UDC 005.330:004.942
DOI: 10.15587/2312-8372.2019.180459

Popova O.

# ADAPTATION OF FLEXIBLE PROJECT MANAGEMENT MODELS BASED ON SCRUM AND KANBAN TECHNOLOGIES

*У даній роботі показано проведення та результат експерименту щодо поєднання двох технологій розробки програмного забезпечення (ПЗ) – Scrum та Kanban – в одну технологію, яка візьме найкращі сторони цих методологій та буде зручною і ефективною у використанні. Об'єктом дослідження є методології розробки Scrum і Kanban. Було проаналізовано існуючі гнучкі методології розробки, зокрема XP, Lean, FDD, а також Scrum і Kanban більш детально. Також була складена порівняльна таблиця останніх двох методологій, де відображено їх відмінності по низці критеріїв. Було виявлено, що в цих методологіях присутні певні недоліки і в новій методології їх би можна було прибрати і таким чином вдосконалити вже існуючі методології для певних цілей. Нова методологія має бути доволі гнучкою та адаптивною для всіх членів команди розробки програмного проекту. Дана методологія має бути зручною у використанні та мати певний сформований набір правил. За допомогою використання мереж Петрі був проведений експеримент, який дозволив наочно продемонструвати, як краще поєднати дані методології з максимальною ефективністю. Спершу методології Scrum та Kanban були зображені у вигляді моделей. Потім на основі цих двох моделей, а також нових правил, була сформована нова модель для методології. Після проведення експерименту за допомогою моделі було виявлено, як саме має виглядати запропонована методологія розробки. Її можна використовувати при розробці ПЗ. Створення даної методології на основі двох вже існуючих є дуже перспективною на сьогодні задачею, адже це має в собі 3 задачі – покращити вже існуючі методології Scrum та Kanban, сформувати набір правил нової методології, а також зробити її максимально гнучкою, адаптивною та корисною.*

**Ключові слова:** *програмне забезпечення, методологія розробки, Agile методологія, методології Scrum та Kanban, мережі Петрі.*

*Received date: 14.06.2019*
*Accepted date: 28.06.2019*
*Published date: 30.08.2019*

## 1. Introduction

Nowadays, it is very important to adhere to a certain methodology during software development for managing product development, which provides, in particular, planning, monitoring and analysis of the implementation of current tasks to achieve the project goal. It is difficult to develop a software product on time and efficiently without using such a methodology. It is important to distribute all the necessary work between the stages, plan each of them and determine how much time is needed to complete each task in a software project. Also it is necessary to control certain parameters during the development. Nowadays, a set of software development methodologies have been developed which includes:

- classical methodologies, for example, Waterfall, Iterative;
- Flexible development methodologies such as Extreme Programming (XP), Lean (Lean software development), FDD (Feature Driven Development), DSDM (Dynamics System Development Method), Crystal, Scrum and Kanban.

But each of these methodologies needs to be adapted to the implementation of the specific project taking into account the peculiar properties of the organization that creates a project. Therefore, the adaptation and use of a specific methodology for software development, as well as the creation of a combined methodology that would be better adapted to the current conditions of implementation of software projects, is an important and relevant task.

There are several different approaches to organizing software development. One of the best approaches is Agile methodology, which is a flexible development methodology, and its representatives for small development teams are Scrum and Kanban. They are used in work on software projects and allow to complete the tasks as quickly and efficiently as possible.

This paper presents a solution that combines these two flexible methodologies, and the result is offered that combines the best features of each methodology and minimizes flaws. And also it allows to enhance the useful qualities of Scrum and Kanban in their combined variant. These qualities were investigated in an experiment which allowed to compare the characteristics on the graph models and to offer a combined variant of the method with better performance.

## 2. The object of research and its technological audit

*Object of research*: Scrum and Kanban development methodologies.

The technical benefit of this research is to develop a new method for design a software project that meets the needs of a team that consists of several small groups. The basis for combining Scrum and Kanban in one methodology is the useful properties of Scrum for small and medium teams that are too difficult for small groups. Instead, Kanban's simplified and optimized organizational procedures may not always be appropriate for a small team. Quite often there is a situation on the project when it is necessary to combine useful properties of the reviewed methodologies and to coordinate with each other in a single organizational technology.

## 3. The aim and objectives of research

*The aim of research* is to conduct an experiment to create a combined software development technology based on the Scrum and Kanban methodologies.

