Journal of Theoretical and Applied Information Technology
31st July 2017. Vol.95. No 14

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

SPECIFIC FEATURES OF GRAPHICAL-ANALYTICAL REPRESENTATION OF A PROCESS MODEL OF MANAGING INNOVATION ACTIVITY IN THE FRAMEWORK OF SOFTWARE DEVELOPMENT FOR IMPORT SUBSTITUTION PROJECTS

OLEG JAKOVLEVICH KRAVETS[1], OLEG KOSORUKOV[2], LARISA UTYUSHEVA[3]

[1] Sustainable Developments in Civil Engineering Research Group, Faculty of Civil Engineering, Ton Duc Thang University, 19 Nguyen Huu Tho, District 7, 700 000 Ho Chi Minh City, Vietnam
[2]Lomonosov Moscow State University,119991, Russian Federation, Moscow, Leninskie Gory, GSP-1
[3]Intelin LLC, 115682, Russian Federation, Moscow, Shipilovskaya str., 64, building 1, office 147
E-mail: olegkravets@tdt.edu.vn[1], kosorukovoa@mail.ru[2], lutysheva841@gmail.com[3]

ABSTRACT

The authors have carried out research on the graphical-analytical representation of a process model of managing innovative activity in the framework of software development for import substitution of high-tech IT projects.

Unlike the known approaches, the offered aspects of graphical-analytical modeling of a process-based approach in the special subject domain provide both new results concerning the life cycle of software products of import substituting technologies and minimization of administrative costs of the development processes.

Taking into account specific features of import substitution projects in the area of IT development, a general graphical diagram of the process model which illustrates composition and the sequence of execution of processes and subprocesses of life cycle of innovative IT developments in the context of a system (for example, in developing import substituting application-oriented or instrumental software as a part of complicated industrial systems) has been proposed.

Features of a life cycle of innovative IT solution (innovative software product) with the use of the presented microclassification have been analyzed.

The presented results are perspective for elaborating hybrid process and project models, as well as for algorithmization of flexible calculations of administrative prime costs.

Keywords: *Control Of Processes, Life Cycle, Innovative Activities, Software, A Process Model Of Control, Import Substitution.*

1. INTRODUCTION

At the moment, efficient management of corporate and sectoral processes (production, innovative, research, etc.) is a key condition for success in the market of cutting-edge, usually knowledge-intensive, technologies, including information technology. Constant changes (primarily in the economic environment) lead to a continuous search for and improvement of models of managing business processes. The import substitution policy is a significant driver of this [12].

Import substitution of software is governed by state and local authorities of the Russian Federation in compliance with a range of regulations adopted in 2015. Since January 1, 2016, all state and municipal bodies, state corporations Rosatom and Roskosmos, governing bodies of state non-budgetary funds, as well as public and budgetary institutions that make purchases in accordance with the requirements of Federal Law No. 44-FZ dated April 5, 2013 "On the contract system of the federal and municipal procurement of goods, works and services" [15] are obliged to respect the ban on software originating from foreign countries for the purpose of the federal and municipal procurement.

The ban was introduced by the Decree of the Government of the Russian Federation No. 1236 dated November 16, 2015 "On introduction of a ban on software originating from foreign countries for the purpose of the federal and municipal procurement" [14].

According to expert estimates, the key problem of implementation of import substituting IT developments for the Russian market is the focus of the majority of large Russian users on Western software. Degree of dependence may vary. For example, Oracle products most often can be hardly transferred to other database management systems or cannot be transferred at all, while Microsoft products rarely can run in non-Windows systems. Therefore, the process of import substitution of foreign IT products is a complex process that requires a systematic approach and the introduction of a digital transformation of business processes and restructuring of information systems.

In modern conditions, the high efficiency of managing production, innovation and research processes cannot be achieved without the use of information technology (IT), which, in turn, rapidly and intensively develops as a major tool for managing large economic, production and other systems, while simultaneously transforming into various information, information-analytical and information management systems [9]. In turn, the creation, application and further development of such systems, based on the use of information technology, require the development of special management models [4].

2. METHODS TO DEVELOP SOFTWARE IN IMPORT SUBSTITUTION PROJECTS

The research is aimed at removing the weak relation between the design approach to software development and cost optimization of the development processes by replacing the standard project development approach with the process-related one.

Among existing models of operations management in the field of production, research, etc., a process model of control is distinguished [2], which, as opposed to the functional management model, for example, allows to formulate the ultimate goal and ways of achieving this goal while simultaneously monitoring the entire management process.

For this reason, the process model of control should be considered as the best model for control over innovative activity when implementing import substitution projects in the development of knowledge-intensive applied IT solutions in the field of information technology.

The primary goal of this article is to justify the approach to the development of a process model of control over innovative activity in the development of knowledge-intensive applied IT solutions in the context of import substitution in the field of information technology.

2.1. Managing software development within the framework of innovative activity

Peculiarities of the life cycle of an innovative IT solution (innovative software product) vary depending on the class of innovative software products, including:

- software products with a relatively short life cycle;
- software products with a long life cycle.

Software products with a short life cycle are created primarily to solve scientific and engineering tasks, obtain specific results of calculations, integrate software into industrial products, including within the framework of import substitution projects in sectors of the Russian industries. Support and modification of such programs are not always mandatory.

Software products with a long life cycle are created for regular operation, replication and managing technological processes. Such programs have a complex structure and can be modified in the process of long-term support and use. Software products of this class can be replicated, registered as intellectual property objects and are accompanied by special documentation; they are software products alienated from the developer.

Primary focus in this article is given to the peculiarities of developing large (complex) software for information control and processing.

2.2. Analytics of activities in software development

The key objects of the proposed process model are the product, the platform and the process. The key activity task in software development is creation by the manufacturer or the company (hereinafter – the business) of software products and systems (hereinafter – the product). The concept of the product is in general equivalent to such dissimilar concepts as products made, works performed, services provided, objects created, etc. Resources are being consumed (used) in the process of product creation, part of which originate from the environment that is external to the business, while another part is consumed from reserves.

In general, products are divided into two types: *new* and *mix*. The *new* type includes products that are newly created in the period under review. The *mix* type includes products obtained through combining, mixing products of the *new* type of one

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

range among themselves or with reserves (balance carried forward) remaining from the previous period.

The platform is what the product (products) is (are) created with; the platform unites such concepts as a unit of technological equipment, a structural unit, an enterprise in general, etc.

Platforms and products are primary objects of the model, the project is a pair (platform, product) where the product is perhaps a composite set of products. The process is a triple (platform, product, process) where the objects are linked by time parameters.

