

UDC 658

doi: 10.15330/jpnu.10.4.93-102

REQUIREMENTS MANAGEMENT METHODS DURING PROJECT IMPLEMENTATION IN CONDITIONS OF UNCERTAINTY

IRYNA PIATNYCHUK, ALEKSANDER IWASZCZUK

Abstract. The purpose of the article is to study the issue of application of software requirements management methods during project management under conditions of uncertainty. It has been established that the management of requirements under conditions of uncertainty should be considered as permanent processes of system management of requirements throughout all stages of the project life cycle, which involves activities aimed at maintaining, tracking and checking requirements from the moment of their formation to their implementation and beyond. Based on the results of the research, it was found that the most common requirements management methods are: flexible (Agile) approach; advanced requirements analysis; the "Gradual Disclosure" method; models of iterative and incremental development; the "Minimum Viable Product" method; "Just-in-Time" method etc. It has been established that the use of software requirements management methods becomes especially relevant in conditions of uncertainty, as these methods allow you to effectively deal with dynamic changes and ensure the flexibility of the project. It was found that requirements management should focus on the organization, communication and development of requirements, among which the following can be distinguished: change management, traceability of requirements, communication, version control of requirements documents, etc. It has been investigated that change management involves making corrections to specified requirements during project implementation, as well as assessing the impact of corrections, obtaining permission to modify and update the necessary documents. Traceability, in turn, means defining and maintaining traceable relationships between requirements and other project artifacts, such as project documentation, test cases, and code. Therefore, traceability makes it possible to guarantee that all requirements are taken into account and corrections can be traced throughout the development of the project. At that time, communication facilitates effective communication between project stakeholders regarding the requirement and involves the conversion of updates, resolution of problematic aspects, and facilitating that all members of the working group have a common understanding of the formulated requirements. Also, control over the versions of requirements documents in order to track corrections in time. The obtained results of the conducted research serve as a basis for further research in the part of more effective use of requirements management methods during the implementation of projects in conditions of uncertainty.

Keywords: requirements management, requirements management methods, project management, project implementation, conditions of uncertainty, software development.

JEL Classification: R58

1. INTRODUCTION

The field of software development and project management of requirements management is characterized by the fact that it has extremely important processes that ensure the effective capture, documentation and tracking of project requirements throughout all stages of their implementation, even under conditions of uncertainty. They are a good foundation for successful project implementation, maintaining clear relationships with stakeholders, managing development efforts, and creating products that meet customer requirements.

It has been established that requirements management means systematic collection, documentation, analysis, prioritization, tracking and control of project requirements. Such requirements will include functional, non-functional, technical and business components. At the same time, the main goal is for all stakeholders to have a clear and common understanding of the project's tasks.

Throughout all phases of the project life cycle, the requirement must be refined, validated, and verified to ensure that it is accurate, feasible, and consistent with the project objectives. Such a process reduces the risks associated with changes in volumes, missed deadlines and customer dissatisfaction in conditions of uncertainty.

So, it is clear that requirements management means systematic and important processes within project management and development of software products, which are aimed at the implementation of several of the most important tasks. Such tasks must ensure that the project requirement is recorded, documented, evaluated and verified in such a way that contribute to the successful implementation of the project and satisfy the needs of customers.

2. THEORETICAL FRAMEWORK

It should be noted that precisely established requirements are a fundamental component in the process of decision-making regarding design and development. When a developer has a clear understanding of what exactly he needs to accomplish, he can focus on building quality products that meet the set parameters. This in turn makes it possible to minimize rework, reduce the number of defects and ensure the launch of a more perfect final product.

Another important component is that requirements management must ensure that the final product meets customer expectations. At the same time, by verifying and establishing customer needs at the early stages of the process under conditions of uncertainty, the project team has the opportunity to make the necessary adjustments and avoid costly changes later. When a launched product meets or exceeds customer expectations, the degree of satisfaction increases, ensuring the development company's overall reputation grows (Piatnychuk et al., 2022; Piatnychuk, & Pyatnychuk, 2021).

