

DECISION SUPPORT TECHNOLOGY FOR SPRINT PLANNING

Melnyk K. V. – PhD, Associate Professor of the Department of Software Engineering and Management Information Technologies, National Technical University “Kharkiv Polytechnic Institute”, Kharkiv, Ukraine.

Hlushko V. N. – Senior Consultant, Solution Architect, GlobalLogic, Kharkiv, Ukraine.

Borysova N. V. – PhD, Associate Professor of the Department of Intelligent Computer Systems, National Technical University “Kharkiv Polytechnic Institute”, Kharkiv, Ukraine.

ABSTRACT

Context. The article describes the relevant planning process of software projects, planning problems and different solutions to these problems basis on use of the Scrum methodology.

Objective. The purpose of the work is to develop the technology for solving the sprint planning task in the face of uncertainty and possible risks from software development standpoint.

Method. The most used software life cycle models are described. The choice of the Scrum as a widely used representative of agile methodology for software development is justified. An analytical review of the methods for estimation of the complexity of user stories is carried out. The major problems of sprint planning are highlighted. The model of the business process to implement an IT-project by Scrum in the form of an BPMN-diagram has been developed. The algorithm to solve the problem of Sprint Backlog planning with uncertainty has been elaborated. The common process of user stories selection from Product Backlog to Sprint Backlog and ways of solving the possible problems are considered. The task of estimation of labor intensity of user stories and the task of risk evaluation in planning are formalized. The technology of user story selection for Sprint Backlog has been developed. Numerical studies of the decision support technology proposed in the article are carried out. It allows suggesting it as the practical tool during sprint planning. The method of adequacy evaluation of proposed technology is offered. The set of key performance indicators for assessing the team performance is selected.

Results. The sprint planning technology was developed, which project managers, product owners and development teams for increasing the effectiveness of decision-making process can use.

Conclusions. The conducted experiments have confirmed the importance of the proposed decision support technology and allow recommending it for use in practice for planning of software projects. Scientific novelty is to improve the sprint planning process with the assistance of the proposed technology, which alleviates uncertainty while defining labor intensity of user stories and decreases time spent on decision making.

KEYWORDS: sprint backlog, planning, uncertainty, labor intensity, user story, user story, selection task, team performance.

ABBREVIATIONS

IS is an Information System;
IT is an Information Technologies;
KPI is a Key Performance Indicators;
LI is a Labor Intensity;
RMS is a Residual Mean Square;
SP is a Story Points;
US is a User Story.

NOMENCLATURE

I is a set of user stories in a sprint;
 $|I|$ is a strength of the set I ;
 $A^{n \times n}$ is a judgment matrix;
 a_{ij} is a paired comparison of i -th US and j -th US
($i, j \in I$);
 $s_i, i \in I$ is a labor intensity of i -th US;
 g_i is an average relative size of labor intensity of i -th US;
 D is a sample variance;
 σ_i is the estimation of RMS of i -th US;
 CI_i is a confident interval of LI of i -th US;
 V is a team velocity;
 p_i is a priority of i -th US;

x_i is a selection indicator of i -th US for implementation in the sprint;
 TP is a team performance;
 w_q is a weight coefficient of q -th key performance indicator;
 u_q is an utility function;
 KPI_q is a value of q -th key performance indicator;
 KPI_q^{worst} is a worst value from set of values of q -th key performance indicator;
 KPI_q^{best} is a best value from set of values of q -th key performance indicator.

INTRODUCTION

The attractiveness of Ukraine in the world market of software development services for foreign companies is constantly growing. The IT share in GDP of Ukraine is 4% by the start of 2019 and it is raising [1]. The number of IT companies, complexity of IT projects, requirements for quality and skills of specialists are increasing. As a result, the software development process is making more and more complex. On the one hand, this process is characterized by the complexity of coordination of IT-professionals, where each member of the team has different experience and qualifications. On the other hand, it is necessary to take into account a large number of require-

ments from users of future programs, which are sometimes controversial. There are many models of software development. The most commonly used software development lifecycle models to date are [2]: Waterfall model, V-model, Incremental iteration model, Prototype model, Spiral model.

All aforementioned models have their advantages and disadvantages for software development. However, Agile methodologies for software development have been getting more and more popularity today. The most utilized among Agile methodologies is the Scrum methodology [3, 4]. It is a flexible development cycle model that allows developers to take advantage of existing coding practices and enables the client to make changes to requirements at any time limiting the negative impact on development teams during the sprint course. Its main feature is involvement of all participants into the process: both client and performer. The use of Scrum allows you to detect and eliminate deviations from the desired result in the earlier stages of software development.

Software development with Scrum consists of small iterations, or sprints, which are essentially small projects. Sprint duration is a fixed time period of 1–4 weeks. It has the same length until the end of the project. When sprint is over, a new working version of the product should be received. The following actions are analysis and refocusing on the new tasks of the next cycle. The effectiveness of sprint and an IT-project, in general, is directly dependent on the planning process, so solving of the sprint planning task is the very important and actual problem nowadays.

The object of study is the planning process of sprint.

The subject of the study is the theoretical and methodological tool for assessing and selection the set of tasks for sprint.

The purpose of this work is to develop a decision support technology for solving the sprint planning task.

1 PROBLEM STATEMENT

The sprint planning activity is a selection of a set of user stories or tasks which development team commits to solve within a single sprint, assessing their complexity and efforts with evaluation of possible risks that may occur while developing software during the sprint. In its turn, it means that the development team literally finds the optimal solution of the planning problem in the face of uncertainty.

The mathematical formulation of the planning task can be presented in the following way:

- to estimate labor intensity of every US $s_i, i \in I$ from the proposed set of user stories;
- to choose subset of user stories $\{US_i\}$ from the proposed set I for next sprint according to labor intensity of every US $s_i, i \in I$ and the team velocity V .

2 REVIEW OF THE LITERATURE

In order to plan a sprint, it is necessary to evaluate the complexity and labor content of user stories. There are

most commonly used methods for evaluating the complexity of a story [5, 6]: T-Shirt Sizes method, Planning Poker method, Dot-voting method, Ordering Rule method. Almost all of these methods are based on heuristic approaches. The aforementioned techniques do not need much time, they are quite accurate for comparison of the work efforts of one US to another, and as such they are used on many IT-projects. Regardless on sufficient number of methods for solving the sprint planning problem, they still remain open some issues that development teams face.