Let's consider a set of products and a structure of platforms for the period corresponding to the period of the product creation process.

Suppose Q is the set of platforms, P is the set of products, $G = P \cup Q$ is the population of the primary objects of the model (the set of platforms-products). Let's introduce the partial ordering relation $q \succ s$ (q is "older" than s) in the set G: if $q,s \in Q$, it means that the platform s is part of the platform q; if $q \in Q$, $s \in P$, then $q \succ s$ means that the product s is produced on the platform q; in other cases, the ratio is not determined. It is clear that there is the largest element k in G, which represents the entire system. Since the set G is finite, there is a direct ancestor for any g, i.e. such an element $q = pred(g)$, where, first of all, $q \succ g$, and secondly, there is no such s that $q \succ s \succ q$. If $g \in P$, pred (g) is the platform on which the product g is directly produced; let's call such a platform as the one producing product g.

Aside from the *structural relationship* of partial ordering, the *consumption ratio* is defined between the objects of the model: if $p \in P$, $g \in G$, then the ratio $p \to g$ means that the product p is consumed

(wholly or partially) by the product or platform g.

Let's call a set $\Omega = (Q, P, \succ, \to)$ a *model scheme*, since this set completely defines the objects of the model and links between them but does not reflect the content information associated with these objects.

2.3. Graph simulation of the development process

The scheme of the model can be represented in the form of a *model graph*, the directed graph G (Figure 1), defined as follows: the elements of the G set connected by the arcs (pred(g),g) form the *tree platforms* – TQP products, which is supplemented by arcs-firmware connecting nodes that are in the relation $p \to g$.

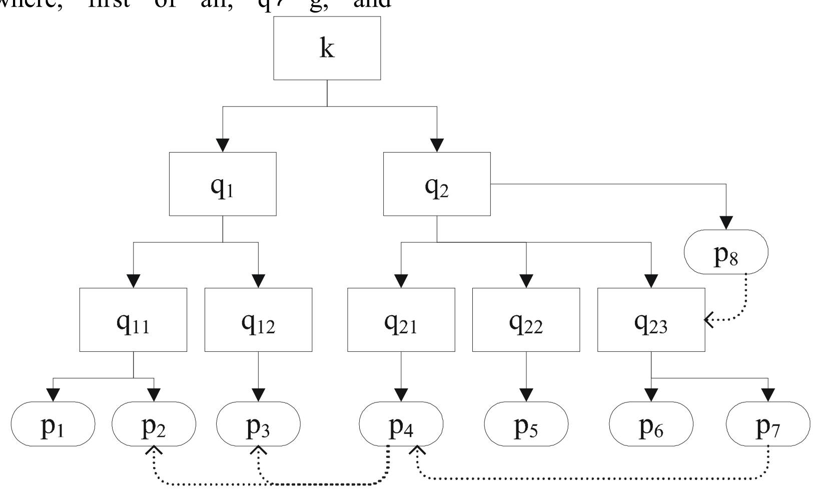

Figure 1. Model graph

Suppose $G = \{x_i\}_{i=1}^{N_P+N_Q+1}$, where N_Q is a number of platforms, N_P is a number of products, and it can be assumed that $x_i = q_i$ at $1 \le i \le N_Q$ and $x_i = p_{i-N_Q-1}$ at $i > N_Q$. Let's define the incidence matrix for the G model graph: $A = \{a_{ij}\}_{i,j=0}^{N_P+N_Q+1}$, where $a_{ij} = 1$, if ($i < j \le N_Q$ and $q_i = pred(q_j)$), or ($i \le N_Q, j > N_Q$ and $q_i = pred(p_{i-N_Q-1})$), or ($i > N_Q, j \le N_Q$ and

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$p_{i-N_Q-1} \to q_j$), or ($i > N_Q, j > N_Q$ and $p_{i-N_Q-1} \to p_{j-N_Q-1}$), in the remaining cases $a_{ij} = 0$.

A node map is a summary of all the information associated with the node. The node-platform map has the following properties: platform name (identifier); platform is an immediate ancestor; for the root of the tree T_{QP}, it is an "empty" platform Λ – a special element of the set Q used for uniformity in situations where the platform does not exist; node resource map.

The resource map contains information about resources consumed by the platform to ensure its operation in the period under review. The node-product map contains the following information: name (identifier); product range (sort); producing platform; product type; product volume (in quantitative and/or aggregate terms); product usage: list of pairs (consumer, consumed volume); related products: list of triples (product range, volume, consumer); final state; initial state; resource map.

It is assumed that two elements of a set P of the same type with a common producing platform belong to different product ranges. This means that the element $p \in P$ represents the entire product of a certain product range and of a certain type, produced directly on some platform. It follows from the foregoing that the product $p \in P$ is uniquely determined by the triple (product range, platform, type).

The volume of a new product type means the volume of the finished product, excluding goods in progress.

The final state of the node-product is the remainder by the end of the period under review: for a mix product type, it is the amount of the product's reserve, while for a new product type, it is goods in progress represented as the resource map, i.e. in the context of used resources. The initial state is the final state of a similar product by the end of the previous period.

If the product is partially (fully) a final product, i.e. is being consumed in the external environment, one of the consumers (the only consumer) in the list of the product use is defined as "external".

Creation of a product p of the new type can be accompanied by the creation of related products; these products are represented in the model as the same node as the main product, information about them is shown in the p node map. A resource map contains information about the resources used to create the product. The resource map of the node-product of the new type consists of several sections. The structure of the resource map is shown in Table 1.

One of the purposes of the model is calculation of the cost of products created by the system, based on the cost of the resources consumed by the system –both reserves, i.e. resources available by the beginning of the period, and those acquired and produced in the period under review.

Table 1. The resource map structure

Resources	Volume	
	Cost	Quantity
▪ Intrinsic value		
▪ resource item		
✓ *resource*		
…		
▪ Related products		
▪ related products		
✓ *product*		
…		
▪ Indirect resources		
✓ resource item		
✓ …		
▪ Total cost		

The system consumes resources to ensure the functioning of the platforms and directly for the products, i.e. resources are associated with model nodes. The total value terms of the set of resources consumed by the node will be called *intrinsic value*, and its constituent resources will be called *intrinsic*, or *direct* resources of the node. Along with intrinsic resources, the cost of products includes *indirect* resources obtained by distributing (allocating) the resources of the platforms to the products produced on these platforms. Along with its intrinsic value, indirect resources make up the *total* cost (cost) of the node-product. For a node-platform, the total cost coincides with intrinsic cost.

