



USING WATERFALL, ITERATIVE AND SPIRAL MODELS IN ERP-SYSTEM IMPLEMENTATION PROJECTS UNDER UNCERTAINTY

Dmitry Yu. Stepanov

RTU MIREA, department of Corporate Information Systems, PhD, professor assistant MIREA

Abstract: *the article considers the use of cascade and multi-pass implementation models of corporate information systems in case of business and technological uncertainty. A review of waterfall, iterative and spiral ERP-systems implementation models is given. The business and technological uncertainties inherent in software systems implementation projects are introduced. The basic principles of development complex applications in ERP-systems are analyzed, including the rules of evolution and functionality. One compares business uncertainty for refined requirements in the waterfall and Agile-based implementation models, which operate with a change request and allocation requirements to a new round of development respectively. It is summarized there is no or minimal technological uncertainty in ERP-systems implementation projects, however high business uncertainty exists, which can not be decreased by any basic implementation approaches. The application area of the waterfall and multi-pass implementation models is clarified for ERP projects from scratch, rollout and evolution under business uncertainty.*

Keywords: *preconfigured software solutions, reduce efforts, cascade model, iterative model, implementation methods, application from scratch, analyze business uncertainty, project preparation, preparing designs, post go-live support, user acceptance test, migration of master data, transactional data, user training, analysis of the feasibility, risk management, characterizing modularity, system control theory, minimal technological uncertainty.*

1. Introduction

Implementation of corporate information systems, in particular, ERP-systems, is carried out mainly with the use of preconfigured software solutions from various vendors. Examples of such program solutions are integrated applications SAP ERP, Oracle eBS, Microsoft Dynamics NAV and 1C ERP. The use of approved software products allows you to reduce efforts during implementation, however the problems of application customization and build are still open [1]. The focus on digitalization of private and public sectors dictates special requirements for company's maturity and hence technological and innovative nature of the enterprise, which is impossible to imagine without using modern software systems. The topics of artificial intelligence, machine learning, the internet of things and cloud solutions are reflected in corporate information systems, which makes it necessary to significantly rethink the approach to planning and implementing software systems.

Realization of all information systems is based on several classical implementation models. Nowadays the most popular are applied methods based on the Agile methodology. The Agile methodology allows you to look at the task of programming complex software solutions differently, despite all project uncertainties. If you carefully read the literature on the Agile methods [2-3], you will misunderstand that this is a panacea for everything, especially in comparison with the classic development methods. Is this really the case? We will try to find an answer to this question in the current article.



2. Problem statement

The purpose of this paper is to demonstrate the usage of the cascade, iterative and spiral models in ERP-systems implementation projects under business and technological uncertainty. The following tasks will be considered to achieve the goal:

- consider waterfall, iterative and spiral implementation models;
- define business and technological uncertainty in ERP-systems implementation projects;
- explore basic principles of developing complex software programs in ERP-systems;
- analyze business uncertainty in cascade and multi-pass implementation models;
- refine the usage of implementation models in ERP-projects.

3. ERP-system implementation methods

There are three classic models of developing software programs regardless of their complexity and business orientation. The waterfall, iterative and spiral models were proposed more than forty years ago and are applicable to implement ERP-systems [4]. All other methods are derived from them (V-model, Scrum, Kanban and many others). It is important to say that previously by implementation of information systems mainly meant development of an application from scratch and configuration played unimportant role. Now the rules of the game have changed: corporate information system is considered as a customized software solution which can be rebuilt. Thus, project of implementation modern ERP-systems is about proper configuration and modification of the package solution. Statistics on the use of foreign ERP-systems from SAP, Oracle and Microsoft companies in Russian enterprises indicate following [5]: 30% of business requirements are covered by ERP-system configuration, 60% of requirements need system modification and remaining 10% are distributed between first two points, depending on the company's industry specifics.

The cascade or waterfall model was proposed in 1970. The implementation project according to this model requires completion of all tasks at the stage, transition to the next stage is possible only if previous ones are successfully completed. It is forbidden to skip the stages, go back to the previous ones and repeat them. The ERP-system implementation project based on this model consists of the following activities:

- project preparation;
- identification, detailing and prioritization of requirements;
- paring designs and functional specifications;
- customization and development of software system;
- performing unit, integration and user acceptance tests;
- data migration and end user training;
- go-live and post go-live support.

