Advances in Social Science, Education and Humanities Research, volume 392
Ecological-Socio-Economic Systems: Models of Competition and Cooperation (ESES 2019)

Teamwork Management in the Field of Software Development

Tatyana Emelyanova
Department of Management and Marketing
Kurgan State University
Energy Retail Company "Vostok"
Kurgan, Russia
dominika2102@mail.ru

Elena Ilchenko
Department of Management and Marketing
Kurgan State University
Kurgan, Russia
elnik01@mail.ru

Zinaida Varlamova
Department of Management and Marketing
Kurgan State University
Kurgan, Russia
varlamova_zn@mail.ru

Lyudmila Paklina
Department of Management and Marketing
Kurgan State University
Kurgan, Russia
paklinala@yandex.ru

***Abstract*—One of the important issues in implementing the strategy of social and economic development of a country and regions is the problem of managing the intellectual resources of companies. Particularly, a growing interest is taken in the team approach in managing the intellectual resources of enterprises in the field of software development. The purpose of the study is to analyze approaches to the management of companies' intellectual resources in the field of software development, as well as formation of a teamwork management model for these enterprises. The article presents the results of studying software development methodologies, as well as a model of teamwork management at enterprises in the field of software development using Scrum methodology. This approach is the most optimal, because allows for creating self-organizing and self-governing teams.**

Keywords—team, management, team building, methodology, software development.

I. INTRODUCTION

In the development of the modern economy, the importance of companies' intellectual resources is increasingly growing. The main tasks of IT management since its inception are to regularly provide IT services and to increase the efficiency of business processes. At the same time, companies are interested in making IT contribution to the results of their activities more measurable, direct and flexible. [1].

In the field of software development, intellectual property management is one of the key success factors for a company in the information industry.

The effectiveness of software development companies depends largely on the correctness of the chosen software development methodology.

II. RESEARCH METHODOLOGY

The solution of the tasks set in the work was carried out on the basis of the application of general scientific research methods: search, grouping, comparative analysis, comparison, generalization of approaches and the development of a scientific hypothesis.

Globally, companies use two main approaches to software development:

- Waterfall development is an approach to software development where the process looks like a flow that goes through the phases of requirement analysis, design, implementation, testing, integration and support;
- Spiral development is an approach in which the work is performed in parallel with the continuous analysis of the results obtained and the adjustment of the previous stages of work. The project with this approach in each phase of development passes through a repeating cycle: Planning — Implementation — Verification — Evaluation. Figure 1 shows the software development methodologies related to waterfall or spiral software development.

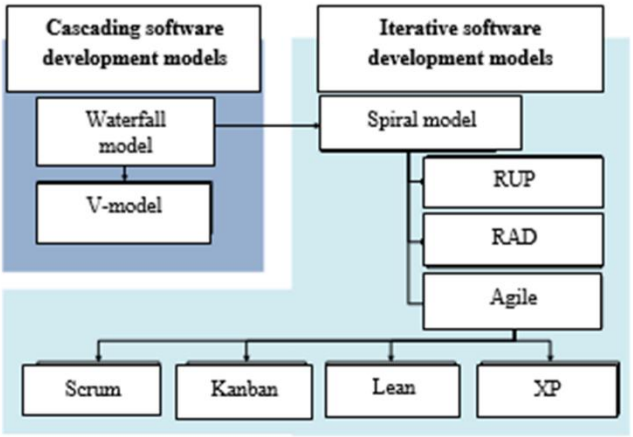

Fig. 1. Varieties of software development methodologies.

Each of the presented models has its own characteristics that affect the software development process. The choice of one of them depends on a large number of aspects, such as the type of project, customer requirements, deadlines, as well as presence of a highly qualified team. Table 1 presents the results of software development methodologies research.