Three tasks have to be solved to achieve this aim:

1. To analyze and to reduce the drawbacks that requires adaptation to different types of teams. Both Scrum and Kanban have their drawbacks that can be taken into account when developing new technology.
2. To add new rules for the combined technology to align the positive qualities of the underlying methodologies and to use them together.
3. To provide a more flexible process for organizing a software project.

## 4. Research of existing solutions of the problem

First of all, as a basis for the offered combined technology, let's provide an overview of Agile's flexible development methodologies. The XP methodology was used for small teams that needed to complete the project in short terms and with a common understanding of the tasks for all team members. The main purpose of this methodology is to cope with the requirements of the software product that are constantly changing and to improve the quality of the development [1]. It is characterized as frequent small releases that implement simple, well-structured solutions, and uses a TDD approach – development through testing, collective code ownership or design patterns ownership, game of planning. The disadvantages of XP are the fact that it is quite difficult to adapt to the project in case of increasing the team and extending the project boundaries. One of the flexible methodologies is Lean (Lean software development). It is originated from a lean manufacturing technology and aims to eliminate all kinds of unnecessary costs. These are unnecessary features, unnecessary processing, unfinished work, time to search for and eliminate defects, which can be reduced by early analysis and identification of errors in the early stages, etc. FDD (Feature Driven Development) is a feature-oriented development that focuses on breaking down a project into small features that are useful from a customer's point of view and are implemented within no more than two weeks. This methodology is a combination of five elements: development of a common model, creation of a functions list, planning, design and development and implementation [2]. The most valuable features of the two methodologies mentioned above have been developed in modern flexible approaches.

The Scrum methodology provides a free set of ordered rules that are based on the best practices and are constantly being refined [3]. An important feature of Scrum is its flexibility and customer focus, as it considers that the client is directly involved in the process [4].

The Kanban methodology is designed for small and very small teams and allows to distinguish from the RUP the most necessary processes for a software project execution in accordance with the «just in time» principle. The last two methodologies are most fully adapted to the modern organizational and technical features of the implementation of software projects and can be used as a basis for the combined technology that is offered.

Scrum is an authoritative flexible development methodology with non-standard team roles distribution and unique iteration organization. The key positions in Scrum teams are occupied by Scrum Master and Product Owner, the development process consists of a certain number of iterations, the iteration begins with planning, then the design and development of a part of the product takes place that ends with a demo and retrospective [5]. The main concepts of Scrum are:

1. Sprint – all the steps that are required to implement entries from a Scrum product block are performed in sprints (also called «iterations») within Scrum. Sprints are always short: usually about 2–4 weeks [6].
2. Sprint backlog is a list of tasks that have to be done in this sprint.
3. Sprint planning is a meeting where sprint planning occurs, that is, the tasks that will be included in it.
4. Scrum board is a tool that helps teams to make elements from Sprint backlog visible [7].
5. Daily Scrum meeting is meeting that usually held in one place and at the same time each day. These meetings are strictly designed for 15 minutes [8].
6. Sprint review (in other words, a demo) is a demonstration of the completed tasks that are discussed between the Scrum team and other sprint review participants [9].
7. Sprint retrospective is an opportunity for the team to review their work and to create a plan to improve some elements in the development process for a future sprint [10].

Kanban is a method of managing software development using «just in time» principle and with a unique overload of team members. The process from the task description to the delivery of the results of its execution to the user is clearly shown in this method to the participants of the process, team members can take work from the queue [11].

It is important for Kanban to do the following:

1. Kanban board – a board that shows the tasks that will be completed, in the process of being completed and are already completed. In addition, it limits the number of tasks that are in column «In progress» [12].
2. Kanban meeting is also important for keeping the team in sync so that they know what each member is doing and what status they have. Ideally, the meeting should last 10–15 minutes [13].

Each of these two software development methodologies has both advantages and disadvantages. In addition, two methodologies have a lot in common, but there are some differences. For clarity, the advantages and disadvantages of each technology are presented using the comparative characteristics in Table 1. This table shows comparison of methodologies based on several criteria. It is shown what is the main thing in the development, tasks planning, sprint importance, sprint length, sprint planning, changes and additions in the sprint, roles, board with tasks and its organization. Also there are team meetings, performance measurement and tasks estimation.