It has been established that clearly organized requirements management processes in conditions of uncertainty make it possible to use project resources more effectively. Resources such as financial, labor, and time resources can be allocated for use based on prioritization of relevant requirements. This, in turn, makes it possible to prevent unnecessary isolation of functional characteristics with lower priorities and to optimize the use of available resource provision.

3. RESEARCH OBJECTIVE, METHODOLOGY AND DATA

Requirements management makes it possible to identify likely risks and uncertainties at the initial stages of the project life cycle. Taking into account the results of a thorough assessment and verification of requirements, developers are able to identify possible obstacles, problems of a technological nature and the reality of implementation. This approach allows developers to plan and address such challenges before they become significant problems.

It is also important to note that requirements management and requirements development are considered distinct but related processes in the fields of project management and software development.

Although they have common features, they are intended for different purposes and stages of the project life cycle. Among the main differences between these concepts, the following can be distinguished.

According to research conducted by IBM in the field of IT, 60% of time spent by software development organizations is a result of an ineffective approach to requirements management (Ryabokin, 2014). In organizations that do not have sufficient business analysis capabilities, projects are three times more likely to fail than to succeed. Failure to fully identify, clearly articulate customer requirements, and track changes in requirements are typical causes of disruption to project timelines and budgets. As a result, errors in the requirements that are revealed at the later stages of the project amount to 30-40% of the total cost of the project budget. With proper requirements definition and management, project cost overruns can be reduced by 20% by reducing the number of inaccurate, incomplete and lost requirements. Therefore, the task of organizing an effective requirements management process is urgent (Ryabokin, 2014).

Thus, the purpose of the article is to study the issue of application of software requirements management methods during project management under conditions of uncertainty.

4. RESULTS AND DISCUSSION

Managing requirements under uncertainty requires special methods and approaches, as uncertainty can significantly complicate the process of defining, gathering and managing software requirements. We consider it necessary to consider some existing methods and approaches to requirements management under conditions of uncertainty.

Methods of managing software requirements during project management under conditions of uncertainty: Agile approach; Requirements Elicitation; Progressive Elaboration; Rational Unified Process; Minimum Viable Product, etc.

Flexible (Agile) approach (Fig. 1). Agile-based software development methodologies such as Scrum, Kanban, and Extreme Programming (XP) are actively used to manage requirements under uncertainty. Agile approaches allow you to change requirements during development and quickly adapt to changing conditions. Figure 1.1 shows the Agile methodology model (Dreissigacker, 2022; Kushnir, 2023).

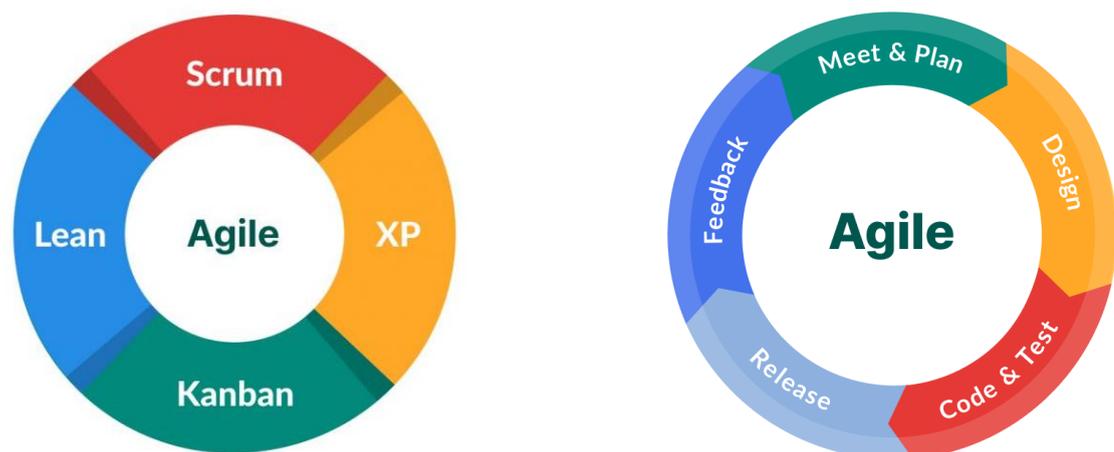


Fig. 1. Software development methodologies based on Agile

Source: (Dreissigacker, 2022; Kushnir, 2023)