TABLE I. RESULTS OF SOFTWARE DEVELOPMENT METHODOLOGIES RESEARCH (COMPLETED BY THE AUTHOR)

Methodology	Advantages	Disadvantages	Team approach implementation
Waterfall model	1. Works well in the projects where requirements can be clearly defined and fixed	1. Ignores dynamic changes 2. Difficult to manage risks	Not necessary
V-model	1. Planning at early stages of developing a system for testing; 2. Easy to use. 3. Simplification of tracking the development process.	1. Difficult to support parallel events; 2. Iterations between phases are not provided 3. Late testing of requirements in the life cycle 4. Absence in the model of actions aimed at risks analysis.	Not necessary
Spiral model	1. Improved risk analysis 2. Good documentation of the development process; 3. Early creation of working prototypes. 4. Possibility of making changes at relatively late stages;	1. Quite expensive to use; 2. Risk management requires involvement of highly qualified specialists; 3. Not suitable for small projects.	Not necessary
RUP	1. Takes into account evolutionary requirements. 2. Function integration is gradual 3. Early product release. 4. Continuing education. 5. Continuous product improvement.	1. An insufficient level of formalism, leading to inconsistent decisions, and, consequently, to unproductive expenditure of resources	Not necessary
RAD	1. Using prototyping 2. Mandatory user involvement in the development process	1. Modules are designed in isolation, which leads to a large number of integration errors. 2. Lack or insufficiency of documentation;	Not necessary
Scrum	1. Ability to quickly launch a project with the highest priority functions and the lowest possible budget; 2. Daily control over the progress, 3. Frequent projects demonstrations. The ability to make adjustments to the requirements document during project implementation	Does not imply a fixed budget and a fixed requirements document, not applicable for working with government orders, does not work with poorly qualified team, underestimated deadlines and budget	Necessary
Kanban	1. Noclear deadlines. 2. Exclusion from production of ineffective stocks and materials, due to this reduced production costs. 3. High program agility	1. Implementation is possible only with a team of 5 people. 2. Not suitable for matrix structures of enterprise organization. 3. Not suitable for long-term strategies.	Necessary
Lean	1. High organization processes allow for avoiding completely unnecessary expenses 2. Quick product release 3. The product with the least number of errors 4. Cost reduction	1. Non-involvement of staff 2. Applies only if experienced developers join the project 3. All decisions should be supported by analytical data	Not necessary
XP	1. Extremely low overhead 2. The project can show exceptional effectiveness	1. Can only be used in a team of experienced developers 2. Cannot split the team into parts, 3. The size of the team is limited to 10-15 people.	Necessary

The study allows us to conclude that each of the models considered is designed to develop a particular class of software with different requirements for development teams. It should be noted that at the present stage more attention is paid to methodologies that implement a team approach. Figure 2 shows the results of an Agile Survey study about the popularity of agile methodologies. [2].

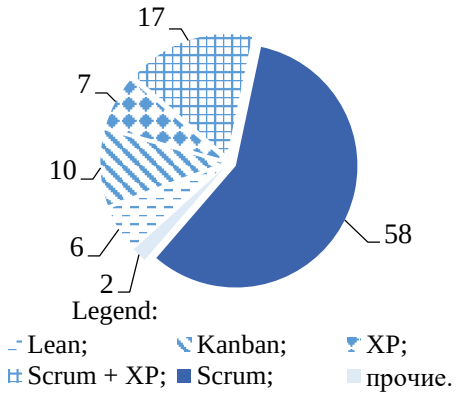

Fig. 2. The results of an Agile Survey research about the popularity of agile methodologies [1]

III. RESEARCH RESULTS

The research findings show that it is practical to use Scrum methodology at the Russian enterprises dealing with software development. Scrum agile methodology demands teamwork of the project developers, focusing on relationship within the team as well as on improving communication among them [3].

In the Scrum methodology, a team is seen as a self-organizing and self-governing structure. The work of the team is assessed as the work of a single group. In Scrum, the contribution of individual members of the project team is not evaluated, as it destroys the self-organization of the team. The structure of the Scrum team is shown in Figure. 3.

