ISSN-L 2664-9969; E-ISSN 2706-5448

UDC 004.03:005.1
DOI: 10.15587/2706-5448.2025.326479

Viktor Levykin,
Maksym Ievlanov,
Ihor Levykin,
Oleksandr Petrichenko

DEVELOPMENT OF A CONCEPT FOR THE TASK OF LIFE CYCLE EFFECTIVE MANAGEMENT OF AN OPERATED INFORMATION SYSTEM

The object of research is the processes of functioning and maintenance, which together determine the operation stage of the information system.

The study is devoted to solving the problem of life cycle formal management of operated information systems of management of enterprises and organizations. Research in this area is mainly aimed at developing models, methods and technologies for managing material products and software applications. Issues of life cycle management of interdisciplinary IT products, such as enterprise management information systems, remain practically unexplored.

The aim and main limitations of classical (permanent) management of the life cycle of an operated information system are determined and formally described. The main disadvantage of such management is the possibility of a significant increase in the number of change requests that arise as a result of changes in business processes and IT infrastructure of enterprises and organizations. Therefore, it was proposed to move from the concept of classical (permanent) management to the concept of life cycle effective management of an operated information system. This concept allows to formally describe the task of life cycle effective management of an operated information system as a task of achieving optimal characteristics of this information system for each of its specific properties and the minimum probability of the existence of unresolved incidents and requests for changes during the operation stage of this information system. Based on the provisions of this concept, formal descriptions of the objective function and the main constraints of the task of life cycle effective management of an operated information system for its individual properties are developed. The use of this concept allows to consider classical (permanent) management as a partial case of life cycle effective management of an operated information system.

Practical application of the proposed formal description of the task of life cycle effective management of an operated information system allows to improve SLM-systems for managing the life cycle of an operated information system without global reengineering of existing systems and technologies for data storage and processing.

Keywords: *information system, system life cycle management, management by properties, objective function, constraints.*

Received: 11.01.2025
Received in revised form: 06.03.2025
Accepted: 27.03.2025
Published: 11.04.2025

How to cite
Levykin, V., Ievlanov, M., Levykin, I., Petrichenko, O. (2025). Development of a concept for the task of life cycle effective management of an operated information system. Technology Audit and Production Reserves, 2 (2 (82)), 66–73. https://doi.org/10.15587/2706-5448.2025.326479

1. Introduction

The modern approach to the creation and functioning of information systems (IS) for the management of enterprises and organizations conceptually considers such systems as a type of system for providing IT services to operators and users [1]. To solve the problem of managing these services throughout their life cycle (LC), a separate type of management system was created – service management systems. According to modern ideas, such systems should be perceived as systems for managing and controlling the activities of a service provider from the point of view of service management [1].

The results of adapting this definition for the IT industry are:

a) the concept of an IT service management system as a system for managing and controlling the activities of managing IT objects of an IT company;

b) an approach to product LC management (PLM);

c) an approach to application LC management (ALM).

Here and further, an IT service will be understood as an IS independent functional task, the use of which is economically and technically feasible for performing a separate activity of an enterprise or organization's process or for managing this activity [2]. The term "PLM" will be understood as an approach to managing processes and methods that are used or created during the LC of products or systems. Its scope of application mainly covers the hardware parts of the system [3]. The term "ALM" will be understood as an approach focused on the software LC. It is designed to monitor, control and manage artifacts and processes that arise during the program LC. These processes include requirements development, source code management, test management, release management, etc. [4].

Although the PLM and ALM approaches arose on the basis of the concept of an IT service management system, the IS created within these approaches have developed in different ways. The reasons for this should be considered different approaches to structuring hardware and software, as well as to combining elements of these services into systems. In addition, the software and hardware LC phases are also different, even if there are similarities [5]. But since modern enterprise and organization management IS consists of various supporting systems, it becomes relevant to create a new approach. The main difference of this approach is the integration of PLM and ALM into the interdisciplinary system life cycle management (SLM) [6].

Modern information technologies (IT) and IS, created on the PLM basis, are aimed at managing large amounts of information and artifacts that arise and accumulate during the product LC [3]. Such PLM-IT and PLM-IS are considered as a center that ensures the existence of compatible data flows (for example, data from CAD products, modeling or architecture data, etc.) and solving the tasks of managing these data flows. This is due to the fact that PLM grew out of CAD (Computer Aided Design) and PDM (Product Data Management). For this reason, PLM-IT and PLM-IS are only limitedly suitable for displaying and supporting the software LC [6].

The use of PLM-IT and PLM-IS for the IS LC management in enterprises and organizations is complicated by the need to manage changes that arise during the creation and operation of such IS. Thus, in [7] the reasons for these complications are the lack of compatibility of product change notifications, low data quality and the lack of direct traceability and tracking of changes to the final product. To eliminate these complications, in [7] it is proposed to integrate PLM-IS and IT asset administration (Asset Administration Shell). Such integration should improve the cooperation of these systems, improve data quality, ensure traceability and increase efficiency within engineering processes. But this approach is not very suitable for the LC management, because it focuses mainly on the hardware management. Unlike PLM, ALM was originally considered as an approach that can theoretically be used without the use of appropriate tools. ALM was mainly proposed to be used to integrate all processes, methods and data that are used and generated within the framework of software development [4]. The modern vision distinguishes three main aspects within ALM: governance, development and operations [6]. The governance aspect considers the tasks of conducting software business analysis, project portfolio management and program portfolio management. The development aspect considers the tasks of software generation (from requirements definition through design and source code development to testing and release). The operation aspect considers the tasks of monitoring and managing software [8]. However, such a distribution of tasks between individual ALM aspects is too limited and does not correspond to the modern practice of maintaining and supporting operated IS. Thus, in particular, during IS operation there is a constant need to improve existing and develop new system functions as a response to changes in the business processes of IS consumer enterprises.

Modern research suggests understanding SLM as a technical and organizational basis for LC management of all system artifacts that are created or improved during the existence of this system, and the tools used to work with these artifacts. This basis provides clear tracking of all system elements. Thus, interdisciplinary SLM is considered the basis for complex digital engineering [9]. But this basis does not have the formal basis necessary for the SLM application as a theoretical and applied tool for solving complex tasks of LC management of the entire IS. Thus, one of the manifestations of this deficiency is the fact that there is no established classification of IT products that can be used to implement the SLM approach.

Unlike theoretical research, applied work in the field of creating IT products that can be used to implement the SLM approach demonstrates significant results. Examples of such IT products include:

– Microsoft products (Microsoft System Center 2016, Microsoft Azure Automation & Control in Microsoft Operations Management Suite);
– specialized products of individual IT companies (Cortex, ActiveBatch Workload Automation, Ignio, VMware Vrealize Orchestrator, etc.);
– products that are descendants of CASE systems and visual modeling environments (CA Process Automation).

