[7].Abed A.A. (2018) Functional Structure of comparators predicate in the compartment identification method. Advanced information systems, 2(2), 78–83.
[8].Bondarenko M.F., Shabanov – Kushnarenko S., Shabanov–Kushnarenko Yu.P. (2009), Practical applications of comparative identification of linear finite– dimensional objects, Bionika intellekta, №2 (71), pp.5–12.
[9].Poole, David (2006), Linear Algebra: A Modern Introduction (2nd ed.), Brooks/Cole, ISBN 0-534-99845-3.-pp. 336.
[10]. Anton, Howard (2005), Elementary Linear Algebra (Applications Version) (9th ed.), Wiley International..–pp. 432.
[11]. Leon, Steven J. (2006), Linear Algebra With Applications (7th ed.), Pearson Prentice Hall.– pp.367.
[12]. Raskin L., Sira O., Katkova T. Comparator identification in the conditions of bifuzzy initial data. Eureka. Physics and Engineering, 2021 (1), 113–124.
[13]. Lev Raskin, Oksana Sira. Performing arithmetic operations over the (L–R) Type fuzzy numbers. 2000.–Vol 3. issue 4.–p.6–//Eastern European.
[14]. Lev Raskin, Oksana Sira. Execution of arithmetic operations involving the second order fuzzy numbers//Eastern European Journal of Enterprise Technologies.– 2020.–Vol.4,issue 4.–p.14–20.
Отримана в редакції 26.07.2024. Прийнята до друку 17.08.2024. Received 26 July 2023. Approved 17 August 2024. Available in Internet 23 October 2024

УДК 004.415.2:004.94

WATERFALL MODEL OVER PCD.UCT MODEL REVIEW

МОДЕЛЬ ВОДОПАДУ НАД ОГЛЯДОМ МОДЕЛІ PCD.UCT

NS Wisidagama[1], Faiz Marikkar[2]
NS Wisidagama[1], Faiz Marikkar[2]

[1]Department of Computer Science, Faculty of Computing, General Sir John Kotelawala Defence University,
[2]Staff Development Centre, General Sir John Kotelawala Defence University
E-mail: [2]faiz@kdu.ac.lk

DOI: 10.15673/atbp.v16i3.2927

Abstract. *The software development life cycle (SDLC) has seen the evolution of numerous models, each tailored to meet specific project needs. Among these, the Waterfall Model has been a longstanding, traditional approach, characterized by its linear and sequential phases. In contrast, the PcD.UcT Model, a relatively newer framework, advocates for a more iterative and user-centered approach. This paper presents a comprehensive review of the Waterfall Model, evaluating its strengths and limitations in modern software development. It contrasts these findings with the PcD.UcT Model, analyzing how its principles better align with contemporary project demands, particularly in terms of flexibility, user engagement, and iterative refinement. Through this comparative analysis, the paper aims to provide insights into the applicability and effectiveness of each model, offering guidance for software development teams in choosing the appropriate methodology for their projects. Since the waterfall development methodology, the key requirements in system development strategies have shifted from processes to users. Because the process as well as the customers are both equally important, employing either a predictive or an adaptive methodology is extremely difficult. The primary goal of this work is mainly combining and then evaluating all the relevant methodologies in order to create a HydroGIS that accurately automates the complexities of the hydrology process in a HydroGIS environment while meeting all user requirements.*

Анотація. *Життєвий цикл розробки програмного забезпечення (SDLC) бачив еволюцію численних моделей, кожна з яких адаптована для задоволення конкретних потреб проекту. Серед них модель водоспаду є давнім традиційним підходом, який характеризується лінійними та послідовними фазами. Навпаки, модель PcD.UcT, відносно новіша структура, виступає за більш ітеративний і орієнтований на користувача підхід. Ця стаття представляє комплексний огляд моделі водоспаду, оцінюючи її сильні сторони та обмеження в сучасній розробці програмного забезпечення. Він порівнює ці висновки з моделлю PcD.UcT, аналізуючи, як її принципи краще узгоджуються з вимогами сучасного проекту, зокрема з точки зору гнучкості, залучення користувачів та ітераційного вдосконалення. Завдяки цьому порівняльному аналізу ця стаття має на меті надати розуміння застосовності та ефективності кожної моделі, пропонуючи керівництво для команд розробників програмного*

забезпечення щодо вибору відповідної методології для своїх проектів. Після застосування методології водоспаду ключові вимоги до стратегій розвитку системи перемістилися від процесів до користувачів. Оскільки як процес, так і клієнти однаково важливі, застосувати або прогнозну, або адаптивну методологію надзвичайно важко. Головною метою цієї роботи є поєднання та подальша оцінка всіх відповідних методологій для створення HydroGIS, яка точно автоматизує складні процеси гідрології в середовищі HydroGIS, задовольняючи при цьому всі вимоги користувачів.

