SZKOŁA GŁÓWNA SŁUŻBY POŻARNICZEJ

TECHNOLOGICZNE, TECHNICZNE I STRATEGICZNE INNOWACJE W RATOWNICTWIE

Redakcja naukowa

Monika Wojakowska
Radosław Fellner
Piec Robert

Warszawa 2023

REDAKTORZY PROWADZĄCY
Ewa Juchimowicz
Radosław Fellner

KOREKTA JĘZYKOWA
Do uzupełnienia

SKŁAD, ŁAMANIE I PROJEKT
OKŁADKI J

Wydanie pierwsze

Warszawa 2023
ISBN ....

WYDAWCA
Szkoła Główna Służby Pożarniczej
01-629 Warszawa
e-mail: wydawnictwo@sgsp.edu.pl
www.sgsp.edu.pl
tel. 22 561 73 83

DRUK
Mazowieckie Centrum Poligrafii
Objętość publikacji: Do uzupełnienia arkuszy wydawniczych

Monografia powstała w ramach Projektu dofinansowanego przez Ministra Edukacji i Nauki ze środków budżetu państwa w ramach programu „Doskonała Nauka”, nr umowy DNK/SP/545240/2022.

THE MAIN SCHOOL OF FIRE SERVICE

TECHNOLOGICAL, TECHNICAL AND STRATEGIC INNOVATIONS IN RESCUE

Editors

Monika Wojakowska
Radosław Fellner
Piec Robert

Warsaw, 2023

Project is co-financed by the Ministry for Education and Science from the national fundings under programmee „Doskonała Nauka", agreement NO: DNK/SP/545240/2022

Spis treści / Table of content

WSTĘP **5**

ROZDZIAŁ 1. Koło Naukowe Ratownictwa Technicznego – w poszukiwaniu innowacyjnych rozwiązań w ratownictwie .... Błąd! Nie zdefiniowano zakładki.
Weronika Madej, Łukasz Szymczyk, Wojciech Kubica

ROZDZIAŁ 2. Inżynierskie rozwiązania stosowane w analizie rozwoju pożaru i planowaniu ewakuacji .... Błąd! Nie zdefiniowano zakładki.
Michał Mikke, Joanna Bąk, Karol Ziębiński, Dominik Strzelczyk, Łukasz Drążek, Adam Romas, Kamil Gąsior, Jakub Rudo

ROZDZIAŁ 3. Zastosowanie bezzałogowych statków powietrznych podczas pożarów wewnętrznych .... Błąd! Nie zdefiniowano zakładki.
Oskar Olejarski

ROZDZIAŁ 4. Wykorzystanie algorytmów komunikacyjnych podczas Medycznych Działań Ratowniczych w Państwowej Straży PożarnejBłąd! Nie zdefiniowano zakładki.
Jan Tymiński

ROZDZIAŁ 5. Organizacja edukacji dla bezpieczeństwa w Polsce do roku 2020 r. - perspektywy rozwoju **77**
Marcin Wiechetek

ROZDZIAŁ 6. Legia Akademicka jako przykład prewencyjnego podejścia do gotowości obronnej obywateli RP **87**
Karolina Stankiewicz

ROZDZIAŁ 7.Strategia przygotowań do działań w warunkach zagrożenia czynnikami CBRN w kontekście działań realizowanych przez PSP w trakcie pandemii COVID-19 .... **99**
Grzegorz Bugaj

ROZDZIAŁ 8. Probabilistyczna ocena bezpieczeństwa konstrukcji w warunkach pożaru **112**
Wojciech Kowalski

ROZDZIAŁ 9. Pomoc na rzecz Ukrainy dzięki współpracy OSP Józefów i SGSP....... 123
Paweł Florek

ROZDZIAŁ 10. Humanitarian aid of the Republic of Poland to Ukraine during Russian aggression in 2022 149
Oleksandra Pekarska, Andrii Havrys

ROZDZIAŁ 11. Information system design for the formation of project teams in safety-oriented systems.. 161
Oleh Kovalchuk

ROZDZIAŁ 12. The network graph traversal method for solving the problem of short-term planning of safety-oriented services development...................................... 172
Yuliia Kordunova, Oleksandr Prydatko, Olga Smotr, Igor Kokotko

ROZDZIAŁ 13. Use of the cavitation effect for rescue operations 182
Andrii Dominik, Yurii Nahirniak, Dmytro Rudenko

ROZDZIAŁ 14. Fire protection of wood using metal-coordinated epoxy-amine compositions……………………………………………………………………………………………………191
Parkhomenko V.-P.O., Lavrenyuk H.O., Myhalichko B.M

THE NETWORK GRAPH TRAVERSAL METHOD FOR SOLVING THE PROBLEM OF SHORT-TERM PLANNING OF SAFETY-ORIENTED SERVICES DEVELOPMENT.