These and similar IT products generally meet the requirements for service management systems defined in [1]. In general, the set of functions of such products is aimed at achieving the following goals:

– increasing the efficiency of the functioning of IT processes of the enterprise-consumer of IT services (technical and economic goal);
– reducing the operational risks of the enterprise-consumer of IT services (technical and economic goal);
– reducing the costs of IT processes of the enterprise-consumer of IT services throughout the entire LC (economic goal);
– compensating for the negative consequences of the complication of managing the IT services system of the enterprise-consumer, in particular, eliminating the "IT blindness" effect (economic goal);
– increasing the level of compliance with standards, including to facilitate further certification (economic and organizational and methodological goal).

But the vast majority of these products are the results of solving individual application tasks or developing a previous product line for a different purpose (CA Process Automation). The application of SLM provisions when creating these products does not find sufficient confirmation.

Unlike applied developments, most theoretical research in the field of PLM, ALM and SML is aimed at developing individual aspects of managing IT services of large industrial systems based on the Internet of Things. For example, in [10] a description of the information layer of the reference architectural model RAMI 4.0 is proposed, which is supported by the Zachman Framework concept. The proposed description, according to the authors, should ensure the practical use of RAMI 4.0, allowing stakeholders to use system engineering based on the model, on the one hand, and include information engineering, on the other. But the description proposed in [10] can be used to develop only individual aspects of industrial systems and does not cover the task of managing industrial IS as a whole. Similar research is being conducted in the IT industry. Thus, in [11] the results of mixed modeling and modeling of continuous service delivery pipeline scenarios as a separate aspect of DevOps are described. But these studies also do not cover the overall task of managing IS and IT, which an IT company uses in its activities. The formation of integral assessments of the progress and results of the functioning of individual IS services currently does not involve the use of formal models [12]. At the same time, studies have shown that the success of projects aimed at improving individual processes is not a coincidence, but a predictable result with clearly defined and measurable characteristics [13]. To solve various IT service management tasks, approaches based on solving multi-criteria optimization tasks are proposed in [14]. However, the difficulties of applying such approaches in managing real web-based IS force to look for other options for solving such problem. Thus, in [15] it is proposed to consider an approach to web service management based on knowledge. However, this approach is not yet fully developed and requires extensive additional research.

A significant number of studies consider the tasks of IS LC management mainly through the prism of instrumental means of its solution. In this case, the multi-criteria decision-making process is usually proposed as the main tool. Decision support systems based on multidimensional data warehouses are proposed to implement this process. An example of such a platform designed to automate the management of the LC stages that describe the development of modern software is considered in [16]. A significant part of the publications is devoted to describing the results of the study of individual tasks and tasks that arise when using tools for solving similar multi-criteria optimization tasks. Algorithms [17] and tools [18] for analyzing and optimizing hyperparameters are considered as similar tools in modern studies. Studies devoted to the use of simulation modeling tools for LC management of individual aspects of an IS [19] are not left without attention. But the main drawback of these and similar studies is the use of existing instrumental means for solving management tasks to solve the task of IS LC management.

 ISSN-L 2664-9969; E-ISSN 2706-5448

At the same time, the assumption about the possibility of using these tools in such studies is recognized as an axiom that cannot be proven, but can be refuted.

The modern circle of researchers recognizes the need for research on the formulation and development of the theoretical basis of SLM. The following directions for further research on the SLM development are indicated in [6]:

- improving the implementation of the knowledge management process in business (in particular, ensuring the interconnection of business goals and indicators with current processes in order to identify awareness of complaints and the potential for business improvement [20]);
- developing an integrated or common ALM-PLM data model (in particular, in order to identify and minimize differences between the descriptions of the hardware and software development processes [21]);
- extending the requirements and use cases of the ALM-PLM system to other LC stages;
- further detailing and refining the SLM, PLM and ALM approaches, developing these approaches through new methods, ideas and processes to support software and hardware development;
- defining KPIs for ALM-PLM integration (in particular, to further evaluate and monitor the use of these approaches to assess their success or potential for improvement [22]).

But the SLM development is impossible without eliminating an important contradiction in the field of IT service management. The essence of this contradiction is as follows. A typical IT service management system, defined in [1], recognizes the goal of management as maintaining user satisfaction with existing IT services at the desired level for the Provider and Consumer of IT services. At the same time, the main quantitative criterion by which it is possible to determine the degree of achievement of this goal is the indicator of the number of reports of events that remained unsolved. But these incident reports arise as a result of the interaction of operators and users with existing IT services and the IT infrastructure that ensures the operation of these IT services. Thus, the IT service management system defined in [1] and the IT products that are created on the basis of the requirements for this system are aimed at maintaining stable and unchanging over time states of the software and hardware of the IT service system. Therefore, compliance with the requirements for a typical service management system defined in [1] does not allow optimizing the set of IT services taking into account the technical and economic features of changing business processes of individual consumers of IT services. Therefore, the study of the general formal formulation of the IS LC management problem is relevant both from a theoretical and practical point of view.

The aim of research is to develop a concept for the task of global LC management of an operated IS as a system of IT services, taking into account the technical and economic features of business processes subject to automation. This will allow automating the solution of the tasks of IS LC management for the management of enterprises and organizations, taking into account the features of SLM and ALM-PLM integration.

To achieve this aim, it is proposed to solve the following objectives:

- to determine the purpose of classical (permanent) LC management of an operated IS as a system of IT services;
- to formulate a concept and develop formal descriptions of the function, purpose and constraints of the task of LC effective management of an operated IS as an IT service system.

2. Materials and Methods

The object of research is the processes of functioning and maintenance of IS, which together determine the stage of IS operation. *The subject of research* is a formal description of the purpose and main limitations of the tasks of IS operation management.

The generalized model of LC management of operated IS is based on the results of improving the methodology for managing the operation of web-based IS. These results in [23] are presented in the form of the following initial assumptions and recommendations:

- operated IS should be considered as a set of interconnected representations at different levels (business level, IT service level, IT service level, IT infrastructure level);
- the main conditional "unit" of describing data processing actions in managed IS of any level of representation is a transaction;
- the dimensionality of the IS operation management problem can be reduced provided that individual IS properties are identified, which characterize the manifestations of individual qualities of this system;
- each individual property of the IS is determined by a set of requirements (functional and non-functional) and a set of requests for change (RFC) of these requirements and individual elements of the system.

For a formal description of the elements of the generalized management model, it was decided to use the ontology of web-based IS service management proposed in [23]. This ontology consists of the following interconnected concepts: "Information System"; "Provider of IS"; "Customer of IS"; "Requirement of IS"; "RFC IS"; "Configuration of IS"; "Presentation Layer"; "Configuration Item"; "Transaction"; "Configuration Item"; "State of IS"; "Planned state"; "Current state"; "Indicator"; "Library of Indicators"; "Property"; "Efficiency" [23]. Each concept is assigned a set-theoretic category, which, in turn, is determined by a set of objects and a set of morphisms.