Intrinsic resources of the node can be divided into *primary* and *secondary resources*: secondary resources consist from the consumption of the products created (by any means, i.e. both products of new and mix types) in the current period, while the remaining resources refer to primary resources. Primary resources, in particular, are the consumption of the balance carried forward – the initial state of the product. Secondary resources are included in the intrinsic value of any node at the full cost of the consumed product.

The node resources are shown in the resource map, where they are grouped by sections (intrinsic resources, related products, indirect resources) and items.

The following principle is implemented to allocate the platform resources to the products created on their basis: resources of a certain platform are allocated to the part of products created on this platform that are used outside this platform.

Consider the example shown in Figure 2. According to the formulated principle, the following occurs in the process of the platforms resource allocation:

- L_1 resources are allocated to the entire product A and are distributed in proportion to the volume between its parts A_1 and A_2;
- L_2 resources are allocated entirely to product B_1;
- C_1 resources are distributed between products A_2 and B_1 used outside this platform;
- C_2 resources are entirely allocated to product B_2;
- H resources are distributed between the end products B_1 and B_2.

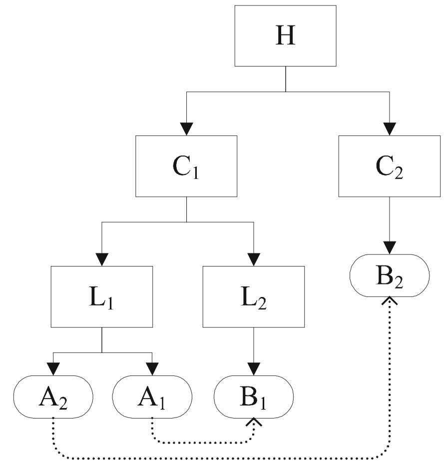

Figure 2. Example of the platform resource allocation

Formalization and implementation (algorithmization) of the principle of resource allocation requires decomposing the graph G by splitting the products that are elements of the set P into their constituent elementary products in accordance with the fact and the place of their consumption.

Graph K, which satisfies the condition (α), is called a calculation model graph; each virtual node-product of the graph K either is consumed *entirely* by a certain node (product or platform) of the graph K, or is the final product consumed in the external environment, or represents the final state of some product-range (Figure 3).

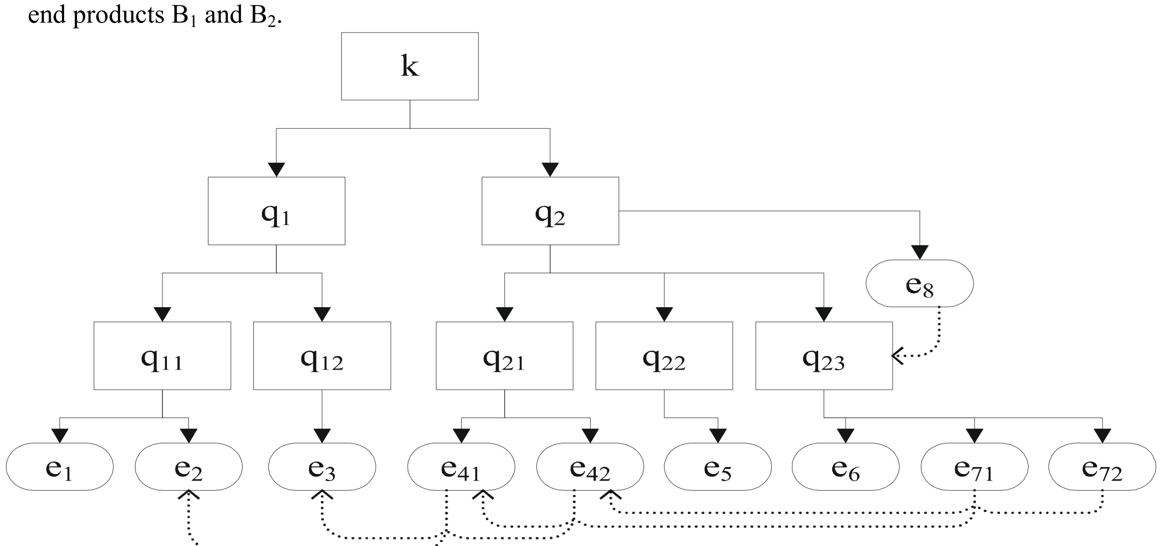

Figure. 3. Calculation graph

Let's denote the incidence matrix of the graph K through $A' = \{a'_{ij}\}_{i,j=1}^{N_Q+N_E+1}$, where N_E is a number of elementary products after decomposition. In terms of the matrix A′ the property (α) means that $\sum_{j=1}^{N_Q+N_E+1} a'_{ij} \le 1$ at $i > N_Q$.

Suppose p∈P is a random product. Math it with a set dec(p) = $\{p^i, i=0..m_p\}$ of elementary (virtual – unlike real products from P) products, defined as follows:

- virtual product p^0=rem(p) represents the final state of the product p in the current period – goods in progress by the end of the period or balance carried forward by the end of the period in
- accordance with the new or mix type of the product p;
- each virtual product p^i, $i=1..m_p$ represents the entire part of the product p or a certain related product that is consumed by one consumer – in particular, an "external" consumer, i.e. is the final product; as such, the related products, if they exist for the product p, will also be represented by separate virtual products.

The set dec(p) = $\{p^i, i=0..m_p\}$ of virtual products, as well as the procedure for the formation of this set, shall be called a decomposition (splitting) of the product p.

The set E = dec(P) of the vertices of the calculation graph K shall be called a decomposition of the set P, or a set of elementary products.

For the sets E and H=E⋃Q, the relations $\succ$, → introduced for the sets P and G=P⋃Q, as well as the notions and constructions defined on their basis, preserve the meaning. The graph K is similar to the graph G in terms that it is a representation of these relations. The graph K can be regarded as a tree of platforms T_Q extended to the tree of platform-products T_{QE} and supplemented with arc-firmware. The defining property (α) of the calculation graph can be formulated as follows: for any p∈E, the relation p→g exists for at most one g, i.e. no more than one firmware originates from one node-product of the graph K.

The resource map and, therefore, the cost of any initial product p∈P are obtained by aggregating the resource maps of the elementary products constituting dec(p).

Suppose q∈Q. T(q) denotes the maximal subtree of the graph K with the root q in the tree of the platforms-products T_{QE}; T(q) also includes all firmware, both ends of which lie in T(q). L(q) denotes the set of leaves of the subtree T(q). For a random set R⊆L(q), the set G(q, R) is defined as a set of nodes of all paths from the node q to the nodes of the set R. The set G(q, R) includes products from R, as well as all platforms belonging to the platform q (including q) and all products created on these platforms that are used to create products from the set R.