**Table 1**

Comparative characteristics of Scum and Kanban

| Scrum | Kanban |
|---|---|
| The main thing in the development | |
| Sprint | Task |
| Tasks planning | |
| A backlog is created with clear tasks to be done in the sprint | There is a workflow that displays all the tasks that need to be completed and simply moves from one status to another |
| Sprint importance | |
| In Scrum everything is done in sprints that last from 2 to 4 weeks. At the end of the sprint there is a sprint review (demo) | Kanban releases as soon as the product is finished. It can be several times a day or once a week |
| Sprint length | |
| Always the same (2–4 weeks) | No clear limitations |
| Sprint planning | |
| Sprint planning is done at the beginning of the sprint | There is no clear planning |
| Changes/additions in the sprint | |
| There cannot be changes/additions. At the beginning of the sprint it is decided how many tasks and what tasks should be done (sprint scope) | It is possible. When the task is completed, a new task can be created; very flexible methodology |
| Roles | |
| There are some roles (Product owner, Scrum master, Scrum team members) | Optional (can be SDM and SRM), the focus is not on role distribution but on delivering the product [14] |
| Board with tasks | |
| Board is new for every sprint | Board remains the same. This is an advantage if the team is small, because it is possible to see all the tasks, their connections, and progress on the same board |
| Organizing board with tasks | |
| Usually consists of columns «To do», «In progress», «In testing» and «Done» | Usually consists of columns «To do», «In progress» and «Done» |
| Meetings | |
| Daily and mandatory. There are the following types of meetings: sprint planning, daily scrum, sprint review, retrospective. Usually daily meetings take no more than 15 minutes | Daily meetings are optional but recommended. There are the following types of meetings: daily meeting, replenishment, delivery planning meeting, service delivery meeting, operations review, risk review, strategy review [15] |
| Performance measurement | |
| It is measured in story points or, more precisely, in the speed of completing sprint tasks | It is measured in the speed of moving the task from status «To do» to «Done» |
| Tasks estimation | |
| Always | The tasks are not always estimated |

It is seen from the Table 1 that each of these methodologies has its advantages and disadvantages. When choosing a methodology it is necessary to ask yourself what it is necessary – continuous development or iterations, structured roles or teams without roles, adaptation to changes or stability.

## 5. Methods of research

This study will use process design, environment design and experiment to identify which Scrum and Kanban sides should be used, how to combine them and under what conditions.

Petri nets method was chosen for modeling. Petri nets are a mathematical instrument for modeling dynamic discrete systems [16]. Petri nets are a bipartite oriented graph which elements (nodes) are used to display certain events [17].

In the process of this modeling it is possible to see how the status of the elements in model changes and how it affects the model in general. Transitions in Petri nets help to figure out how the system will behave in a particular environment. Petri nets show a transition in a network where labels from the entry transition positions go to the exit positions.

Petri nets consist of four elements:

– set of positions $P$;

– set of transitions $T$;

– input function $I$;

– output function $O$ [18].

The undeniable advantage of Petri nets is a mathematically accurate description of the model. It makes possible to perform scientific analysis with the help of modern computing (also with parallel architecture) [19].

It is possible to simulate the task to determine what way is the best to combine Scrum and Kanban using Petri nets. First of all, Scrum methodology model can be shown based on the comparative Table 1 above. Fig. 1 shows the whole Scrum process: an idea that is formed on the basis of what the end users, clients and team want, creating a general list of product development tasks for the whole period (user stories; tasks that are prioritized by business value), creating a similar list for a specific sprint at the meeting, conducting a sprint that lasts 1–4 weeks. Every day there is a meeting where three issues are discussed in the team – what has been done, what obstacles exist and what is planned to be done by the next meeting. Tasks have 4 statuses – To do, In progress, In testing and Done, which are displayed on the board with tasks. After the sprint there is a meeting where the results of the new functionality are shown and discussed. A retrospective is also held at the end. The whole process is based on the fact that the methodology is quite flexible and is targeted at different teams.

A model for Kanban development methodology is also shown (Fig. 2). First, there is an analysis of what needs to be done. This is followed by the design, development, testing, UAT and SIT testing and deployment. It is worth to mention that there are no restrictions for sprint tasks, so change requests tasks may also be added, as well as bugs (non-functional program elements) or maintenance tasks.

A new development methodology will be created based on Scrum and Kanban models.