Ключові слова: модель PcD.UcT, методології розробки програмного забезпечення, водоспадна модель
Keywords: PcD.UcT Model, Software Development Methodologies, Waterfall Model

Introduction

In the realm of software development, the choice of development methodology significantly influences the efficiency, effectiveness, and adaptability of the development process. Among the various methodologies, the Waterfall Model has long been recognized as one of the earliest and most straightforward models (Senarath, 2021). Introduced as a linear and sequential approach, the Waterfall Model divides the software development process into distinct phases: Requirements Analysis, System Design, Implementation, Testing, Deployment, and Maintenance. Each phase must be completed before the next one begins, and there is minimal overlap between stages. This methodology's simplicity and clarity have made it a popular choice for projects with well-defined requirements and stable environments (Saravanos and Curinga, 2023).

The limitations of the Waterfall Model, particularly its lack of flexibility and difficulty accommodating changes, have led to the development of more adaptive methodologies. The iterative and incremental models, such as the Spiral Model (Boehm, 1988) and Agile methodologies (Beck et al., 2001), offer more flexibility by allowing for iterative refinement and continuous user feedback. These models address the need for adaptability and responsiveness in software development, contrasting with the Waterfall Model's linear approach.

The PcD.UcT Model, as discussed by Pradeep and Wijesekara (2017), represents an advanced evolution in software development methodologies by combining elements of predictive and adaptive approaches. The model integrates the iterative nature of prototyping with the structured phases of the Waterfall Model, creating a hybrid framework designed to enhance user engagement and adaptability. Pradeep and Wijesekara (2017) highlight that the PcD.UcT Model leverages iterative cycles to address the limitations of traditional methodologies, offering a more flexible and user-centered approach. This model's ability to adapt and incorporate user feedback throughout the development process provides a significant advantage over more rigid methodologies.

However, as the complexity of software systems has increased and the demand for flexibility has grown, newer models have emerged to address the limitations of the Waterfall Model. One such model is the PcD.UcT Model, which integrates principles from iterative and prototype- based development processes (Hidayati and Sismadi, 2020). The PcD.UcT Model combines the structured approach of the Waterfall Model with the iterative and user-centric methodologies of prototyping, aiming to provide a more adaptive and user- focused development framework. Pradeep and Wijesekara (2017) underscores the importance of balancing structure and flexibility in software development. Their work illustrates how the PcD.UcT Model effectively combines the best aspects of both iterative and predictive approaches, allowing for continuous refinement and adaptation based on user feedback. This model represents a significant advancement in addressing the challenges faced by traditional methodologies, particularly in complex and evolving project environments.

This review paper explores the Waterfall Model in detail, examining its advantages and limitations within the context of modern software development. It then compares the Waterfall Model with the PcD.UcT Model, highlighting how the latter's hybrid approach addresses some of the former's shortcomings. By evaluating both models' methodologies, the paper seeks to provide insights into their applicability, especially in projects requiring flexibility and user engagement. The objective of this paper is to offer a comprehensive comparison between these two methodologies, assessing their effectiveness in different scenarios and providing guidance for selecting the most suitable approach based on project requirements and constraints (Kavindya et al., 2022).

The development of software methodologies has evolved significantly over the decades, with each approach offering unique benefits and addressing specific project needs. The Waterfall Model, introduced in the early days of software engineering, remains one of the most foundational methodologies. According to Fagarasan et al., in 2021, the Waterfall Model, also known as the linear sequential model, is characterized by its structured and sequential phases. This approach emphasizes a systematic flow from requirements gathering through design, implementation, and testing to deployment and maintenance. The simplicity of the Waterfall Model makes it particularly suited for projects with well-understood requirements and low levels of uncertainty. However, its rigid structure can be a drawback in dynamic environments where requirements may change (Hidayati and Sismadi, 2020).