The use of the ontology proposed in [23] allows to form a theoretical and categorical description of an operated IS according to the "Digital Twin" concept. This concept proposes to solve the tasks of managing complex real-world objects by using their electronic representation – a digital analogue, which is actually no different from such an object [24]. Although the "Digital Twin" concept is used mainly for managing material objects (aircraft, cars, power plants, industrial robots, etc.), a number of modern studies recognize the expediency of using this concept in software engineering [25]. The generalized model of IS LC management, taking into account the features of the "Plan – Do – Check – Act" cycle, is presented in the form of a supercategory, which has the form [23]:

$$M_O = \left[L_P, L_D, L_{Ch}, L_A, F_{L_D}^{L_P}, F_{L_{Ch}}^{L_D}, F_{L_D}^{L_{Ch}}, F_{L_A}^{L_{Ch}}, F_{L_P}^{L_A} \right], \qquad (1)$$

where L_P – supercategory, which formally describes the "Plan" stage; L_D – supercategory, which formally describes the "Do" stage; L_{Ch} – supercategory, which formally describes the "Check" stage; L_A – supercategory, which formally describes the "Act" stage; $F_{L_D}^{L_P}$ – single covariant functor, which establishes a connection between supercategories L_P and L_D; $F_{L_D}^{L_{Ch}}$ – single covariant functor, which establishes a connection between supercategories L_{Ch} and L_D; $F_{L_{Ch}}^{L_D}$ – single covariant functor, which establishes a connection between supercategories L_D and L_{Ch}; $F_{L_A}^{L_{Ch}}$ – single covariant functor, which establishes a connection between supercategories L_{Ch} and L_A; $F_{L_P}^{L_A}$ – single covariant functor, which establishes a connection between supercategories L_A and L_P.

Functors $F_{L_D}^{L_P}$, $F_{L_{Ch}}^{L_D}$, $F_{L_A}^{L_{Ch}}$ and $F_{L_P}^{L_A}$ connect supercategories L_P, L_D, L_{Ch} and L_A according to the control cycle of an operated IS. The functor $F_{L_D}^{L_{Ch}}$ establishes a connection between supercategories L_{Ch} and L_D to describe situations when the IS operation is carried out according to the developed plans and there is no need to adjust these plans.

To detail the formal description of the supercategories L_P, L_D, L_{Ch} and L_A introduced in [23], the category of description of the states of an operated IS L_{St} was divided into two subcategories: the subcategory of description of the planned state of an operated IS L_{St}^P and the subcategory of description of the current state of an operated IS L_{St}^C. The subcategory L_{St}^P has the form:

$$L_{St}^{P}=\begin{bmatrix} Ob_{Conf}, Ob_{P\_L}, Ob_{CI}, Ob_{Tr}, Ob_{St}, Ob_{Pl\_St}, H_{Ob_{Conf}}^{Ob_{P\_L}}, H_{Ob_{Conf}}^{Ob_{CI}}, \\ H_{Ob_{Conf}}^{Ob_{Tr}}, H_{Ob_{Tr}}^{Ob_{CI}}, H_{Ob_{St}}^{Ob_{Conf}}, H_{Ob_{Pl\_St}}^{Ob_{St}}, H_{1_{Ob}} \end{bmatrix}, \quad (2)$$

where Ob_{Conf} – a subset of objects of the subcategory L_{St}^{P}, which describes the "Configuration of IS" concept; Ob_{P_L} – a subset of objects of the subcategory L_{St}^{P}, which describes the "Presentation Layer" concept; Ob_{CI} – a subset of objects of the subcategory L_{St}^{P}, which describes the "Configuration Item" concept; Ob_{Tr} – a subset of objects of the subcategory L_{St}^{P}, which describes the "Transaction" concept; Ob_{St} – a subset of objects of the subcategory L_{St}^{P}, which describes the "State of IS" concept; Ob_{Pl_St} – a subset of objects of the subcategory L_{St}^{P}, which describes the "Planned State" concept; $H_{Ob_{Conf}}^{Ob_{P\_L}}$ – a subset of morphisms of the subcategory L_{St}^{P}, defined between subsets Ob_{P_L} and Ob_{Conf}; $H_{Ob_{Conf}}^{Ob_{CI}}$ – a subset of morphisms of the subcategory L_{St}^{P}, defined between subsets Ob_{CI} and Ob_{Conf}; $H_{Ob_{Conf}}^{Ob_{Tr}}$ – a subset of morphisms of the subcategory L_{St}^{P}, defined between subsets Ob_{Tr} and Ob_{Conf}; $H_{Ob_{Tr}}^{Ob_{CI}}$ – a subset of morphisms of the subcategory L_{St}^{P} defined between the subsets Ob_{CI} and Ob_{Tr}; $H_{Ob_{St}}^{Ob_{Conf}}$ – a subset of morphisms of the subcategory L_{St}^{P} defined between the subsets Ob_{Conf} and Ob_{St}; $H_{Ob_{Pl\_St}}^{Ob_{St}}$ – a subset of morphisms of the subcategory L_{St}^{P} defined between the subsets Ob_{St} and Ob_{Pl_St}, $H_{1_{Ob}}$ – a subset of unit morphisms defined on selected subsets of objects of the subcategory L_{St}^{P}. The subcategory L_{St}^{C} has the form:

$$L_{St}^{C}=\begin{bmatrix} Ob_{Conf}, Ob_{P\_L}, Ob_{CI}, Ob_{Tr}, Ob_{St}, Ob_{C\_St}, H_{Ob_{Conf}}^{Ob_{P\_L}}, H_{Ob_{Conf}}^{Ob_{CI}}, \\ H_{Ob_{Conf}}^{Ob_{Tr}}, H_{Ob_{Tr}}^{Ob_{CI}}, H_{Ob_{St}}^{Ob_{Conf}}, H_{Ob_{C\_St}}^{Ob_{St}}, H_{1_{Ob}} \end{bmatrix}, \quad (3)$$

where Ob_{Conf} – a subset of objects of the subcategory L_{St}^{C}, which describes the "Configuration of IS" concept; Ob_{P_L} – a subset of objects of the subcategory L_{St}^{C}, which describes the "Presentation Layer" concept; Ob_{CI} – a subset of objects of the subcategory, L_{St}^{C} which describes the "Configuration Item" concept; Ob_{Tr} – a subset of objects of the subcategory L_{St}^{C}, which describes the "Transaction" concept; Ob_{St} – a subset of objects of the subcategory L_{St}^{C}, which describes the "State of IS" concept; Ob_{C_St} – a subset of objects of the subcategory L_{St}^{C}, which describes the "Current State" concept; $H_{Ob_{Conf}}^{Ob_{P\_L}}$ – a subset of morphisms of the subcategory L_{St}^{C}, defined between subsets Ob_{P_L} and Ob_{Conf}; $H_{Ob_{Conf}}^{Ob_{CI}}$ – a subset of morphisms of the subcategory L_{St}^{C}, defined between subsets Ob_{CI} and Ob_{Conf}; $H_{Ob_{Conf}}^{Ob_{Tr}}$ – a subset of morphisms of the subcategory L_{St}^{C}, defined between subsets Ob_{Tr} and Ob_{Conf}; $H_{Ob_{Tr}}^{Ob_{CI}}$ – a subset of morphisms of the subcategory L_{St}^{C} defined between subsets Ob_{CI} and Ob_{Tr}; $H_{Ob_{St}}^{Ob_{Conf}}$ – a subset of morphisms of the subcategory L_{St}^{C} defined between subsets Ob_{Conf} and Ob_{St}; $H_{Ob_{C\_St}}^{Ob_{St}}$ – a subset of morphisms of the subcategory L_{St}^{C} defined between subsets Ob_{St} and Ob_{C_St}. $H_{1_{Ob}}$ – a subset of unit morphisms defined on selected subsets of objects of the subcategory L_{St}^{C}.