To reduce the subjectivity of the judgments of stakeholders, it is necessary to use formal methods based on different mathematical models. There are many scientific articles showed the way of calculating efforts or labor intensity of user stories. The journal publications [7, 8] proposes using of Bayesian network for effort calculation; the reports at scientific conferences [9, 10] demonstrate unusual implementation of Bloom's Taxonomy for computation of complexity of user stories; the works [11, 12] reveal how to construct and use fuzzy logic framework for complexity calculation. These scientific works show the applicability of the proposed methods for assessing the complexity of user stories, but they require a very long period of time for preparation. For each IT-project, the team has to build own model or own framework, and this requires additional financial and time resources, as well as the availability of experienced specialists in the mathematical field. Such a disadvantage can lead to increase project times, which is unacceptable to Product Owner.

All of the aforementioned methods and approaches have a disadvantage: there are no recommendations for planning tasks when the complexity of the estimated user stories exceeds the average sprint velocity for the team. Scrum methodology [3, 4] proposes to decompose large user stories in such cases, and then choose a set of stories, the complexity of which corresponds to the velocity of the sprint. The problem there is that the task of optimal user story selection is task in the face of uncertainty, because the likelihood of different choices is unknown most of the time. In such cases, the team and the Product Owner are guided only by benefits they get in return, with no risk assessment recommendations in every case.

3 MATERIALS AND METHODS

In general, the model of the business process of software development using the Scrum methodology can be represented in the following form (Fig. 1). The Business Process Model and Notation Specification [13] was chosen to model the diagram.

Before the start of software development, Product Owner and development team conducts the first meeting. It is called the Kickoff Meeting, which provides the opportunity for Product Owner to explain vision and scope of the project. The following meeting is the Product Backlog Grooming Meeting, which is devoted to creation of Product Backlog – the main document of the project. Product Backlog can be considered as the software re-

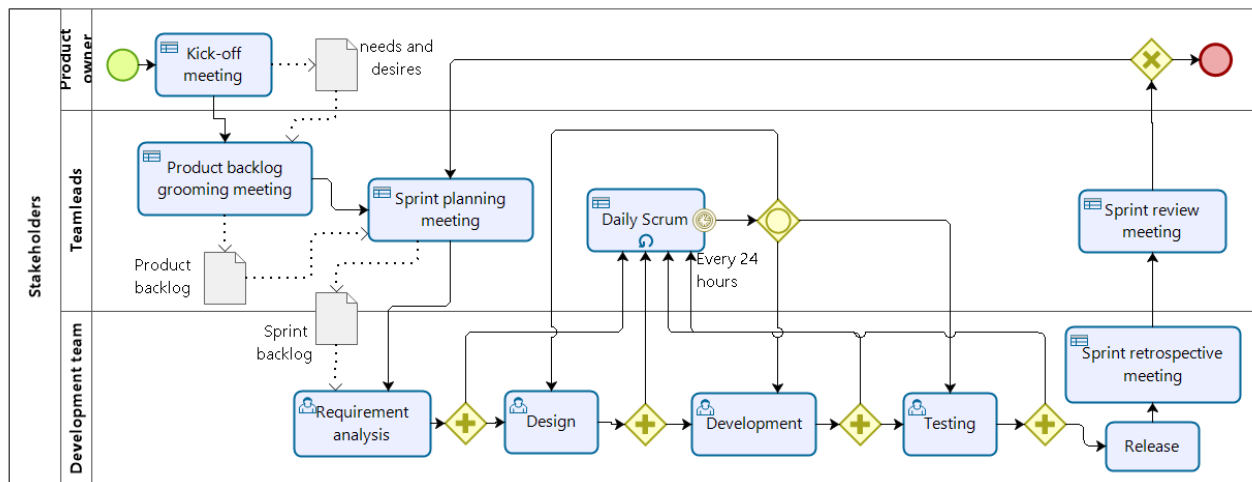


Figure 1 – Scrum model of software development

quirement specification, which consists of epics and user stories. Sprint Planning Meeting is carried out before starting of the sprint. The main purpose is to prioritize and evaluate the content of Product Backlog and form the Sprint Backlog. It means that set of user stories from Product Backlog are selected for the next sprint. The biggest problem at this stage is to correctly evaluate the complexity of each story as well as to assess related uncertainty if any. Every day, a Daily Scrum is conducted to determine the status and progress of work during the sprint, identify early obstacles, and make decisions to change the strategy needed to meet sprint's goals.

The Sprint Retrospective meeting is undertaken on the last day of the sprint. It goes as following:

- team members answer two questions: “what has been done well in the past sprint?” and “what needs to be improved in the next one?”;
- highlights improvements of the development process;
- evaluates the efficiency of the team in the past sprint and predicts the expected performance in the next sprint;
- identifies existing problems, proposes possible solutions and assigns team members responsible for them;
- makes estimates of the probability of completion of all necessary work on the product.

Sprint Review Meeting is conducted at the end of the sprint. It may be used by the team to demonstrate the version of the product to all interested stakeholders.

Thus, solving the Sprint planning task allows to increase the effectiveness of decision-making process by project managers. So, the main purpose of this study is to develop decision support technology for sprint planning.

Referring to [3, 4], the general model of solving the sprint planning task can be represented in the following form (Fig. 2).

The first step of planning is to prioritize each user story in Product Backlog. Product Owner handles this activity. Due to the prior prioritization, all user stories are sorted by importance to the business. Typically, Sprint Backlog creation is the selection of user stories with the highest priority from Product Backlog, unless otherwise discussed with the client.

According to the algorithm above, the next step is to solve two problems:

- 1) estimation of labor intensity of each user story from Sprint Backlog;
- 2) evaluation of uncertainty and possible risks from software development standpoint.

The determined estimations of Sprint Backlog should be compared with the team velocity. Velocity is a measure of the amount of work the team can do during a single sprint. Depending on the results, there are three options:

- 1) The labor content of all Sprint Backlog stories is roughly equal to the team's predicted speed and effort. In this case, the Sprint Backlog is not changed, and the team works as usual.
- 2) If the estimated labor content of Sprint Backlog less than the velocity of the team, then Sprint Backlog is filled with the next user story from the ordered Product Backlog list.
- 3) A difficult situation arises when the number and complexity of user stories in Sprint Backlog are much greater than the speed of the team. Therefore, the task selection of the highest priority user stories arises, but their number must be as high as possible. Otherwise, Product Owner changes the priorities of user stories or decomposes some of them, and the stories are re-evaluated.

Solving of the problems from above results in creation of Sprint Backlog. After the sprint, the sprint results are analyzed, and the velocity is modified. Its updated value is used for further calculations in the next sprint. Velocity is calculated by totaling the points for all fully completed user stories.