Senior lieutenant, PhD student Yuliia Kordunova
Lviv State University of Life Safety
kordunovayulia@gmail.com

Lieutenant colonel, Ph.D, associate professor Oleksandr Prydatko,
Lviv State University of Life Safety
o_prydatko@ukr.net

Colonel, Ph.D, associate professor Olga Smotr
Lviv State University of Life Safety
olgasmotr@gmail.com

PhD student Igor Kokotko
Lviv State University of Life Safety
igorkokotko@gmail.com

Introduction

It is known that estimating the software development duration in dynamic conditions is an extremely uncertain, but very important stage in the process of developing new software. Changing requirements, limited time and material resources, risk and quality management – all these conditions have an impact on the timing of new software implementation and release. Today, there are two classic approaches (methodologies) in managing the software development process – flexible and cascade, each of them has its own principles and limitations. In particular, in flexible management, the development team and the execution time are fixed, and the requirements are variable. In cascade – on the contrary, a clear list of works (requirements for the product) is fixed, and the team and time can change.

However, during the development of safety-oriented services, which are developing by the employees of the State Emergency Service of Ukraine, was revealed a problem of inconsistency and ineffectiveness of these methods in the management of the specialized software. As in this case the dynamism is characterized not only by the scope and content of the works, but also time resource. That is why there is a need to develop new methods and approaches to estimate the duration of the safety-oriented services development, adapted to the specifics of the State Emergency Service of Ukraine and correlated with the principles of flexible software life cycle management methodology.

Literature Review

As of today, there is a large number of scientific works in which the issue of short-term planning of the software development process was raised. In particular, work [1] proved that planning occupies a central place in Agile projects, and all other areas revolve around it. Works [2, 3, 4] are aimed at using an expert method of estimating time resources according to the Agile methodology. In the article [5], it is proposed to use the block diagram tool for decision-making in interdisciplinary research collaboration, and in the work [6], the regression equation was improved for estimating the complexity of software development created according to the flexible methodology. The work [7] describes user story estimation based on the complexity decomposition using Bayesian networks.

There are also a number of works [8 – 10] devoted to the network graphs use in the process of project planning. In particular, the authors in work [9] chose a method for solving the planning problem based on the application of the network planning method, which is based on the idea of optimizing the critical path with the involvement of additional limited funds. This approach does not correlate with the development of specialized software. The article [8] describes the process of developing a computer program for solving network optimization problems, but the shortest paths, in this case, are calculated using Dijkstra's algorithm, which is not universal and does not work for network planning of the software development process. The article [10] is based on PERT network planning methods, the use of elements of graph theory and the Gantt chart method. This approach is relevant to the cascade management model, which is characterized by the sequential execution of tasks and a defined execution time.

Materials and Methods

Based on the conducted literature review, it can be concluded that, as of today, there are a large number of different methods for estimating the duration of software development, but none of them correlates with the development of safety-oriented services. The main goal of the work is to develop an adaptive method of short-term planning of the life cycle of safety-oriented software systems in a dynamic environment.

To achieve the goal, it was decided to use the mathematical apparatus for modeling dynamic discrete systems: Petri nets and the mathematical apparatus of graph theory: network planning.

First of all it is need to say that the software development planning process consists of defining product requirements in the user stories form. In the Agile methodology, it is not customary to break down requirements into technical tasks, because this approach does not allow a holistic view of the implementation of certain functions of the program. A user story is a short and simple description of a product's features from the perspective of a user who is looking for new opportunities. It includes the entire spectrum of development: from design to testing. This approach will allow the entire team to participate in the process of evaluating and developing this functionality. Once defined, the user story is broken down into small tasks (functions) that will be performed directly by different team members. Based on it, it is clear that there is a high probability that several jobs can be performed in parallel (which will allow significant time savings), and there may also be jobs whose execution depends on previous ones. That is why determining the critical path that will take the most time to complete is extremely important. But first of all, there is a need to develop a unified algorithm for network graph construction. Its result will help to find its main parameters much faster and with minimum mistakes. A graphical representation of the software development process is shown in Figure 1.