The Scrum team is cross-functional, which is another feature of this methodology. It includes people with various skills - developers, analysts, testers. There are no predefined and imparted roles in the team that limit the scope actions of the team members. The team consists of engineers who contribute to the overall success of the project in accordance with their abilities and design requirements.

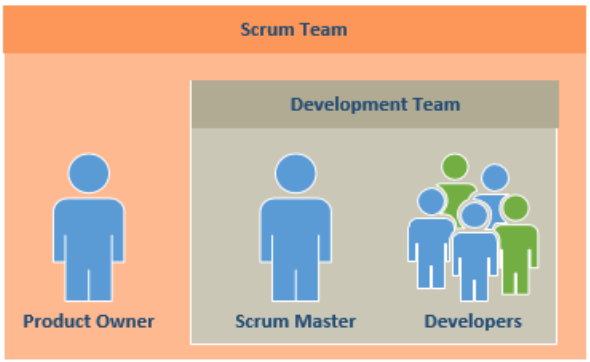

Fig. 3. Scrum team structure.

Scrum uses an iteratively incremental approach to optimize predictability and risk management [4]. The software product development model on Scrum-based methodology is depicted in Figure. 4.

Fig. 4. Software product development model on Scrum methodology.

Considering the features of Scrum methodology, one can imagine a model for implementing this approach for a particular enterprise in the field of software development (Fig.5).

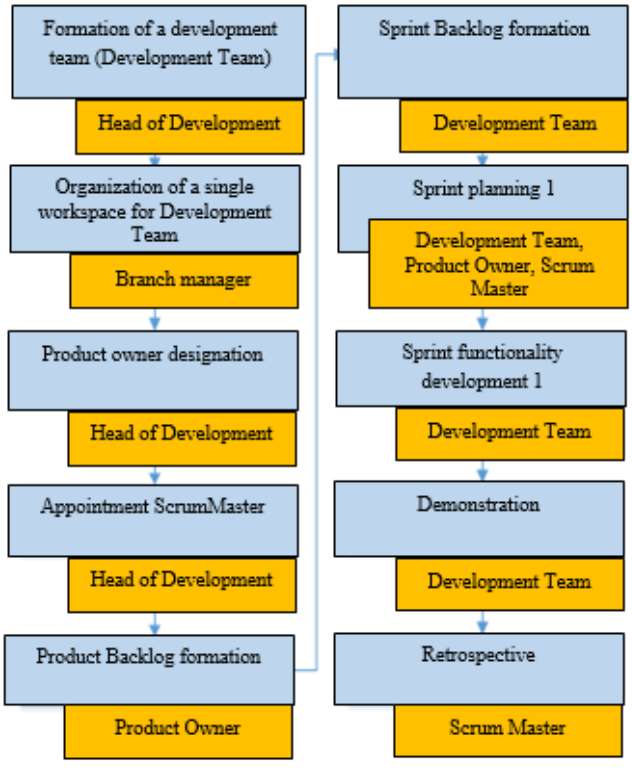

Fig. 5. Scrum model for an enterprise in the sphere of development.

Implementation of Scrum methodology should begin with creating a Development Team, as the quality of the future product depends on teamwork. The Development Team is one work unit. This work unit is self-sufficient, self-governing and self-organizing.

The main functions of the Development Team can be identified as follows:

- Development of the user-requested features for the customer;
- tracking your own performance (in collaboration with Scrum Master);
- decision-making on product development and design;
- element evaluation Product Backlog;
- responsibility of the Development Team for the result before the Product Owner[3].

When forming a development team, it is important to move away from the usual understanding of titles. Each team member has some unique skills, but, in addition to this, he also has related knowledge. This is what allows the team to be one.

Scrum methodology requires special attention to organization of a single workspace for the Development Team. When implementing Scrum methodology, it is necessary to use the practice of “open space”. So that all members of the development team can freely communicate with each other, solving problems on the go. A single space will allow the team members to constantly interact with each other, presenting a single communication process, implementing a complete feedback loop.