The model is used in implementation projects of ERP-systems, integrated with a variety of third-party subsystems, where even the most minor amendments in requirements lead changes in the project time, scope and costs. Due to this, the waterfall model is often used in «from scratch» and «rollout» implementation projects characterized by quite a lot of effort. The main weak point of the model is that there is no feedback loop between project stages. Therefore, errors made during ERP-system conceptual design are detected only during user acceptance test, when it is impossible to make significant changes to the program code. Examples of the waterfall model are implementation methodologies such



as ASAP (Accelerated SAP) from SAP AG, MDSS (Microsoft Dynamics Sure Steps) from Microsoft and AIM (Application Implementation Method) as part of Oracle Unified Method concept [6].

A distinctive feature of the iterative model is providing a feedback between the project stages, returning to previous stages as well as multiple stages. The model was proposed in 1957. The development process is reduced to a series of fast, independent and adaptive iterations that rapidly bring semi-ready software solution [7]. Each iteration ends with a demonstration of the intermediate product to the customer for quick gathering comments and error detection. During iterations the understanding of final software product may change, so it is allowed and encouraged to add new requirements. The duration of each iteration varies between 1-4 weeks, and the number of iterations is calculated based on the end date of the project or budget. The ERP-system implementation project according to the iterative model consists of the following steps:

- identification, detailing and prioritization of business requirements;
- calculation of number and duration of development iterations;
- realization of a part of ERP-system, functional and integration testing, followed by demonstration of semi-finished software product to the customer in order to clarify requirements to be implemented in further iterations of development, transfer of the software solution to the productive environment (for each iteration of development) ;
- user acceptance test;
- migration of master and transactional data, end user training;
- go-live and support.

As you can see from the above list, there is no design and documentation of the solution, this is indeed the case. However, since post go-live support ends up transferring the software system to customer support, in practice, documentation still has to be prepared. In comparison with the waterfall model, the iterative scheme is used in evolution projects where requirements can be formulated only superficially by end users. The increase in efforts, the use of expert evaluation and rough criteria to complete the project are the main disadvantages of the iterative model.

The spiral implementation model was proposed in 1986 and combines both a waterfall and an iterative approach. The focus of this model is on risk assessment and decision-making readiness. The spiral model is often considered as a special case of the iterative model, each cycle of development of this model is comparable to iteration and is characterized by:

- definition of goals and objectives;
- qualitative and quantitative risk management;
- modification and testing of solution;
- evaluation of the results and planning of the next round of development.

The ERP-system implementation projects based on the spiral model are similar to the iterative scheme with the only difference that each round of development, in addition to risk management, includes:

- assessment of a need for the next round of development;
- assessment of understanding of system requirements;
- analysis of the feasibility to complete the project.

Managing deadlines, resources, budget, quality, communications and other project parameters, as well as assessing the need to complete the next development rounds require a lot of effort. The model is applied in projects for the



evolution of ERP-systems [8]. For small subprojects, time spent on processing and evaluating the project parameters will exceed the time needed to finalize the software system itself.

Iterative and spiral models are often called multi-pass models, the waterfall scheme is called one-pass model. As mentioned above, iterative and spiral models have eliminated main drawback of the cascade scheme: lack of a feedback loop, which does not allow to correct errors from previous stages in time. The formalization of the principles providing feedback loop is reflected in the concept of flexible development, called Agile [2]. The Agile methodology includes a few application methods, such as Scrum, Kanban, FDD (Feature Driven Development), as well as XP (eXtreme Programming) and RAD (Rapid Application Development). The Agile approach has four core values and twelve principles declared in the manifesto [9]. The implementation of information systems in line with these values and principals should become more efficient. The features of implementing ERP-systems based on the Agile manifesto include:

- maximum involvement of customers in project tasks;
- multiple modification of program prototypes that are regularly demonstrated to the customer and transferred to the production environment;
- readiness and encouragement of changes in business requirements, providing the necessary feedback loop.