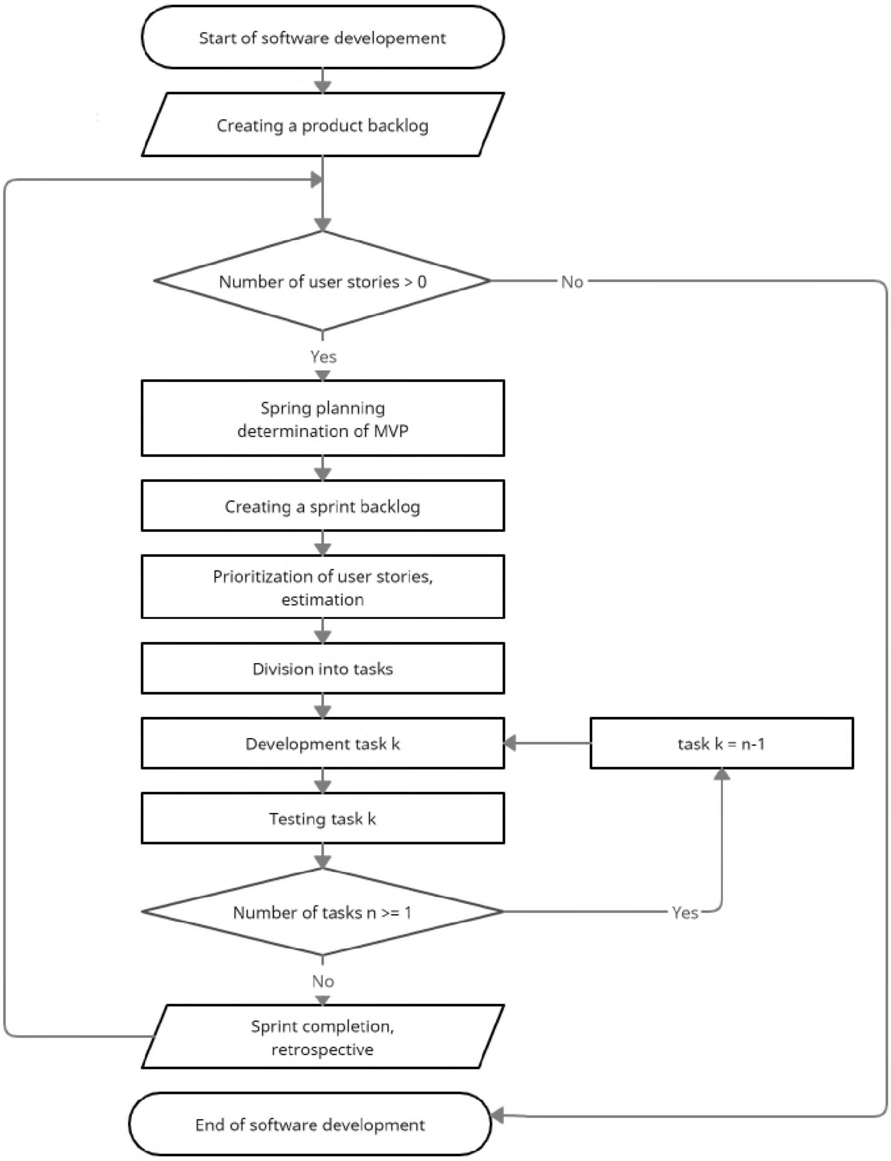

Figure 1. The process of software development planning.

From this block diagram, we can conclude that the software development process in dynamic conditions is a highly loaded distributed parallel system. To form a clear idea about the organizational and technical processes that arise in such systems, it is necessary to conduct simulations of parallel processes. An excellent toolkit for performing such tasks is the Petri net modeling apparatus. The Petri net allows you to display certain states of the modeled system, where the transition between different states of the system requires mandatory fulfillment of conditions (they are laid in the network in advance).

Let us present the process of developing specialized software systems as a discrete system, the transitions between states of which are the implementation of individual parts of the software product functionality (Figure 2).