Then, taking into account previously developed categorical descriptions of operated ISs, it is proposed to describe the supercategory L_P as follows [23]:

$$L_P=\left[L_{IS}, L_{St}^{P}, L_{Ind}, F_{L_{St}^{P}}^{L_{IS}}, F_{L_{Ind}}^{L_{St}^{P}}\right], \quad (4)$$

where L_{IS} – a category that describes the operated IS at different levels of representation; L_{Ind} – a category that describes the indicators that characterize the operated IS and its configuration elements; $F_{L_{St}^{P}}^{L_{IS}}$ – a single covariant functor that establishes a connection between the categories L_{IS} and L_{St}^{P}; $F_{L_{Ind}}^{L_{St}^{P}}$ – a single covariant functor that establishes a connection between the categories L_{St}^{P} and L_{Ind}.

It is proposed to describe the supercategory L_D as follows [23]:

$$L_D=\left[L_{IS}, L_{St}^{C}, L_{Ind}, F_{L_{St}^{C}}^{L_{IS}}, F_{L_{Ind}}^{L_{St}^{C}}\right], \quad (5)$$

where $F_{L_{St}^{C}}^{L_{IS}}$ – a single covariant functor that establishes a connection between categories L_{IS} and L_{St}^{C}; $F_{L_{Ind}}^{L_{St}^{C}}$ – a single covariant functor that establishes a connection between supercategories L_{St}^{C} and L_{Ind}.

The supercategory L_{Ch} is proposed to be described as follows [23]:

$$L_{Ch}=\left[L_{IS}, L_{St}^{P}, L_{St}^{C}, L_{Ind}, F_{L_{St}^{P}}^{L_{IS}}, F_{L_{St}^{C}}^{L_{IS}}, F_{L_{Ind}}^{L_{St}^{P}}, F_{L_{Ind}}^{L_{St}^{C}}\right]. \quad (6)$$

The supercategory L_A is proposed to be described as follows [23]:

$$L_A=\left[L_{IS}, L_{St}^{P}, L_{St}^{C}, L_{Ind}, F_{L_{St}^{P}}^{L_{IS}}, F_{L_{St}^{C}}^{L_{IS}}, F_{L_{Ind}}^{L_{St}^{P}}, F_{L_{Ind}}^{L_{St}^{C}}\right], \quad (7)$$

where L_{Eff} – a category that describes the IS effectiveness by its individual properties and as a whole; $F_{L_{Eff}}^{L_{IS}}$ – a single covariant functor that establishes a connection between supercategories L_{IS} and L_{Eff}.

The diagram of the relationships between the elements of the generalized model of IS LC management is shown in Fig. 1.

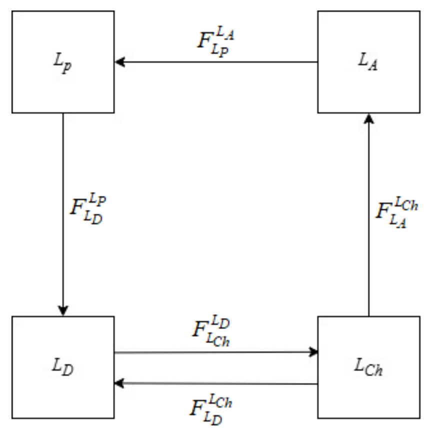

Fig. 1. Diagram of relationships between elements of the generalized model of IS LC management

The diagram shown in Fig. 1 establishes transitions between the main states of the "Plan – Do – Check – Act" cycle of IS LC management. In this diagram, states are defined as supercategories L_P, L_D, L_{Ch} and L_A, describing the corresponding stages of the mentioned cycle. Transitions between these states are shown as functors $F_{L_D}^{L_P}$, $F_{L_{Ch}}^{L_D}$, $F_{L_D}^{L_{Ch}}$, $F_{L_A}^{L_{Ch}}$ and $F_{L_P}^{L_A}$. Thus, model (1) can be represented as a finite state machine, transitions between the states of which are associated with changing the alphabets of the description of these states, although they retain their structural features. Such a representation allows in further studies to consider model (1) as the basis for a formal description of automated IS LC management based on the "Plan – Do – Check – Act" cycle. This description can exist provided that it is proven possible to transform the corresponding finite state machine into a consistent set of simpler finite state machines that describe the IT processing of data structures at each stage of the cycle.

3. Results and Discussion

3.1. Formal description of the purpose of classical life cycle management of an operated information system as an IT service system

For a formal description of the purpose of classical LC management of an operated IS, let's introduce a detailed description of the categories L_{IS}, L_{Ind} and L_{Eff}. The category L_{IS}, taking into account the concepts of the ontology of web-based IS service management identified in [23] and the connections between these concepts, will have the form:

$$L_{IS}=\begin{bmatrix} Ob_{IS}, Ob_{Pr}, Ob_{Cust}, Ob_{rec}, Ob_{RFC}, Ob_{Conf}, Ob_{P\_L}, Ob_{CI}, Ob_{Tr}, \\ H_{Ob_{IS}}^{Ob_{Pr}}, H_{Ob_{Cust}}^{Ob_{IS}}, H_{Ob_{rec}}^{Ob_{IS}}, H_{Ob_{rec}}^{Ob_{Pr}}, H_{Ob_{IS}}^{Ob_{Cust}}, H_{Ob_{Conf}}^{Ob_{rec}}, H_{Ob_{Conf}}^{Ob_{P\_L}}, \\ H_{Ob_{Conf}}^{Ob_{CI}}, H_{Ob_{Conf}}^{Ob_{Tr}}, H_{Ob_{Tr}}^{Ob_{CI}}, H_{Ob_{RFC}}^{Ob_{Pr}}, H_{Ob_{RFC}}^{Ob_{Cust}}, H_{1_{Ob}} \end{bmatrix}, \quad (8)$$

where Ob_{IS} – a subset of category L_{IS} objects that describes the "Information System" concept; Ob_{Pr} – a subset of category L_{IS} objects that describes the "Provider of IS" concept; Ob_{Cust} – a subset of category L_{IS} objects that describes the "Customer of IS" concept; Ob_{rec} – a subset of category L_{IS} objects that describes the "Requirement of IS" concept; Ob_{RFC} – a subset of category L_{IS} objects that describes the "RFC IS" concept; $H_{Ob_{IS}}^{Ob_{Pr}}$ – a subset of category L_{IS} morphisms defined between subsets Ob_{Pr} and Ob_{IS}; $H_{Ob_{Cust}}^{Ob_{IS}}$ – a subset of category L_{IS} morphisms defined between subsets Ob_{IS} and Ob_{Cust}; $H_{Ob_{rec}}^{Ob_{IS}}$ – a subset of category L_{IS} morphisms defined between subsets Ob_{IS} and Ob_{rec}; $H_{Ob_{IS}}^{Ob_{Cust}}$ – a subset of category L_{IS} morphisms defined between subsets Ob_{Cust} and Ob_{IS}; $H_{Ob_{RFC}}^{Ob_{Pr}}$ – a subset of category L_{IS} morphisms defined between subsets Ob_{Pr} and Ob_{RFC}; $H_{Ob_{RFC}}^{Ob_{Cust}}$ – a subset of category L_{IS} morphisms defined between subsets Ob_{Cust} and Ob_{RFC}; $H_{1_{Ob}}$ – a subset of unit morphisms defined on selected subsets of category L_{IS} objects.