It is worth noting that Agile also means a strategic approach to software development and project management, which emphasizes the ability to quickly adapt to changes. In the context of requirements management, it is defined as follows ("What is Agile Software Development", 2022):

1. Collaboration with interested parties: The Agile approach emphasizes close interaction with the

client, users, and other stakeholders throughout the development process. Requirements may change during the project and it is important to identify and account for these changes.

2. Iterations and incremental development: the work product is developed iteratively, in small parts (increments) that are quickly implemented and evaluated. This allows flexibility to adapt requirements based on experience and changing user needs.

3. Changeability of requirements: An Agile approach to requirements management involves accepting that requirements can change. It is important to have mechanisms to effectively identify, evaluate and integrate changes in the development process.

4. Simplifying documentation: Agile emphasizes human interaction more than documentation. Although documentation is important, an agile approach favors simplified and clear documentation, as well as active communication between team members.

5. Testing and Quality Assurance: Agile approach includes continuous testing and quality assurance throughout the development process. This helps to identify and correct problems quickly, ensuring high product quality.

Consequently, an agile approach to requirements management is the basis for many software development methodologies, such as Scrum, Kanban, etc. ("What is Agile and how to apply it in business", 2021). Its core principles are aimed at building effective and adaptive teams to achieve successful results in a changing development environment.

Requirements Elicitation is the process of systematically collecting, analyzing, documenting and specifying requirements for a system or software. This process is a key stage in requirements management and is an important component of software product or system development ("What Is Elicitation: Top Requirement Elicitation Techniques for 2023", 2023).

In Fig. 2 shows the method of advanced analysis of requirements (Requirements Elicitation).

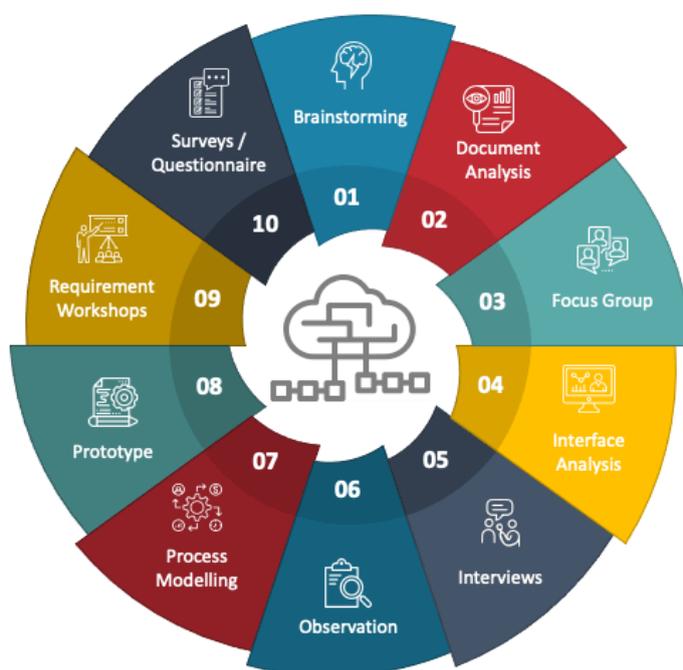


Fig. 2. Requirements Elicitation

Source: ("What Is Elicitation: Top Requirement Elicitation Techniques for 2023", 2023)

Advanced requirements analysis includes several important stages ("What Is Elicitation: Top Requirement Elicitation Techniques for 2023", 2023; Laplante, 2009):

1. Collection of requirements. This phase includes interaction with stakeholders to identify their needs and expectations from the system. Methods such as interviews, focus groups, questionnaires, and

prototyping can be used to effectively gather requirements.