Suppose q is a random platform, R is a certain set of products produced on this platform, i.e. R⊆L(q). A pair X = (q, R) shall be called a process, if for every p∈R from p→s follows s ∉ G(q, R), i.e. products from R are consumed outside the process X.

Suppose q∈Q. R(q) denotes the set of vertices p∈L(q), where the relation p→s does not exist for the vertices s from the tree T(q). Assume that R(q) is non-empty, since R(q)=∅ would mean that all products created on the platform q serve only to ensure its functioning. It is obvious that the pair (q, R(q)) satisfies the definition of the project. Such a project can be called a basic project on the platform q and denoted as T(q), since the process model is a subtree T(q) (Figure 4).

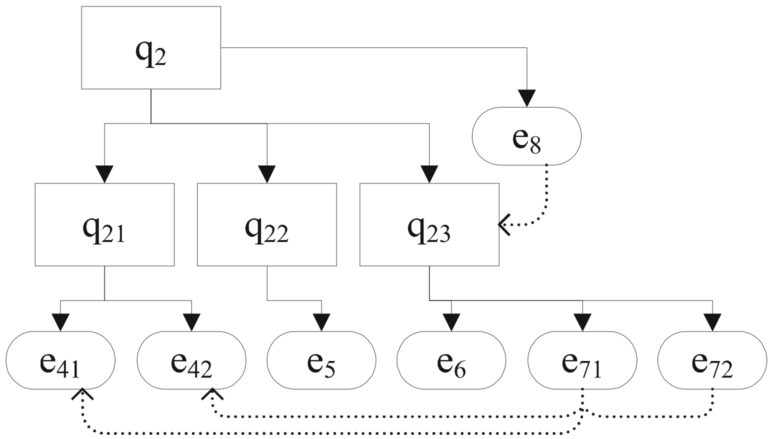

Figure. 4. Graph of the base process $T(q_2)$

As such, a structural model of activity in software development based on the graph approach was developed, which ensures a unified description of the resource consumption. Algorithms to redistribute the contribution of consumed resources between various levels of the enterprise structural units can be developed on the basis of the structural model, which ensures the formation of the value indicators for the products created, taking into consideration the hierarchy of the business structural units in the context of items for subsequent decision-making.

3. DISCUSSION AND RESULTS IN THE DEVELOPMENT OF GRAPHIC-ANALYTICAL PROCESS MODELS

It must be particularly noted that only the international standard ISO/IEC 12207:2008 "Information technology. System and software engineering – Software life cycle processes" [3], which is also in force on the territory of the Russian Federation as GOST R ISO/IEC 12207-2010 "Information technology. System and software engineering – Software life cycle processes" [10], has clarified the methods of control of the life cycle of software products.

The methodology of the above standard allows to develop the models of control of life style processes of software products, which are part of an independent project, such as an import substitution project that includes software products.

3.1. Model of control over the process of software creation in terms of import substitution

The process of software creation is a major top-level process. A project on import substitution, within which the creation process occurs, introduces some special conditions and requirements into this process model of control, which need to be analyzed before developing an implementation strategy.

Since this process is essentially a framework process in the implementation of import substitution projects, as well as other projects involving the development of IT solutions, the key successive outputs of this process include:

- analysis of requirements and conditions of the import substitution project;
- definition of the implementation strategy in terms of import substitution;
- definition of restrictions on the technology of the project realization;
- production of the software component;
- packaging and storage of the software component in accordance with the agreement on its supply.

Visualization of the model of control over the process of software creation in terms of import substitution is shown in Figure 5.

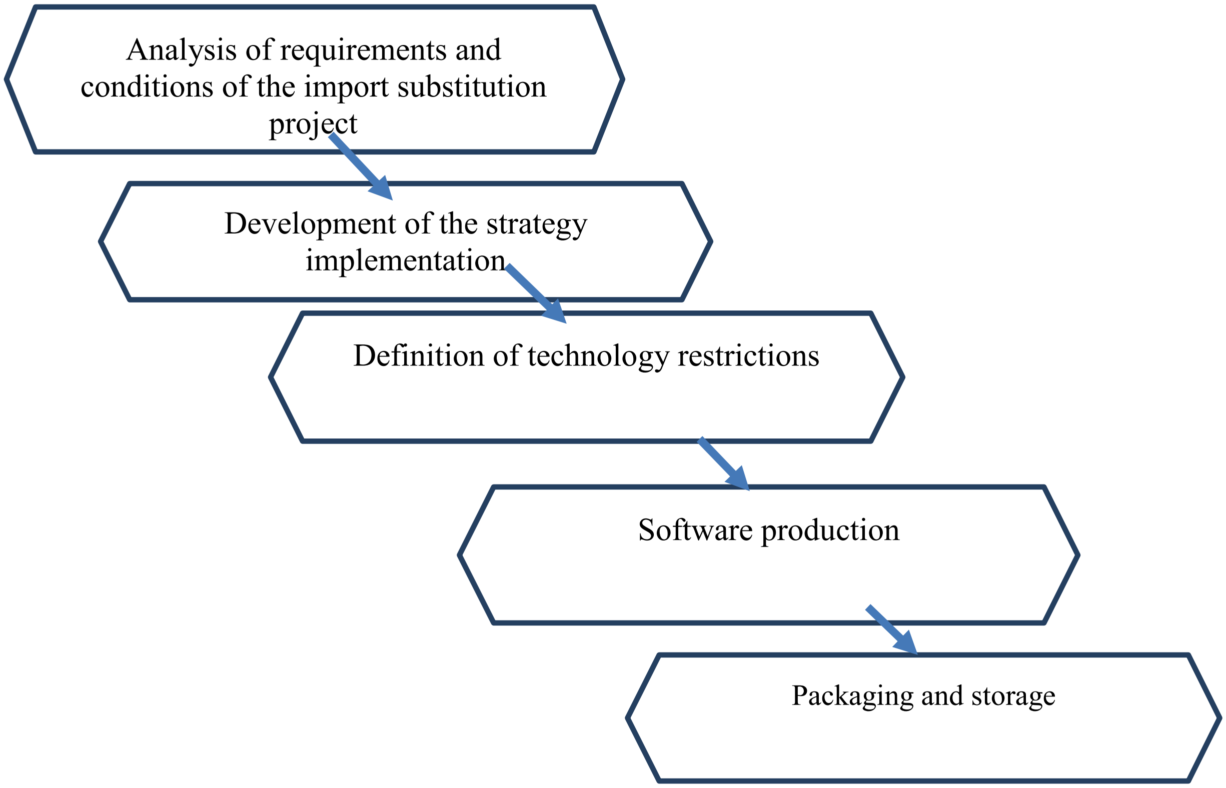

Figure 5. Cascade model of control over the process of software creation in terms of import substitution

3.2. Model Of Control Over The Process Of Analyzing The Requirements For Software In Terms Of Import Substitution

As noted above, the goal of the process of analyzing the requirements for software is to define and set the requirements for individual elements of the IT system. The import substitution project, under which the process of analyzing the requirements for software occurs, introduces some special conditions and requirements into this process model of control, which need to be analyzed before developing a process of analyzing the requirements for software and taken into account at each stage of the process.