In summary, while the Waterfall Model provides a clear and structured approach to software development, its limitations in handling change have led to the emergence of more flexible methodologies. The PcD.UcT Model offers a promising alternative by integrating iterative and adaptive principles with traditional development practices. This hybrid approach aims to improve responsiveness to user needs and enhance overall project outcomes.

Waterfall Model

As a Process Model, Waterfall Model was the firstly introduced software model. It can also be also known as the linear sequential life cycle development model. Waterfall model is very simple and easy to understand, and it can be used in software development. In waterfall model, each and every phase should be fully completed before proceeding to the next. Waterfall model was the first SDLC model which was used in the software development process. It primarily represents the process of developing a software as a linear and as well as a sequential flow. It means that no phase of the software

development process can begin until all previous phases have been completed completely. The most important factor regarding waterfall model is that the phases do not overlap with each other.

Waterfall Model - Requirements Analysis and Specification

In this phase, requirements are gathered and analyzed. Every requirement of the potential system is captured and then reported to a requirement specification document.

The system design phase

Depicts all of the requirements' specifications from the very first stage, that are being analyzed and the design of the system is being prepared. This design helps to define the system's overall architecture, as well as the hardware specification and system requirements. During this phase, the whole system is primarily implemented using the system design inputs. The system is initially broken down into little programs known as units and are then incorporated into the very next step, which utilizes all of the system's inputs. The task of developing and testing each unit's functionalities is defined to as unit testing. All the units which were produced in implementation stage are being integrated into a single system after unit testing. After integration, the whole testing is conducted for identifying system flaws and failures. Once after all functional and non-functional testing has been completed, the system is being implemented by the developers, and the finished result is deployed in the environment of the customer or launched to the market. During the maintenance period, multiple problems occur in the client's environment. To fix these problems, patches are issued. To improve the products, the most real recent version has been issued (Figure 1).

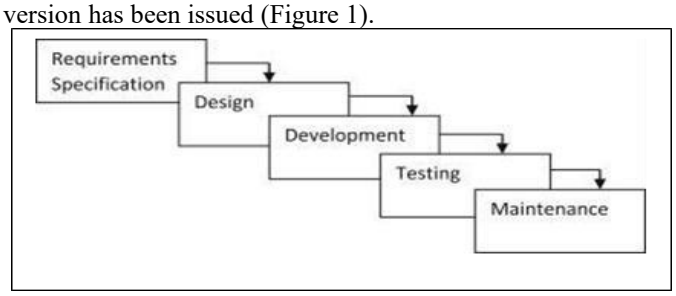

Fig.1 Waterfall Model – Architectual Design

All of these stages are connected to each other, allowing progress to be tracked as it moves progressively downward through the stages. The "Waterfall model" refers to the fact that the next phase begins only once the previous stage begins only once the previous stage is completed. n this model stages do not overlap under this paradigm.

Waterfall Model - Detailed Design

When the system design has been finished, which specifies hardware and system needs such as data layers, programming languages, network infrastructure, user interface, etc. It helps to shape the overall system architecture. This phase analyses the first phase's required specifications and prepares the system design. This system design aids in defining the overall system architecture, as well as specifying hardware and system requirements. Design phases can be divided into two categories: High-level design phase and Low-level design phase The initial step of system design and architecture is called high-level design. It contains a list of modules, their functions, and their connections as well as architecture diagrams and databases tables. This stage is completed with the creation of a High-level Design document. The result of actual software components is part of the low-level design process. Individual modules are created from the high-level design from the previous phase. The programmer can code directly from the lower-level design document (pseudo-code), which describes each module. It also includes details about each module's interface, error messages, dependencies, as well as inputs and the outputs.

Waterfall Model - Implementations (installation)

The system is first created as a series of independent programs, which are known as units. These are then merged using system design inputs in the upcoming stage. Generating and testing each unit to verify appropriate operation is known as unit testing. The waterfall approach, according to most software developers, is reasonably simple to apply. The first step is to assemble a large team of developers, testers, analysts, and managers who will work on diverse projects. The project manager is the most important person in every water-fall model project. The reason for this is when we use water-fall model once a stage is completed there is no going back to the precious stage. Therefore, all the requirements should be clearly identified at the beginning. He is the one who is accountable for the final product.

