DOI: 10.17747/2618-947X-2024-3-270-279
УДК 338.984
JEL M15

Technological features of creating IT products within the framework of implementing IT projects

V.S. Nikolaenko[1, 2, 3]
[1] Tomsk State University of Control Systems and Radioelectronics (Tomsk, Russia)
[2] Tomsk Polytechnic University (Tomsk, Russia)
[3] Siberian State Medical University (Tomsk, Russia)

Abstract
The article examines the characteristics of ECM programs, such as incrementalism and high technology, as well as the technological features IT project management that create similar characteristics. The purpose of the article is to analyse the technological features of IT project management that arise during the creation of ECM programs. The study found that waterfall and flexible approaches, fourteen techniques and four lifecycle models are used to create ECMr programs. In addition, it was found that every IT project implemented according to the waterfall life cycle model, regardless of its size, complexity, duration, type, management methods and number of team members, goes through the six phases, such as starting an IT project, determining the requirements for the created ECM program, planning, coding, testing and closing the IT project. From the results obtained, it could be concluded that the concepts and techniques for creating ECM programs, as well as lifecycle models, are necessary knowledge competencies that all participants in IT projects must possess. Inadequate possession or lack of these skills jeopardises the ability to achieve the planned project objectives, to produce a working program code and also to fulfil the obligations of the transaction.
Keywords: IT project, Project Management, Waterfall, Agile, Scrum.

For citation:
Nikolaenko V.S. (2024). Technological features of creating IT products within the framework of implementing IT projects. *Strategic Decisions and Risk Management,* 15(3): 270-279. DOI: 10.17747/2618-947X-2024-3-270-279. (In Russ.)

Acknowledgment
The work was carried out within the framework of the state task ‘Science’, project FEWM-2023-0013.

在IT项目实施框架内创建IT产品的技术特点

V.S. Nikolaenko[1, 2, 3]

[1] 托姆斯克国立系统管理与无线电电子大学（（俄罗斯，托木斯克）
[2] 托木斯克理工大学（俄罗斯，托木斯克）
[3] 西伯利亚国立医科大学（俄罗斯，托木斯克）

简介

本文讨论了计算机软件的特性，如增量和高可制造性，以及 IT 项目管理的技术特点。 本文旨在分析在 IT 项目过程中创建计算机程序的技术特点。研究发现，级联和敏捷概念（Waterfall和Agile）、约14种技术（XP、RUP、AUP、RAD、DSDM、Scrum、DAD、Kanban、Lean SD、FDD、MDD、DevOps、MSF 和 Oracle CDM）和4种生命周期模型（V 模型、Boehm 生命周期模型、迭代和级联生命周期模型）用于创建计算机软件。 此外，研究还发现，任何采用级联生命周期模式的信息技术项目，无论其规模、复杂程度、持续时间、类型、管理方法和团队成员数量如何，都要经历六个阶段：信息技术项目的开始、确定要创建的计算机程序的要求、规划、编码、测试和信息技术项目的结束。 研究结果使我们得出结论，计算机软件制作的概念和技术以及生命周期模型是信息技术项目的所有参与者都必须掌握的必要知识能力。如果不具备这些能力或缺乏这些能力，就有可能无法实现计划的项目目标，无法获得可行的软件代码，也无法履行交易契约。

关键词: IT 项目、项目管理、Waterfall, Agile, Scrum.

引用文本：

Nikolaenko V.S. (2024). 在IT项目实施框架内创建IT产品的技术特点。战略决策和风险管理, 15(3): 270–279. DOI: 10.17747/2618-947X-2024-3-270-279. (俄文)

致谢

这项工作是在国家任务 "科学 "项目 FEWM-2023-0013 框架内完成的。

The analysis of the works of O'Connell (2005), Chernikov, Dashitsyrenov (2017), Marchenko et al. (2020) and Bazarova, Rochev (2022) has enabled us to conclude that the main technological properties of computer programs are incrementality and high technological efficiency.