To get comments on the development, the program prototype is periodically shown to the customer. Bugs identified during the demonstration are included in the next iteration as additional refined business requirements. It is the customer who distributes the requirements for each iteration, moreover, priority of business needs may change once iteration is completed. Thus, the client decides what and when will be implemented in terms of the functionality of the software solution.

4. Uncertainty in the development of software systems

Misunderstanding of what a developed function or graphical user interface should look like to meet the initial business requirements can be avoided by regular communication with the client. Let's call this type of uncertainty technological, which characterizes misunderstanding of what a developed solution should look like and what it should do.

Hundreds and sometimes thousands of requirements are identified in ERP projects from scratch and rollout. Prioritization of business requirements is carried out according to the following approach: legal requirements that are mandatory in Russian Federation processed at first, then requirements that ensure business continuity and heavily reduce manual work. Such requirements are classified as obligatory for implementation. Usually, mandatory business requirements are clearly formulated and do not need to be clarified. Optional requirements have a low priority.

A lot of questions in the implementation project are caused by low-priority business requirements. Roughly formalized requirements are characterized by uncertainty, which is often called business uncertainty. Business uncertainty is defined by unformalized, contradictory and often incomplete requirements for developing program. There are technological and business uncertainties defined by the uncertainty of instruments and the finite uncertainty in [2]. The cascade implementation model minimizes both uncertainties already at the initial stages with the help of assumptions. On the other hand, iterative and spiral models allow for uncertainty and reduce it from iteration to iteration.

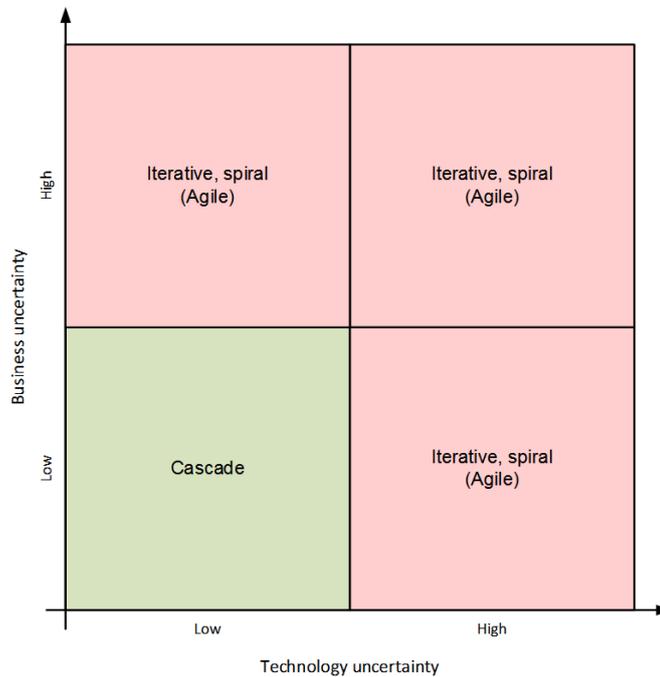


Fig. 1. *The use of cascade, iterative and spiral models for developing software systems under uncertainty*

There is a matrix demonstrating the use of waterfall and multi-pass models to implement information systems under uncertainty (see figure 1,) described in paper [10]. The matrix is applicable to develop any software system and is an adapted version of the strategic management and organizational dynamics model introduced by Ralph Stacy. It was said before, only 30% of business requirements can be covered by ERP-system configuration, the rest requires development. The feature of ERP-system customization is that technological uncertainty is minimized, due to configuration allows you to cover requirements mainly one way without any other options. Thus, the Agile methodology is reasonable to use in ERP-system modification projects but not in customization projects. Taking into account the paper [11], explaining the use of the Agile Scrum method in SAP projects, it can be formulated that the Agile approach is applicable for projects of the evolution of ERP-systems.

The use of ERP-systems allows you to automate production, sale, banking and many other types of organizations. The functionality of the information system is built based on existing business objects interacting with each other and describing business processes, as well as potential enhancement points used to enrich the software system. Modification of any ERP-system is carried out under technical constraints and has predictable number of changes, so it is wrong to talk about the presence of technological uncertainty. Summarizing the above hypotheses, let's shape the following statement: ERP-system implementation projects characterized by a limited number of business objects and functions to meet new requirements have minimal technological uncertainty and high business uncertainty.