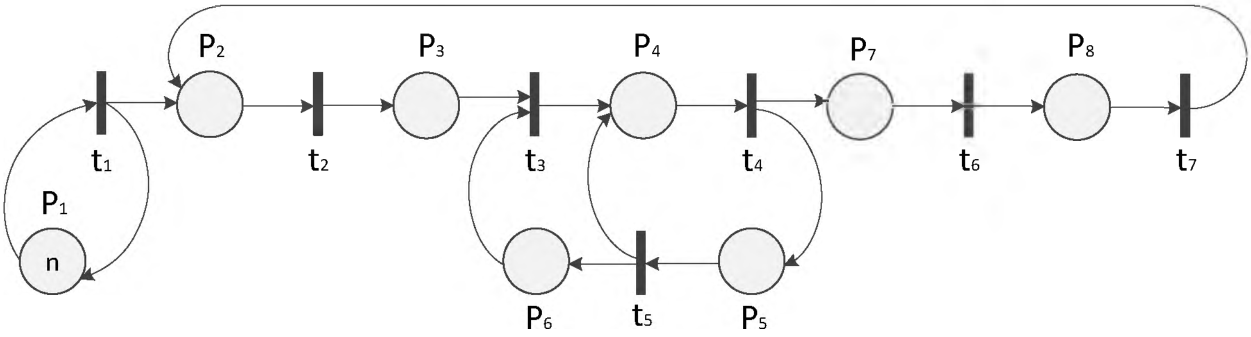

Figure 2. Petri model of the safety-oriented software systems development process (scaled to a sprint)

Let's describe the model presented in Figure 2. Position P1 involves the formation of a product backlog, which includes a list of all tasks for the successful project implementation. Transition t1 indicates that the development team has received a defined list of tasks from the backlog. Position P2 involves the planning of a separate sprint, the condition for which is the selected tasks for sprint t2. The P3 position is focused on the sprint backlog formation and the task distribution among the members team. Successfully distributed tasks are positioned by transition t3. The direct implementation of sprint tasks related to the development of the software system is implemented within the scope of position P4. Completed tasks within transition t4 may have branching for testing P5 and sprint review P7. The tested fragment of the t5 system, provided that the tasks are performed well and there are no violations of the requirements, returns to position P4. If a discrepancy is detected, the system model provides for a transition to position P6, the condition of which is to correct errors and finalize requirements. The sprint review position P7 ends with the successful acceptance of completed tasks t6, and after the completion of the P8 retrospective, the completion of the sprint is fixed by transition t7. After fulfilling the described list of conditions and transitions, the discrete-cyclical system provides planning and the start of a new sprint with a list of new tasks in position P2. The scope and content of tasks are selected as a result of the activation of position P1 and transition t1, which also have the characteristic of cyclicity to determine the new scope of work per sprint.

To make a study changes in various states of the system (described by the network) is possible by combining matrices. The first "positive" matrix indicates exactly which processes are derived after the execution of a certain event (transition). Such processes precede the next position - the condition for the transition.

$$R^{+} = \begin{pmatrix} & t_1 & t_2 & t_3 & t_4 & t_5 & t_6 & t_7 \\ P_1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ P_2 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ P_3 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ P_4 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ P_5 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ P_6 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ P_7 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ P_8 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}. \quad (1)$$

The second "negative" matrix indicates which condition (position) precedes the specified transition.

$$R^- = \begin{pmatrix} & t_1 & t_2 & t_3 & t_4 & t_5 & t_6 & t_7 \\ P_1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ P_2 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ P_3 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ P_4 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ P_5 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ P_6 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ P_7 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ P_8 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \quad (2)$$

For further modeling of system states, it is necessary to find the difference of matrices. The matrix R is constructed by subtracting the matrix R^- from R^+.

The result of the matrix difference is represented by the expression:

$$R = \begin{pmatrix} & t_1 & t_2 & t_3 & t_4 & t_5 & t_6 & t_7 \\ P_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ P_2 & 1 & -1 & 0 & 0 & 0 & 0 & 1 \\ P_3 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ P_4 & 0 & 0 & 1 & -1 & 1 & 0 & 0 \\ P_5 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ P_6 & 0 & 0 & -1 & 0 & 1 & 0 & 0 \\ P_7 & 0 & 0 & 0 & 1 & 0 & -1 & 0 \\ P_8 & -1 & 0 & 0 & 0 & 0 & 1 & -1 \end{pmatrix}. \quad (3)$$

Matrix (3) provides the basis for further studies of system performance in its various states. For this, it is necessary to determine the product of the matrix and the vector, which describes the state of the system in a certain period of time. However, when modeling the system's performance, the conditions for performing parallel transitions should be taken into account. Auxiliary tools for this (except for Petri nets) are the conceptual apparatus of graph theory and their algorithmic representation (which will be displayed later). The Petri net itself allows us to imagine the general sequence of processes and their interrelationships, which is actually the subject of further research.