Incrementality of computer programs is defined as the possibility of adding new data and commands to the program code in order to expand functionality and rectify software bugs. A case in point is the computer game Cyberpunk 20777[1], which was released at the end of 2020. Despite the game's relatively long release cycle, CD Projekt RED has consistently released updates to improve technical specifications, fix bugs and add new content. For instance, in September 2022, the developer released patch 1.6, which added content from the Cyberpunk: Edgerunners series, which premiered on Netflix in autumn 2022.

The concept of incrementality allows for the decomposition of a desired computer program into discrete user stories, which can then be delivered to stakeholders in a piecemeal fashion. As Vigers and Beatty observe in their work, stakeholders frequently present a multitude of conflicting user, functional, and business requirements. Incrementalism offers a straightforward method for eliminating these inconsistencies. In particular, it helps us to first create user stories that fulfil the requirements of all parties and then, with each update, to incrementally resolve critical contradictions [Vigers and Beatty, 2022].

It is important to highlight that the defining characteristic of incrementality sets computer programs apart from the outputs of more traditional project types, such as construction or education. In particular, if separate parts of a computer program can be developed in parallel, in classical projects the desired result is only obtained when a certain sequence of actions is performed.

Another property of computer programs is their high manufacturability. This implies that only those specialists who have the necessary professional competences can be involved in creating the program code. For instance, a programmer[2] should possess a minimum level of professional qualification, whereas a testing specialist[3], a database

[1] https://www.cyberpunk.net/ru/ru/.
[2] Professional standard 06.001 'Programmer'. https://clck.ru/PaFBJ.
[3] Professional Standard 06.004 'Information Technology Testing Specialist'. https://clck.ru/PaFa5.

Table 1
The main advantages and disadvantages of the cascade approach to programming for ECM

Advantages/Disadvantages	Comments
The main advantages of the cascade approach to programming for ECM	
Differentiation of the phases of computer software development	Clear boundaries of the phases make it possible to determine their exact start and end dates, which not only increases customer loyalty, but is also one of the essential conditions for concluding contracts (Article 432 of the Civil Code of the RF[a])
Fixed Price	Determining a fixed price increases the loyalty of customers of computer programs and the competitiveness of organisations involved in their creation
Quality of project documentation	As a rule, different workers take part in the process of creating computer programs, so high-quality project documentation is necessary for their coordinated and synchronous work
Harmonious integration of new participants into the IT project team	Each phase ends with the production of a set of design documents sufficient for another team to continue development
The main disadvantages of the cascade approach to programming for ECM	
Difficulty in changing previously approved user, functional and business requirements	As a general rule, changes to requirements can only be made once a specific IT result has been produced.
Duration of creation of a computer program	The development of computer software is a lengthy process, which increases the probability of such risks as changes in legislation, transformation of the business structure and customer interests, departure of key employees, etc.
Deviation from planned timelines	Computer software creation rarely meets the planned deadlines. In particular, The CHAOS Manifesto analytical reports state that the average deviation from the planned terms in IT projects is 89%[b]

[a] Civil Code of the Russian Federation of 30.11.1994 No. 51-FZ. https://clck.ru/MsKTf.
[b] The Standish Group International. https://clck.ru/3Div55.

administrator[4] and a system analyst[5] should have completed secondary specialised education. A specialist in graphic user interface design[6] should have undertaken up to one year of professional training, while an IT[7] project manager should have obtained a higher education qualification at the bachelor's degree level.

The work emphasises that the success of IT projects is dependent on the professional competences and experience of programmers (Chernikov & Dashitsyrenov, 2017). Failure to meet the required qualification level will inevitably result in the introduction of numerous defects and a reduction in the performance of the developed programs.

Furthermore, the high manufacturability of computer programs is evidenced by the capacity to create program code remotely [Konobevtsev et al., 2019]. The remote form of IT project team organisation offers a number of advantages over classical projects. For example, it allows for the recruitment of qualified personnel from diverse geographical locations, the implementation of performance-based remuneration, and the reduction of software development costs by eliminating the expenses associated with office rental, electricity, internet, and other utilities.