2. Requirements analysis. At this stage, an in-depth analysis of the information obtained at the stage of gathering requirements is carried out. Requirements can be classified, specified and prioritized. It is also important to check requirements for inconsistency and consistency.

3. Documentation of requirements. Collected and analyzed requirements should be documented in detail. This may include creating requirements specifications, diagrams, tables, and other documents that clearly define what the system must do.

4. Specification of requirements. Sometimes problematic aspects or ambiguities may arise during requirements analysis. Clarification of requirements involves solving these issues by communicating with interested parties and clarifying the details of requirements.

5. Approval of requirements. After specifying the requirements and documenting them, they must be approved by all interested parties. This is an important stage, because the approved requirements become the basis for the further development of the project.

Advanced requirements analysis is used in many software development methodologies, including agile approaches (Agile), traditional waterfall methods (Waterfall) and others. This process helps ensure that the product or system being developed meets the expectations of customers and users, and is a key element of effective requirements management (Yatsyshyn, Petryk, & Markovets, 2017).

The method of "Progressive Elaboration" in requirements management indicates that requirements can be revealed, clarified and developed gradually over time. This principle is especially relevant in flexible (Agile) software development methodologies, but can also be applied in other approaches. In Fig. 3 shows the general scheme of gradual disclosure (Tsonev, 2019).

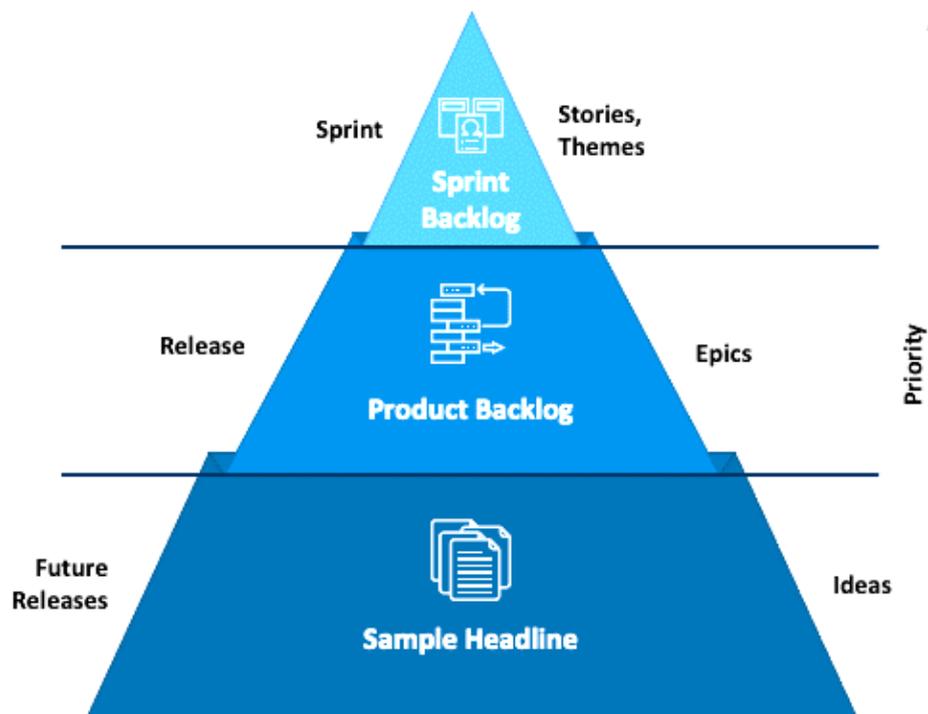


Fig. 3. Progressive Elaboration

Source: (Tsonev, 2019)

The main ideas of the "Gradual disclosure" method ("Top 12 Best Software Development Methodologies & its Advantages / Disadvantages", 2017):

1. Changing requirements over time. Project requirements often cannot be fully defined at the beginning of the process. Progressive disclosure involves the assumption that new requirements or refinements to existing ones may arise during development or even after the product is released.