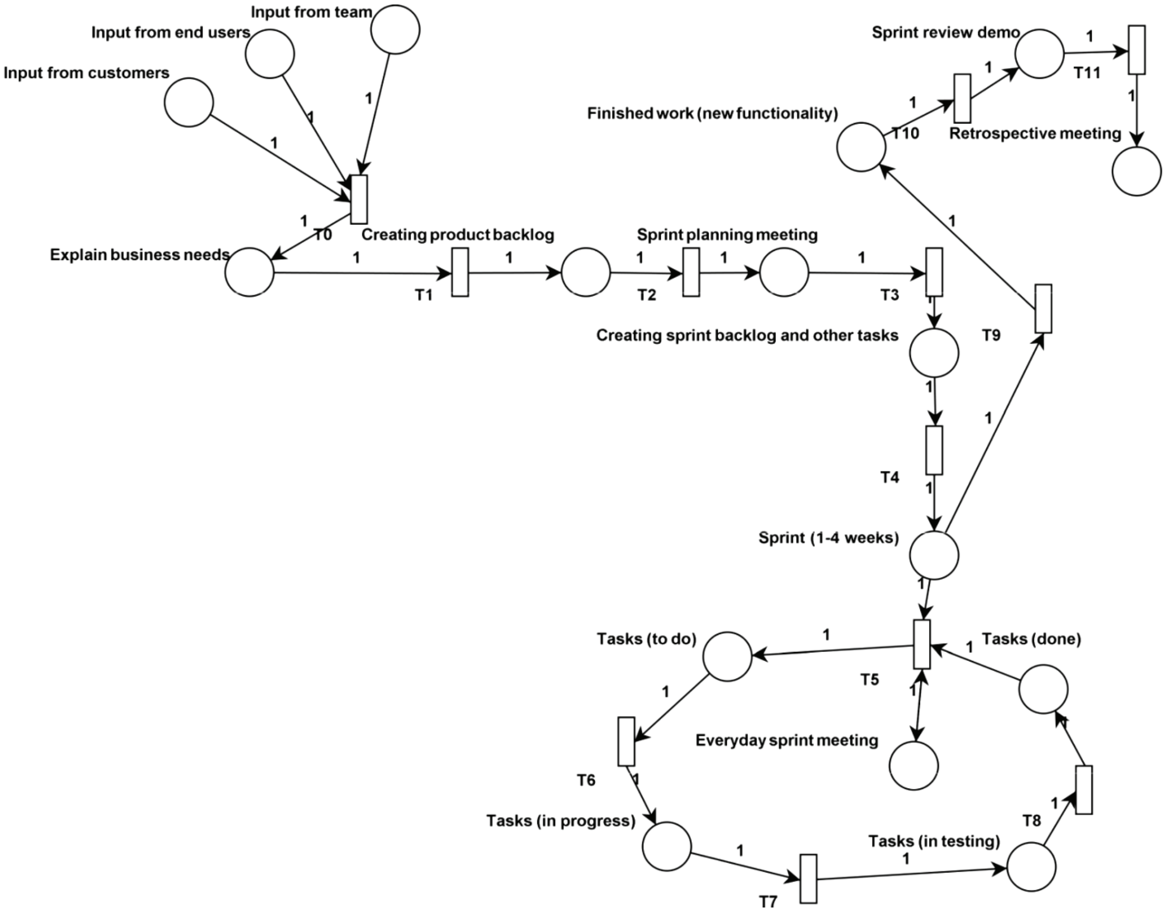

**Fig. 1.** Scrum methodology model

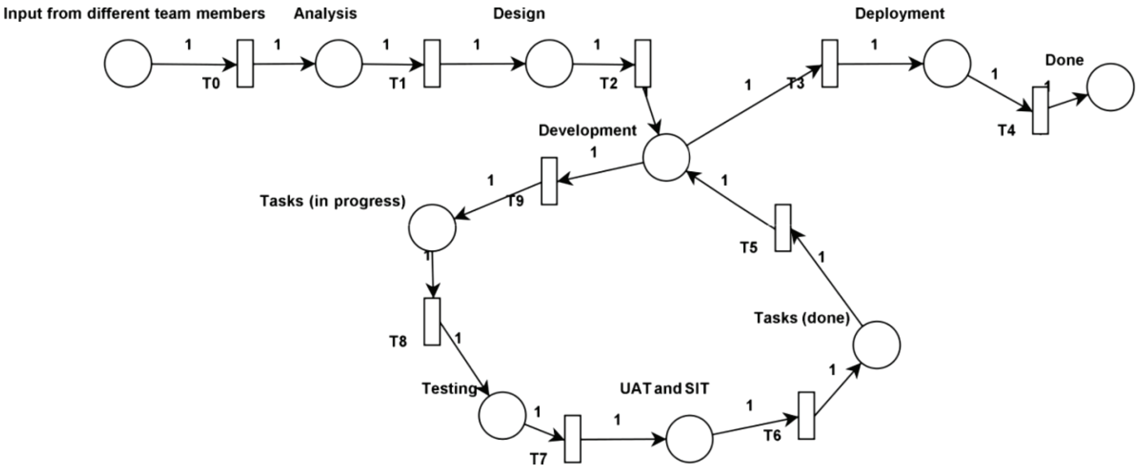

**Fig. 2.** Kanban methodology model

## 6. Research results

It is decided to take the best sides of each methodology by combining the data of the two methodologies. This is the hypothesis for the experiment. The following items from the Table 1 that is located above are selected to create a new methodology based on Scrum and Kanban:

1. Sprint is the main thing in the development. This hypothesis was taken from the Scrum methodology. It is better to develop the product gradually using an iterative method.

2. Tasks planning. It would be a good idea to create the main tasks that need to be completed, and as they are done, it is possible to take tasks from the backlog as well as bugs and change request tasks. So, on the one hand, it is the part of the Kanban methodology, and on the other hand, it is a new requirement.

3. Sprint importance. It is suggested to take the Scrum methodology rule and do all tasks using sprints that would last 2 weeks and finally have a sprint review for clients.

4. Sprint length. It is suggested to use 2 weeks as the length of the sprint, but with the following corrections: if the task was completed earlier, you have to take next tasks from the backlog, change requests tasks or bugs. If team did not have enough time, the task has to be moved to the next sprint.

5. Sprint planning. Planning should be done to structure the product development. There must be sprint planning, but also tasks that do not need planning and can be completed as a secondary priority (as in the Kanban methodology).

6. Changes/additions to the sprint. Changes are possible as it is in the Kanban methodology.

If all tasks are completed, it is possible to complete next tasks that were not included to the planning. There may also be a replacement for any sprint tasks if it is required.

1. Roles. It is offered, as in the Scrum methodology, to split people into two types for this methodology – Product Owner, Scrum Master (Project Manager), and the development team. Each person in the team will have a clear role, but each team member can test tasks for better performance. This methodology should be very flexible for teams and take into account everyone's interests.

2. Board with tasks. The board must be changed each sprint so that there are no large number of tasks and it can be clearly seen what is being done right now in the sprint. But at the same time, key tasks can be displayed on top of each status column and can be attached to the board.

3. Organizing a board with tasks. There should be four status columns as it is in Scrum methodology – To do, In progress, In testing and Done.

4. Meetings. Scrum has a good practice of frequent meetings that help to interact in a structured way and to know how things are going in the team. In addition, sprint review meetings and retrospective meetings for discussion of the sprint are very useful. However, if a meeting is considered as unnecessary (for example, a retrospective), it may not be held as it is in the Kanban methodology.

5. Performance measurement. It would be good to use a number of tasks in the sprint to measure performance. That is because movement of the task from status «To do» to «Done» should not be taken as a unit of the measurement because the tasks can be of different sizes and complexity.

6. Tasks estimation. In order to know how many tasks can be accomplished in a sprint they always need to be estimated and some more time should be added. This principle is taken from the Scrum methodology.

Thus, the basic principles that the new methodology should have were identified. As it is possible to see from the previous points, the Scrum methodology was taken as a basis, which was supplemented by some Kanban methodology rules as well as its own rules. Next, a model will be created using Petri nets.

As a result of the study, a model for the new methodology is designed. It is designed based on two methodologies – Scrum and Kanban. The formed model is shown in Fig. 3.