New state laws that are obligatory in Russian Federation, their variability, ambiguity are the main reasons for business uncertainty. If it is not clear what will change in the law, when the changes will come into force, whether there will be a transition period, it is quite difficult to design and build a program that meets the originally formulated requirements. This leads to the fact that quite working applications are created, but they will not be able to ensure end-

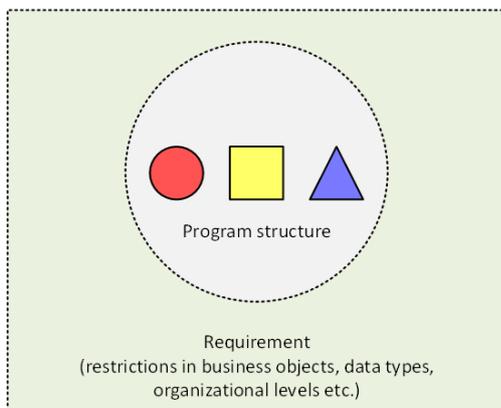
to-end business processes of the enterprise. The question of implementing applications is related to the basic principles of software development, let's look at them in more detail.

5. Principals of design and build of software systems

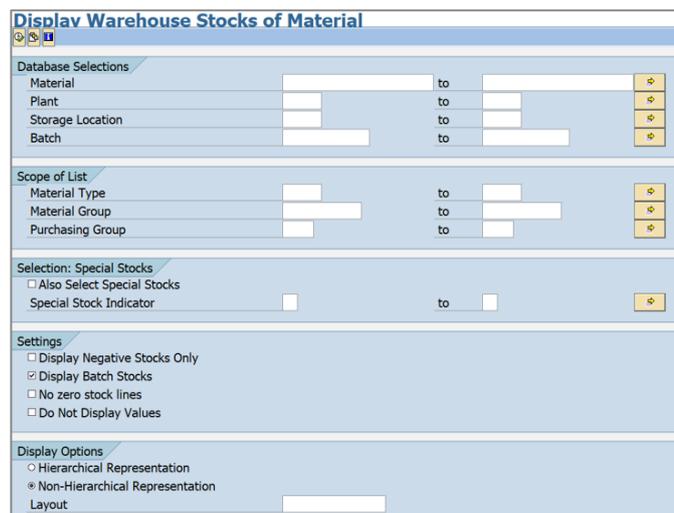
There are several rules that the designed and developed RICEF program must comply with (RICEF is an acronym of the possible software types) [12-13]. For instance, the application must be scalable with an increase in the number of users, flexible to future changes, solve problems optimal way and so on. The principles of software development are grouped into the following topics:

- system control theory, specifying program control, feedback loop and compensation principals;
- system analysis, defining the rules of functionality, evolution and uncertainty;
- programming, characterizing modularity, functional selectivity and reliability of software systems.

It is necessary to consider the rules of evolution and functionality from the above list in more details. Each program undergoes changes, for example, due to new legal requirements, an increase in the number of end users or data volume. The principle of evolution determines the need to design ERP programs taking it into account. Therefore, each program in ERP-systems successfully operates with all kinds of objects and data types, as well as for various organizational levels of the enterprise.



A)



B)

Fig. 2. Graphical illustration of the principal of functionality (A) and an example of its realization in the SAP ERP system (B)

The structure of the developed program should be completely determined by the initially formulated business requirements, which is the basis of the principle of functionality. Thus, a complex program consisting of a great number of buttons, menu items and screens is the result of an unsuccessful attempt to resolve various business problems by means of a single software program (see figure 2). The development of an automated workplace that combines many programs, each of them can be run independently, is an exception to this rule.