2. Clarification and expansion of requirements. Instead of spending a significant amount of time defining all the requirements at the outset, teams can start with a basic understanding and then gradually refine and develop it based on the knowledge and experience gained.

3. Cooperation with interested parties. The principle emphasizes the importance of continuous engagement with stakeholders to ensure that their needs and expectations meet evolving requirements.

4. Flexibility and adaptability. Teams must be flexible and ready to adapt to changes in requirements. This allows us to respond more quickly to new requirements and changing conditions, ensuring that the product meets the current needs of users.

It is worth noting that the principle of “Gradual disclosure” is taken from flexible methodologies, where changes in requirements are considered the norm, not the exception. This emphasizes the dynamics and evolution of requirements throughout the development process, allowing for better adaptation to changes in the project environment and user requirements.

Iterative and incremental development models are software development strategies often used in requirements management. These models allow you to effectively manage change and improve the product by adding new features iteratively or incrementally. For example (Kyrylenko, 2018):

1. Iterative model: involves constant repetition of iterations, each of which is a mini-project that includes stages of development, testing and implementation. At each iteration, requirements may be changed or new requirements may be added. Requirements can be refined and supplemented during each iteration, which allows you to effectively respond to changes and ensure better adaptation to user needs.

2. Incremental model: involves adding new functionality or increments (parts) to an already existing product. At the same time, new requirements can be included in new increments, which allows for gradual improvement of the product. The requirements for each increment can be defined and specified before starting work on it.

Both models are designed to improve product quality and collaboration with the client or stakeholders. They allow for greater flexibility and adaptability to change, which can be particularly useful in projects where requirements may change frequently or may not be fully defined at the outset. In Fig. Figure 4 shows the diagram of the iterative model.



Fig. 4. Rational Unified Process

Source: (Kyrylenko, 2018)

Among the models of iterative and incremental development, it is possible to single out the use of models such as RUP (Rational Unified Process) or Spiral, which allows software to be developed iteratively and incrementally, which makes it possible to take into account uncertainty in the development process (Likarenko, 2020).

For example, Rational Unified Process (RUP) is an iterative and incremental approach to software

development that was developed by Rational Software Corporation (acquired by IBM) and presented as one of the project management and software development methodologies. Requirements management in RUP includes several important elements: requirements collection; requirements analysis; specification of requirements; validation and verification of requirements. RUP aims to create high-quality software products by using flexible, iterative and incremental development methods. This approach allows for adaptation to changes in requirements and ensures better cooperation with customers and stakeholders (Kuzminykh, Koval, & Taranenko, 2023).

In turn, the “Spiral” model is an iterative model of software development that combines elements of other approaches, such as incremental, evolutionary and flexible (Agile) approaches. This model was introduced by Barry Boehm in 1986 and focuses on risk management in development. Management of requirements in the “Spiral” model includes the following stages and principles: definition of goals; risk analysis; development and testing; evaluation and planning of the next iteration; repetition. Requirements management in the “Spiral” model is designed to help effectively identify and clarify requirements at each stage of development, as well as manage risks that may arise due to uncertainty of requirements or other factors (Ovsyankin, 2019).

The MVP (Minimum Viable Product) method is an important element of requirements management and software development. MVP involves creating the minimum necessary functionality that satisfies the basic needs of the user or solves a specific problem. The use of the MVP principle in requirements management has several important aspects (Shapovalova, 2022):

1. Quick access to the market. MVP allows you to bring a product to market as quickly as possible without waiting for all features to be fully developed. This helps to validate the idea and receive surface feedback from users.

2. Testing hypotheses and requirements. MVP allows you to test hypotheses about what really matters to users and experiment with different product concepts. Using an MVP, you can quickly and efficiently find out whether the product meets the needs of the market.

3. Gradual development. Using MVP allows you to gradually add new functionality and improve the product based on feedback and real-world user experience. Each iteration may include new requirements that are based on received data and changed conditions.

4. Cost minimization. Creating an MVP allows you to avoid unnecessary costs for the development of functionality that may not be important to users or may be changed during the development process.