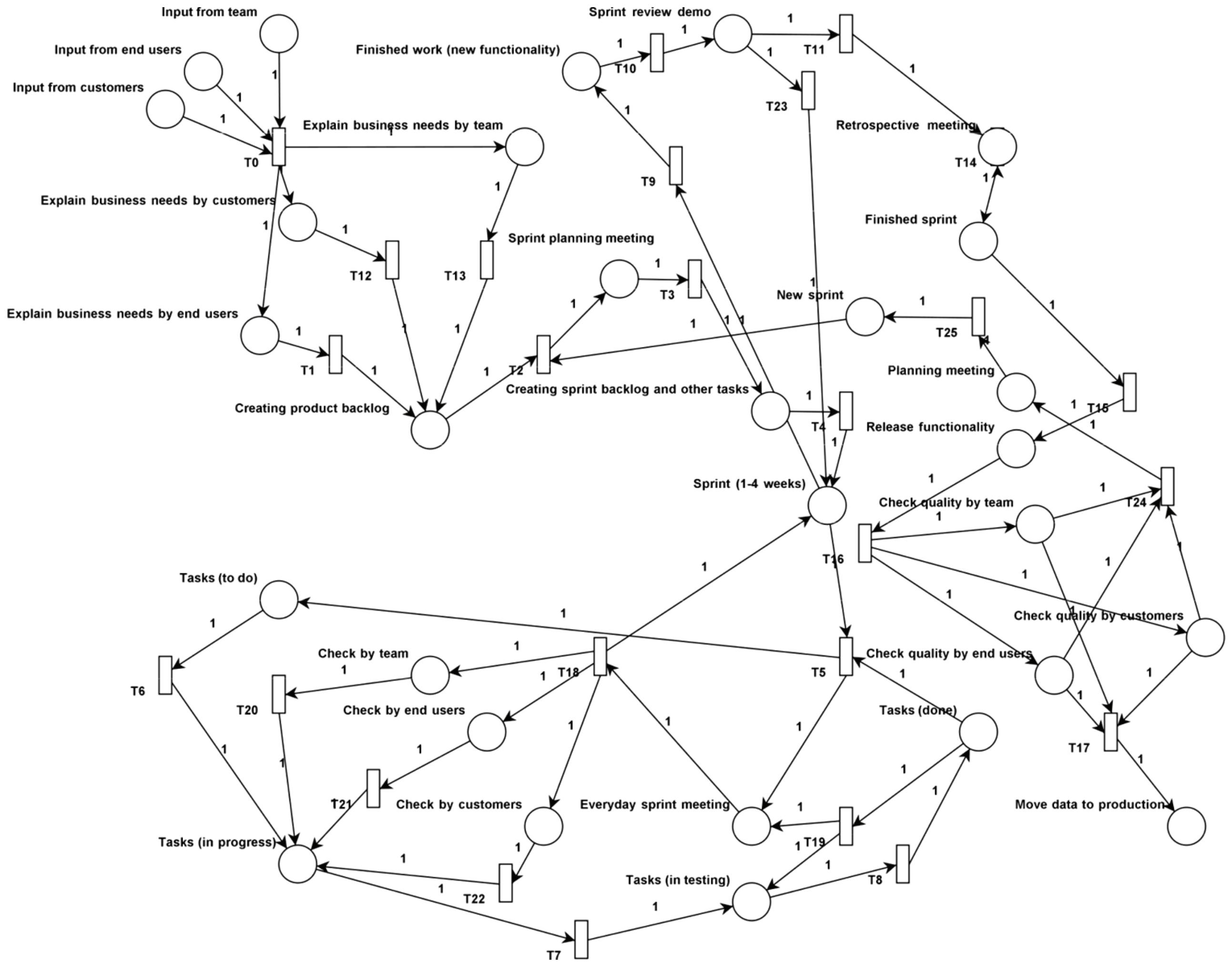

**Fig. 3.** New methodology model based on Scrum and Kanban

The basic properties of the elements of this model are shown below.

First, business goals are explained by business users, customers, team members and other people. Each person contributes to the software development at an early stage.

Then, after defining the main business goals, the backlog with prioritized requirements should be created. Due to the fact that the business goals were created with the help of all participants, the very important role of each team member can be seen.

After that, a sprint is planned, where a list of tasks for a specific sprint is created. The whole team is also involved in this process.

The sprint should last 1–4 weeks. Every day there is a meeting where the team learns what was done, what each participant is doing and what will be done, as well as whether there are any obstacles. If there is no need for a meeting, it will not take place.

Then all the tasks are done. They move from «To do» status to «In progress», «In Testing» and «Done». If the planned tasks are completed, then it is possible to do other backlog tasks as well as change requests tasks and bugs.

As the new functionality has been developed, sprint review (demo) takes place, and if it is necessary to do something afterwards, the sprint continues. If not, there is a retrospective and a sprint completion. After that, a new sprint begins where there are all the relevant stages of planning, execution and verification.

When all the sprints have been completed, a software product is released. At this stage, the work is checked and if it is necessary to do something else, a planning meeting takes place and sprint continues. If there is no need to continue working on something, then there is a release.

This model shows that it fully meets the requirements of the new methodology. It is very flexible and adaptable. This methodology is suitable for small teams and takes into account the opinion of each participant. As a result, this methodology can really be used to develop a software product.

## 7. SWOT analysis of research results

*Strengths.* The newly created methodology greatly improves existing development methodologies, it has its own set of rules and makes it easy for teams to use it because it is a flexible methodology.