The next step in the implementation of the Scrum methodology is appointment of the product owner. Product Owner is the owner of the product. This is not a customer, but a full member of the team, which has not only great responsibility, but also great restrictions. Product Owner is a link between the customer and the Development Team. Final decisions on the development of the required functionality are made by the product owner. The main responsibility of Product Owner is the creation and control of Product Backlog. The product owner reflects all his decisions in Product Backlog.

Product Owner's primary obligations and responsibility for managing Product Backlog should be:

- definition of product backlog elements;
- correct arrangement of elements to optimize achievement of the goal;
- Ensuring clarity and transparency of Product Backlog;
- ensuring transparency and understandability of the requirements that all Scrum Team will have to work on;
- general optimization to achieve the highest work value from the Development Team;
- responsibility for understanding the backlog by the Development Team [4].

In order to appoint Product Owner, you must determine the list of necessary personal qualities that he must possess.

In his article, Roman Pichler, a leading expert on Scrum and Agile at Yandex, indicated that the Product Owner should be responsible, active and interested. In addition, the applicant for this role must share the principles of Agile. The product owner should be well aware of everything that concerns the product: its development, business processes, and the mood of users [5].

The question of whether the Product Owner should have programming skills is controversial. Some authors argue that the owner of the product should be chosen among experienced developers, while others say that the Product Owner is a managerial position, which means that the best option is an employee from the management industry.

Expressing the author's position, it should be said that the Product Owner should have basic knowledge in the field of software development, but do not forget that Scrum is a team work, and if the team can be trusted, then programming skills are not needed. The Product Owner will be the center of information exchange and the first one to go with problems, which means that he should be an open, purposeful, stress-resistant employee, and most importantly with communication skills.

The next step in implementing Scrum methodology is to appoint a Scrum Master. We can say that the Scrum Master will be one of the most important people in the scrum methodology. The main feature is that the Scrum Master does not give tasks, but eliminates problems that appear within the team. Questions that may arise during the workflow, as a consequence, give rise to problems, and the purpose of the Scrum Master to identify them and share them. This openness leads directly to trust within the team that will be created within the software development company. The resulting trust creates an atmosphere that will positively affect the speed and quality of work.

The main functions of the ScrumMaster can be identified as follows:

- elimination of problems arising within the team;
- identification of hidden issues;
- creation of friendly relationships in the team;
- process and task tracking;
- change the status of tasks in the sprint;
- conducting a Daily Scrum Meeting;
- organization of meetings before sprints;
- assist the Product Owner with Backlog [6].

It should be noted that Scrum Master should interact not only with the development team, but also with the Product Owner. It helps the product owner understand how to create Backlog to maximize the value of the product. The Scrum Master should try to find more efficient methods of maintaining the Backlog, using agile methods in development and management.

The next step in implementing Scrum is sprint planning. But this process cannot be started without the two components Product Backlog and Sprint Backlog, which in Scrum are called artifacts.

Product Backlog is an ordered set of items, a task queue, and a list of all the features that stakeholders want from a product. [3] This list contains brief descriptions of all the desired features of the product.

Product Backlog should be understood by absolutely everyone, for which the Product Owner is responsible. Product Backlog transparency will help both the team and the customer understand it. Often, customers and specialists speak completely different languages, and this “language barrier” mainly interferes with work. If somewhere initially there was a misunderstanding, and the interpretation of the customer’s wishes was not correct, then after a sprint there will be a finished product, which will go far away from the customer’s wishes.

Backlog items are “user stories”. Such elements are ordered according to their “weight”. The higher the specific element in the backlog, the sooner the developers will work on it. The upper positions will be more detailed and clearer in comparison with the lower elements. All must be understood by the customer and stakeholders.

The Sprint Backlog is a list of specific tasks for implementing selected elements of the product backlog. This is a list of tasks that the team will do in the next sprint, as well as a description of how the task will be implemented by the team. Each sprint team creates a new sprint backlog based on the product backlog. Sprint planning (Sprint Planning Meeting) is divided into two parts, as shown in Figure 6.