Let's consider the tasks presented above in details.

The task of evaluating uncertainty is to predict implementation of a sprint in the context of incomplete information. There is a risk of failing of team commitments when there is not enough input data for sprint planning. Therefore, one of the important tasks in sprint planning is to assess the uncertainty to mitigate any possible risks that may arise in the following cases:

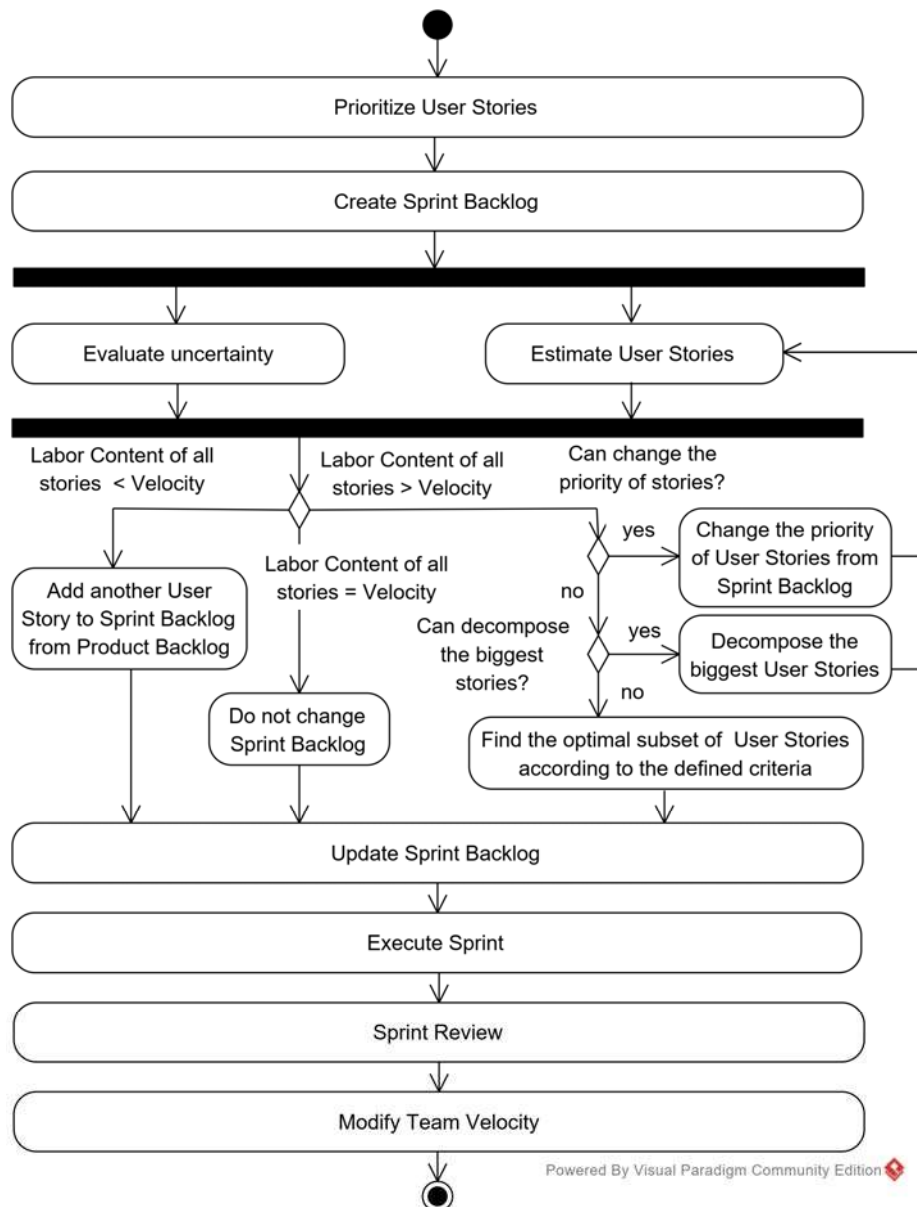


Figure 2 –The algorithm of solving the sprint planning task

1) Changing the team composition:

- temporary occupation of a team member on project tasks unrelated to development, for example, demonstration to a client of a product demo, which may happen even in another country;
- missing employees due to improving their skills or supporting the IT company by attending an IT conference, conducting an online training, involvement into pre-sales activities for new clients, etc.;
- temporary absence of a team member due to the urgent need to use his expertise on other projects;
- temporary absence of a team member due to illness;
- lowering of team member performance due to objective or subjective reasons.

2) Changing of the Product Backlog before the start of the sprint:

- some user stories can be added, deleted or modified on demand of the client, it leads to changing of the labor

intensity of these user stories, labor intensity of the sprint and the whole project;

- changing of project scope;
- changing user story priorities in Product Backlog can cause Sprint Backlog failure;
- incorrect estimation of complexity and labor content of the user stories in Sprint Backlog;
- misinterpretation of the client’s wishes, in other words, changing of the content of some user stories.

In each case, the Product Owner should decide how to assess any uncertainty and mitigate risks. Uncertainty can be taken into account in two ways.

The first way is to reduce velocity of the next sprint, in case if probability of complications is big enough.

The second way of evaluating of sprint risks is to raise the labor intensity of the user story:

$$LI \text{ of } US = f(\text{complexity}, \text{risk}, \text{efforts}).$$

Then the final estimation of the user story should be slightly increased by a couple of units used by the team for the evaluation. However, the adjusted estimates should be used with care in the decision making process as of their expert judgment nature.

We can use different, more formal way of uncertainty evaluation – calculation of interval estimation instead of point estimation:

$$\text{Confident interval} = [LI \text{ of } US \pm \text{uncertainty}]. \quad (1)$$

As a measure of uncertainty, it is suggested to use the residual mean square for estimates of the confidence interval. The RMS shows the average deviation of user story estimation from the mathematical expectation of a set of user stories in Sprint Backlog. Therefore, when Sprint Backlog is being evaluated with a high likelihood of time failure, it is highly recommended to use the pessimistic values of the labor intensity of the user story. It means the usage of the maximum value from the confidence interval.

The RMS can also be used to analyze the estimations: the large value of RMS characterizes the imbalance of the user story estimations. This means that the labor intensity of some user stories is very high. According to the Scrum methodology [3, 4, 14], implementation of user story should not exceed one working day or 12 hours. So, the user story should be divided into separate tasks and re-evaluated.

Consider the task of estimating the labor intensity of the user story.