Waterfall Model - Testing and Integration

The customer reviews the product at this stage to confirm that it meets the requirements stated at the start of the waterfall process. To do this, the finished product is delivered to the customer. During this waterfall phase, the project is tested to ensure that it fits the needs of the customer. The QA engineers arrive on the scene at this point. Before delivering a product to the customer(s), it must be carefully tested to ensure that it is free of faults and meets all requirements, resulting in a positive user experience between the user and the software. The testing team will construct test cases based the product manager's design papers, personals, and the use case scenarios. The application's coding is now complete, and testing of the written code begins. Testing observing whether the designed software has any flaws and comparing it with the specifications listed. A successful stage ensures the consumer's satisfaction with the completed software. If a problem is encountered, the development process must be recommenced at the initial stage.

PcD.UcT model

The PcD.UcT is a combination of the Waterfall, prototype, and repetitive development processes, based on the EPLCM the development team initially develops a structure relies on the customer's main feedback and then develops additional

models. Until the completed output is introduced, each with additional functions or upgrades. When The PcD.UcT model deems with the Evolutionary Prototyping Life Cycle Model in the table below. The PcD.UcT. Building steps can be imagining, predictive and flexible.

Maintenance

Laterally, the customer deploys the final product this step commenced. Each group would introduce continuing the level up procedure and to release the newer stages the product and bugs are encountered and user feedback for changes is given. This makes way for the last step of the waterfall model, in which the software is organized at the consumer's side after complete testing. After the program's deployment, routine maintenance is carried out. If the customer asks for any alteration after the product is received, the entire process should start once again. In this step, our client makes deploy the completed product. As well as the client makes use of the developed application. New arrangements would also be done in this as matter of poor communication in initial steps. When the maintenance phase has begun, the consumer uses the product on a regular basis, facing bugs, inadequate functions, and other errors that occurred during manufacturing. These arrangements will be performed as requested by the production team carried out until the client is satisfied. In this phase, the customer makes use of the completed software.

Model description and analysis

The Waterfall Model is a linear, sequential approach to software development that progresses through a series of distinct phases: Requirements Analysis, System Design, Implementation, Testing, Deployment, and Maintenance. Each phase must be completed before moving to the next, ensuring a structured and organized development process. This model's simplicity and clearly defined stages make it well-suited for projects where requirements are well-understood and unlikely to change, allowing for precise planning and predictable outcomes. However, the Waterfall Model's rigidity can limit its effectiveness in dynamic environments where flexibility and ongoing user feedback are required.

The PcD.UcT Model represents a hybrid approach that merges elements from both iterative and predictive methodologies to create a more adaptable and user-focused framework. By integrating the structured phases of the Waterfall Model with the principles of iterative development, the PcD.UcT Model provides a balance between planning and flexibility. This model addresses the limitations of traditional approaches by incorporating continuous user feedback and allowing for adjustments throughout the development cycle. As a result, the PcD.UcT Model enhances responsiveness to changing requirements, making it particularly effective for projects in dynamic environments where both structure and adaptability are crucial for success.

The PcD.UcT Model effectively combines both predictive and adaptive frameworks, integrating the strengths of each to create a versatile approach to software development. The Predictive Cycle within the PcD.UcT Model mirrors the structured, linear approach of the Waterfall Model, providing a clear framework for planning, design, and initial development. This cycle ensures that core requirements are thoroughly defined and systematically met, maintaining a level of organization and control that is critical for project success. Meanwhile, the Adaptive Cycle introduces flexibility by allowing for continuous refinement and iteration based on user feedback and evolving requirements. This dual-cycle approach enables the PcD.UcT Model to handle both predictable and uncertain elements of a project, balancing the need for a well-defined structure with the ability to adapt to changes. As a result, the PcD.UcT Model offers a comprehensive framework that is better suited for complex projects in dynamic environments, where both stability and adaptability are essential.

The Adaptive Cycle introduces iterative development and prototyping, allowing for flexibility and continuous improvement based on user feedback. This cycle involves the iterative development of prototypes to refine and enhance the system based on user input and evolving requirements. Development Phases: The PcD.UcT Model follows several key phases, which integrate both predictive and adaptive elements: Initial Planning and Design: Similar to the Waterfall Model, this phase involves defining core requirements and system architecture. However, it also incorporates feedback mechanisms to adjust the initial design based on stakeholder input. Prototyping and Iteration: The model emphasizes iterative prototyping, where prototypes are developed and refined through multiple iterations. This allows for continuous user feedback and adjustments, improving the system's alignment with user needs.