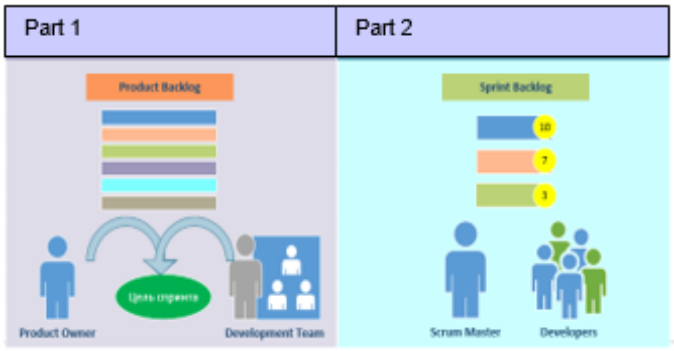

Fig. 6. Sprint planning.

In the first part, Product Owner reviews the elements from Product Backlog that need to be presented and discussed at this meeting. The product owner describes what he wants to see. The product owner does not have to describe each Product Backlog item. For Product Owner, a good guideline is talking about the tasks that will be distributed into two sprints.

It is at this meeting that questions will be asked and tasks will be discussed in a chaotic manner, since after discussing one task, another clarifying question may come to mind. In fact, the purpose of such clarifications is to clarify any ambiguity. At the end of the first part, the team must form a sprint goal.

The goal of a sprint is to formulate one or two sentences that describe what the development team plans to achieve during a Sprint. The goal is described in conjunction with the team and the product owner.

In the second part, the team will form a Sprint Backlog. At this stage, the Product Owner interference is unacceptable. The duration of work on a task will be estimated. The flexibility of Scrum methodology is shown everywhere, including the duration of sprint planning, which depends on duration of the future sprint. The scheduling duration formula is as follows: 1-week Sprint = 2 hours Sprint Planning Meeting [6].

After sprint planning, the tasks that are entered in the Sprint Backlog should be placed on the scrum Board. It is an integral artifact of Scrum methodology and serves as a center of information about the project and tasks in the sprint.

The tasks get onto the board in the form of User Story (user stories). Each card has mandatory attributes: description, weight and priority. When the task passes the next stage, it is relocated to the corresponding column. By moving tasks between columns, the team will be able to evaluate their progress. With a simple look at the board, it will become clear at what stage the project is as a whole and what is currently happening with each task.

Using scrum boards has the following advantages:

- development processes become more transparent. All participants know the status of each task and can influence the processes;
- a sequence of tasks appears within the project. It is necessary to ensure a stable level of quality and meet the expectations of all stakeholders;
- the use of an online whiteboard will allow the company's management to know at any time at what stage the development of the project is without distracting employees from software development;
- product development documentation is maintained. All relevant information on projects and tasks are stored and distributed centrally.

The next stage after the sprint planning is the stage of developing the required functionality. In the Scrum print process, special 15 - minute meetings should be held every day, i.e. Daily Scrum Meeting. They are needed in order to openly show the problems that arise in the process of the team, in order to eliminate them on time. Also, these meetings show how the work on the project as a whole is going. Thus, it is possible to formulate the purpose of such a meeting as follows: to adjust and understand the work of the team, to find out what its current problems are, and to propose solutions [7].

Participation in a Daily Scrum Meeting is mandatory for the entire Development Team, the Scrum Master and Product Owner are also involved. Other persons (customers, marketers, etc.) may be present at the meeting, but are not allowed to speak.

Having successfully completed all the tasks from the sprint backlog, the team proceeds to the next stage of scrum methodology implementation, i.e. demonstration of the product (Sprint Reviews Meeting). Sprint Review Meeting should be carried out at the end of each sprint and is of an overview nature. At the meeting, the team will be able to evaluate what they have done, and most often it will be presented in the form of a demonstration of the new product features [8].