User stories in Sprint Backlog are evaluated in units used by the team: man-hours or story points [5, 6, 14]. Although the Scrum methodology recommends story points as an abstract metric for assessing the labor content of user story, some IT companies use time as a unit of story complexity. In the latter case, there are a certain number of hours to complete the story. A more qualified developer can complete the stories in a part of the allotted time, and then begin to perform the next task or switch to tasks that are not directly related to the sprint goals. Moreover, a developer with no experience can spend extra time for solving a specific problem. Thus, estimations of the user story are not objective in this case. This disadvantage can be minimized by using a different rating scale based on comparisons of labor content to implement user story. It is suggested to use Story Points as a unit of measurement.

1 SP is the unit of labor intensity of the story or the effort of the whole team to implement the simplest requirement or user story.

The number of Story Points to develop the same functionality differs from team to team, but this does not mean that time costs will be different, as each team means its value for 1 SP. Assessing stories in SP makes sense only within the same project and the same development team, because the labor content of the tasks is compared with each other.

Thus, in order to evaluate the complexity of the user story, it is necessary to make subjective paired comparisons on the selected scale. Paired Comparison Method is © Melnyk K. V., Hlushko V. N., Borysova N. V., 2020
 DOI 10.15588/1607-3274-2020-1-14

one of the appropriate decision making tools because of its simplicity and effectiveness. It lets to describe values and compares them to each other. In [15] the scale of comparison for subjectively paired comparisons was proposed:

- equal importance – 1;
- moderate importance – 3;
- strong importance – 5;
- very strong importance – 7;
- extreme importance – 9;
- these marks for intermediate cases – 2, 4, 6, 8.

The effectiveness of this scale has been proven by comparison with many other scales in many applications [16].

The process of pairwise comparison is conducted as following. All user stories need to be compared with each other. The obtained estimates are entered into the judgment matrix. When comparing an element with itself, the ratio equals 1. If the first user story is more important than the second, then an integer from the scale is used, otherwise the inverse value is used. The lower off-diagonal elements are determined by the upper off-diagonal elements. The number of different paired comparisons in a rank-ordering of N objects is $N(N-1)/2$. Then the judgment matrix is used to calculate the estimation of user stories. Most scientific papers uses eigenvector to calculate values of user stories [15]. However, Crawford and William in work [16] showed that the geometric mean vector is computationally easier than eigenvector and statistically preferable to the eigenvector.

Let's formalize the process of user story evaluation based on the mathematical apparatus proposed in [15] and [16].

Let us denote I as the set of user stories in the current sprint, the complexity of which should be evaluated, whereas $|I| = n$ is the strength of the set of user stories. Then a_{ij} is the result of a paired comparison of the i -th and j -th ($i, j \in I$) user stories, which is written to the judgment matrix $A^{n \times n}$, where $a_{ij} = \frac{1}{a_{ji}}$ and $a_{ii} = 1$.

The average relative size g_i of LI of i -th US is calculated as geometric mean of judgments

$$g_i = \left(\prod_{j=1}^n a_{ij} \right)^{\frac{1}{n}}, i \in I. \quad (2)$$

Let us assume $k, k \in I$ is the number of the US chosen as the standard. There is recommendation for the first sprint to choose a standard user story, which labor content is known from team past experience on similar projects, or such story that has the minimal labor content. Standard US for the following sprints is the one which complexity in SP can be found in the easiest way based on the experience of the previous sprints.

Suppose $s_k, k \in I$ is a known value of LI in story points of standard US. According to aforementioned des-

ignations the labor intensity of i -th US can be calculated by the following formula [17]:

$$s_i = \frac{s_k g_i}{g_k}, i \neq k, i \in I, k \in I, \quad (3)$$

where $g_k, k \in I$ was calculated by the formula (2).

It is necessary to use formula (1) to determine uncertainty of the complexity of user stories, where uncertainty is determined by residual mean square. Analyzing the input data, it is clear that it is impossible to find the RMS, but only RMS estimation by means of the sample variance. The following formula was proposed in science paper [13] for finding the sample variance D_A of the judgment matrix $A^{n \times n}$ for creation of the sprint backlog:

$$D_A = \frac{1}{\frac{n(n-1)}{2} - (n-1)} \sum_{i < j}^n \left(\ln a_{ij} - \ln \frac{g_i}{g_j} \right)^2. \quad (4)$$

Sprint backlog is formed in such a way that there is no idleness of the team. So, the implementation of one user story is a process independent from implementation of another user story. Assume that each story contributes equally to the overall variance. In this case, the sample variance of the sprint backlog is the sum $D[s_i]$ of the sample variances of the every user story

$$D_A = \sum_{i=1}^n D[s_i] = n D[s_i].$$

The estimation of RMS of i -th US is calculated in following way:

$$\sigma_i = \sqrt{D[s_i]} = \sqrt{\frac{D_A}{n}}. \quad (5)$$

Then confident interval CI_i can be found such as:

$$CI_i = [s_i \pm s_i \sigma_i]. \quad (6)$$

Thus, the algorithm of finding estimates of the labor intensity of user stories has been presented.

Consider the **task of selection of user story** from Sprint Backlog in the case when it is not necessary to change the priorities of a user story.

In general, the scale of priority evaluation may differ from one IT project to another. Product Owner chooses a way to evaluate the user stories herself.

Let's x_i is the variable that indicate whether or not the i -th user story is selected for implementation in the current backlog sprint: $x_i = 1$ when i -th user story is selected and $x_i = 0$ – otherwise. So, the variable can only take two values

$$x_i = \{0, 1\}, i \in I. \quad (7)$$

Sprint backlog has to include the highest number of top priority user stories:

$$\sum_{i=1}^n p_i x_i \rightarrow \max. \quad (8)$$

The labor content of sprint backlog should not exceed the team velocity

$$\sum_{i=1}^n s_i x_i \leq V. \quad (9)$$

Based on the aforementioned objective functions and constraints, the model of selection of user stories for the sprint backlog can be defined in the following way: find the set of user stories satisfying the objective function (8) and constraint (9) under condition (7). This task belongs to the class of integer programming problems with Boolean variables.

4 EXPERIMENTS

Let's consider usage of the decision support technology for sprint planning by example. There is IT-project for creating IS. It is known, that team have chosen Scrum as model of software development lifecycle. The team conducted several sprints, so velocity now is equal to $V = 60 SP$. The product owner for the current sprint have proposed five user stories. He evaluated priority of each user story on the 5-point scale: $p_1 = 4; p_2 = 5; p_3 = 3; p_4 = 5; p_5 = 3$. The team has compared US to each other. The judgment matrix of pairwise comparisons is presented in Table 1.