This article aims to examine the technological aspects of developing computer programs as part of information technology (IT) projects. In order to achieve this objective, the author of this article has addressed the following issues:

- the advantages and disadvantages of the concepts of software programming have been revealed;
- the techniques used for the creation of computer software (XP, RUP, AUP, RAD, DSDM, Scrum, etc.) have been analysed;
- the main models of IT project life cycles have been identified.

The incremental nature and high technological efficiency of computer software have stimulated the development of various concepts of computer software creation, including the Waterfall and Agile models.

The concept of cascading programming for ESM (Waterfal). It is thought that the concept of the cascade (classical, waterfall) approach to programming for ECM

[4] Professional Standard 06.011 'Database Administrator'. https://clck.ru/qNbpz.
[5] Professional Standard 06.022 'System Analyst'. https://clck.ru/PaFVa.
[6] Professional Standard 06.025 'Specialist in Graphic User Interface Design'. https://clck.ru/PaFGs.
[7] Professional standard 06.016 'Project manager in the field of information technologies'. https://clck.ru/PaFDk.

Table 2
Principles of the flexible programming approach to ECM

Name of the operating principle	Description of the operating principle
Frequent and continuous delivery of computer program parts	Frequent and continuous delivery of the computer programs is important to customers and users because it enables stakeholders to clarify user, functional and business requirements and shorten the return on investment period.
Maximum openness to the possibility of changing requirements	The overarching objective of the Agile concept of software program creation is to achieve the highest level of stakeholder satisfaction. Consequently, Agile endeavors to address all user, functional and business requirements wherever feasible.
Flexible processes	The principle of maximum openness to the possibility of changing requirements at any phase of the IT project lifecycle implies that processes are required to adapt promptly to new and/or altered requirements.
Systematic delivery of the current IT result	To manage risks such as bugs, defects, software code inaccuracies and vulnerabilities, failure to meet stakeholder expectations, litigation, etc., Agile involves the systematic delivery of a valid IT deliverable.
Maximum stakeholder involvement	Collaboration and open communication in Agile is more important than hierarchy and contractual constraints, so the process of creating valuable computer program for all stakeholders is possible if all interests and opinions have been taken into account.
Self-organisation of the IT project team	The Agile methodology does not prescribe a specific set of processes for implementing IT projects. Instead, it provides a general framework that outlines the key stages of the development process, including planning, requirements analysis, design, programming, testing and documentation. For instance, the decomposition of user stories is the responsibility of the IT project team. User stories are defined as a description of the requirements for the computer program being developed. To illustrate, a user story might be a bank client wanting to receive messages about changes in the status of a loan application to promptly dispose of funds. It is also worth noting that during a sprint[a], IT project participants make their own decisions about what planned work will be done and in what sequence it will be done.
Face-to-face communication	Agile believes that the most effective and efficient way for stakeholders to interact is through face-to-face communication
Effectiveness	The primary indicator of progress is the delivery of functional computer program components to stakeholders in a systematic manner.
Continuous professional development of IT project participants	Maximum openness to the possibility of changing user, functional and business requirements stimulates IT project team members to continuous professional development and learning new features of computer program creation.
Keep it short and simple (KISS)	E. Raymond states in his writings that when designing computer programs, it is essential to ensure maximum simplicity and transparency of the program code [Raymond, 2003].

[a] Sprint is the time interval during which the IT project team performs the planned amount of work..

was first developed in 1970 by the American scientist W.W. Royce [Royce, 1970]. In Royce's view, the process of creating a program code is akin to a continuous flow of water, with each phase building upon the previous one and commencing only when the preceding phase is complete. The key advantages and disadvantages of the cascade approach to computer program creation are presented in Table 1.

The concept of flexible computer software development (Agile). The agile concept of creating computer programs was developed in February 2001 by C. Beck, M. Beadle, E. W. Bennekum et al. [McConnell, 2021]. The authors used the incrementality property to formulate the basic principles of agile IT project management, which differ from the cascade concept. Table 2 provides examples of the principles underlying the concept of agile computer software development.