The mathematical apparatus of graph theory allows one to easily model and interpret the necessary processes (events or phenomena), to implement their visual representation without the need for physical design. In particular, the adapted method of calculating the parameters of the network schedule will allow quick in a dynamic environment to ensure the re-planning of the stages of project implementation, as well as to determine the critical scope of the mandatory works. Network planning will allow monitoring of exceeding the time possible risks of work execution in a dynamic environment, as well as prioritizing the phasing of their execution.

As in network planning, the main elements of the network model are works and events, in the context of Agile planning, a work is a user story or task that the user story has been broken down into. As a unit of measurement, story points (hereinafter - s. p.) are used, which mean a relative value, in particular, a combination of the laboriousness of the development of the function, the complexity of the development, and the risk associated with it. Each user story or task must have a start and end execution, for this is used events in the network graph.

To obtain a network planning graph it is necessary:

1. Prioritize user stories and functions that need to be performed.
2. Evaluate user stories and functions in the story points
3. Identify previous user stories or functions to be performed for specific user story

At the beginning of building the network graph, number the initial event j = 1, and the previous event (i = 0) is absent. From it, draw a vector that will indicate the first user story or function and its weight in the Story Points. If several tasks can be performed in parallel, the required number of vectors is built. Since each user story must end with a final event, mark the event on the graph and increase its sequence number by 1 (do this for all parallel works, increasing j by 1). If it is needed to build a user story, which is possible only after the previous one end, look for the final event of the work that interests you on the graph, and from it, draw the vector of the desired user story. If the execution of the user story depends on the completion of several previous ones, solve this problem using fictitious work. From each previous user story draw a vector, which means a fictitious work whose weight is 0 s. p., and end them in one final event, which will be the beginning of the new user story you need. Thus, build a network planning graph for the number of user stories and functions that interest you (within the scope of the sprint) [11]. The algorithm for constructing a network graph (network model) of a certain amount of work on the sprint is shown in Figure 3.