In result of the process of analyzing the requirements for software, the following main outputs of this process should be provided, from the point of view of analyzing the requirements:

- general requirements for the process of analyzing the requirements for software in terms of import substitution should be formulated;
- requirements for individual program elements of the system and their interfaces in the context of import substitution should be formulated;
- requirements for software correctness and testability should be analyzed;
- the impact of requirements for software on the functionality of the import substitution project should be assessed;
- the compatibility between the requirements for software and the requirements for the import substitution project should be defined;
- priorities for implementation of requirements for software in terms of import substitution should be determined.

Visualization of the cascade model of control over the process of analyzing the requirements for software in terms of import substitution is presented in Figure 6.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

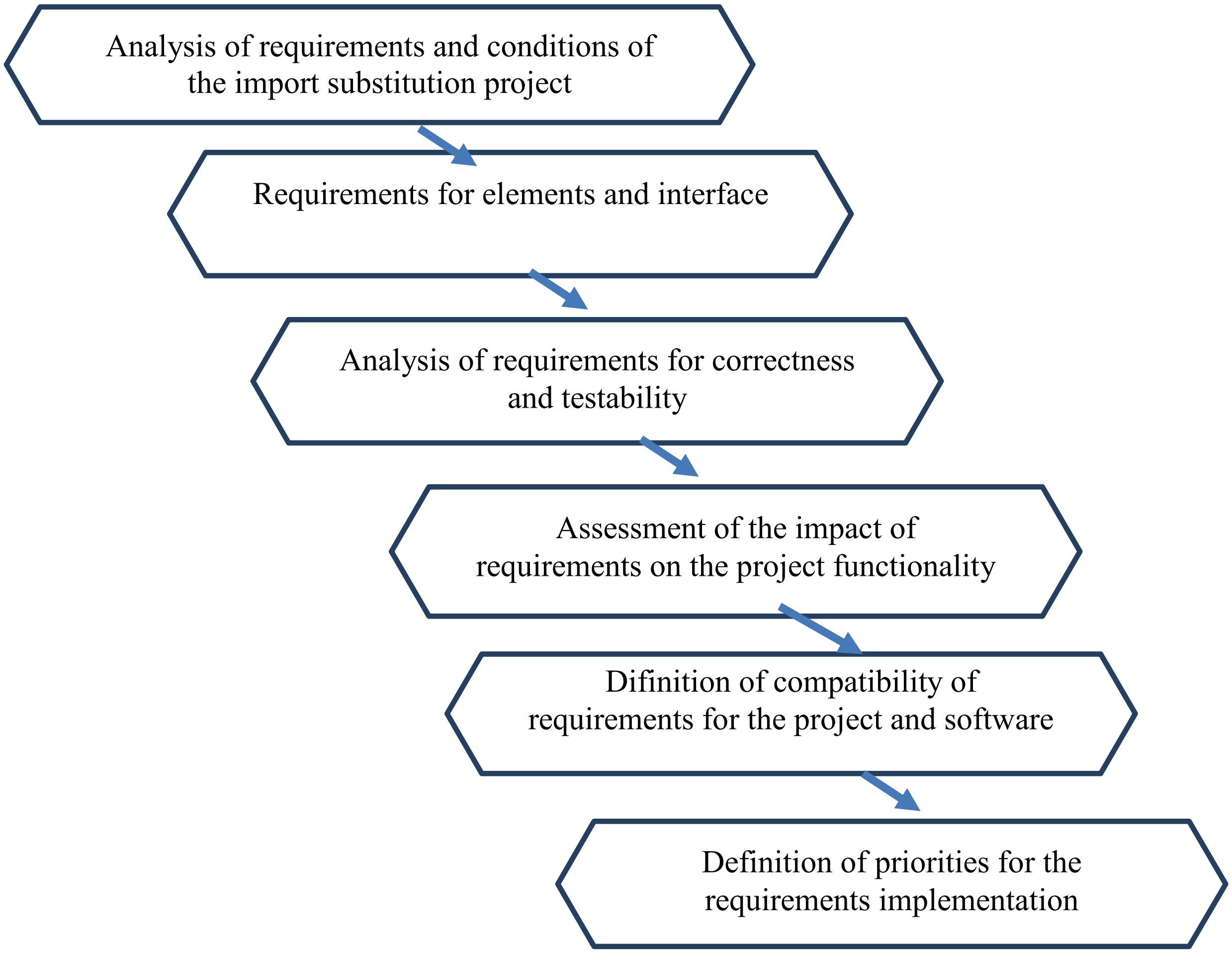

Figure 6. Cascade Model Of Control Over The Process Of Analyzing The Requirements For Software In Terms Of Import Substitution

3.3. Model of control over the process of designing the software architecture in terms of import substitution

The process of designing the software architecture is an element of a unified process of software implementation, which is implemented in accordance with the requirements of the import substitution project.

The import substitution project, under which the process of analyzing the requirements for software occurs, introduces some special conditions and requirements into this process model of control, which need to be analyzed before developing a process of analyzing the requirements for software and taken into account at each stage of the process.

The main goal of designing the software architecture in terms of import substitution is to develop and ensure the implementation of the project on software creation, taking into consideration the requirements of the import substitution project. Meanwhile, one of the key tasks is to provide an opportunity to verify the software part of the product created in accordance with the requirements of the import substitution project.

According to the structure of the life cycle process, the model of control over the process of designing the software architecture in the cascade model of control over the software development in terms of import substitution takes the form shown in Figure 7.