*Weaknesses.* With all the adaptability and flexibility of the methodology, it should be applied correctly, depending on the type of software, the purpose of the project and the development team.

*Opportunities.* Among the new benefits is that this methodology can be used to conveniently and efficiently manage applications that are Scrum-fitting on one side and Kanban-fitting on the other one. The Scrum methodology is based on this.

*Threats.* From an objective point of view, the risks are possible if do not follow some of the rules of this methodology. For example, incorrect planning, bad meetings attendance, no work analysis.

## 8. Conclusions

1. The disadvantages of the Scrum and Kanban methodologies have been eliminated and reduced as a result of analysis, selection of methods for conducting the research and completing the research. In addition, the new methodology is suitable for different types of teams and improves their performance. Using the existing Scrum and Kanban methodologies, hypotheses are offered to create a new methodology, and these hypotheses are confirmed as a result of the experiment.

2. A set of rules is formed for the new development methodology, as Scrum and Kanban methodologies are not always suitable for project development. Therefore it is decided to add new rules. These rules are based on the advantages and disadvantages of the Scrum and Kanban methodologies.

3. A more flexible process of organizing software by setting the parameters of the combined technology is provided. It is considered to increase flexibility by structurally-parametric analysis of existing Scrum and Kanban methodologies in this software development technology.

### References

1. Granko, O. (June 23, 2017). Ekstremal'noye programmirovaniye (XP). *Blog. PM Resheniya*. Available at: https://worksection.com/blog/extreme-programming.html
2. Kratko o metodologiyakh razrabotki PO: Waterfall, Lean i Feature Driven Development. (November 30, 2017). *Blog. IT Gil'diya*. Available at: https://habr.com/ru/company/it-guild/blog/341932/
3. «Scrum. Revolyutsionnyy metod upravleniya proyektami». Kniga za 15 minut. (December 15, 2015). *Blog. MakeRight*. Available at: https://habr.com/ru/company/makeright/blog/297250/
4. Sutherland, J. (2014). *Scrum: The Art of Doing Twice the Work in Half the Time*. New York: Random House, 256.
5. Granko, O. (June 8, 2017). Scrum chi ne Scrum – yakiy pidkhid obrati. *Blog. PM Rishennia*. Available at: https://worksection.com/ua/blog/scrum.html
6. *What is a sprint*. Available at: https://www.scrum-institute.org/What_is_a_Sprint.php. Last accessed: 22.07.2016.
7. *What is Scrum*. Available at: https://www.scrum.org/resources/what-is-scrum. Last accessed: 01.08.2016.
8. *Daily Scrum Meeting*. Available at: https://www.mountaingoatsoftware.com/agile/scrum/meetings/daily-scrum. Last accessed: 03.08.2016.
9. *Sprint demo*. Available at: https://innolution.com/resources/glossary/sprint-demo. Last accessed: 10.08.2016.
10. *What is a sprint retrospective*. Available at: https://www.scrum.org/resources/what-is-a-sprint-retrospective. Last accessed: 11.08.2016.
11. Kanban (rozrobka). *Vikipediia*. Available at: https://uk.wikipedia.org/wiki/%D0%9A%D0%B0%D0%BD%D0%B1%D0%B0%D0%BD_(%D1%80%D0%BE%D0%B7%D1%80%D0%BE%D0%B1%D0%BA%D0%B0). Last accessed: 12.08.2016.
12. *Kanban Encyclopedia: Concepts and Terms*. Available at: https://kanbanize.com/kanban-resources/getting-started/kanban-encyclopedia/. Last accessed: 12.08.2016.
13. Razbirayemsya v Scrum i Kanban. (December 16, 2016). *Netologiya*. Available at: https://netology.ru/blog/scrum-kanban
14. *What is Kanban board*. Available at: https://kanbanize.com/kanban-resources/getting-started/what-is-kanban-board. Last accessed: 22.08.2016.
15. *Kanban Explained for Beginners*. Available at: https://kanbanize.com/kanban-resources/getting-started/what-is-kanban. Last accessed: 26.08.2016.
16. Merezhi Petri – instrument dlia modeliuvannia dinamichnikh sistem v yekonomitsí. (15.03.2017). *Naukova spilnota*. Available at: http://www.spilnota.net.ua/ru/article/id-1759/
17. *Prosti merezhi Petri*. Available at: https://studopedia.su/6_41767_prosti-merezhi-petri.html. Last accessed: 02.09.2016.