The comparison of complexity and ease of use between the Waterfall and PcD.UcT models reveals significant differences in their practical application. The Waterfall Model, with its straightforward, linear approach, is relatively simple to understand and implement, making it suitable for projects with well-defined requirements and minimal changes. However, its rigidity can lead to challenges when adapting to dynamic environments. In contrast, the PcD.UcT Model, while more complex due to its iterative nature and emphasis on user feedback, offers greater flexibility and adaptability, making it ideal for projects that require continuous refinement and responsiveness to changing requirements. Case studies reviewed in this analysis highlight the effectiveness of each model across various real-world projects. These studies provide valuable insights into how the Waterfall Model excels in controlled, predictable environments, while the PcD.UcT Model proves more effective in projects that demand frequent updates, user involvement, and flexibility. Together, these insights demonstrate the contexts in which each model is most advantageous, emphasizing the importance of choosing the right approach based on project needs and conditions.

Discussion

The Waterfall Model and the PcD.UcT Model represent two distinct approaches to software development, each with its own strengths and weaknesses. The Waterfall Model, known for its linear and sequential structure, offers a clear, step-by-step approach that includes phases such as requirements gathering, design, implementation, testing, deployment, and maintenance (Aswin et al., 2024). Its strength lies in its simplicity and straightforward management, making it ideal for

projects with well-defined requirements and fixed timelines. However, its rigidity can be a limitation, as it does not easily accommodate changes once a phase is completed, potentially leading to costly modifications if requirements evolve during the project.

In contrast, the PcD.UcT Model (Prototyping, Continuous Development, User-Centric Testing) emphasizes an iterative and flexible approach, beginning with prototyping and followed by cycles of integration, testing, and refinement based on continuous user feedback (Pradeep and Wijesekera, 2020). This model's adaptability makes it well-suited for dynamic environments where requirements are expected to change frequently. It also enhances risk management by identifying and addressing issues early through iterative testing. While the PcD.UcT Model promotes user satisfaction through ongoing involvement, it can pose challenges in terms of cost control and timeline predictability due to its iterative nature.

To effectively compare these models, a framework based on criteria such as flexibility, risk management, cost and time efficiency, and user satisfaction can be employed. Ultimately, the choice between the Waterfall and PcD.UcT models depends on the specific needs of the project: the Waterfall Model is preferable for projects with stable requirements and a need for strict control over budget and timelines, while the PcD.UcT Model is better suited for projects requiring adaptability, continuous user feedback, and iterative development.

The comparative analysis of the Waterfall Model and the PcD.UcT Model also involves evaluating their flexibility, user engagement, and project suitability. Flexibility is a critical factor in modern software development, as it determines a model's ability to accommodate changes and updates during the development process. The Waterfall Model, with its rigid, sequential phases, is less adaptable to changes once a phase is completed, making it less ideal for projects with evolving requirements. In contrast, the PcD.UcT Model offers greater flexibility by integrating iterative prototyping and phased development, allowing for continuous refinement based on user feedback and changing needs. This iterative approach provides a more dynamic framework that is particularly useful in scenarios where requirements may evolve during development.

Table 1: Comparison of PcD.UcT over "Waterfall Model"

Stages	**Waterfall model**	**PcD.UCT**
Requirements Analysis and Specification	Every requirement is captured and documented.	Business logic became less complex because of familiarity. Then follow the step one by one. It became an international methodology to analyse the part when the problems occur, and user requirements change. Changing the environment according to the user specification. Because of this iteration, these steps become less complex and easy to understand
Architectural Design	The architecture of a system is the set of structures required to reason about it, which includes hardware and software elements and their related properties.	Protype in Process-centaric and User-centearic approach is targeting the system's next stage, prototype development, in the UcT development cycle. After identifying user demands, the system's next stage, prototyping, incorporates architectural design.
System-Design	This phase examines the most required and essential requirements/ specification from the preceding phases before beginning to develop the system. After specifying the software and hardware components, this design is critical in the creation of the overall system architecture.	The thorough design in the model PcD.UcT is primarily focused on the system's next stage,prototype development in the UcT development cycle. Then, in the next step of the system, when constructing the prototype, the user requirements are defined, and a thorough design is being created.
Coding	In the next phase, the system will be coded as discrete programs known as units, which will then be merged using inputs from the system design.	In this model coding occurs during the algorithm development stage of the PcD development cycle. In the algorithm development stage, after determining the process requirements, code for the prototype is developed.
Testing	Unit testing is used to test all of the developed units. After that, the entire system is integrated and tested for any problems or malfunctions.	System testing is carried out in this model in three different methods. To begin, we'll use the UcT development cycle. The purpose of the prototype evaluation is to put the developed prototype to the test with the user. Secondly, the testing phase of the PcD development cycle is then completed to test the system's newly design algorithm. Finally, after implementing both cycle optimizations, a test is performed to confirm whether the customer requirements have been met by the final product to be delivered.