The category L_{Ind}, taking into account the concepts of web-based IS service management ontology identified in [23] and the relationships between these concepts, will have the form:

$$L_{Ind}=\left[Ob_{St},Ob_{Ind},Ob_{Lib},H_{Ob_{St}}^{Ob_{Ind}},H_{Ob_{Lib}}^{Ob_{Ind}},H_{1_{Ob}}\right],\qquad(9)$$

where Ob_{St} – a subset of category L_{Ind} objects that describes the "State of IS" concept; Ob_{Ind} – a subset of category L_{Ind} objects that describes the "Indicator" concept; Ob_{Lib} – a subset of category L_{Ind} objects that describes the "Library of Indicators" concept; $H_{Ob_{St}}^{Ob_{Ind}}$ – a subset of category L_{Ind} morphisms defined between subsets Ob_{Ind} and Ob_{St}; $H_{Ob_{Lib}}^{Ob_{Ind}}$ – a subset of category L_{Ind} morphisms defined between subsets Ob_{Ind} and Ob_{Lib}; $H_{1_{Ob}}$ – a subset of unit morphisms defined on selected subsets of category L_{Ind} objects.

The category L_{Eff}, taking into account the concepts of web-based IS service management ontology identified in [23] and the relationships between these concepts, will have the form:

$$L_{Eff}=\begin{bmatrix}Ob_{St},Ob_{Pl\_St},Ob_{C\_St},Ob_{Ind},Ob_{P\_L},Ob_{Prop},Ob_{Eff},Ob_{RFC},\\ H_{Ob_{Pl\_St}}^{Ob_{St}},H_{Ob_{C\_St}}^{Ob_{St}},H_{Ob_{St}}^{Ob_{Ind}},H_{Ob_{Prop}}^{Ob_{St}},H_{Ob_{Prop}}^{Ob_{P\_L}},H_{Ob_{Prop}}^{Ob_{Eff}},\\ H_{Ob_{Pl\_St}}^{Ob_{Eff}},H_{Ob_{C\_St}}^{Ob_{Eff}},H_{Ob_{RFC}}^{Ob_{St}},H_{1_{Ob}}\end{bmatrix},\qquad(10)$$

where Ob_{Prop} – a subset of category L_{Eff} objects that describes the "Property" concept; Ob_{Eff} – a subset of category L_{Eff} objects that describes the "Efficiency" concept; $H_{Ob_{Prop}}^{Ob_{St}}$ – a subset of category L_{Eff} morphisms defined between subsets Ob_{St} and Ob_{Prop}; $H_{Ob_{Prop}}^{Ob_{P\_L}}$ – a subset of category L_{Eff} morphisms defined between subsets Ob_{P_L} and Ob_{Prop}; $H_{Ob_{Prop}}^{Ob_{Eff}}$ – a subset of category L_{Eff} morphisms defined between subsets Ob_{Eff} and Ob_{Prop}; $H_{Ob_{Pl\_St}}^{Ob_{Eff}}$ – a subset of category L_{Eff} morphisms defined between subsets Ob_{Eff} and Ob_{Pl_St}; $H_{Ob_{C\_St}}^{Ob_{Eff}}$ – a subset of category L_{Eff} morphisms defined between subsets Ob_{Eff} and Ob_{C_St}; $H_{Ob_{RFC}}^{Ob_{St}}$ – a subset of category L_{Eff} morphisms defined between subsets Ob_{St} and Ob_{RFC}; $H_{1_{Ob}}$ – a subset of unit morphisms defined on selected subsets of category L_{Eff} objects.

Using a generalized model of the LC management of an operated IS (1) allows to formally describe the global task of LC management of an operated IS as an IT service system. In this case, let's proceed from the assumption that the main approach to managing an IT service system is the "Plan – Do – Check – Act" cycle.

As noted above, the classical management of an operated IT service system is management aimed at minimizing the number of incident reports that remain unresolved. Therefore, it is proposed to consider the target state of an operated IT service system as a state in which the number of incident reports and RFCs of the managed IT service system tends to 0. Let's call such a state of an operated IS as an IT service system stable.

The aspiration of an operated IS to a stable state is proposed to be described by a goal function that has the following form:

$$\left|(Tr_i)_{i=1,n}\right|+\left|(RFC_j)_{j=1,n}\right|\rightarrow 0,\qquad(11)$$

where Tr_i – transaction, the execution of which generates the i-th incident; n – number of incidents that occur during the IS operation; RFC_j – j-th RFC; m – number of RFCs that occur during the IS operation.

The objective function (11) of LC sustainable management of an operated IS is limited by the following conditions:

a) for each incident during its elimination, the planned and current states in which the operated IS may be during the control of its operation must be determined:

$$\forall Tr_i\in Ob_{Tr}\in L_{IS}\in L_D\exists\left(L_{St}^P\in L_{Ch},L_{St}^C\in L_{Ch}\right);\qquad(12)$$

b) for each RFC during its implementation, the planned and current states in which the operated IS may be during the control of its operation must be determined:

$$\forall RFC_i\in Ob_{RFC}\in L_{IS}\in L_D\exists\left(L_{St}^P\in L_{Ch},L_{St}^C\in L_{Ch}\right);\qquad(13)$$

c) the current state of the IS during the control of its operation must fully correspond to its planned state (based on the characteristics of the cycle and the "Plan – Do – Check – Act" model (1)):

$$\left|(L_{St}^P\in L_{Ch}-L_{St}^C\in L_{Ch})\bigcap(L_{St}^C\in L_{Ch}-L_{St}^P\in L_{Ch})\right|=0.\qquad(14)$$

Such a statement of the task of LC management of an operated IS will allow obtaining an IS that will be best adapted to the requirements of its users. But such adaptability does not guarantee effective IS operation, because the set of requirements and RFCs themselves may not be aimed at achieving effective planned and current states of an operated IS. Therefore, an attempt to achieve goal (11) may lead to an increase in RFC. This increase is due to the fact that the number of RFCs that will arise during the IS operation can increase only because previous solutions to the management task will not give the desired coincidence of planned and current states. This situation will be especially pronounced in cases where the business processes of the automation object, the IS itself and the IT infrastructure in which the IS is operated are in a state of constant change.