5. Focus on the main thing. MVP helps to focus on core requirements and functionality, without getting distracted by details or excessive feature development.

In Fig. 5 shows a generalized scheme of MVP (“Top 12 Best Software Development Methodologies & its Advantages / Disadvantages”, 2017).

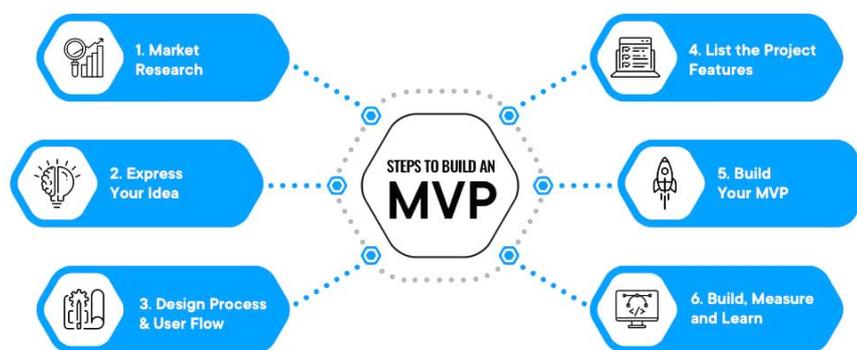


Fig. 5. Minimum Viable Product

Source: (“Top 12 Best Software Development Methodologies & its Advantages / Disadvantages”, 2017)

Using the MVP principle in requirements management allows you to effectively interact with users, implement basic requirements and quickly respond to changes in their needs.

The “Just-in-Time” method in software requirements management defines a strategy when requirements are defined, documented and implemented only at the moment of their real need, that is, as close as possible to the time when they become necessary for the further development of the system. This approach arose in the context of flexible (Agile) development methodologies, where the emphasis is on flexibility, adaptability and quick response to changes. In Fig. 6 shows the generalized “Just-in-Time” scheme (Choudhary, 2022).

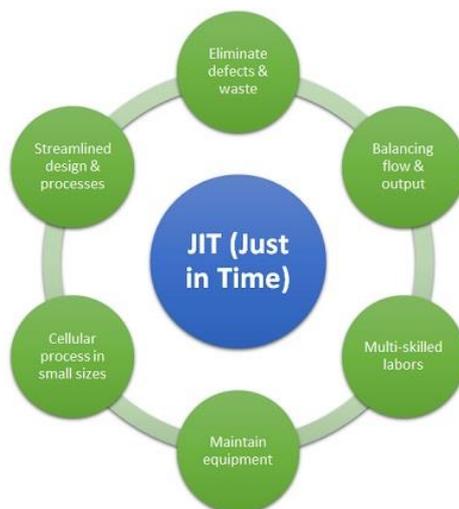


Fig. 6. Just-in-Time

Source: (Choudhary, 2022)

Key aspects of the “Just-in-Time” principle in requirements management (“JIT (Just In Time) – Definition, Importance & Example”, 2023):

1. Determination of requirements if necessary. Requirements are defined only when it becomes clear that their definition has become important for the continuation of work on the project. Avoiding the detailed definition of all possible requirements at the beginning of the project.

2. Emphasis on variability and adaptability. The “Just-in-Time” approach is designed to take into account that requirements can change during the project life cycle, and they must be ready for changes at any moment.

3. Iterative and incremental development. Just-in-Time is often used in conjunction with iterative and incremental development, where functionality is added incrementally and requirements are refined along the way.

4. Minimizing redundancy. Avoiding detailed definition and documentation of requirements that may become obsolete or unnecessary due to changes in the project context.

5. Active involvement of interested parties. Change in requirements can occur as a result of active interaction with stakeholders, rather than pre-planned planning.

The “Just-in-Time” principle helps teams to be more flexible, to respond more quickly to changes in requirements and to maintain the relevance of product functionality at each stage of its development.