It is important to note that in the domestic literature, the Agile concept is most often referred to as the methodology of agile software development [Aubrey, 2019]. However, according to the classical interpretation, methodology is understood as a set of methods, means and technologies of cognition used for the purpose of organizing and constructing a research [Luzgina, 2018; Smagina, 2020]. The author considers the utilisation of the notion of 'concept' to be more appropriate, given that Agile constitutes a set of principles, approaches, best practices, ideas and ways of creating computer software. In this regard, Agile should be understood as a concept of flexible creation of computer software, which includes a set of special principles, approaches, best practices, ideas and ways to achieve project goals.

In addition to the technological properties that have led to the differentiation of computer software creation concepts,

Table 3
Techniques for writing programs for ECM

Technique name	Technique description
eXtreme Programming, XP	The name is based on the idea of using only the best practices of creating software code, taking the process of its creation to a new level - extreme. For example, when checking the created program code, it is recommended to involve two programmers at the same time, so that one is busy creating the program and his partner is busy checking it [Beck, 2003]. This best practice is commonly referred to as pair programming. Among the advantages of the XP technique, it is worth mentioning that the first version of a computer program for ECM is obtained quickly, which enables users to start its testing
IBM Rational Unified Process (RUP)	The program creation for ECM comprises nine processes and four phases. The principal processes encompass business modelling, requirements management, analysis and design, implementation, testing, deployment, project work management, change management and infrastructure (the internal environment of software code creation). The phases are as follows: initial phase, refinement, design and implementation.
Agile Unified Process (AUP)	AUP is a simplified version of RUP. AUP, which includes 7 processes: modelling, implementation, testing, deployment, configuration management, project management, and infrastructure creation [Edeki, 2013]
Rapid Application Development (RAD)	RAD is recommended for short-term IT projects characterised by tight deadlines, limited budget, small scope of work, graphical interface and low computational complexity [Beynon-Davies et al., 1999].
Dynamic Systems Development Method (DSDM)	DSDM is based on the RAD concept of rapid application development. The toolkit is based on its own life cycle, which consists of the following phases: assessment of technical feasibility; business case; creation of a functional model; design; development.
Scrum	Scrum consists of such elements as roles, artefacts and processes. There are three main roles in Scrum - product owner, scrum master and project team. The artefacts are the prioritised list of requirements (product backlog), the list of requirements that have been selected for sprint (sprint backlog) and the incremental code. The processes in Scrum are related to communication. In particular, a short meeting of the project team to synchronise work (scrum meeting), sprint planning (project team meeting to decompose user story) and retrospective analysis [Kohn, 2011]
Disciplined Agile Delivery (DAD)	The DAD toolkit developed by IBM is based on Scrum. The main difference is the extended IT project lifecycle, which starts with the classical project initiation and ends with the use of the obtained IT result by the user
Kanban	Kanban is based on Japanese lean manufacturing technology - 'just-in-time'. The main advantage of the toolkit is an even workload among the project team members. The tasks are entered into a separate list (pull) as they become available
Lean Software Development (Lean SD)	Lean SD was developed in [Poppendieck, Poppendieck, 2010]. The technique is based on traditional Lean principles such as elimination of wastage and pauses in the development process, emphasis on learning, extremely delayed decision making based on facts, extremely fast delivery of a working IT result to the customer, motivation of the project team, integration, holistic vision
Feature Driven Development (FDD)	FDD uses an ideal labour input model to easily monitor the progress of IT result creation, where 1% is allocated to subject area analysis, 40% to design, 3% to design verification and refinement, 45% to software code creation, 10% to software code testing and refinement, and 1% to implementation
Model Driven Development (MDD)	This technique of creating computer programs is characterised by an abstract description of the desired result, where some aspects may be left out. This is done in order to simplify the design and documentation process
Development and Operations (DevOps)	The main feature of DevOps is the maximum integration between programmers and information technology specialists in the maintenance of information systems
Microsoft Solutions Framework (MSF)	Microsoft Corporation offers MSF for managing the work of a team of organisation-level programmers. According to MSF authors, there are many reasons that lead to failures: errors in forecasts, changes in requirements, inaccurate specifications, etc. MSF consists of 3 models: team model, project team model and design process model
Oracle Custom Development Method (Oracle CDM)	Oracle CDM consists of 3 models: Classic, Fast Track and Lite. Each model has its own set of phases and processes. For example, Classic has 6 phases and 11 processes. Classic is used for long-term (duration of more than 6 months) and complex IT projects. Fast Track is oriented to custom development projects with duration not exceeding 6 months. Fast Track development life cycle consists of 3 phases (requirements modelling, design and creation, implementation) and 11 processes. Lite is used for short-term projects and consists of 2 phases (prototyping and building, implementation) and 9 processes