3.2. Concept and model of the task of effective LC management of an operated information system as an IT service system

To overcome this shortcoming of sustainable LC management of an operated IS, it is proposed to apply elements of the theory of potential efficiency. This theory allows to formally describe the task of effective LC management of an operated IS based on relatively simple models that describe individual qualities of such systems [26].

The application of the theory of potential efficiency allows to formulate the concept of effective LC management of an operated IS. This concept defines the effectiveness of IS LC management as the probability of a beneficial exchange between the system and its environment. Such an environment for web-based IS of enterprise management is the IT infrastructure and business processes of the enterprise-consumer of this IS. The IS itself, according to this concept, is considered as a set of IT services provided by the IT company-Supplier of this IS in response to the requirements of the Consumer.

Unlike the existing concept of classical (permanent) management, the proposed concept of effective LC management of an operated IS uses the additional concept of "system property" [24]. At the same time, only those system properties that characterize the manifestation of individual IS qualities are of interest to the Consumer and Supplier of IS. IS qualities, in turn, are determined by a set of requirements (functional and non-functional) that are imposed on the IS before the start of its operation, as well as a set of IS RFCs and its individual configuration elements formed during the IS operation. In this case, effective LC management of an operated IS will be defined as actions to achieve the planned results for each individual IS property and its configuration elements.

Then, achieving the global goal of effective LC management of an operated IS will be considered as the task of achieving optimal characteristics of this IS for each of its specific properties and the minimum probability of the existence of unresolved incidents and RFCs during the operation stage of this IS. The formal description of the objective function of this task will look like:

$$\begin{cases} f_{Eff}^{1}(Ob_{\text{Prop}}^{1} \in L_{Eff}, Ob_{Eff} \in L_{Eff}) \to opt^{1}, \\ f_{Eff}^{2}(Ob_{\text{Prop}}^{2} \in L_{Eff}, Ob_{Eff} \in L_{Eff}) \to opt^{2}, \\ \ldots \\ f_{Eff}^{z}(Ob_{\text{Prop}}^{z} \in L_{Eff}, Ob_{Eff} \in L_{Eff}) \to opt^{z}, \\ \sum_{i=1}^{n} P(Tr_i) + \sum_{j=1}^{m} P(RFC_j) \to \min, \end{cases} \quad (15)$$

where 1,2,..., z – identifiers of individual properties of an operated IS; $P(Tr_i)$ – probability of no beneficial exchange between the operated IS and its environment as a result of transaction Tr_i execution; $P(RFC_j)$ – probability of no beneficial exchange between the operated IS and its environment as a result of RFC implementation RFC_j.

The objective function of effective LC management of an operated IS (15) is limited by the following conditions:

a) for any specified k-th property of an operated IS, there must be a set of states of this IS, being in which the operated IS will be considered effective for this property:

$$\begin{aligned} &\forall ob_{Eff}^{k} \in Ob_{Eff} \in L_{Eff} \in L_A \to \\ &\to opt^{k} \exists \left(Ob_{St} \in L_{IS} \in L_D, Ob_{Ind} \in L_{IS} \in L_D \right), \end{aligned} \quad (16)$$

where $Ob_{Ind} \in L_{IS} \in L_D$ – a set of indicators by which, during the IS operation, the fact of this IS being in an effective state is established;

b) for each incident during its elimination, planned and current states must be determined, being in which the operated IS can be considered effective:

$$\forall Tr_i \in Ob_{Tr} \in L_{IS} \in L_D \exists \left(L_{St}^{P} \in L_{Ch}, L_{St}^{C} \in L_{Ch}, L_{Eff} \in L_A \right); \quad (17)$$

c) for each RFC during its implementation, planned and current states must be determined, being in which the operated IS can be considered effective:

$$\forall RFC_j \in Ob_{RFC} \in L_{IS} \in L_D \exists \left(L_{St}^{P} \in L_{Ch}, L_{St}^{C} \in L_{Ch}, L_{Eff} \in L_A \right); \quad (18)$$

d) the current state of the IS in the process of controlling its operation must approach its planned state (based on the features of the cycle and the "Plan – Do – Check – Act" model (1)) with some permissible small deviation:

$$\left| (L_{St}^{P} \in L_{Ch} - L_{St}^{C} \in L_{Ch}) \bigcap (L_{St}^{C} \in L_{Ch} - L_{St}^{P} \in L_{Ch}) \right| \le \varepsilon. \quad (19)$$

The task of sustainable LC management of an operated IS can be considered as a partial case of the task (15)–(19). For this case, the objective function (15) will have the form:

$$\begin{cases} f_{Eff}^{1}(Ob_{\text{Prop}}^{1} \in L_{Eff}, Ob_{Eff} \in L_{Eff}) \to opt^{1}, \\ f_{Eff}^{2}(Ob_{\text{Prop}}^{2} \in L_{Eff}, Ob_{Eff} \in L_{Eff}) \to opt^{2}, \\ \ldots \\ f_{Eff}^{z}(Ob_{\text{Prop}}^{z} \in L_{Eff}, Ob_{Eff} \in L_{Eff}) \to opt^{z}, \\ \sum_{i=1}^{n} P(Tr_i) + \sum_{j=1}^{m} P(RFC_j) \to 0, \end{cases} \quad (20)$$

and condition (19) will be reduced to condition (14).

The proposed formal description of the task of effective LC management of an operated IS will allow obtaining an IS that will, during its operation:

a) be as well adapted as possible to the requirements of its users and the features of the IT infrastructure and business processes of the automation object existing during operation;

b) be in those states that will be recognized as effective by both the Consumer and the Supplier (including based on their strategies and goals within the IS LC [2]).

3.3. Discussion of the research results

The study considers two main concepts of LC management of an operated IS as an IT service system – classical (permanent) and effective management. The concept of classical management is based on the currently existing point of view on the purpose of the IS operation as an IT service system. This point of view is the need to maximize the satisfaction of IS users based on the results of each iteration of the "Plan – Do – Check – Act" cycle. It is proposed to consider the IS state as a formal sign of such satisfaction as the IS state in which the number of incident reports and RFCs of an operated IS tends to 0. Based on this sign, a formal description (11) of the purpose of classical (permanent) LC management of an operated IS and limitations (12)–(14) is compiled.

