Analysis of technology selection problems for software development

Igor Kovalenko\textsuperscript{a}, Anastasiia Asieieva\textsuperscript{b}, Inessa Kulakovska\textsuperscript{b,}\textsuperscript{*}

\textsuperscript{a}Department of Software Engineering, 
\textsuperscript{b}Department of Intelligent Information Systems, Petro Mohyla Black Sea National University, 
68 Desantnikov St., 10, Mykolayiv, Ukraine, 54003

Abstract

The developers have quite a wide selection of technologies for implementing software for securing information systems. The complexity of such systems also grows with the selection. The practical value of the system is to use it when deciding on the choice of information systems software development technology.

Keywords: software; development; service; software life cycle; Pareto optimality.

1. INTRODUCTION

1.1. Formulation of a scientific problem

Today, developers have a fairly wide selection of technologies for implementing software for providing information systems. The complexity of such systems also increases with the selection. at the given moment, based on the analysis of the literary data, the development of the software security (software) remains not sufficient. We know that 30–40 \% of all projects are not completed yet. The 70 \% of the projects do not complete the submitted tasks completely, and the average project ends with a delay of 22 \%. In 10 \% of the results, the result does not meet the requirements. In 12 \%, the customer was not sufficiently involved in the work to ensure the product’s characteristics. In 22 \% of the proposals, not all changes made were taken into account.

\textsuperscript{*}Corresponding author

Email addresses: ihor.kovalenko@chmnu.edu.ua (Igor Kovalenko), armandi@ukr.net (Anastasiia Asieieva), inessa.kulakovska@chmnu.edu.ua (Inessa Kulakovska)

ORCID: 0000-0003-2655-6667 (Igor Kovalenko), 0000-0002-4641-7093 (Anastasiia Asieieva), 0000-0002-8432-1850 (Inessa Kulakovska)

\textsuperscript{1}doi: http://dx.doi.org/10.26693/cse2020.01.046
When implementing projects from the development of software, the provision of information systems (Software and IS) is advisable to use different technologies that will reduce the percentage of errors. There remains an open-ended question of choosing a technology for software development. One technology may be the crunching option for one task, and the other technology may be hindered or in general lead to a draft. There are many project management methodologies to choose from when starting a new project. The purpose of this article is to analyze the most well-known agile and Waterfall methodologies to determine which is most appropriate for a software project. A practical study is planned by analyzing the results of an expert survey designed to take into account the experience of developers with the methodologies mentioned above. We will focus on agile and Waterfall methods to analyze the results of the study. The purpose of the work is to improve the choice of software technology by providing a decision support system that will lead to a reduction in the risk of unsuccessful design.

To do this, it is necessary to analyze the existing software development technologies. To choose the actual platform for creation of decision support system, to investigate the issues of choice in the minds of richness. Analyze the mathematical model of decision-making and create your own decision support system, analyze the results and make recommendations for their practical application. The Pareto-optimality method and the Pareto set narrowing method based on information about the relative importance of the criteria became the method for solving this problem. The method of research is a computer program for modeling DSS.

1.2. Analysis of research

The study is based on the writings of such prominent authors in the field of software development as K. Peterson [1], D. Schor [2], L. Mark [3], K. Beck [4], B. Bohem [5]. In his research, Williams [6, 7] presents several well-known planned and rapid SDMs (System Development Model). The two agile and Waterfall models in the study are compared [8].

2. Theory and methods

2.1. Summary of background material and rationale for research findings

Most often, when it is necessary to make a decision on the methodology selection, there is too much heterogeneous information and it is difficult to understand what is best for the project. There is a scheme of methodology selection that allows us to draw attention to some of the most important aspects.

There is no universal set of conditions for all situations when choosing a particular methodology. in each case, the RM (ODA) should focus on the specifics of its project. The scheme of methodology selection merely identifies the main aspects and will recall the features of the underlying methodologies. It is impossible to treat it as the only correct guide for choosing a methodology. Moreover, it is not necessary to extend it to such complex projects that modern customers need. Choosing a methodology, you need to adapt it to your project. You can throw something away, add something from other methodologies, make something your own.