Table 1 – The judgment matrix

№ US	1	2	3	4	5
1	1,00	3,00	3,00	0,50	2,00
2	0,33	1,00	0,50	0,25	0,50
3	0,33	2,00	1,00	0,33	0,50
4	2,00	4,00	3,00	1,00	2,00
5	0,50	2,00	2,00	0,50	1,00

In order to obtain the numerical values of the labor intensity s_i of the formula (3), the geometric mean of the labor intensity g_i is calculated by using the formula (2). If this is the first sprint, and there is no information about the complexity of the user stories, then it is recommended to take as the standard a story with a minimum geometric mean value, and to take its complexity as 1 SP. Nevertheless, by the statement of the task, it is known that $s_2 = 10 SP$, therefore, it acts as a standard for this sprint. To determine the risks the sample variance and estimation of RMS for each user story can be found by formulas (4) and (5) accordingly.

To find the interval estimations, which maximum value the product owner can use as a pessimistic estimations of user stories, the limit values have been calculated by formula (6) basing on obtained results.

According to labor complexity of each US and its priority, it is necessary to use the model of selection of user stories (7)–(9). It allows choosing the set of US from proposed sequence of US for current sprint, if summary labor content of proposed sequence of user stories is much greater than the team velocity.

5 RESULTS

The results of the calculations of the labor intensity of user stories from proposed IT-project for the current sprint are presented in Table 2.

Table 2 – Complexity of user stories

№ US	g_i	s_i	D_A	σ_i	$s_i\sigma_i$	CI_i	
						min	max
1	1,55	33	0,08	0,13	4	29	37
2	0,46	10			1	9	11
3	0,64	13			2	11	15
4	2,17	47			6	41	53
5	1,00	21			3	18	24

The value of estimation of RMS σ_i is low, it means that obtained estimations are well-balanced.

The graphical representation of the confidence intervals of the estimates of user stories is shown on the Fig. 3 based on the obtained results from the Table 2. It can be seen, that for the fourth user story has the largest difference between minimum and maximum estimations. Product owner may use this information to analyze the user stories, for example, to decompose 4th US into sub-stories, thereby reducing uncertainty.

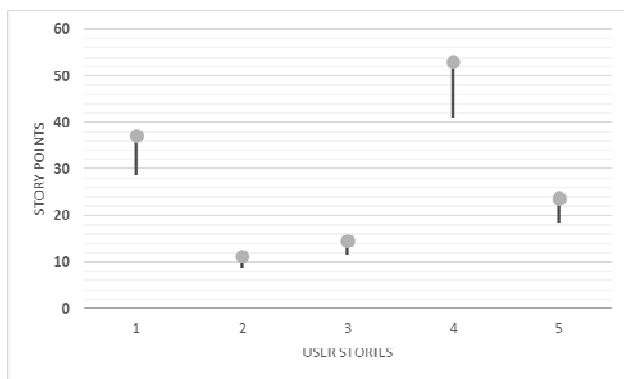


Figure 3 – Interval estimations of user stories

During the previous several sprints, the team showed an average speed of work equal to $V = 60SP$. Knowing the complexity of a sprint backlog, one can see that $\sum_{i=1}^5 s_i = 124SP \geq V = 60SP$. Therefore, it is necessary to change the story priorities and reduce the sprint backlog, or to select user stories for the sprint with current priorities. In this particular case, it is not possible to change the priorities according to the conditions of the example, so it is necessary to solve the task of selection of user stories.

Based on priorities and calculated labor intensity of each user story, the objective function (8) can be defined as follows:

$$4x_1 + 5x_2 + 3x_3 + 5x_4 + 3x_5 \rightarrow \max .$$

The constraint (9) has the following form:

$$33x_1 + 10x_2 + 13x_3 + 47x_4 + 21x_5 \leq 60 .$$

Taking into account condition (7) the solution of the given problem by simplex method is as following:

$$x_1 = 1; x_2 = 1; x_3 = 1; x_4 = 0; x_5 = 0 .$$

Basing on the calculated results, the user story #4 with high priority has not been selected as the candidate for the sprint backlog. This is because its labor intensity is almost the same as the velocity of the team. In this case, the Product Owner can:

- divide this story into individual cases;
- refine the functionality of this story;
- change priority.

The results for this story are consistent with the recommendations obtained in the evaluation of the complexity of the sprint backlog.

Thus, technology for supporting manager decisions in sprint planning has been considered.

6 DISCUSSION

To assess the adequacy of the developed decision support technology in sprint planning, it is necessary to compare the results of work of the team before applying the technology and after. One of the ways to verify the effectiveness of using the proposed technology in an IT company is to evaluate work efficiency using Performance Management [18]. Performance management is the process of calculating and improving team performance to achieve the goals of the IT-company. To assess team performance, it is necessary to identify set of key performance indicators. We can use the following KPI:

- KPI_1 is a meeting time per sprint;
- KPI_2 is a percentage of missed tasks;
- KPI_3 is the team velocity;
- KPI_4 is a customer satisfaction;
- KPI_5 is a team satisfaction.

To evaluate the team performance TP , the article [18] proposes using of universal mathematical model from [19] as a convolution criterion for KPI:

$$TP = \sqrt{\sum_{q=1}^5 (w_q u_q)^2} . \quad (10)$$

The utility function u_q can be found in the following way:

$$u_q = \frac{KPI_q - KPI_q^{worst}}{KPI_q^{best} - KPI_q^{worst}} .$$

Let's consider limit values for proposed KPI. If the team spends more than 2/3 working time on different project meetings, there is the chance the project will be behind the schedule. One of the possible reason is the "analysis paralysis", which means the inability to develop or decide due to overthinking available alternatives and possible outcomes [20]. Another possible reason is "scope creep", it refers to changes, continuous or uncontrolled growth in a project's scope, at any point after the project begins [21]. In the common case, the number of working hours in each sprint is equal to 160 hours, so

$KPI_1^{worst} \geq 100$ hours. If the $KPI_2^{worst} \geq 60\%$, the project will be behind the schedule as well. The value of team velocity depends on team, project, Product Owner, so best and worst values can be found from set of previous values. The KPI_4 and the KPI_5 should be evaluated by 10-point scale, where 1 is the minimal value for metrics and 10 is the maximum accordingly, so KPI_4^{worst} and KPI_5^{worst} are equal to 1.