5. CONCLUSION

Therefore, Management of requirements under conditions of uncertainty should be considered as continuous processes of system management of requirements in all stages of the life cycle of the project, which involves activities aimed at supporting, ensuring and verifying requirements from the moment of their formation to their implementation and beyond. At the same time, management requirements should focus on organization, communication and development of requirements, among which the

following can be distinguished. Methods of managing software requirements during project management in conditions of uncertainty: Agile approach; Identification of requirements; Progressive development; Rational unified process; Minimum Viable Product, etc.

The use of requirements methods during the implementation of projects becomes especially relevant in conditions of uncertainty, after which these methods can effectively cope with dynamic ones and ensure the flexibility and adaptability of the project. In general, in conditions of uncertainty, effective requirements management must ensure flexibility, speed up change review time, and ensure project effectiveness in an uncertain environment.

REFERENCES

- [1] Choudhary, B. (2022). Top 15 Software Development Models (Not Just Agile And Waterfall). *Finoit*. <http://surl.li/ovpsz>
- [2] Dreissigacker, U. (2022). Why Agile is so Popular in Project Management. *GanttPRO*. <https://blog.ganttpro.com/en/why-agile/>
- [3] JIT (Just In Time) – Definition, Importance & Example (2023). *MBASkool*. <http://surl.li/ovpuc>
- [4] Kushnir, A. (2023). Types of Software Development Models. *Bamboo Agile*. <http://surl.li/ovpvg>
- [5] Kuzminykh, V. O., Koval, O. V., & Taranenko, R. A. (2023). IT project management models and tools: teaching. Kyiv. <http://surl.li/ovpvz>
- [6] Kyrylenko, L. V. (2018). Standardization of information technologies in Ukraine. II All-Ukrainian Science-Pract. conf. *Theoretical and practical problems of using mathematical methods and computer-oriented technologies in education and science*. pp. 152–155. (in Ukr.)
- [7] Laplante, P. (2009). *Requirements Engineering for Software and Systems* (1st ed.). Redmond, WA: CRC Press, 268 p.
- [8] Likarenko, Ya. (2020). Iterative development model. 5 lessons we learned. DOU. <http://surl.li/ovpxe> (in Ukr.)
- [9] Ovsyankin, A. M. (2019). Selection of effective models of the requirements management system in projects. *NTU «Khpі»*, 1 (1326), pp. 55–62. (in Ukr.) <https://doi.org/10.20998/2413-3000.2019.1326.9>
- [10] Piatnychuk, I., & Pyatnychuk, H. (2021). Modern information technologies and services in public institutions. *Journal of Vasyl Stefanyk Precarpathian National University*, 8, 3, pp. 43-51. <https://doi.org/10.15330/jpnu.8.3.43-51>
- [11] Piatnychuk, I., Boryshkevych, I., Sala, D., Tomashevska, A., & Hryhoruk, I. (2022). Online Tools in Providing Feedback in Management. *Journal of Vasyl Stefanyk Precarpathian National University*, 9,3, pp. 6-17. <https://doi.org/10.15330/jpnu.9.3.6-17>
- [12] Ryabokin, Yu. M. (2014). Requirements management software. *Project management, system analysis and logistics*, 14(1). <http://surl.li/ovqcy> (in Ukr.)
- [13] Shapovalova, M. (2022). What does MVP mean in software development? *Definition & Guide to create a powerful product*. *Voypost*. <https://www.voypost.com/blog/what-does-mvp-mean>
- [14] Top 12 Best Software Development Methodologies & its Advantages / Disadvantages. (2017). *AadiIT Services*. <http://surl.li/ovqge>
- [15] Tsonev, N. (2019). Top 12 Software Development Methodologies. *Businessmap*. <http://surl.li/ovqhb>
- [16] What is Agile and how to apply it in business (2021). *Brainrain*. <http://surl.li/ovqjc> (in Ukr.)
- [17] What is Agile Software Development? (2022). *Inflectra*. <http://surl.li/ovqia>
- [18] What Is Elicitation: Top Requirement Elicitation Techniques for 2023. *Simplilearn*, 2023. <https://www.simplilearn.com/what-is-elicitation-article>
- [19] Yatsyshyn, V. V., Petryk, N. M., & Markovets, O.O. (2017). Management of software requirements based on the Requirements Management Maturity model. Materials of the VI International Scientific and Technical Conference of Young Scientists and Students. *Actual tasks of modern technologies*. <http://surl.li/ovqlv> (in Ukr.)