User engagement is another important aspect, as it directly impacts the final product's alignment with user needs and expectations. The Waterfall Model typically involves users only during the initial requirements-gathering phase, which can result in a product that may not fully meet user needs if those needs change. The PcD.UcT Model, however, places a strong emphasis on continuous user involvement and feedback throughout the development process, ensuring that the final product is user-focused and adaptable to changes. The practical applications and relevance of both models are evident in today's rapidly changing technological landscape, where flexibility and user-centric design are increasingly important. For example, the HydroGIS software, which boasts a 92% user-friendliness rating and high accuracy in hydrological and GIS calculations, demonstrates the effectiveness of the PcD.UcT Model. Six software development projects currently undergoing testing with this model emulate its principles, highlighting its practical utility in adapting to changing requirements and ensuring high user satisfaction. This section aims to provide a balanced evaluation of both models, considering their strengths, weaknesses, and suitability for different types of projects.

Conclusion

In conclusion, this review paper provides an analysis of the Waterfall and PcD.UcT models, highlighting their strengths and limitations in software development. The Waterfall Model offers clear control and a structured schedule, guiding the product through distinct phases with specific deadlines. However, its rigidity and limited ability to accommodate changes make it less suitable for projects where requirements may evolve. The PcD.UcT Model, which blends iterative prototyping with a phased approach, addresses these limitations by providing a more flexible and user-centered development process. By allowing for continuous refinement and user feedback, the PcD.UcT Model proves to be a more effective framework for modern software development, especially in dynamic environments where adaptability and user engagement are crucial.

Reference

[1]. Aswin, K.Q., Kurniadi, F.I. and Arifin, S., 2024, April. Application of waterfall methods in a product registration. In AIP Conference Proceedings (Vol. 3024, No. 1). AIP Publishing.

[2]. Fagarasan, C., Popa, O., Pisla, A. and Cristea, C., 2021, August. Agile, waterfall and iterative approach in information technology projects. In IOP Conference Series: Materials Science and Engineering (Vol. 1169, No. 1, p. 012025). IOP Publishing.

[3]. Hidayati, N. and Sismadi, S., 2020. Application of Waterfall Model In Development of Work Training Acceptance System. INTENSIF: Jurnal Ilmiah Penelitian Dan Penerapan Teknologi Sistem Informasi, 4(1), pp.75-89.

[4]. Kavindya, A.K., Samarasinghe, V.I., Sanja, H.A.A., Deraniyagala, D.P. and Gunasekara, S.S.J., 2022. Study on Evolutionary Prototype Model over PcD. UcT.

[5]. Pradeep, R.M.M. and Wijesekera, N.T.S., 2020. Incorporating stakeholder concerns in Land Information Systems for urban flood management. Array, 8, p.100037.

[6]. Pradeep, R. M. M., & Wijesekara, N. T. S. (2017). Predictive cum Adaptive Systems Development Methodology for HydroGIS Tool Development. In 10th International Research Conference. Rathmalana: General Sir John Kotelawala Defence University

[7]. Saravanos, A. and Curinga, M.X., 2023. Simulating the Software Development Lifecycle: The Waterfall Model. Applied System Innovation, 6(6), p.108.

[8]. Senarath, U.S., 2021. Waterfall methodology, prototyping and agile development. Tech. Rep., pp.1-16.

Отримана в редакції 26.07.2024. Прийнята до друку 17.08.2024. Received 26 July 2023. Approved 17 August 2024. Available in Internet 23 October 2024