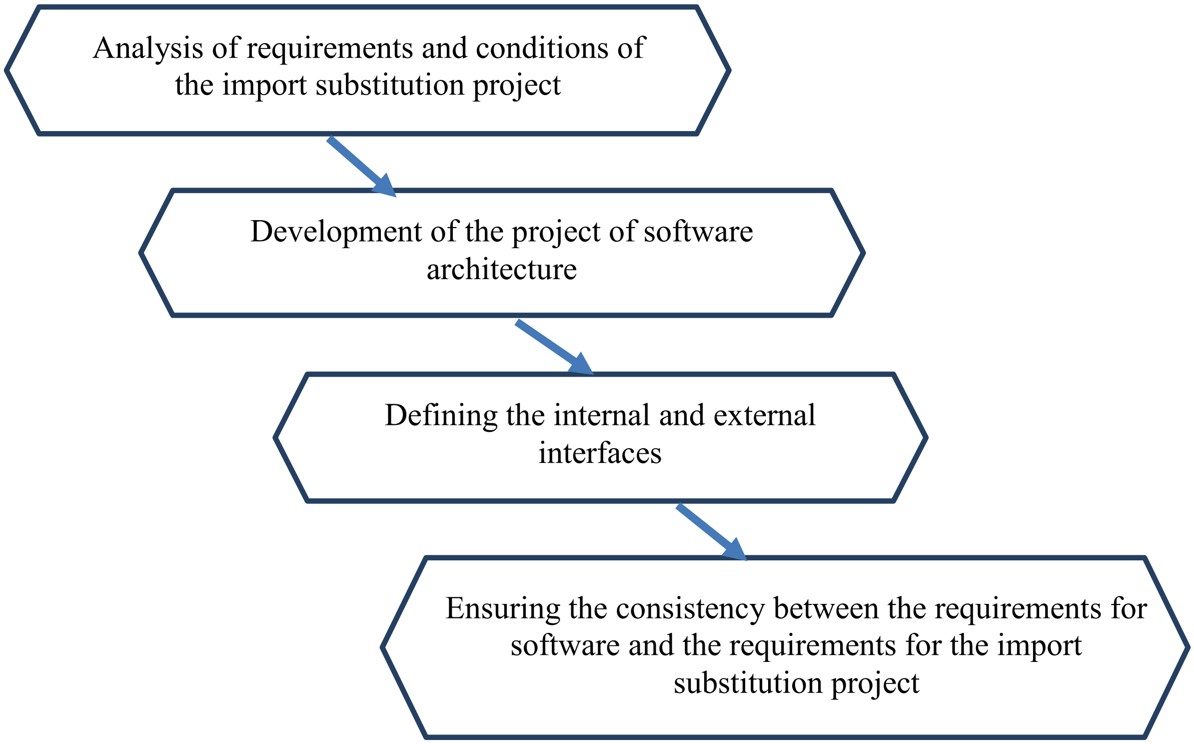

Figure 7. Cascade Model Of Control Over The Process Of Designing The Software Architecture In Terms Of Import Substitution

3.4. Model of control over the process of detailed software design in terms of import substitution

The goal of control over the process of detailed software design, which is essentially the key process in the software life cycle (innovative IT solution), is to ensure the creation and use of software in accordance with the requirements of the import substitution project, which are implemented and can be verified against the set requirements and the developed software architecture, as well as to ensure the subsequent coding and testing.

The import substitution project, under which the process of analyzing the requirements for software occurs, introduces some special conditions and requirements into this process model of control, which need to be analyzed before developing a process of detailed software design and taken into account at each stage of the process.

According to the abovementioned outputs of the process of detailed software design, a cascade model of control over this process in terms of import substitution includes three results:

- analysis of special requirements for the process of detailed software design, imposed by the import substitution project;
- developed detailed project of each software component describing the generated software modules;
- developed external interfaces of each software module, which ensure compatibility and traceability between detailed design, requirements of the import substitution project and design of architecture.

In this regard, the visualization of the cascade model of control over the process of detailed software design in terms of import substitution takes the form shown in Figure 8.

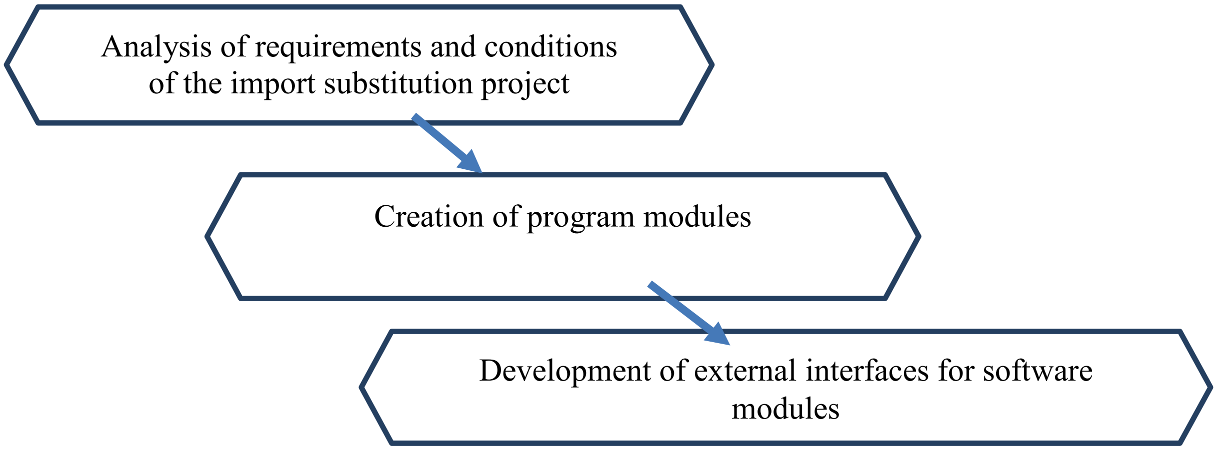

Figure 8. Cascade Model Of Control Over The Process Of Detailed Software Design In Terms Of Import Substitution

3.5. General conceptual graphical scheme of the process model illustrating the composition and sequence of execution of the processes and subprocesses of the life cycle of the innovative IT development in terms of import substitution

The conceptual graphical scheme of the process model illustrating the composition and sequence of execution of the processes and subprocesses of the life cycle of the innovative IT development in terms of import substitution includes only processes and subprocesses that are directly related to software development in terms of import substitution.

At the same time, in case of successful implementation of import substitution project that provides for the development and installation of software, for example, into some hardware and software complex, information retrieval system, etc. (applied software and software tools), an effect arises associated with the fact that innovation product developed within the import substitution project (almost any product produced within the import substitution project is innovative by definition – at least for the manufacturing business) acquires properties determined by the requirements of the import substitution project. This means that the processes of the innovation product life cycle in the context of the system will be determined by the software installed in it and developed within the import substitution project.