Table 4
Lifecycle models for IT projects

Lifecycle Model	Description of the Lifecycle Model
V-shaped	IT project phases follow each other sequentially in a strictly defined order [Balaji, Sundararajan, 2012]. Requirements testing occurs too late in the life cycle, so it is impossible to make changes without affecting the project work plan
Spiral	The spiral model emphasises analysis, design and risk management [Boehm, 1981]. At each turn of the spiral, a new version of the computer program is created. One turn of the spiral represents a complete project cycle based on the cascade concept
Iterative	The principle of operation of the iterative LC model is similar to the spiral model. The difference is that in the iterative model the result of work is a mock-up-prototype of a fragment of a future computer program. During the development process iterations are performed until the mock-up-prototype acquires all the necessary properties in accordance with the ToR. As a rule, the iterative model is applicable for short-term (less than 2 months) and medium-term (2-6 months) IT projects
Cascade	The lifecycle phases occur in a sequential order. Despite a number of drawbacks, the cascade model can be applied to any type, size and complexity of IT project. The following phases are considered universal in the context of project management: project initiation, project organisation and preparation, project execution and project closure.

Table 5
Phases of the cascade model lifecycle for creating ECM programs according to national standards

Lifecycle Model	Description
Formation of requirements (pre-design phase)	The applicant contractor(s) shall conduct a survey of the customer's existing infrastructure, identify current problems, threats and opportunities, justify the need to create an IT result, identify user, functional and business requirements
Development of the automated system concept	An applicant for contractors (executors) creates an As Is model, e.g. in the form of an IDEF0 model, works out options of possible IT solutions that satisfy user, functional and business requirements, and develops a 'To Be' model [Kovalenko, Chokla, 2020]
Development of ToR	By virtue of GOST 19.101-77[a] and GOST 15.016-2016[b] a statement of work is a document that formalises the purpose and scope of a computer program, technical and special requirements. In the literature it is also possible to meet other names of TOR. For example, in [Vigers, Beatty, 2022] the technical specification is called a specification of requirements for a computer program
Development of preliminary design	According to GOST 2.119-2013[c] and GOST2.103-2013[d] the conceptual design is understood as a set of design documents that contain basic solutions that give a general and preliminary idea of the purpose, design, principle of operation of a computer program, as well as data defining their basic parameters. The preliminary design allows selecting a suitable IT solution for further development of the program[e]
Technical project development	In accordance with GOST 2.120-2013[f] the technical project is a document where the final technical solutions are fixed in accordance with the ToR and sketch of the project. It should be noted that the concept, TOR, sketch and technical design are parts of the project documentation. Project documentation is understood as documentation in text and graphic form containing information necessary for the development, maintenance and operation of the IT result
Documentation	Working documentation is the materials in text and graphic form, according to which the IT result is created

Table 5 – *remainder*

Модель ЖЦ	Описание
Commissioning	During this phase, installation and commissioning works are carried out, preliminary tests and pilot operation are carried out, employees are prepared and trained, and acceptance of the created computer program is carried out. By virtue of clause 1.4 of GOST 34.201-89[g] acts of work completion, acceptance into pilot operation, acceptance into industrial operation, etc. are developed.
Maintenance	This phase is characterised by the performance of works (rendering of services, supply of goods) in accordance with warranty obligations, as well as post-warranty maintenance