To evaluate the team performance, it was considered three sprints: first sprint was conducted without proposed technology, the team worked as usual; during the second and third sprints, the team used decision support technology for sprint planning. The results of the evaluation of the team performance in several sprints calculated by using formula (10) are shown in the Table 4.

Table 4 – Team performance

Key performance indicator	Number of sprint	Value of KPI_q	u_q	w_q	TP
KPI_1	I	28	0,92	0,1	I – 0,26 II – 0,35 III – 0,38
	II	25	0,96		
	III	22	1,00		
KPI_2	I	23	0,76	0,2	
	II	15	0,92		
	III	11	1,00		
KPI_3	I	60	0,00	0,2	
	II	72	0,86		
	III	74	1,00		
KPI_4	I	6	0,56	0,3	
	II	7	0,67		
	III	7	0,67		
KPI_5	I	5	0,44	0,2	
	II	5	0,44		
	III	7	0,67		

The analysis of results of the team performance evaluation from the Table 4 demonstrates the positive dynamics of changes of the team productivity on 9–12%.

The commonly used approaches to decision-making in sprint planning [5–12] allow determining the complexity of user stories in specified units. In comparison to them, the proposed technology allow calculation of the complexity of user stories, takes into account uncertainty of the current sprint, and selects a set of user stories from the sprint backlog when the total complexity of the sprint backlog exceeds the team velocity.

The proposed technology enables increasing of the team productivity and provides additional information for sprint planning. The using of the decision support technology for sprint planning and the obtained results show the feasibility of using the proposed technology in real conditions.

CONCLUSIONS

In the course of this research the decision-making technology for solving the planning problem in the face of uncertainty has been proposed. For this purpose, an analytical review of the methods of estimating the labor intensity of user stories has been conducted. It has revealed the shortcomings of existing approaches. Due to the development of technology of decision support in software development the planning task in the face of uncertainty has been further developed. A sprint planning algorithm has been developed. Formalization of the process of estimation of the labor content of user stories based on the previous experience has been presented. Sprint Backlog reshuffling model in case of extra labor effort needed for its implementation in comparison with team velocity has been developed.

The scientific novelty of the obtained results consists in improvement of the sprint planning process with the assistance of the proposed technology, which helps to reduce uncertainty while defining labor intensity of user stories and Sprint Backlog as a whole. Numerous studies have shown that the use of the proposed technology requires only handy tools, such as Microsoft Excel, OpenOffice Calc, LibreOffice Calc, PlanMaker and others, which do not require from project managers and Product Owners any specific mathematical skills. The results show the practical significance of the approach for IT companies and the ability to use the proposed technology in software development projects to increase the effectiveness of decision-making process in uncertainty for project managers, product owners and development teams.

Prospects for further research consists in creating an IS that will reduce the time spent on data processing about US, and will automate the decision-making process for planning the sprint.

ACKNOWLEDGEMENTS

The article is supported by the state budget scientific research project of National Technical University «Kharkiv Polytechnic Institute» of Software Engineering and Management Information Technologies department «Development of models and methods of collecting and automated processing of business information in Web» (state registration number 0119U002556).

REFERENCES

1. Chastka IT-industriyi v ekonomitsi Ukrainy [Electronic resource]. – Access mode: <https://ua.112.ua/suspilstvo/chastka-it-industrii-v-ekonomitsi-ukrainy-standovyt-4-vvp-kubiv-480452.html>
2. Software Engineering Body of Knowledge (SWEBOK) [Electronic resource]. Access mode: <https://www.computer.org/education/bodies-of-knowledge/software-engineering>
3. A Guide to the Scrum Body of knowledge (SBOKTM Guide) [Electronic resource]. Access mode: <http://www.scrumstudy.com/SBOK/SCRUMstudy-SBOK-Guide-2016.pdf>

4. Sutherland J., Sutherland J. J. Scrum: A revolutionary approach to building teams, beating deadlines and boosting productivity. United States, Random House, 2014, 256 p.
5. 7 Agile Estimation Techniques – beyond Planning Poker [Electronic resource]. Access mode: <https://technology.amis.nl/2016/03/23/8-agile-estimation-techniques-beyond-planning-poker/>
6. Agile Estimation Techniques: A True Estimation in an Agile Project [Electronic resource]. Access mode: <https://www.softwaretestinghelp.com/agile-estimation-techniques/>
7. Lopez-Martinez J., Ramirez-Noriega A., Juarez-Ramirez R. et al.] User stories complexity estimation using Bayesian networks for inexperienced developers, *Cluster Computing*. 2018, Vol. 21, pp. 715–728. DOI: 10.1007/s10586-017-0996-z
8. Karna H., Gotovac S. Estimating software development effort using Bayesian networks, *Telecommunications and Computer Networks (SoftCOM) : 23rd International Conference on Software, Split, 16–18 September 2015 : proceedings. Split: IEEE*, 2015, pp. 229–233. DOI: 10.1109/SOFTCOM.2015.7314091
9. [Barrera F. E., García M. A., González H. G. et al. Agile Evaluation of the Complexity of User Stories Using the Bloom's Taxonomy, *Computational Science and Computational Intelligence (CSCI'17) : 4th Annual International Conference, Las Vegas, 14–16 December 2017 : proceedings. Las Vegas, Nevada: Conference Publishing Services (CPS)*, 2017, pp. 1047–1050. DOI:10.1109/csci.2017.182
10. Castillo-Barrera F. E., Amador-García M., Pérez-González H. G. et al. Adapting Bloom's Taxonomy for an Agile Classification of the Complexity of the User Stories in SCRUM, *Software Engineering Research and Innovation (CONISOFT'18) : 6th International Conference, San Luis Potosi, 24–26 October 2018, proceedings. San Luis Potosi, IEEE*, 2018, pp. 139–145. DOI: 10.1109/CONISOFT.2018.8645899
11. Nassif A. B., Capretz L. F., Ho D. Estimating software effort based on use case point model using Sugeno Fuzzy inference system, *International Conference on Tools with Artificial Intelligence (ICTAI 2011) : 23rd IEEE International Conference, Boca Raton, 7–9 November 2011 : proceedings. Boca Raton, Florida, USA, IEEE*, 2011, pp. 393–398. DOI: 10.1016/j.asoc.2016.05.008
12. Arora M., Verma S., Kavita An efficient effort and cost estimation framework for Scrum Based Projects, *International Journal of Engineering & Technology*, 2018, Vol. 7, No. 4.12, P. 52–57. DOI: 10.14419/ijet.v7i4.12.20992.
13. Business Process Model and Notation [Electronic resource]. Access mode: <http://www.bpmn.org/>
14. Estimate a Story [Electronic resource]. Access mode: <https://www.quickscrum.com/ScrumGuide/175/sg-Estimate-A-Story>
15. Saaty T. L. The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation. London, McGraw-Hill, 1980, 287 p.
16. Crawford G., Williams C. The Analysis of Subjective Judgment Matrices : tech. report : R-2572-1-AF, Rand Corp. Santa Monica, Calif, 1985. [Electronic resource]. Access mode: <https://www.rand.org/pubs/reports/R2572-1.html>
17. Eduardo M. Sizing User Stories Using Paired Comparisons. [Electronic resource]. Access mode: https://www.researchgate.net/publication/222140271_Sizing_user_stories_using_paired_comparisons
18. Mel'nik K. V. Otsenka effektivnosti meditsinskoy informatiionoy tekhnologii, *Matematicheskoye modelirovaniye protsessov v ekonomike i upravlenii innovatsionnymi proyektami (MMP-2013) : 1 Mezhdunarodnaya nauchno-prakticheskaya konferentsiya, Alushta, 9–15 Sentyabrya 2013 g. : tezisy dokladov. Khar'kov, KHNURE*, 2013, pp. 122–123.
19. Ovezgel'dyyev A. O., Petrov E. G., Petrov K. E. Sintez i identifikatsiya modeley mnogofaktornogo otsenivaniya i optimizatsii. Kiev, «Naukova dumka», 2002, 163 p.
20. Berteig M. Pitfall of Scrum: Excessive Preparation/Planning [Electronic resource], Access mode: <http://www.agileadvice.com/tag/analysis-paralysis/>
21. Lewis J. P. Fundamentals of Project Management. Warszawa, AMACOM, 2002, 128 p.