Iryna Piatnychuk, PhD in Economics, Head of the Department of Management and Business Administration, Vasyl Stefanyk Precarpathian National University, Ivano-Frankivsk, Ukraine;

ORCID ID: 0000-0003-2876-6422

Aleksander Iwaszczuk, PhD in Economics, Assistant professor, Poland;

ORCID ID: 0000-0002-0695-8864

Address: Iryna Piatnychuk, Vasyl Stefanyk Precarpathian National University, 57, Shevchenko Str, Ivano-

Frankivsk, 76018, Ukraine.

Aleksander Iwaszczuk, Cracow University of Technology, College of Social Sciences, 24 Warszawska Str., Krakow, 31-155, Poland.

E-mail: iryna.piatnychuk@pnu.edu.ua, aleksander.iwaszczuk@gmail.com

Received: October 02, 2023; **revised:** November 09, 2023; **accepted:** December 14, 2023; **published:** December 31, 2023

П'ятничук Ірина, Іващук Олександр. Методи управління вимогами під час реалізації проектів в умовах невизначеності. *Журнал Прикарпатського університету імені Василя Стефаника*, **10** (4) (2023), 93-102.

Метою статті є дослідження питання щодо застосування методів управління вимогами до програмного забезпечення під час управління проектами в умовах невизначеності. Встановлено, що управління вимогами за умов невизначеності слід розглядати як постійні процеси системного керування вимогами впродовж усіх стадій життєвого циклу проекту, що передбачає діяльність, яка спрямована на підтримання, простежування й перевірку вимог від моменту їх формування і до їхньої імплементації й далі. За результатами проведеного дослідження виявлено, що найпоширенішими методами управління вимогами є: гнучкий (Agile) підхід; розширений аналіз вимог; метод «Поступового розкриття»; моделі ітеративного та інкрементного розроблення; метод «Мінімально життєздатний продукт»; метод «Just-in-Time» та ін. Встановлено, що використання методів управління вимогами до програмного забезпечення набуває особливої актуальності в умовах невизначеності, оскільки ці методи дозволяють ефективно впоратися з динамічними змінами та забезпечити гнучкість проекту. Виявлено, що управління вимогами має зосереджуватись на організації, комунікуванні й розвиванні вимог, серед яких можна виокремити наступні: управління змінами, відстежуваність вимог, комунікація, контроль за версіями документів вимог та ін. Досліджено, що управління змінами, що передбачає внесення коректив до визначених вимог в ході реалізації проекту, а також передбачає здійснення оцінки впливу коректив, одержання дозволу на модифікування й оновлення потрібних документів. У свою чергу, відстежуваність означає визначення й підтримку відстежуваних взаємозв'язків вимог з іншими артефактами проекту, а саме: проектна документація, випадки під час тестування й коди. Тому відстежуваність дає змогу гарантувати, що усі вимоги урахується і корективи можна простежувати впродовж розроблення проекту. У той час, комунікація сприяє ефективному спілкуванню між стейкхолдерами проекту стосовно вимоги й передбачає конверсію оновлень, розв'язання проблемних аспектів й сприяння тому, щоб всі учасники робочої групи мали єдине розуміння щодо сформованих вимог. Також проведення контролю за версіями документів вимог для того, відстежувати корективи в часовому вимірі. Отримані результати проведеного дослідження виступають основою для наступних досліджень у частині більш ефективного використання методів управління вимогами під час реалізації проектів в умовах невизначеності.

Ключові слова: управління вимогами, методи управління вимогами, управління проектами, реалізація проекту, умови невизначеності, розробка програмного забезпечення.