[a] State Standard of the Union of Soviet Socialist Republics. Unified system of program documentation. Types of programs and program documents. GOST 19.101-77. M.. Standardinform, 1980.
[b] Interstate standard. System of product development and putting products into production. Technical Assignment. Requirements for content and design. GOST 15.016-2016. Moscow, Standardinform, 2020.
[c] Interstate standard. Unified system of design documentation. Sketch project. GOST 2.119-2013. M., Standardinform, 2018.
[d] Interstate standard. Unified system of design documentation. Stages of development. GOST 2.103-2013. M., Standardinform, 2019.
[e] State Standard of the Union of Soviet Socialist Republics. Ensuring manufacturability of product design. General requirements. GOST 14.201-83. M., Standardinform, 2009.
[f] Interstate standard. Unified system of design documentation. Technical project. GOST 2.120-2013. Moscow, Standardinform, 2015.
[g] State Standard of the Union of Soviet Socialist Republics. Information technology. Complex of standards for automated systems. Types, completeness and designation of documents when creating automated systems. GOST 34.201-89. M., Standardinform, 1989.

Table 6
Phases of the IT project lifecycle

Lifecycle Phase	Description
Starting an IT project	Для этой фазы характерно проведение переговоров между заинтересованными сторонами, обычно между заказчиком и подрядчиком (исполнителем, поставщиком), в рамках которых утверждаются требования к программе для ЭВМ, даты начала и окончания работ, цена, способы нивелирования и ослабления коммерческих, комплаенс- и проектных рисков и др. Как правило, по окончании фазы между сторонами заключается гражданско-правовой договор (контракт)
Defining the requirements for the computer program for ECM to be created	The identified user, functional and business requirements are formalised in ToRs
Planning	Based on the TOR and communications with stakeholders, the IT project manager defines the concept and technique of creating a computer program, prepares the necessary resources and basic project documents
Coding	The phase entails the development of software code in accordance with user, functional, and business requirements.
Testing	At this stage, the software code is subjected to testing to ascertain the degree of correspondence between the actual behavior of the developed algorithms and their anticipated behaviour[a]. In accordance with clause 4.2 of GOST 34.603-92[b] the documentation required for testing includes the TOR, acceptance certificate for experimental operation, working logs of experimental operation, and so forth.
End of IT project	In consideration of the specific requirements of compliance, this phase encompasses the acceptance of the computer program that has been created and the signing of the pertinent documents.

[a] Software Engineering. Software Engineering - Guide to the Software Engineering Body of Knowledge (SWEBOK). ISO/IEC TR 19759:2005. 2005.
[b] State Standard of the Union of Soviet Socialist Republics. Information technology. Types of tests of automated systems. GOST 34.603-92. M., Standardinform, 1993.

the implementation of IT projects is also influenced by a number of other factors, including uncertainty and variability in user, functional and business requirements; intellectual labour; different management styles; limited resources; cross-communication with stakeholders; and others. Taking these factors into account in management has stimulated the development of many different techniques of creating computer programs: XP, RUP, AUP, RAD, DSDM, Scrum and others.

The list of techniques of computer software creation is presented in Table 3.

The technological characteristics of computer software, the differentiation of the concepts and the variety of techniques used to create such software have influenced the logical relationships between the phases of IT projects, thus stimulating the creation of different life cycle models.

A project lifecycle is defined as a series of phases through which a project progresses from its inception to its completion. A project phase (or stage) can be defined as a set of operations that culminate in the achievement of planned results. Typically, a project phase ends with a checkpoint (or milestone) [Isaev et al., 2021]. Project phases can be sequential, iterative and/or overlapping [Polonsky, Vasiliev, 2018]. Table 4 provides examples of lifecycle models relevant to IT projects.

The selection of the appropriate IT project lifecycle model is typically based on four key criteria: cost, risk, quality and speed of development. These criteria are interrelated, so it is not possible to achieve all four goals simultaneously. For instance, if the deadline for the final version of the program needs to be brought forward, this will require additional human resources, which will increase the cost of the IT project.