Unlike the concept of permanent management, the proposed concept of effective LC management of an operated IS consists in presenting this system as a product with a set of individual properties. In this case, each IS property is considered as a manifestation of the individual qualities of this system. Each quality, in turn, is proposed to be considered as a result of fulfilling a set of requirements and IS RFC and its configuration elements. This concept allows to formally describe the task of LC effective management of an operated IS as a task of achieving optimal characteristics of this IS for each of its specific properties and the minimum probability of the existence of unresolved incidents and RFCs during the operation stage of this IS. In this case, the condition of implementation in the "Plan – Do – Check – Act" control cycle at the "Check" stage of classical (constant) control with an admissible small deviation must be met. Based on this concept, a formal description of the objective function of the effective LC management of an operated IS (15) and constraint systems (16)–(19) has been developed. The obtained results allow to define the task of classical (constant) LC management of an operated IS (20) as a partial case of the task of effective management. It should be noted that a significant number of modern studies in the field of PLM, ALM and SLM use the mathematical apparatus of graph theory for the formal description of the tasks to be solved. Examples of such studies are [27, 28]. In contrast, in this study, the mathematical apparatus of category theory is used to formally describe the LC management tasks of an operated IS. The theoretical and categorical models of an operated IS and the LC management tasks of this IS used and developed in this study are based on the ontology of web-based IS service management proposed by one of the authors of the study in [23] and allow to establish:

- the boundaries of domains and data structures used to describe individual ontology concepts;
- the rules for transforming domains and data structures when creating connections between different ontology concepts.

The application of the models developed in this study makes it possible to transition to effective LC management of an operated IS without global reengineering of systems and technologies for data storage and processing. The vast majority of IT products for LC management of an operated IS are highly specialized and relatively new products that occupy a small market segment. Therefore, systems for LC management of an operated IS are most often the result of integrating a large number of such highly specialized products. The proposed theoretical and categorical models allow in the following cases:

- to determine the domains and data structures for technologies for integrating heterogeneous IT products into a single system;

– to ensure the transition from classical to effective LC management of an IT service system without serious changes in the list of highly specialized products in operation.

Also important from an applied point of view is the proposed presentation of the existing task of classical (permanent) LC management of an operated IS as a partial case of the task of effective management. This representation allows to consider the design and implementation of an effective LC management system for operated IS as a gradual evolutionary development of the previously created and already operating system for the classical LC management of IT services.

The main limitation of this study is the need to use complete descriptions of all elements of this IS to solve the task of effective LC management of an operated IS. This means that during at least one iteration of the "Plan – Do – Check – Act" cycle, IS administrators must interact with a complete digital model of this IS at all levels of representation to solve individual tasks of LC management. The development and maintenance of such a model is a complex and costly task. But this limitation is methodological in nature and is valid for any work in the field of change management of developed or operated IS (which is confirmed, for example, by studies [29, 30]).

Another feature of the obtained results is their orientation on the LC management of an operated IS in conditions of constant changes in business processes, functions and elements of IT infrastructure. This feature is a consequence of the generalization of applied experience in supporting and ensuring the functioning of IS management of enterprises and organizations in the conditions of martial law, which is in force in Ukraine. It should also be noted that the conditions of martial law in force in Ukraine have determined the implementation of relationships between the authors in the process of this study exclusively with the use of tools for remote communication.

Although the obtained research results are methodological in nature, they should be recognized as important for further theoretical and applied research in the field of creating PLM-, ALM- and SLM-IS and IT. The main further directions of development of this study are:

– determination of the set of properties of an operated IS and performance indicators for these properties;
– development of new and improvement of existing models, detailing the formal description of the task of effective management;
– development and implementation of methods and algorithms for solving tasks of effective management of various IS and IT options.

4. Conclusions

A formal statement of the task of classical (constant) LC management of an operated IS as an IT service system has been developed. A formal description of the objective function (11) and constraints (12)–(14) of this task has been proposed using the apparatus of category theory and set theory. The developed formal statement is based on the existing definition of the goal of such management as maximizing the satisfaction of IS users based on the results of each iteration of the "Plan – Do – Check – Act" cycle of LC management of an operated IS. It is proposed to consider the IS state as a formal sign of such satisfaction as the number of incident reports and RFCs of an operated IS tending to 0. The results allow to formally describe the LC management task of an operated IS in conditions of constant (unchangeable) business processes and IT infrastructure of enterprises and organizations as objects of automation.

The concept of the task of effective LC management of an operated IS as an IT service system has been formulated. The formulated concept allows to formally describe the task of LC effective management of an operated IS as a task of achieving optimal characteristics of this IS for each of its specific properties and the minimum probability of the existence of unresolved incidents and RFC during the operation stage of this IS. A formal description of the objective function (15) and constraints (16)–(19) of this task is proposed using the apparatus of category theory, set theory and elements of probability theory. It is determined that the task of classical (constant) LC management of an operated IS is a partial case of the proposed task of effective management. Practical application of the proposed formal description of the LC effective management task of an operated IS allows to improve SLM-systems for LC management of an operated IS without global reengineering of existing systems and technologies for data storage and processing.

Acknowledgments

The authors of the study express their gratitude to the Candidate of Technical Sciences, Senior Researcher Olha Neumyvakina for participating in obtaining and preliminary processing of the research results.

Conflict of interest

The authors declare that they have no conflict of interest regarding this study, including financial, personal, authorship or other, which could affect the study and its results presented in this article.

Financing

This study was performed within the framework of the economic contract research work "Study of the results of monitoring a web-based information system in operation", which was carried out at the Kharkiv National University of Radio Electronics.

Data availability

The manuscript has no related data.

Use of artificial intelligence

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

References

1. *ISO/IEC 20000-1. Information technology – Service management – Part 1: Service management system requirements* (2018). Geneva: ISO Copyright Office, 96.
2. Levykin, V. M., Evlanov, M. V., Kernosov, M. A. (2014). *Patterny proektirovaniia trebovanii k informatcionnym sistemam: modelirovanie i primenenie*. Kharkov: OOO "Kompaniia "Smit", 320.
3. Stark, J. (2020). *Product Lifecycle Management (Volume 1)*. Cham: Springer International. https://doi.org/10.1007/978-3-030-28864-8
4. Schwaber, C. (2006). *The Changing Face of Application Life-Cycle Management*. Forrester Research Inc. Available at: https://www.yumpu.com/en/document/view/13866040/download-the-changing-face-of-application-lifecycle-mks Last accessed: 07.02.2025
5. Rizzo, S. (2016). *Why ALM and PLM need each other*. Siemens Whitepaper. Available at: https://polarion.plm.automation.siemens.com/hubfs/Docs/Whitepapers/why-alm-and-plm-need-each-other-whitepaper.pdf Last accessed: 07.02.2025
6. Wyrwich, F., Kharatyan, A., Dumitrescu, R. (2024). Interdisciplinary system lifecycle management – a systematic literature review. *Proceedings of the Design Society, 4,* 2765–2774. https://doi.org/10.1017/pds.2024.280
7. Liepert, C., Stary, C., Lamprecht, A., Zügn, D.; Elstermann, M., Lederer, M. (Eds.) (2025). Interoperable Product Change Management Within Engineering: A Digital Twin Approach. Subject-Oriented Business Process Management. Models for Designing Digital Transformations. *S-BPM ONE 2024. Communications in Computer and Information Science. Vol. 2206.* Cham: Springer. https://doi.org/10.1007/978-3-031-72041-3_17
8. Chappell, D. (2010). *What is Application Lifecycle Management?* David Chappell and Associates. Available at: http://davidchappell.com/writing/white_papers/What_is_ALM_v2.0--Chappell.pdf Last accessed 07.02.2025
9. Eigner, M. (2021). *System Lifecycle Management: Digitalisierung des Engineering*. Berlin, Heidelberg: Springer Vieweg. https://doi.org/10.1007/978-3-662-62183-7