Having regard to the above, the general conceptual graphical scheme of the process model of the life cycle of the innovative IT development in terms of import substitution includes the life cycle processes in the context of the system that are not involved in the LC processes in the context of software, while the general graphical scheme itself acquires the following form (Figure 9). Figure 9 also shows process groups that are formally not related to the development of an innovative IT solution outside the import substitution project (for example, systemic or open-source import substitution software). But in terms of the import substitution project, which involves the development of the applied software or software tools, the requirements of import substitution conditions also apply to the characteristics of the system (for example, the hardware and software complex), modifying them in accordance with the requirements of the import substitution project. The processes of software life cycle in terms of import substitution are marked with filling.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

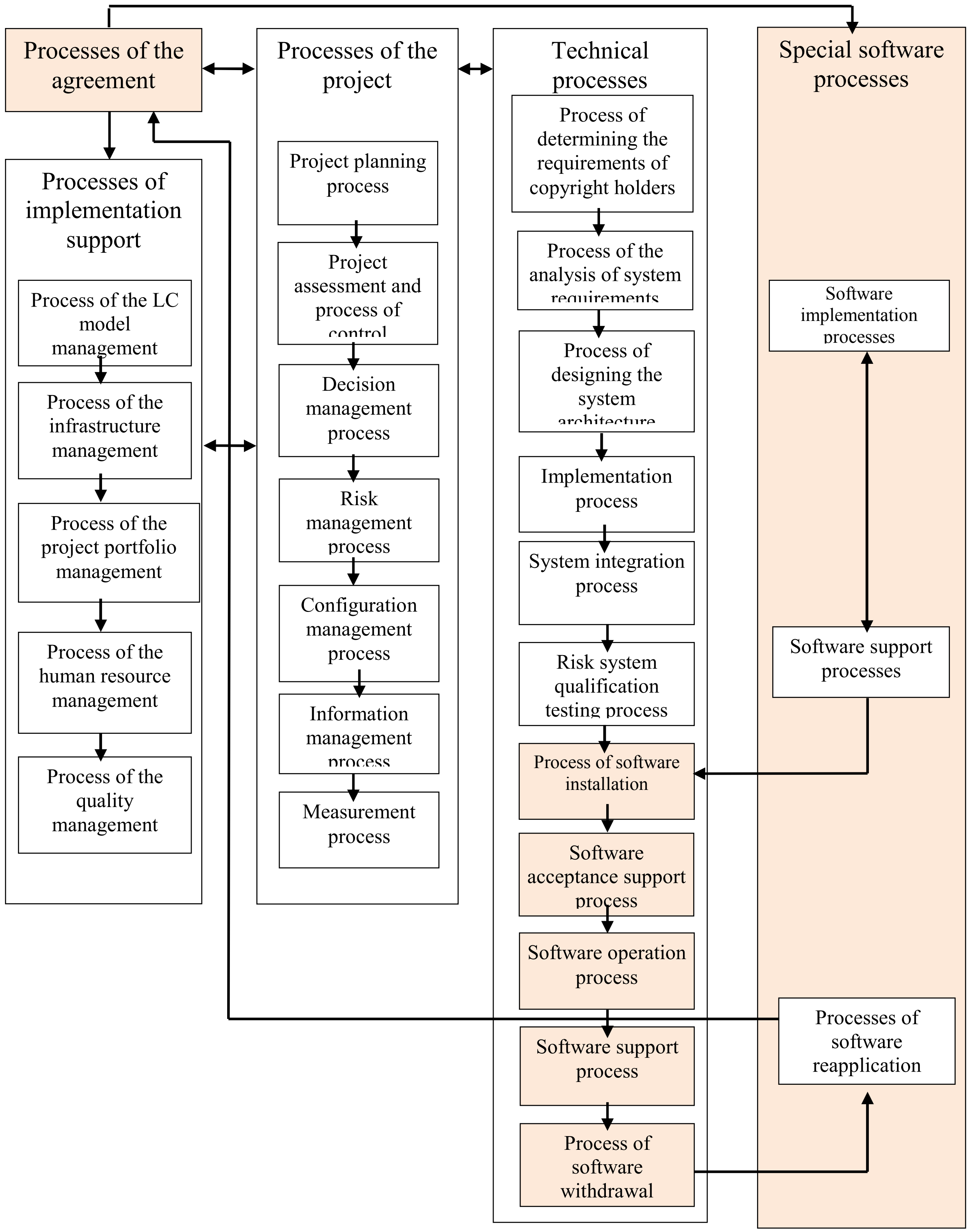

Figure 9. General Conceptual Graphical Scheme Of The Process Model Of The Life Cycle Of The Innovative IT Development In Terms Of Import Substitution Taking Into Consideration Processes In The Context Of The System

3.6. Results and discussion

Graphical-analytical modeling of the components of the system of control over the creation of software for innovative projects in terms of import substitution combines a number of previously known solutions and contains new results.

A structural model of activity in software development based on the graph approach was developed, which ensures a unified description of the resource consumption. Algorithms to redistribute the contribution of consumed resources between various levels of the enterprise structural units can be developed on the basis of the structural model, which ensures the formation of the value indicators for the products created, taking into consideration the hierarchy of the business structural units in the context of items for subsequent decision-making.

A model of control over the process of software creation in terms of import substitution was created, which defines the outputs as requirements and conditions of the import substitution project; implementation strategy in terms of import substitution; definition of restrictions on the technology of the project realization; production of the software component; packaging and storage of the software component in accordance with the agreement on its supply.

A model of control over the process of analyzing the requirements for software in terms of import substitution was presented, which is intended to define and set the requirements for individual elements of the IT system.

The model of control over the process of designing the software architecture in terms of import substitution was updated, taking into account the need to verify the software part of the product created in accordance with the project requirements.

The cascade model of control over the process of detailed software design in terms of import substitution was analyzed, which ensures the creation and use of software in accordance with the project requirements.

In general, process models are widely represented in non-industrial applications as well: in economics [1], management theory [5, 11], technical diagnostics [13], etc. However, integration of the process-related models aimed at cost optimization of the development processes have not been studied before. Thus, in work [16], cost difference between the tasks and the processes is considered.

Research [17] is devoted to comparing various financial processes with other organizations from the point of view of cost efficiency, productivity, speed of work and frequency of errors.

Work [18] proposes a soft method of calculating the predicted value of the process.

Several authors have discussed various aspects of process modeling in software development. For example, project software errors identification was studied in [6], the formation of event maps for top-level process models is discussed in [7], and a comparative analysis of various process models in software development is presented in [8]. Despite the process-oriented nature of the studies, they are strictly object-oriented, and do not consider process-based approach to development in relation to the cost aspects.