Our analysis of the national standards revealed that the standard practice in the domestic market for creating computer programs is based on the cascade model of the life cycle[8], which comprises eight phases[9]. Examples of these phases are presented in Table 5.

In light of the analysis of national standards for the systematisation of computer software creation methods, the author of this article identifies six phases of the IT project life cycle as being of particular relevance:

- start of the IT project,
- defining the requirements for the computer program to be created,
- planning,
- coding,
- testing,
- end of the IT project (Table 6).

A review of the technological features of IT project management reveals the following conclusions.

- The primary characteristics of computer software development within the context of IT projects are incrementality and high manufacturability (Nikolaenko, 2020; Nikolaenko & Sidorov, 2023).
- A variety of techniques and methodologies are employed in the creation of computer software, including the cascading and agile concepts of the waterfall and agile models, as well as 14 techniques such as XP, RUP, AUP, RAD, DSDM, Scrum, DAD, Kanban, Lean SD, FDD, MDD, DevOps, MSF and Oracle CDM. Additionally, there are over four lifecycle models, including the V-model, Boehm's lifecycle model, iterative and cascading lifecycle models.
- Any IT project implemented according to the cascade lifecycle model, regardless of its scale, complexity, duration, type and management methods, passes through six phases: the beginning of the IT project, definition of requirements to the computer program to be created, planning, coding, testing and the end of the IT project.

In light of the above, it can be concluded that a comprehensive understanding of computer software concepts, techniques and life cycle models is a fundamental competency for all those involved in IT projects. A lack of these skills can jeopardise the achievement of planned project objectives, the production of functional software code and the fulfilment of transactional obligations. This, in turn, can lead to the subsequent materialisation of compliance and project risks.

References

Bazarova A.M., Rochev K. V. (2022). IT-technologies in the management of production processes at enterprises of the fuel and energy complex. In: *Management of sustainable development of the fuel and energy complex-2021.* Materials of the II All-Russian Scientific and Practical Conference. Ukhta, 97-103. (In Russ.)

Beck K. (2003). *Extreme programming: Development through testing.* St. Petersburg, Piter. (In Russ.)

Vigers K., Beatty D. (2022). *Development of software requirements.* St. Petersburg, BHV. (In Russ.)

Isaev E.A., Pervukhin D.V., Rytikov G.O., Filyugina E.K., Hayrapetyan D.A. (2021). Evaluation of the effectiveness of information systems. *Business Informatics,* 15(1): 19-29. (In Russ.)

[8] National Standard of the Russian Federation. Information technologies. System and software engineering. Life cycle management. Part 2: Guidelines for the application of ISO/IEC 15288. GOST P 57102-2016/ISO/IEC TR 24748-2:2011. M., Standardinform, 2016.

[9] State Standard of the Union of Soviet Socialist Republics. Information technology. Complex of standards for automated systems. Automated systems. Stages of creation. GOST 34.601-90. M., Standardinform, 1992.

Kovalenko V.V., Chokla D.S. (2020). Transformation of processes and organizational structure of a WEB-based model business system based on the results of business analysis. In: *Modern technologies in science and education-2020,* III International Scientific and Technical Forum, collection of works. Ryazan, Book Jet, 195-198. (In Russ.)

Con M. (2011). *Scrum. Flexible software development.* Moscow, Williams. (In Russ.)

Konobevtsev F.D., Lass N.A., Gurova E.V., Romanova I.A. (2019) Remote work: Technologies and experience of organization. *Bulletin of the University,* 7: 9-16. (In Russ.)

Luzgina Ya.A. (2018) Concept, methodology and methods of information support of innovation activity. In: *Materials of the 16th All-Russian Scientific and Practical Conference «Problems of economic development and entrepreneurship»,* 58-64. (In Russ.)

McConnell S. (2021). *Even more efficient Agile.* St. Petersburg, Piter. (In Russ.)

Marchenko D.S., Grigoriev A.V., Rochev K.V. (2020). Information system for storing autorotation data. *Information Technologies in Management and Economics,* 3(20): 21-39. (In Russ.)