Received 15.01.2020.
Accepted 25.02.2020.

УДК 004.02

ТЕХНОЛОГІЯ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ПРИ ПЛАНУВАННІ СПРИНТУ

Мельник К. В. – канд. техн. наук, доцент кафедри Програмної інженерії та інформаційних технологій управління, Національний технічний університет «Харківський політехнічний інститут», Харків, Україна.

Глушко В. Н. – Senior Consultant, Solution Architect, GlobalLogic, Харків, Україна.

Борисова Н. В. – канд. техн. наук, доцент кафедри Інтелектуальних комп'ютерних систем, Національний технічний університет «Харківський політехнічний інститут», Харків, Україна.

АНОТАЦІЯ

Актуальність. У статті описується актуальний процес планування розробки програмного забезпечення, проблем планування і різні рішення цих проблем на основі використання методології Scrum.

Мета. Метою даної роботи є розробка технології для вирішення задачі планування спринту в умовах невизначеності і можливих ризиків з точки зору розробки програмного забезпечення.

Метод. Описано найбільш використовувані моделі життєвого циклу програмного забезпечення. Обґрунтовано вибір Scrum, як найбільш часто використовуваного представника гнучкою методології розробки програмного забезпечення. Проведено аналітичний огляд методів оцінки складності для історій користувача. Виділено основні проблеми планування спринту. Розроблено модель бізнес-процесу для реалізації IT-проекту по Scrum у вигляді BPMN-діаграми. Розроблено алгоритм вирішення проблеми планування Sprint Backlog в умовах невизначеності. Розглядається загальний процес вибору для користувача історій з Product Backlog для Sprint Backlog і шляхи вирішення можливих проблем. Формалізовані задача оцінки трудомісткості історій користувача і задача оцінки ризиків при плануванні. Була розроблена технологія вибору історій користувача для Sprint Backlog. Проведено чисельні дослідження технології підтримки прийняття рішень, яка була запропонована

в статті. Це дозволяє пропонувати її в якості практичного інструменту при плануванні спринту. Запропоновано метод оцінки адекватності запропонованої технології. Обрано набір ключових показників ефективності для оцінки продуктивності команди.

Результати. Була розроблена технологія планування спринту, яку можуть використовувати керівники проєктів, власники продуктів і команди розробників для підвищення ефективності процесу прийняття рішень.

Висновки. Проведені експерименти підтвердили значимість запропонованої технології підтримки прийняття рішень і дозволяють рекомендувати її для практичного використання при плануванні програмних проєктів. Наукова новизна полягає в поліпшенні процесу планування спринту за допомогою запропонованої технології, яка усуває невизначеність при визначенні трудомісткості користувальницьких історій і скорочує час, що витрачається на прийняття рішень.

КЛЮЧОВІ СЛОВА: спринт беклог, задача планування, невизначеність, трудомісткість історія користувача, задача вибору, ефективність команди.

УДК 004.02

ТЕХНОЛОГИЯ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ ПРИ ПЛАНИРОВАНИИ СПРИНТА

Мельник К. В. – канд. техн. наук, доцент кафедры Программной инженерии и информационных технологий управления, Национальный технический университет «Харьковский политехнический институт», Харьков, Украина.

Глушко В. Н. – Senior Consultant, Solution architect, GlobalLogic, Харьков, Украина.

Борисова Н. В. – канд. техн. наук, доцент кафедры интеллектуальных компьютерных систем, Национальный технический университет «Харьковский политехнический институт», Харьков, Украина.

АННОТАЦИЯ

Актуальность. В статье описывается актуальный процесс планирования разработки программного обеспечения, проблем планирования и различные решения этих проблем на основе использования методологии Scrum.

Цель. Целью данной работы является разработка технологии для решения задачи планирования спринта в условиях неопределенности и возможных рисков с точки зрения разработки программного обеспечения.

Метод. Описаны наиболее используемые модели жизненного цикла программного обеспечения. Обоснован выбор Scrum, как наиболее часто используемого представителя гибкой методологии разработки программного обеспечения. Проведен аналитический обзор методов оценки сложности пользовательских историй. Выделены основные проблемы планирования спринта. Разработана модель бизнес-процесса для реализации IT-проекта по Scrum в виде BPMN-диаграммы. Разработан алгоритм решения проблемы планирования Sprint Backlog в условиях неопределенности. Рассматривается общий процесс выбора пользовательских историй из Product Backlog для Sprint Backlog и пути решения возможных проблем. Формализованы задача оценки трудоемкости пользовательских историй и задача оценки рисков при планировании. Была разработана технология выбора пользовательских историй для Sprint Backlog. Проведены численные исследования технологии поддержки принятия решений, которая была предложена в статье. Это позволяет предлагать ее в качестве практического инструмента при планировании спринта. Предложен метод оценки адекватности предлагаемой технологии. Выбран набор ключевых показателей эффективности для оценки продуктивности команды.