The most diverse are the classic cascade (waterfall) model and modern agile methodology. On the one hand (cascading) everything is planned down to the smallest detail, we define tight deadlines, we have a fixed budget, but problems come when we need to make some changes or change the direction of software development altogether. On the other hand (agile) we have a flexible sprint system, the customer pays for the sprints, it is easy enough to make changes, but
it is difficult to set a timetable for a product and it is difficult to calculate the budget. Figure 1 shows the coordinate plane on which the known methodologies are located. The abscissa axis is defined as from “low formalized” to “highly formalized” methodologies. The ordinate axis is defined as from “iterative” to “cascading” methodologies. As you can see from the graph, we have two diametrically opposite methodologies, xp (extreme programming) and cascading. XP is aimed at rapid development with small iterations and makes changes quite easy, and the cascade, on the contrary, requires well-defined conditions, has defined steps that do not allow you to go back.

2.2. Cascade methodology

One of the first began to apply the cascade model, where each work is performed once and in the order shown in fig. 2 [9].
It is considered that each work should be done so carefully that after its completion and proceeding to the next stage, it will not be necessary to return to the previous one. The developer verifies the intermediate result by known verification methods and fixes it as a ready reference for the next process. According to this model of work, the tasks of the development process are usually performed in sequence, as shown in the diagram. However, ancillary and organizational processes (requirements control, quality management, etc.) are generally performed in conjunction with software development processes. In this model, the return to the original process is assumed after the error correction and correction. The peculiarity of this model is the fixation of sequential processes of software development. It is based on the model of the factory, where the product goes from the stage of design to production, and then it is passed on to the customer in the form of a finished product, where replacement is not provided, although you can submit another similar product.

The disadvantages of this model are:

- the process of creating software does not always fit into such a rigid form and sequence of actions;
- do not take into account the changing needs of users, unstable environmental conditions that affect changes in software requirements during their development;
- a significant gap between the time of making the error (for example, during the design process) and the time of its detection (during maintenance), which leads to significant processing of the software.

When applying Waterfall models, the following risk factors may be observed. The requirements for the PS are not sufficiently clearly formulated or do not take into account the prospects for development of OS, environments:

- a cumbersome system that does not allow component decomposition may cause problems with its placement in memory or on platforms not provided in the requirements;
- making rapid changes to technology and requirements can impair the development of individual parts of the system or system as a whole;
- the product obtained may not be suitable for use as a result of misunderstandings by the developers of the requirements or functions of the system or insufficient testing [1].

The advantages of implementing the system with a cascade model are:

- all tasks of the subsystems and systems are implemented simultaneously, so that no task can be forgotten, which helps to establish stable links between them;
- a fully-fledged documentation system makes it easier to keep track of, test, fix bugs and make changes, not deliberately, but purposefully, starting with requirements such as adding or replacing some features and repeating the process.

The cascade model can be considered as a model of the LCD, suitable for the creation of the first version of the software in order to check the functions implemented in it. During maintenance and operation, various types of errors can be detected, correcting which will cause all processes to be repeated, starting with the refinement of requirements.

3. Agile methodology

Agile methodology is used to design an orderly management of the development process that allows for permanent changes to the project development [9, 10, 11, 12]. This methodology
is one of the conceptual frameworks for the creation of various software development projects. This model is used to minimize the risk of developing a product in short time intervals called iterations and usually lasts from one week to one month. The life cycle of the methodology is shown in fig. 3.

This model should be applied when the needs of users are constantly refined in a dynamic process. Agile changes are made at a lower cost due to constant sprints. Unlike the cascade model, in a flexible model, only a small amount of planning is enough to start a project [2, 10, 12].

The advantages of the model:
- flexible methodology has an adaptive approach that allows to change customer requirements;
- direct communication and constant feedback from customers or their representatives leave no room for uncertainty.

The disadvantages of the model:
- this methodology focuses on creating software earlier than documentation, hence there may be a lack of documentation;
- the development process can get out of hand if the customer does not clearly represent the final result of the project.