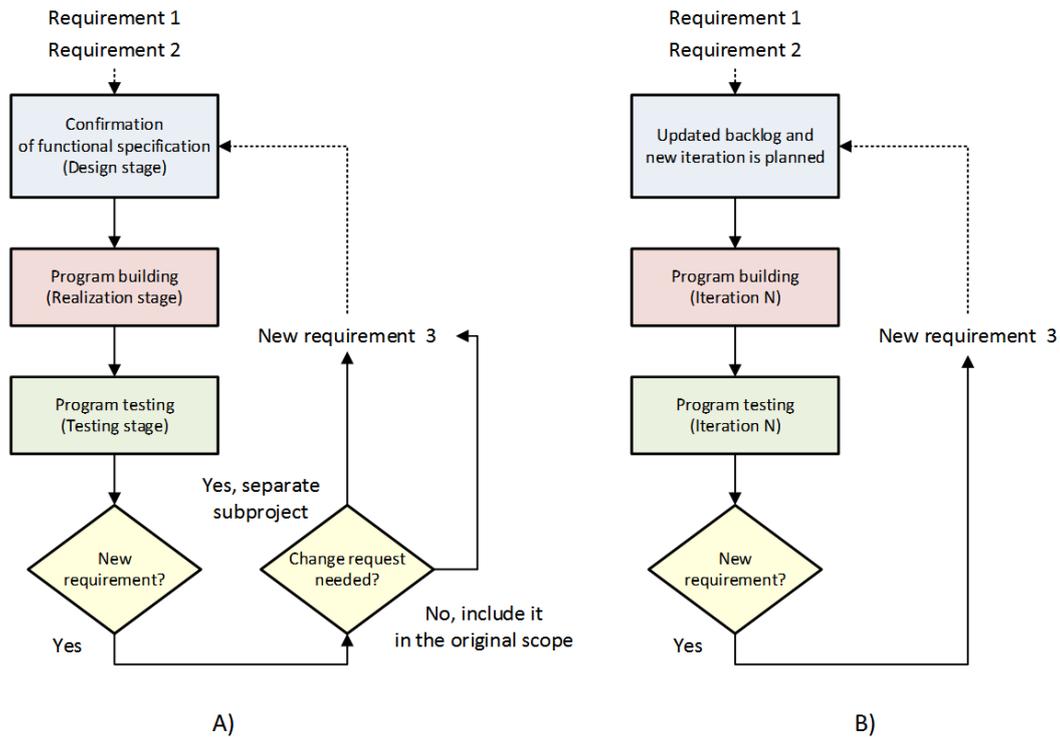


Fig. 3. Managing new business requirement in one-pass (A) and (B) multi-pass implementation models

The procedure of realization business requirements in software systems based on waterfall and multi-pass implementation models using the principle of functionality can be presented as follows. Once requirements are shaped, functional specifications are to be prepared or their implementation is planned for the next development rounds. Then the application is built and tested which leads to clarification of requirements, i.e. new requirements are identified, detailing the original ones. This helps us to formulate the following statement: the waterfall model in ERP-system implementation projects allows you to realize refined requirements under separate independent subprojects by creating change requests, unlike multi-pass models, where new requirements are prioritized and processed in the next development rounds.

New requirements are often included free of charge in the current scope of ERP project implemented using the waterfall model, when the total efforts to realize refinements do not exceed the agreed amount, for example, three man-days (see figure 3). Following the above figure, it can be summarized that both models allow you to manage changes in the initial requirements, providing visibility, flexibility and traceability through the use of regulated procedures.

6. Business uncertainty in the development of ERP-systems

The above statement applies if refined business requirements complement the original ones. Using the principle of evolution as a basis for designing and building software systems allows you to preserve the original architecture of the application without violating the rule of functionality. There is high business uncertainty in real ERP-system implementation projects. For instance, there are initial requirements 1-2 for a process, for which software development is designed and built according to the principal of functionality (see figure 4). Business uncertainty leads to a completely new requirement 3, which is related to the same process and requires a considerable reworking of the original architecture of the solution. In this case, the behaviour is similar to the one given in figure 3, but there is a

dilemma: to completely modify the existing program and save time by violating the principal of functionality or to develop a brand-new application that implements all three requirements, ensuring compliance with the rule of functionality, but requiring a lot of effort.