10. Binder, C., Neureiter, C., Lüder, A. (2022). Towards a domain-specific information architecture enabling the investigation and optimization of flexible production systems by utilizing artificial intelligence. *The International Journal of Advanced Manufacturing Technology, 123 (1-2),* 49–81. https://doi.org/10.1007/s00170-022-10141-2
11. Colantoni, A., Berardinelli, L., Garmendia, A., Bräuer, J. (2022). Towards Blended Modeling and Simulation of DevOps Processes: the Keptn Case Study. *MODELS '22: Proceedings of the 25th International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings, Association for Computing Machinery.* New York, 784–792. https://doi.org/10.1145/3550356.3561597
12. Gulzar, K., Ruusu, R., Sierla, S., Aarnio, P., Karhela, T., Vyatkin, V. (2018). Automatic Generation of a Lifecycle Analysis Model from a First Principles Industrial Process Simulation Model. *2018 IEEE 16th International Conference on Industrial Informatics (INDIN).* Danvers, 741–746. https://doi.org/10.1109/indin.2018.8471980
13. Calderon, N. N., Kajko-Mattsson, M., Nolan, A. J. (2015). Successful process improvement projects are no accidents. *Journal of Software: Evolution and Process, 27 (11),* 896–911. https://doi.org/10.1002/smr.1738
14. Reiff-Marganiec, S., Tilly, M. (Eds.) (2012). *Handbook of Research on Service-Oriented Systems and Non-Functional Properties: Future Directions.* IGI Global. https://doi.org/10.4018/978-1-61350-432-1
15. Driss, M., Aljehani, A., Boulila, W., Ghandorh, H., Al-Sarem, M. (2020). Servicing Your Requirements: An FCA and RCA-Driven Approach for Semantic Web Services Composition. *IEEE Access, 8,* 59326–59339. https://doi.org/10.1109/access.2020.2982592
16. Kienzle, J., Combemale, B., Mussbacher, G., Alam, O., Bordeleau, F., Burgueno, L. et al. (2022). Global Decision Making Over Deep Variability in Feedback-Driven Software Development. *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering.* New York. https://doi.org/10.1145/3551349.3559551
17. Moosbauer, J., Binder, M., Schneider, L., Pfisterer, F., Becker, M., Lang, M. et al. (2022). Automated Benchmark-Driven Design and Explanation of Hyperparameter Optimizers. *IEEE Transactions on Evolutionary Computation, 26 (6),* 1336–1350. https://doi.org/10.1109/tevc.2022.3211336
18. Garouani, M., Ahmad, A., Bouneffa, M., Hamlich, M., Bourguin, G., Lewandowski, A. (2022). Using meta-learning for automated algorithms selection and configuration: an experimental framework for industrial big data. *Journal of Big Data, 9 (1).* https://doi.org/10.1186/s40537-022-00612-4
19. Bush, B., Stright, D., Huggins, J., Newes, E. (2022). Simulation process and data flow for a large system dynamics model. *Simulation, 98 (9),* 823–833. https://doi.org/10.1177/00375497221093381
20. Ebert, C. (2013). Improving engineering efficiency with PLM/ALM. *Software & Systems Modeling, 12 (3),* 443–449. https://doi.org/10.1007/s10270-013-0347-3
21. Deuter, A., Imort, S. (2020). PLM/ALM Integration With The Asset Administration Shell. *Procedia Manufacturing, 52,* 234–240. https://doi.org/10.1016/j.promfg.2020.11.040
22. Deuter, A., Otte, A., Höllisch, D. (2017). Methodisches Vorgehen zur Entwicklung und Evaluierung von Anwendungsfällen für die PLM/ALM-Integration. *Wissenschaftsforum Intelligente Technische Systeme (WInTeSys).* Paderborn, 211–222. https://doi.org/10.17619/UNIPB/1-93
23. Petrichenko, O. V. (2021). Improving enterprise IT-service management methodology. *Management Information System and Devises, 177,* 4–12. https://doi.org/10.30837/0135-1710.2021.177.004
24. Kanaga Priya, P., Reethika, A.; Mishra, A., El Barachi, M., Kumar, M. (Eds.) (2024). A Review of Digital Twin Applications in Various Sectors. *Transforming Industry using Digital Twin Technology.* Cham: Springer, 239–258. https://doi.org/10.1007/978-3-031-58523-4_12
25. Guinea-Cabrera, M. A., Holgado-Terriza, J. A. (2024). Digital Twins in Software Engineering – A Systematic Literature Review and Vision. *Applied Sciences, 14 (3),* 977. https://doi.org/10.3390/app14030977
26. Fleishman, B.; Patten, B., Jorgenson, S. (Eds). (1995). Stochastic Theory of Complex Ecological Systems. *Complex Ecology.* Prentice Hall PTP, Prentice Hall Inc, A. Simon & Schuster, Englewood Cliffs, New Jersey, 07632, 166–224.
27. Zimmermann, T. C., Konietzko, E., Lindow, K. (2024). Graph-based Parameter Management for Configuration Controlled Multi-level Modeling of Cyber-physical Systems. *2024 19th Annual System of Systems Engineering Conference (SoSE),* 270–274. https://doi.org/10.1109/sose62659.2024.10620964
28. Katzung, S., Cinkaya, H., Kizgin, U. V., Savinov, A., Baschin, J., Vietor, T. (2024). AI-based analysis and linking of technical and organisational data using graph models as a basis for decision-making in systems engineering. *Proceedings of the Design Society, 4,* 2625–2634. https://doi.org/10.1017/pds.2024.265
29. Rostami, K., Stammel, J., Heinrich, R., Reussner, R. (2017). Change Impact Analysis by Architecture-based Assessment and Planning. *Lecture Notes in Informatics, Proceedings – Series of the Gesellschaft fur Informatik.* Hannover, 267, 69–70.
30. Rostami, K., Heinrich, R., Busch, A., Reussner, R. (2017). Architecture-Based Change Impact Analysis in Information Systems and Business Processes. *2017 IEEE International Conference on Software Architecture (ICSA).* Gothenburg, 179–188. https://doi.org/10.1109/icsa.2017.17

Viktor Levykin, *Doctor of Technical Science, Department of Information Control Systems, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine, ORCID: https://orcid.org/0000-0002-7929-515X*

✉***Maksym Ievlanov,*** *Doctor of Technical Science, Department of Information Control Systems, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine, e-mail: maksym.ievlanov@nure.ua, ORCID: https://orcid.org/0000-0002-6703-5166*

Ihor Levykin, *Doctor of Technical Science, Department of Media Systems and Technologies, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine, ORCID: https://orcid.org/0000-0001-8086-237X*

Oleksandr Petrychenko, *PhD, Department of Information Control Systems, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine, ORCID: https://orcid.org/0000-0002-1319-5041*

✉*Corresponding author*