An article [8] analyzes the most well-known agile and Waterfall methodologies to determine which is most appropriate for a software project. A practical study is proposed by analyzing the results of a survey designed to take into account the experience of developers in the methodologies mentioned above. Based on the results of the study, it was concluded that the perfect solution does not exist, because many factors must be taken into account. The results of the comparison are shown in table 1.
Table 1: Comparison of agile and cascading technologies.

| Content          | Agile                                                                 | Waterfall                                                                 |
|------------------|----------------------------------------------------------------------|---------------------------------------------------------------------------|
| Creation, developers | 2001 IT Team (US)                                                  | 1956, 1961, 1970 G. Bennington, Hozier, W. Walker Royce                   |
| Principles of application | - the highest priority in meeting customer needs;   | - rigid sequence of stages of development;                                   |
|                   |  - throughout the project, the team and customer interact with each other and each other on a daily basis; | - moving to a new stage - only after the successful completion of the previous one; |
|                   |  - running a product is a major indicator of progress;          | - fixed cost of the product;                                               |
|                   |  - only a self-motivated, motivated team can trust the work;    | - the customer is not involved in the direct development process;           |
|                   |  - optimum release time for a work product - from 2 weeks to 2 months. | - changes can only be made after completion of the entire development process. |
| Advantages        | - high level of interaction between project team members;       | - clear and clear outline of the workflow;                                 |
|                   |  - quick result (working code) as a result of sprints;          | - the ability to calculate the exact amount of resources spent on the project; |
|                   |  - stimulating change and improvement of the product during its development; | - does not require the expense of establishing communication between all team members. |
|                   |  - direct involvement of the customer in the workflow.          | - the priority of a formal approach to workflow consistency;                |
| Disadvantages     | - the risk of endless product changes;                          | - the inability to make changes to the customer before the completion of product development; |
|                   |  - great dependence on skill level and team experience;        | - in case of lack of resources, the quality of the project suffers due to the shortening of the testing phase. |
|                   |  - it is almost impossible to accurately calculate the total cost of the project. | - the shortening of the testing phase. |


Continuation of Table 1

| Criteria ”Yes”                                                                 | Agile                                                                                                           | Waterfall                                                                                                                |
|--------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------|
| – an experienced, highly skilled team is working on the project;               | – Most or all of the project work is outsourced;                                                                 | – you have a clear concept for the product you want to get;                                                             |
| – you are working on a startup;                                                | – you are not limited in time and resources of product creation;                                                | – you are not limited in time and resources of product creation;                                                         |
| – you need to get a working version of the product quickly;                    | – the customer acts as a partner, not an investor;                                                                | – building a product or business is based on a strict sequence of tasks.                                                  |
| – the customer acts as a partner, not an investor;                             | – the product is being developed in an area prone to constant change.                                             |                                                                            |
| – the product is being developed in an area prone to constant change.          |                                                                            |                                                                            |
| Criteria ”No”                                                                 | – Most or all of the project work is outsourced;                                                                 | – you have a clear concept for the product you want to get;                                                             |
| – you are not ready to spend additional resources to establish daily stable   | – you are not limited in time and resources of product creation;                                                | – you are not limited in time and resources of product creation;                                                         |
| communication between all participants of the process;                         | – building a product or business is based on a strict sequence of tasks.                                         |                                                                            |
| – the product must be created by a specific deadline;                          | – you need detailed documentation on all development processes.                                                  | – financial resources are not a key constraint in your project.                                                          |
| – the project budget is strictly limited;                                      |                                                                            |                                                                            |
| – you need detailed documentation on all development processes.                |                                                                            |                                                                            |

The problems of decision-making in the recent times are increasingly attracting the attention of scientists. The choice of technology for the creation of software for securing information systems has not been left out of the way. At this moment, more and more new technologies and they are becoming more sophisticated and are the solution for a small circle of tasks. Executives (ODA) are faced with the problem of making a decision regarding the application of a separate technology for the creation of a final product. Decision-making is a complex and ambiguous dynamic process that arises when it is necessary to choose the best, in a sense, an option among many alternatives to achieve the desired or desired result.

To implement the criteria, which the experts, who were involved in the preliminary survey, drew attention to. The BCMCP (Business Continuity Management Consulting Practice) concept is to make a conscious choice from many alternatives to one. This choice is made by the ODA, which is striving to achieve its intended purpose. ODA can be either a specific individual or a group of people who make the decision at the same time. The Pareto-optimality method and the Pareto set construction method are selected as the method for solving this problem based on the relative importance of the criteria. Criteria for solving the problem have been formed. Multicriteria problems are analyzed.

The following local criteria for evaluating software development technology were identified.

1. Ability to select individual parts of programs as modules.
2. Control of correctness of work with data types.
3. Working with complex structure data.
4. Control of interfaces of program modules at separate compilation.
5. Readability of programs.
6. Programmer error protection.
7. Technology flexibility.
8. Completeness of implementation of functionality.