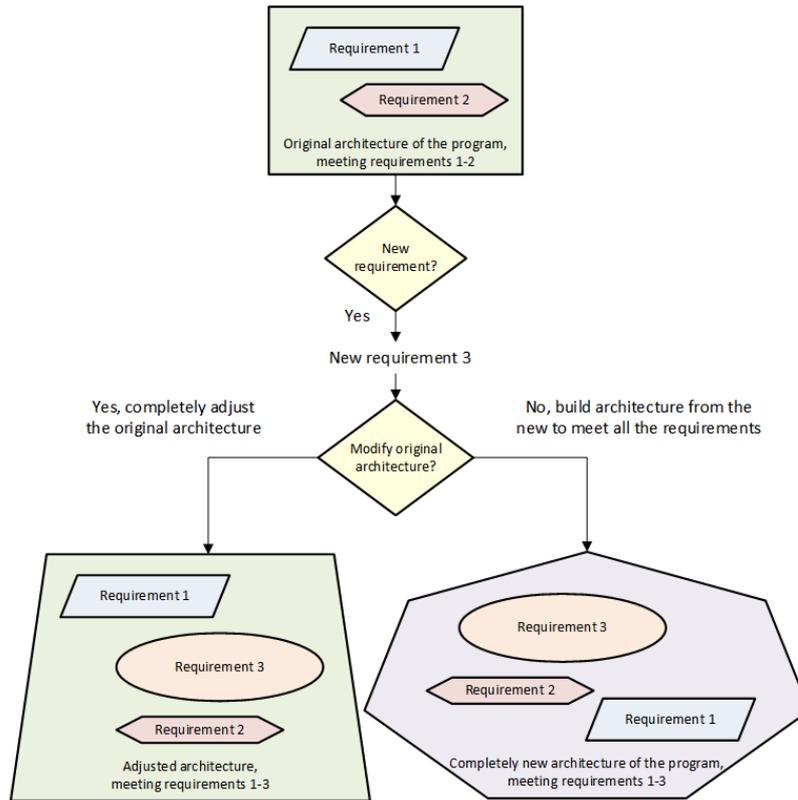


Fig. 4. Violation of the principal of functionality when processing a new requirement

It does not matter which method is chosen, there will be a need to retest the functionality related to requirements 1-2. Therefore, business uncertainty leads to significant changes in the already built software system, zeroing out previous results. Such changes are impossible to predict and the use of modern Agile approaches cannot avoid it. Let's summarize the above in the following statement: business uncertainty creates new requirements. To meet these requirements, essential changes are needed in already developed applications. The amendments cannot be avoided or planned in any ERP-system implementation models.

Table 1. The use of ERP-system implementation models in case of minimal technological uncertainty

Business uncertainty	Model	Project type	Realization type
High	Waterfall	From scratch and rollout	Configuration and development
		Evolution	Configuration
	Waterfall, iterative and spiral (Agile methodology)	Evolution	Development
Low	Waterfall	From scratch and rollout	Configuration and development
		Evolution	Development

Despite the growing popularity of multi-pass development models, the Agile methodology does not provide significant advantages in ERP-system implementation projects in comparison with the waterfall approach. The Agile methodology is focused on a high technological uncertainty, which is avoided in ERP-projects. None of the classical implementation models can reduce the high business uncertainty, so their use is equivalent. Waterfall and multi-pass models allow you to effectively solve the problems of developing complex software systems, while using different organizational procedures and technical methods. Taking into account the introduced statements, we will refine the use of the basic ERP-systems implementation models described in [11] (see table above).