18. *Seti Petri. Struktura i pravila vypolneniya setey Petri.* Available at: https://itmodeling.fandom.com/ru/wiki/%D0%A1%D0%B5%D1%82%D0%B8_%D0%9F%D0%B5%D1%82%D1%80%D0%B8._%D0%A1%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%B0_%D0%B8_%D0%BF%D1%80%D0%B0%D0%B2%D0%B8%D0%BB%D0%B0_%D0%B2%D1%8B%D0%BF%D0%BE%D0%BB%D0%BD%D0%B5%D0%BD%D0%B8%D1%8F_%D1%81%D0%B5%D1%82%D0%B5%D0%B9_%D0%9F%D0%B5%D1%82%D1%80%D0%B8. Last accessed: 07.09.2016.

19. *Merezhi Petri.* Available at: https://stud.com.ua/98834/informatika/merezhi_petri. Last accessed: 10.09.2016.

***Popova Olena,*** *Department of Software Engineering, The Bohdan Khmelnytsky National University of Cherkasy, Ukraine, e-mail: olena.popova2016@gmail.com, ORCID: https://orcid.org/0000-0002-8838-8219*

UDC 336.74, 338.1, 004.02
DOI: 10.15587/2312-8372.2019.180552

**Kruhlov A.,**
**Pyroh M.**

# ANALYSIS OF PROBLEMS OF FORECASTING OF FINANCIAL INSTRUMENTS IN STOCK MARKETS

*Об'єктом дослідження є процеси прогнозування фінансових інструментів на фондових ринках в умовах невизначеності. Високий ступінь невизначеності на фондових ринках значно ускладнює процес прогнозування динаміки фінансових інструментів. Дана проблема має значення, як для держав, так і для інвестиційних компаній. А також для інших учасників ринку, яким необхідно приймати довгострокові інвестиційні рішення, засновані на превентивних заходах щодо зниження впливу ризиків фінансових криз на їх діяльність. У даній роботі автори аналізують ряд прогностичних моделей, які застосовуються в сфері розрахунків числових рядів. В контексті прогнозування цін на фондових ринках виявлені сильні і слабкі сторони популярних на практиці моделей. Наведено їх математичні функції, пояснені алгоритми розрахунку та дані авторські висновки про ступінь ефективності застосування окремих моделей в сфері фінансових інструментів.*

*В ході дослідження автори вивчили ряд різних наукових праць з даної проблеми і провели аналіз отриманих відомостей. Отриманий результат аналізу показав, що процеси прийняття рішень при прогнозуванні змін фінансових інструментів будуть ускладнені наявністю зовнішніх чинників, але також ці зовнішні чинники є результатом діяльності окремих учасників ринку. Це пов'язано з тим, що при прогнозуванні фінансових інструментів на фондових ринках можна нівелювати псевдо-випадкові події зовнішнього середовища. Багато існуючих рішень прогнозування допускають низьку точність при моделюванні прогнозу, тому раціональніше використовувати мультиагентні технології. Завдяки ним забезпечується більша точність показників, в порівнянні з аналогічними методами, такими як економетричні моделі (найбільш відомі з них : ARCH, GARCH, VAR).*

*Отримані у роботі результати досліджень можна використовувати для прогнозування фінансових криз, а також для розробки методів протидії ним.*

**Ключові слова:** *моделі прогнозування, фондовий ринок, ланцюг Маркова, довгострокові інвестиційні рішення, мультиагентні технології.*

*Received date: 22.05.2019*
*Accepted date: 28.06.2019*
*Published date: 30.08.2019*

## 1. Introduction

Thanks to the rapid development of information technologies, it became possible in a matter of seconds to analyze a large amount of information, build complex mathematical models, and solve multicriteria optimization problems. Scientists involved in the cyclical development of the economy began to develop theories, believing that tracking trends in a number of economic variables would clarify and predict periods of ups and downs. One of the objects for study is the stock market. Multiple attempts have been made to build such a mathematical model that would successfully solve the problem of forecasting the price of financial instruments [1]. In particular, «technical analysis» has become widespread. The relevance of this study lies in the fact that at the moment, with the pace of development of the world economy, stock and foreign exchange markets, many forecasting methods have appeared. In this work, the author identifies the most accurate of them on the basis of a number of studied works and constructed analytical models.

## 2. The object of research and its technological audit

*The object of research* is the forecasting processes of financial instruments in stock markets in the face of uncertainty. A high degree of uncertainty in the stock markets