The choice of the Pareto set is as follows:
- all alternatives are compared with each other by all criteria;
- if by comparison of any alternatives it is found that one of them is not better than the other
  by no one criterion, then it can be excluded from consideration;
- the excluded alternative does not need to be compared to other alternatives, as it is clearly
  unpromising.

The set reduction is as follows.
1. First of all, it is necessary to establish pairs of "unequal" according to ODA criteria. For example, let’s see a pair of them, which consists of the $i$ and $j$ criterion, and in this case, according to the intuitive idea of ODA about the importance of the $i$ criterion is more important than the $j$ one.
2. It is now possible to begin to determine the specific value of the coefficient of relative importance of the $i$ criterion in comparison with the $j$. It should be borne in mind that the more this coefficient is found, the more meaningful the information will be and thus the greater the degree of constriction of the Pareto set can be expected. The degree of narrowing is calculated by the following formula:

$$\theta_{ij} = \frac{1}{\omega_i + 1}$$

3. Suppose that the above method reveals a whole set of information about the relative importance of the criteria, which consists in the fact that the $i$ criterion is more important than the $jk$ criterion with a given coefficient of relative importance $\theta_{ij} \in (0, 1)$, $k = 1, 2, ..., M$, where $M \leq \frac{m}{2}$. It is considered that none of the criteria can be more important than himself, that is, for any number $k = 1, 2, ..., mj$ equality is not fulfilled $i_k = j_k$. The following are some of the less important criteria (numbers of a specific set $j_1, j_2, ..., j_m$) by the formula

$$\hat{f} = \theta_{ij} \cdot f_i + 1 - \theta_{ij} \cdot f_j$$

and substitute them into the original vector criterion $f$ instead of the former $f_{jk}$. As a result of the substitution, a new vector criterion $\hat{f}$ is formed. Next, find the Pareto set for this new vector criterion. Thus, the Pareto set will be narrowed by using a set of mutually independent information about the relative importance of the criteria.

Certain aspects of the use of expert multicriteria ODA methods need further elaboration. In future research on the topic, I suggest focusing on:
- ensuring completeness and consistency of expert data;
- increasing the level of trust in them by decision makers.
4. RESULTS AND DISCUSSIONS

4.1. Solving the problem using the Pareto-optimality method

In this task of choice, we will choose from 4 technologies for the development of software for information systems: \( y^{(1)} \) – cascade technology, \( y^{(2)} \) – coiled, \( y^{(3)} \) – V-shaped, \( y^{(4)} \) – it packing technology.

To solve this problem, 20 experts were recruited, who gave their expert evaluation for each technology with eight criteria: \( f_1 \) – modular principle, \( f_2 \) – control of correctness of work with data types, \( f_3 \) – work with data of complex structure, \( f_4 \) – control of interface of modules separate compilation, \( f_5 \) – programmability, \( f_6 \) – programmer error protection, \( f_7 \) – technology flexibility, \( f_8 \) – functionality implementation. The experts rated their score with 10 points in 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 points.

Number of criteria \( m = 8 \). Let us denote the set of five possible vectors (estimates) of the respective technologies for the development of information system software through \( Y = y^{(1)}, y^{(2)}, y^{(3)}, y^{(4)} \). To average the bill count for all criteria, we use the formula:

\[
a_{ij} = \frac{1}{n} \sum_{i=0}^{n} a_{ij}
\]

where \( a_{ij} \) - expert rating, \( n \) - number of experts.

In accordance with the algorithm described above, we assume that \( P(Y) = Y \) and start comparing vectors with others. We compare vector \( y^{(1)} \) with vectors: \( y^{(2)}, y^{(3)}, y^{(4)} \):

- pair \( y^{(1)} \) and \( y^{(2)} \) cannot be compared in relation \( \geq \);
- pair \( y^{(1)} \) and \( y^{(3)} \) cannot be compared in relation \( \geq \);
- pair \( y^{(1)} \) and \( y^{(4)} \) cannot be compared in relation \( \geq \).

Let’s compare the vector \( y^{(2)} \) with vectors \( y^{(3)} \) and \( y^{(4)} \):

- pair \( y^{(2)} \) and \( y^{(3)} \) cannot be compared in relation \( \geq \);
- pair \( y^{(2)} \) and \( y^{(4)} \) cannot be compared in relation \( \geq \).

Let’s compare the vector \( y^{(3)} \) with vectors \( y^{(4)} \): \( y^{(3)} \) and \( y^{(4)} \) cannot be compared in relation \( \geq \).