During Sprint Review Meeting, the project is evaluated against the sprint goal that was identified during the planning. Ideally, the software development team should complete all the tasks from the Backlog Sprint, but this is not the main indicator of productivity. The important thing is that the sprint goal was achieved.

The final step in the sprint is a Sprint Retrospective Meeting, which should also be held on the last day of the sprint. If the Sprint Review Meeting aims to look at the result of the product, then the retrospective will be meant to look at the result of the team. No matter how well the team works, there is always an opportunity to improve performance. A good Scrum team will always look for opportunities to improve, and for this, a special time is allocated in the Scrum methodology to stop and think about how the team works, what can be improved and by what means and methods.

Options for conducting a retrospective may be different. In fact, to conduct more effective meetings, it is the Scrum Master that is needed, after a brainstorming session at the Sprint Retrospective Meeting, voting can begin on specific issues, which will be taken into account in the next sprint. The following retrospective will look at the remaining items from the last meeting.

IV. Conclusion

Thus, the introduction of agile Scrum methodology in software development enterprises will bring the following benefits:

- having a short-term plan in the form of a sprint backlog, which the team always undertakes to complete on time, will increase customer satisfaction.

- the team will better understand the task. This will affect the quality of the final product, reduce the time to explain problems and find errors. As a result, there will be a decrease in volumes of incidents from technical support.
- development without haste and incorrectly set deadlines will allow a necessary testing of the product, which will have a positive impact on the end user of the company's software development services. Customer service time is reduced because the number of program failures that directly affect customer service time will be minimized.
- load balancing within the development team. The amount of free resources determines that the production-teams will take as many tasks as they can perform, and in reserve time will take additional or urgent tasks.
- development of communication between the team, management and the customer. Everyone is aware of their role in the overall process, planning and distribution of work will become more transparent, which will allow everyone to assess their contribution and the contribution of other team members to creation of the final product.

Therefore, the application of Scrum methodology in enterprises in the field of software development will provide a systematic and integrated approach to software development.

Taking into account the trend of expansion of innovative activity of companies the areas of further research may be the issues of personal communications in solving problems of software development [9].

REFERENCES

[1] V. Nissen, T. Lezina, and A. Saltan, "The Role of IT-Management in the Digital Transformation of Russian Companies," Foresight and STI Governance, Vol. 12, No 3, pp. 53–61, 2018. https://doi.org/10.17323/2500-2597.2018.3.53.61

[2] B. Volfson, Gibkiye metodologii razrabotki. Saint Petersburg: Piter, 2017. (in russ.)

[3] K. Schwaber and J .Sutherland, The Scrum Guide, The definitive Guide to Scrum: The Rules of the Game. Scrum.Org and Scrum Inc, 2014.

[4] M. Cohn, "Scaling Agile to Work with Distributed Teams." https://www.mountaingoatsoftware.com/presentations/scaling-agile-and-working-with-a-distributed-team

[5] E. Stellman, D. Grin, "Postigaya Agile. Tsennosti, printsipy, metodologii. Moscow: Mann, Ivanov i Ferber, 2017. (in russ.)

[6] S. Denning, Epokha Agile. Moscow: Mann, Ivanov i Ferber, 2019. (in russ.)

[7] D. Sazerlend, Scrum: Revolyutsionnyy metod upravleniya proyektami. Moscow: Mann, Ivanov i Ferber, 2017. (in russ.)

[8] O. Kopylova, "Gibkiye metodologii dlya organizatsii upravleniya razrabotki programmnogo produkta," Vestnik sovremennykh issledovaniy, No. 1.3 (28), pp. 90-91, 2019. (in russ.)

[9] Ye. I. Kudryavtseva and N. V. Volkova, "The Role of Network Communication Activities to Establish the Project Team," Russian Management Journal, Vol. 17, No. 1, pp. 47-70, 2019. https://doi.org/10.21638/spbu18.2019.103