Результаты. Была разработана технология планирования спринта, которую могут использовать руководители проєктов, владельцы продуктов и команды разработчиков для повышения эффективности процесса принятия решений.

Выводы. Проведенные эксперименты подтвердили значимость предложенной технологии поддержки принятия решений и позволяют рекомендовать ее для практического использования при планировании программных проєктов. Научная новизна заключается в улучшении процесса планирования спринта с помощью предлагаемой технологии, которая устраняет неопределенность при определении трудоемкости пользовательских историй и сокращает время, затрачиваемое на принятие решений.

КЛЮЧЕВЫЕ СЛОВА: спринт беклог, задача планирования, неопределенность, трудоемкость истории пользователя, задача выбора, эффективность команды.

ЛІТЕРАТУРА / LITERATURA

1. Частка IT-індустрії в економіці України [Electronic resource]. Access mode: <https://ua.112.ua/suspilstvo/chastka-it-industrii-v-ekonomitsi-ukrainy-stanovyit-4-vvp-kubiv-480452.html>
2. Software Engineering Body of Knowledge (SWEBOOK) [Electronic resource]. – Access mode: <https://www.computer.org/education/bodies-of-knowledge/software-engineering>
3. A Guide to the Scrum Body of knowledge (SBOKTM Guide) [Electronic resource]. – Access mode: <http://www.scrumstudy.com/SBOK/SCRUMstudy-SBOK-Guide-2016.pdf>
4. Sutherland J. Scrum: A revolutionary approach to building teams, beating deadlines and boosting productivity / J. Sutherland, J. J. Sutherland. – United States : Random House, 2014. – 256 p.
5. Agile Estimation Techniques – beyond Planning Poker [Electronic resource]. – Access mode: <https://technology.amis.nl/2016/03/23/8-agile-estimation-techniques-beyond-planning-poker/>
6. Agile Estimation Techniques: A True Estimation in an Agile Project [Electronic resource]. – Access mode: <https://www.softwarerequestinghelp.com/agile-estimation-techniques/>
7. User stories complexity estimation using Bayesian networks for inexperienced developers / [J. Lopez-Martinez, A. Ramirez-Noriega, R. Juarez-Ramirez et al.] // Cluster Computing. – 2018. – Vol. 21. – P.715-728. DOI: 10.1007/s10586-017-0996-z
8. Karna H. Estimating software development effort using Bayesian networks / H. Karna, S. Gotovac // Telecommunications and Computer Networks (SoftCOM) : 23rd International Conference on Software, Split, 16–18 September

- 2015 : proceedings. Split: IEEE, 2015. – pp. 229–233. DOI: 10.1109/SOFTCOM.2015.7314091
9. Agile Evaluation of the Complexity of User Stories Using the Bloom's Taxonomy / [F. E. Barrera, M. A. García, H. G. González et al.] // Computational Science and Computational Intelligence (CSCI'17) : 4th Annual International Conference, Las Vegas, 14–16 December 2017 : proceedings. Las Vegas, Nevada: Conference Publishing Services (CPS), 2017. – P. 1047–1050. DOI:10.1109/csci.2017.182
 10. Adapting Bloom's Taxonomy for an Agile Classification of the Complexity of the User Stories in SCRUM / [F. E. Castillo-Barrera, M. Amador-García, H. G. Pérez-González et al.] // Software Engineering Research and Innovation (CONISOFT'18) : 6th International Conference, San Luis Potosi, 24–26 October 2018 : proceedings. San Luis Potosi : IEEE, 2018. – P. 139–145. DOI: 10.1109/CONISOFT.2018.8645899
 11. Nassif A. B. Estimating software effort based on use case point model using Sugeno Fuzzy inference system / A. B. Nassif, L. F. Capretz, D. Ho // International Conference on Tools with Artificial Intelligence (ICTAI 2011) : 23rd IEEE International Conference, Boca Raton, 7–9 November 2011 : proceedings. Boca Raton, Florida, USA: IEEE, 2011. – P. 393–398. DOI: 10.1016/j.asoc.2016.05.008
 12. Arora M. An efficient effort and cost estimation framework for Scrum Based Projects / M. Arora, S. Verma, Kavita // International Journal of Engineering & Technology. – 2018. – Vol. 7, № 4.12. – P. 52–57. DOI: 10.14419/ijet.v7i4.12.20992.
 13. Business Process Model and Notation [Electronic resource]. – Access mode: <http://www.bpmn.org/>
 14. Estimate a Story [Electronic resource]. – Access mode: <https://www.quickscrum.com/ScrumGuide/175/sg-Estimate-A-Story>
 15. Saaty T. L. The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation / T. L. Saaty. – London: McGraw-Hill, 1980. – 287 p.
 16. The Analysis of Subjective Judgment Matrices : tech. report : R-2572-1-AF / G. Crawford, C. Williams / Rand Corp. – Santa Monica, Calif, 1985. [Electronic resource]. – Access mode: <https://www.rand.org/pubs/reports/R2572-1.html>
 17. Eduardo M. Sizing User Stories Using Paired Comparisons. [Electronic resource] / M. Eduardo. – Access mode: https://www.researchgate.net/publication/222140271_Sizing_user_stories_using_paired_comparisons
 18. Мельник К. В. Оценка эффективности медицинской информационной технологии / К. В. Мельник // Математическое моделирование процессов в экономике и управлении инновационными проектами (ММП-2013) : 1 Международная научно-практическая конференция, Алушта, 9-15 Сентябрь 2013 г. : тезисы докладов. – Харьков : ХНУРЭ, 2013. – С. 122–123.
 19. Овезгельдыев А. О. Синтез и идентификация моделей многофакторного оценивания и оптимизации / А. О. Овезгельдыев, Э. Г. Петров, К. Э. Петров. – К. : «Наукова думка», 2002. – 163 с.
 20. Berteig M. Pitfall of Scrum: Excessive Preparation/Planning [Electronic resource] / M. Berteig. – Access mode: <http://www.agileadvice.com/tag/analysis-paralysis/>
 21. Lewis J. P. Fundamentals of Project Management / J. P. Lewis. – Warszawa: AMACOM, 2002. – 128 p.
 22. Berteig M. Pitfall of Scrum: Excessive Preparation / Planning [Electronic resource] / Access mode: <http://www.agileadvice.com/tag/analysis-paralysis/>, 17.01.2020.
 23. James P. Lewis. Fundamentals of Project Management / P. Lewis James. – AMACOM, 2002. – 128 p. ISBN 978-0814471326.