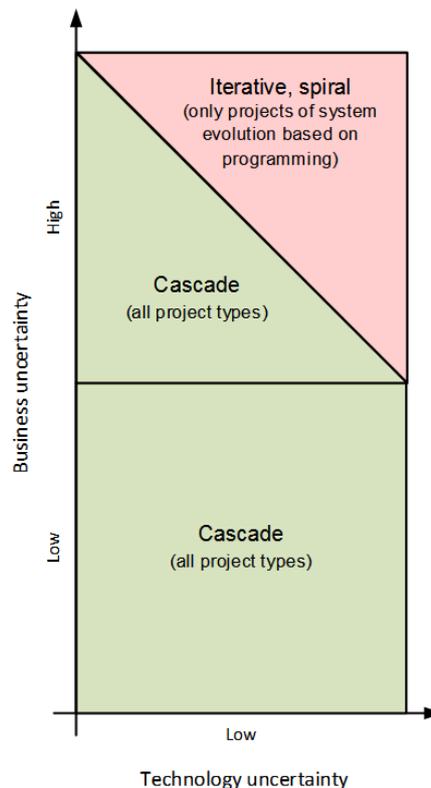


Fig. 5. *The use of waterfall and multi-pass models in ERP-system implementation projects*

Figure 1, presented in chapter 4, demonstrates a matrix of the use of information system implementation models depending on business and technological uncertainties. Let's update the matrix to use it in ERP-systems implementation projects based on the table (the table is given above). To do this, it is necessary to exclude the area of high technological uncertainty, add the cascade model to the first sector and limit the use of Agile methodology only to ERP-systems evolution projects that require development. An updated matrix of the use of waterfall, iterative and spiral models in ERP projects under uncertainty is shown in the figure 5.

7. Conclusions

The functionality of corporate information systems is limited by a given number of technical tools that allow you to implement various requirements for business processes. There is minimal technological uncertainty in ERP-systems implementation projects. However, implementation projects are characterized by business uncertainty admitting fuzzy initial requirements for the system. The cascade implementation model is applicable in the case of minimal business



uncertainty. The average level of uncertainty allows the use of both waterfall and multi-pass models, each model can manage refined requirements. The high business uncertainty violates the fundamental principle of functionality using in program development, thus the choice of the implementation model is not so critical.

Summarizing the results of this paper, it should be noted each ERP-systems implementation model has advantages and disadvantages, allows developing software solutions using different techniques and, unfortunately, is helpless against uncertainty of business requirements [14]. The choice of the implementation model should be made consciously, based on the project regulations and objectives, considering the experience of the project team and technical feasibility, and not guided by fashion trends and novelties.

References

1. Sudakov V A 2016 Corporate information systems (Moscow: Publisher “MAI”) p 96 (in Russian).
2. Cohn M 2005 Agile estimating and planning (Publisher “Pearson”) p 368.
3. Sutherland J 2014 Scrum: the art of doing twice the work in half the time. (Publisher “Cornerstone Digital”) p 258.
4. Ostroukh A V and Surkova N E 2019 Information system design (Moscow: Publisher “Lan”) p 164 (in Russian).
5. Kalyanov G N 2014 Consulting: from business strategy to corporate information system. (Moscow: Publisher “Goryachaya liniya – Telecom”) p 210 (in Russian).
6. Samara T 2018 ERP and information systems. Integration or disintegration (Publisher “John Wiley & Sons Limited”) p 145.
7. Habadi A. An introduction to ERP-systems: architecture, implementation and impacts. International Journal of Com-puter Applications. 2017. Vol. 167. Issue 9. P. 1-4.
8. Ali M and Miller L 2017 Journal of Enterprise Information Management 30(1) 666-692.
9. Principles behind the Agile Manifesto. URL: <http://agilemanifesto.org/iso/en/principles.html>.
10. Schwaber K and Beedle M 2001 Agile software development with scrum. Pearson 176 p.
11. Stepanov D Yu and Velsovskiy A V 2018 Corporate information systems 1 9-17 (in Russian).
12. Stepanov D Yu 2019 Corporate information systems 3(7) 29-52 (in Russian).
13. Stepanov D Yu 2014 Actual problems of modern science 78(4) 258-268 (in Russian).
14. Badewi A, Shehab E, Zeng J and Mohamad M 2018 Business Process Management Journal 24(1) 266-294.

Paper details

Dmitry Yu. Stepanov. Using waterfall, iterative and spiral models in ERP-system implementation projects under uncertainty // Journal of Physics: Conference Series. 11. «XI International Conference on High-Performance Computing Systems and Technologies in Scientific Research, Automation of Control and Production, HPCST 2021». – 2021. – p.120-129. – URL: <https://stepanovd.com/science/article/116-2021-7-uncertainty>.