So \( p(y) = y^{(1)}, y^{(2)}, y^{(3)}, y^{(4)} \). Narrowing has not taken place of all four technologies and a choice should be made. Suppose the OPR has made such a decision that the criterion \( f_8 \) – completeness of implementation of functionality is more important than \( f_3 \) – work with complex structure data on the second. OPR gave the following parameters: \( w_8 = 4 \), \( w_3 = 1 \). Let calculate the coefficients of relative value, by the (1):

\[
\theta_{83} = \frac{1}{\frac{4}{1} + 1} = \frac{1}{5} = 0.2
\]
We calculate $\hat{f}_3$ the new coefficients by the (2) (see Table 3):

- $y^{(1)}$: $f_3 = 3.5 \cdot 0.2 + (1 - 0.2) \cdot 5.0$
- $y^{(2)}$: $f_3 = 5.0 \cdot 0.2 + (1 - 0.2) \cdot 4.5$
- $y^{(3)}$: $f_3 = 3.5 \cdot 0.2 + (1 - 0.2) \cdot 5.5$
- $y^{(4)}$: $f_3 = 5.3 \cdot 0.2 + (1 - 0.2) \cdot 5.4$

As we can see, all vectors remain Pareto-optimal. The next decision was the criterion $f_4$ – control of software module interfaces is more important than separate compilation.$f_1$ is the modular principle on the fourth. OPR gave the following parameters: $w_8 = 6$, $w_3 = 1$. Let calculate the coefficients of relative value, according to the (1):

$$\theta_{41} = \frac{1}{\frac{6}{1} + 1} = \frac{1}{7} = 0.14$$

The results of calculation of coefficients $\hat{f}_1$ of relative value, according to (2) are in Table 4).

- $y^{(1)}$: $f_3 = 4.5 \cdot 0.14 + (1 - 0.14) \cdot 5.5$
- $y^{(2)}$: $f_3 = 6.3 \cdot 0.14 + (1 - 0.14) \cdot 4.5$
- $y^{(3)}$: $f_3 = 5.0 \cdot 0.14 + (1 - 0.14) \cdot 7.3$
- $y^{(4)}$: $f_3 = 7.2 \cdot 0.14 + (1 - 0.14) \cdot 7.1$

In vector $y^{(3)}$ the Pareto optimality is out. That is, we exclude this vector.

The OPR then decided upon the criterion $f_2$ – controlling the correctness of working with data types on the fifth is more important than the criterion $f_5$ – readability of programs. OPR
gave the following parameters: \( w_2 = 3, w_5 = 1 \). Then we calculate the coefficients of relative value, according to the formula (1):

\[
\theta_{25} = \frac{1}{\frac{3}{4} + 1} = \frac{1}{\frac{7}{4}} = 0.25
\]

\[
y(1) : f_3 = 6.3 \cdot 0.25 + (1 - 0.25) \cdot 6.3
\]
\[
y(2) : f_3 = 7.8 \cdot 0.25 + (1 - 0.25) \cdot 5.4
\]
\[
y(4) : f_3 = 9.8 \cdot 0.25 + (1 - 0.25) \cdot 5.3
\]

The results of calculate new coefficients \( \hat{f}_5 \) are in Table 5.

| \( f_1 \) | \( f_2 \) | \( f_3 \) | \( f_4 \) | \( f_5 \) | \( f_6 \) | \( f_7 \) | \( f_8 \) |
|---|---|---|---|---|---|---|---|
| \( y^{(1)} \) | 5.36 | 6.3 | 4.7 | 4.5 | 6.3 | 6.4 | 4.1 | 3.5 |
| \( y^{(2)} \) | 4.75 | 7.8 | 4.84 | 6.3 | 5.4 | 6.0 | 9.0 | 5.0 |
| \( y^{(4)} \) | 7.17 | 9.8 | 5.38 | 7.2 | 5.3 | 6.5 | 9.4 | 5.3 |

Comparing vectors, we can see that the vectors in (1) and (2) are no longer Pareto-optimal. That is, we exclude these vector. That is, as a result of the narrowing of the Pareto sets on the basis of information about the relative importance of the criteria, we conclude that in this case the technology in \( y^{(4)} \), that is, iterative technology, is the most optimal one.