Nikolaenko V. (2020). Risk, risk management and uncertainty: Clarifying the concepts. *Public Administration. Electronic Bulletin,* 81: 91-119. (In Russ.)

O'Connell F. (2005). *How to successfully manage projects. Silver bullet.* Moscow, Kudits-Obraz. (In Russ.)

Aubrey K. (2019). *All about Agile. The art of creating an effective team.* Moscow, Eksmo. (In Russ.)

Polonsky A.Yu., Vasiliev M.M. (2018). Analysis of software development methodologies. *Alley of Science,* 6(22): 1084-1094. (In Russ.)

Poppendik M., Poppendik T. (2010). *Lean software production: From idea to profit.* Moscow, Williams. (In Russ.)

Smagina S.M. (2020). On the concepts of 'method of scientific research', 'methodology of scientific research' and 'logic of scientific research'. In: *Logic and methodology of scientific research:* Collection of scientific articles and reports of the International Scientific and Practical Conference. Orel, The Central Russian Institute of Management, RANEPA, 112-116. (In Russ.)

Chernikov B.V., Dashitsyrenov Z.D. (2017). Analysis of modern methods of quality management and their application to the field of higher education. In: *Managing the development of large-scale systems MLSD'2017:* Proceedings of the 10th International Conference. Moscow, V.A. Trapeznikov Institute of Management Problems, RAS, 217-219. (In Russ.)

Balaji S., Sundararajan M.M. (2012). Waterfall Vs V-model Vs Agile: A comparative on SDLC. *International Journal of Information Technology and Business Management,* 2(1): 26-30.

Beynon-Davies P., Carne C., Mackay H., Tudhope D. (1999). Rapid application development (RAD): An empirical review. *European Journal of Information Systems,* 8: 211-223.

Boehm B.W. (1981). *Software engineering economics.* Englewood Cliffs, NJ, Prentice-Hall.

Edeki C. (2013). Agile unified process. *International Journal of Computer Science and Mobile Applications,* 1(3): 13-17.

Nikolaenko V., Sidorov A. (2023). Analysis of 105 IT project risks. *Journal of Risk and Financial Management,* 16(1), 33: 1-20.

Raymond E. (2003). *The art of Unix programming.* Addison-Wesley.

Royce W.W. (1970). *Managing the development of large software systems*.

About the author

Valentin S. Nikolaenko
Candidate of economic sciences, associate professor at the Department of Automation of Information Processing, Tomsk State University of Control Systems and Radioelectronics (Tomsk, Russia); associate professor at the Business School of Tomsk Polytechnic University (Tomsk, Russia); associate professor at the Department of Economics, Sociology, Political Science and Law of Siberian State Medical University (Tomsk, Russia). ORCID: 0000-0002-1990-4443; Web of Science Researcher ID: J-8521-2015; SPIN: 9301-1835; Author ID: 745788; IRID: 283767926; Scopus Author ID: 57193434445.
Research interests: risk-management, national security, economic security, information law and intellectual property protection, civil law, project management.
valentin.s.nikolaenko@tusur.ru

作者信息

Valentin S. Nikolaenko
经济学副博士, 托姆斯克国立系统管理与无线电电子大学（（俄罗斯，托木斯克）信息处理自动化系教授; 托木斯克理工大学（俄罗斯，托木斯克）商学院副教授; 西伯利亚国立医科大学（俄罗斯，托木斯克）经济学、社会学、政治学和法学系副教授。ORCID: 0000-0002-1990-4443; Web of Science Researcher ID: J-8521-2015; SPIN: 9301-1835; Author ID: 745788; IRID: 283767926; Scopus Author ID: 57193434445.
科学兴趣领域: 风险管理、国家安全、经济安全、信息法和知识产权保护、民法、项目管理。
valentin.s.nikolaenko@tusur.ru

The article was submitted on 16.08.2024; revised on 17.09.2024 and accepted for publication on 21.09.2024. The author read and approved the final version of the manuscript.
文章于 16.08.2024 提交给编辑。文章于 17.09.2024 已审稿。之后于 21.09.2024 接受发表。作者已经阅读并批准了手稿的最终版本。