017_8
8. Shah, J., Kama, N., Ismail, S. A. (2018). An Empirical Study with Function Point Analysis for Software Development Phase Method. Proceedings of the 7th International Conference on Software and Information Engineering. https://doi.org/10.1145/3220267.3220268
9. Shah, J., Kama, N. (2018). Extending Function Point Analysis Effort Estimation Method for Software Development Phase. Proceedings of the 2018 7th International Conference on Software and Computer Applications. https://doi.org/10.1145/3185089.3185137
10. Lima Júnior, O. de S., Muniz Farias, P. P., Dias Belchior, A. (2018). A Fuzzy Model for Function Point Analysis to Development and Enhancement Project Assessments. CLEI Electronic Journal, 5 (2). https://doi.org/10.19153/cleiej.5.2.4
11. Zhang, K., Wang, X., Ren, J., Liu, C. (2021). Efficiency Improvement of Function Point-Based Software Size Estimation With Deep Learning Model. IEEE Access, 9, 107124–107136. https://doi.org/10.1109/access.2020.2998581
12. Rohayani, H., Gaol, F. L., Soewito, B., Spits Warnars, H. L. H. (2017). Estimated measurement quality software on structural model academic system with Function Point Analysis. 2017 International Conference on Applied Computer and Communication Technologies (ComCom). https://doi.org/10.1109/comcom.2017.8167085
13. de Freitas, M., Fantinato, M., Sun, V., Thom, L. H., Garaj, V. (2020). Function Point Tree-Based Function Point Analysis: Improving Reproducibility Whilst Maintaining Accuracy in Function Point Counting. Lecture Notes in Business Information Processing, 182–209. https://doi.org/10.1007/978-3-030-40783-4_10
14. Prokopova, Z., Silhavy, P., Silhavy, R. (2018). Influence Analysis of Selected Factors in the Function Point Work Effort Estimation. Advances in Intelligent Systems and Computing, 112–124. https://doi.org/10.1007/978-3-030-00184-1_11
15. Alkazemi, B. Y., Nour, M. K., Sahraoui, A.-E.-K. (2014). Guidelines for Designing Reusable Software Components. International Journal of Computer and Information Technology, 03 (06), 1356–1361. Available at: https://www.ijcit.com/archives/volume3/issue6/Paper030626.pdf
16. Meyer, M. H., Osiyevskyy, O., Libaers, D., van Hugten, M. (2017). Does Product Platforming Pay Off? Journal of Product Innovation Management, 35 (1), 66–87. https://doi.org/10.1111/jpim.12378
17. Systems and software engineering – Software life cycle processes (ISO/IEC/IEEE 12207:2017). Available at: https://www.iso.org/ru/standard/63712.html
18. Ievlanov, M., Solovieva, E. (2014). Unification of methods for estimating the cost of creating modern information systems. Visnyk Kremenchutskoho natsionalnoho universytetu im. Mykhaila Ostrohradskoho, 5/2014 (88), 62–67. Available at: http://visnikkrnu.kdu.edu.ua/statti/2014_5_62.pdf
19. Ievlanov, M., Vasiltcova, N., Neumyvakina, O., Panforova, I. (2022). Development of a method for solving the problem of it product configuration analysis. Eastern-European Journal of Enterprise Technologies, 6 (2 (120)), 6–19. https://doi.org/10.15587/1729-4061.2022.269133