5. Conclusions

The growing demand for software is creating the need for the right deployment of processes across development teams so that they can meet the needs of the business and create a quality, supported and functional product. Existing development methodologies have been created and tested over time, but their shortcomings are sometimes critical to their users, which opens up space for improvement and new approaches to software development.

Two SRWPs focused on the Waterfall model and Agile methodologies were presented. Both models have their advantages and disadvantages. Small projects are always suitable for the agile approach and almost never for the Waterfall approach. A large and complex project with several teams working simultaneously on different parts of the program is almost always a Waterfall project. This ambiguity needs to change procedures in the approach to choosing software development technology. We decide on the choice of technology for software development by analyzing the opinions of development experts and bringing them to Pareto optimization. The BCPPR’s PR concept is to make a conscious choice from many alternatives to one. This choice is made by the ODA, which is striving to achieve its intended purpose. ODA can be either a specific individual or a group of people who make the decision at the same time.

As a result, it is proposed to use the Pareto method of optimality to narrow the criteria by using a set of mutually independent information on the relative importance of the criteria. Certain aspects of the use of expert multicriteria ODA methods need further elaboration. In future research on the subject, we propose to focus on ensuring the completeness and consistency of expert data and increasing the level of trust in it by decision makers (ODA).
6. **Conflict of interest statement**

None declared.

**References**

[1] K. Petersen, C. Wohlin, D. Baca, The waterfall model in large-scale development, in: F. Bomarius, M. Oivo, P. Jaring, P. Abrahamsson (Eds.), Product-Focused Software Process Improvement. PROFES 2009. Lecture Notes in Business Information Processing, Vol. 32, Springer, Berlin, Heidelberg, 2009, pp. 386–400. doi: 10.1007/978-3-642-02152-7_29.

[2] J. Shore, S. Warden, The Art of Agile Development, 1st Edition, O’Reilly Media, Inc., Sebastopol, CA, 2007, p. 438 [cited 13 Mar 2020]. url: https://poetiosity.files.wordpress.com/2011/04/art_of_agile_development.pdf.

[3] M. L. Layton, Agile Project Management For Dummies, 1st Edition, For Dummies, Hoboken, NJ, 2012, p. 360.

[4] K. Beck, C. Andres, Extreme Programming Explained: Embrace Change, 2nd Edition, Addison-Wesley Professional, Boston, MA, 2004, p. 216.

[5] B. Boehm, J. A. Lane, S. Koolmanojwong, R. Turner, The Incremental Commitment Spiral Model: Principles and Practices for Successful Systems and Software, 1st Edition, Addison-Wesley Professional, Boston, MA, 2014, p. 332.

[6] L. Williams, survey of plan-driven development methodologies, 2004 [Internet], Ira A Fulton Schools of Engineering. Arizona State University, Mesa, Arizona, 2004 [cited 13 Mar 2020]. url: http://pooh.poly.asu.edu/Ser515/Schedule/docs/PlanDrivenSurvey-PSP-TSP-RUP-Williams2004.pdf.

[7] L. Williams, survey of agile development methodologies, 2007 [Internet], semanticscholar.org, 2007 [cited 13 Mar 2020]. url: https://pdfs.semanticscholar.org/29f6/01214845d84fe495593fb093863da5a36c.pdf.

[8] B.-l. ndrei, n.-C. asu Pop, S.-C. Gheorghe, o.-n. Boiangiu, study on using waterfall and agile methods in software project management, Journal of information Systems and Operations Management 13 (1) 2019. 125–135 [cited 13 Mar 2020]. url: http://jisom.rau.ro/Vol.13%20No.1%20-%202019/JISOM-SU19-A12.pdf.

[9] M. Treskin, Rabbitmq: introduction to amqp, July 2009 [Internet], (in Russian), habr.com, Moscow, Russia, 2009 [cited 13 Mar 2020]. url: https://habr.com/ru/post/64192/.

[10] M. Cohn, Succeeding with Agile: Software Development Using Scrum, 1st Edition, Addison-Wesley Professional, Boston, MA, 2009, p. 504.

[11] Google, Angularjs, 2018 [Internet], google.com, 2018 [cited 13 Mar 2020]. url: https://angularjs.org/.

[12] D. Cohen, M. Lindvall, P. Costa, Agile software development. a dacs state-of-the-art report, Jan 2003 [Internet], Fraunhofer Center for Experimental Software Engineering Maryland, Maryland, 2003 [cited 13 Mar 2020]. url: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.94.7&type=pdf.