Unlike the known approaches, the offered aspects of graphical-analytical modeling of a project-based approach in the special subject domain provide both new results concerning the life cycle of software products of import substituting technologies and minimization of administrative costs of the development processes.

In the light of the above, it seems advisable to create special software that can be integrated into the structure of applied software packages and support the estimation of the cost of production by a commercial industrial enterprise based on a process approach to structuring its operation processes and described by invariance to the phases of business operation and taking into account the hierarchy of business units in the redistribution of contribution of the consumed resources between the various levels of structural units.

4. CONCLUSION

Given the specifics of import substitution projects in the field of IT development, which generally include the development of three main types of software (system software, applied software and software tools), a general graphical scheme of the process model illustrating the composition and sequence of execution of the processes and subprocesses of the life cycle of the innovative IT development in terms of import substitution in the context of the system (for example, in development of the import substituting applied software or tool software) has been proposed.

The peculiarities of the life cycle of an innovative IT solution (innovative software

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

product) with the use of the presented microclassification have been analyzed.

Prospects for the study include the development of the hybrid process-design models, as well as the implementation of algorithmization of flexible calculation of administrative costs with subsequent software implementation.

5. FUNDING SOURCES:

The article is sponsored by the Ministry of Education and Science of the Russian Federation on the basis of the materials of the project "Building a process model of control over the innovative activity in terms of import substitution" (Agreement on granting the subsidy No. 03.573.21.0013 dated December 14, 2015, unique R&D identifier RFMEFI57315X0013).

REFERENCES:

[1]. Brian, B. Using Process Modeling Notation to Map the Buying and Selling of Complex Software Solutions: A Qualitative Study's Implications for Practice and Pedagogy. Technical Communication, Volume 63, Number 3, pp. 212-230(19), 2016

[2]. Dawson, C. and Dawson, R. Software Development Process Models: A Technique for Evaluation and Decision-Making. Knowledge and Process Management, Volume 21, Issue 1, pp. 42–53, 2014

[3]. ISO/IEC 12207:2008 "Information technology. System and software engineering - Software life cycle processes". Retrieved November 6, 2016, from https://www.iso.org/obp/ui/#iso:std:iso-iec:12207:ed-2:v1:en.

[4]. Romansky, R. An application of procedure for analytical evaluation of computer structures. International Journal on Information Technologies & Security, № 2, pp. 55-63, 2013.

[5]. Luo, L., Yao, Y. and Gao, F. Cost-Effective Process Modeling and Optimization Methodology Assisted by Robust Migration Techniques. Ind. Eng. Chem. Res., vol. 54 (21), pp 5736–5748, 2015

[6]. Selvaraj, M. G. Devi Shree Jayabal, Thenmozhi Srinivasan, and Palanisamy Balasubramanie. Predicting Defects Using Information Intelligence Process Models in the Software Technology Project. The Scientific World Journal, Volume 2015 (2015), Article ID 598645, pp. 6

[7]. Mitsyuk A.A., Shugurov I., Kalenkova A.A., van der Aalst W. Generating Event Logs for High-Level Process Models. Simulation Modelling Practice and Theory, Vol. 74, pp. 1-16, 2017

[8]. Raval, R. R. and Rathod, H. M. Comparative Study of Various Process Model in Software Development. International Journal of Computer Applications, vol. 82(18), pp. 16-19, November 2013.

[9]. Doneva, R., Gaftandzhieva, S. and Totkov G. Fetch project: the maturity of quality management through dynamic evaluation of the project progress. International Journal on Information Technologies & Security, Number 3, pp. 29-38, 2016.

[10]. GOST R ISO/IEC 12207-2010 "Informatsionnaya tekhnologiya. Sistemnaya i programmnaya inzheneriya – Protsessy zhiznennogo tsikla programmnykh sredstv [Information technology. System and software engineering – Software life cycle processes]". Retrieved November 6, 2016, from http://gostpdf.ru/gost-12207-2010/download.

[11]. Evdokimova E.N. Evolyutsiya protsessnogo podkhoda v upravlenii i perspektivy yego razvitiya [Evolution of the process-based approach in management and prospects for its development]. Managing economic systems, vol. 28, pp. 117-124, 2011.

[12]. Kalyuzhny, K.A. Sostoyaniye i perspektivy importozameshcheniya v rossiyskoy IT-otrasli [State and prospects of import substitution in the Russian IT sector]. Science. Innovation. Education, No. 2(20), pp. 85-103, 2016.

[13]. Kornyushko, V.F., Porodnikova, P.A. and Kostrov, A.V. Podkhod k razvitiyu sistemy upravleniya testirovaniyem programmnykh sredstv [Approach to the development of a software testing management system]. Software products and systems, No. 4. pp. 126-132, 2015

[14]. Decree of the Government of the Russian Federation No. 1236 dated November 16, 2015 "Ob ustanovlenii zapreta na dopusk programmnogo obespecheniya, proiskhodyashchego iz inostrannykh gosudarstv, dlya tseley osushchestvleniya zakupok dlya obespecheniya gosudarstvennykh i munitsipalnykh nuzhd [On introduction of a ban on software originating from foreign countries for the purpose of the federal and municipal procurement]". Retrieved November 6, 2016, from http://d-russia.ru/wp-

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

content/uploads/2015/11/ac872y0wqioFnrRUeTnpGjEavWCfgEAo.pdf.

[15]. Federal Law No. 44-FZ dated April 5, 2013 "O kontraktnoy sisteme v sfere zakupok tovarov, rabot, uslug dlya obespecheniya gosudarstvennykh i munitsipalnykh nuzhd [On the contract system of the federal and municipal procurement of goods, works and services]". Retrieved November 6, 2016, from http://publication.pravo.gov.ru/Document/View/0001201506300082?index=0&rangeSize=1.

[16]. Steven Bragg. The difference between job costing and process costing. March 31, 2013. https://www.accountingtools.com/articles/what-is-the-difference-between-job-costing-and-process-costi.html.

[17]. Mary Driscoll. Metric of the Month: Accounts Payable Process Cost. March 31, 2015. http://ww2.cfo.com/expense-management/2015/06/metric-month-accounts-payable-process-cost/.

[18]. Amitava Ray, Bijan Sarkar, Subir Kumar Sanyal, (2010) "Process cost prediction: a soft computing approach", International Journal of Intelligent Computing and Cybernetics, Vol. 3 Issue: 3, pp.431-448, doi: 10.1108/17563781011066710