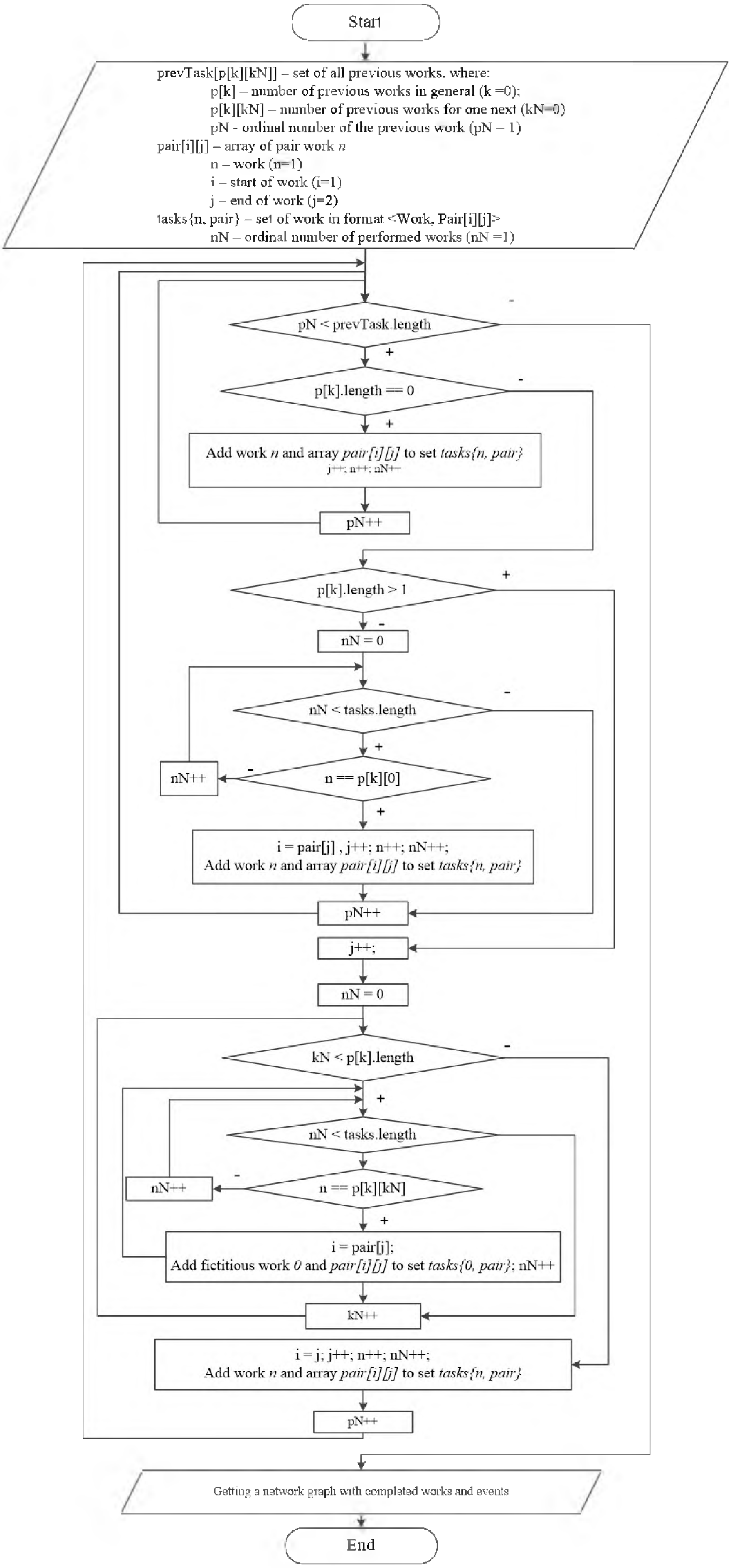

Figure 3. Algorithm for building a network graph

This algorithm will make it possible to build a mathematical set of events (which will mean the start and end of work) and work, which in turn will be the basis for determining the indicators necessary for planning the software development process (critical path of the development process, reserves of work and events). Actually, the adapted methods will be the beginning of development an information system to support decision-making, which could make operational re-planning of the project stages of implementation in a dynamic environment, determine the critical scope of mandatory work, tracking in a dynamic environment of the possible risks of overtime, prioritize the phasing of their implementation. This system is quite needed for developers in the State Emergency Service of Ukraine.

Conclusions

Based on the results of this work, it needs to conclude that the modern methods and approaches of software life cycle management do not match the development of safety-oriented services. That is because in addition to variable requirements, execution time is fundamentally important. The methods of the software development planning process and network graph construction developed by us will minimize risks and time losses in the case of changing requirements and a dynamic environment. These methods will make it possible to increase the efficiency of planning individual stages of software development and make their adjustments in real-time. In the future, on their basis, it is planned to develop an information system to support decision-making, regarding the management of the life cycle of the security-oriented systems development in dynamic conditions, which are relevant for the State Emergency Service of Ukraine.

References

1. S. Boral, Domain V: Adaptive Planning, “Ace the PMI-ACP” 2016, pp. 201 - 261
2. V. Lenarduzzi et.al., Functional size measures and effort estimation in agile development “A replicated study” 2015, No. 212 pp. 105 – 116.
3. V. Sachdeva, Requirements prioritization in agile:Use of planning poker for maximizing return on investment, “Advances in Intelligent Systems and Computing” 2017, pp. 403–409.
4. E. Kula et.al., Factors Affecting On-Time Delivery in Large-Scale Agile Software Development, “IEEE Transactions on Software Engineering” 2022, No. 48 (9), pp. 3573-3592.
5. U. Jansen, W. Schulz, Flowchart tool for decision making in interdisciplinary research cooperation, “International Conference on Digital Human Modeling and Applications in Health, Safety, Ergonomics and Risk Management” 2017, pp. 259– 269.
6. L. Latanska, M. Balashov, Improvement of the regression equation for estimating the labor intensity of software development created according to the flexible methodology, “Materials of the 11th International Scientific and Practical Conference «Free and Open Source Software»”, Kharkiv 2019, p. 100.
7. M. Dur´an et al., User story estimation based on the complexity decomposition using bayesian networks “Programming and Computer Software” 2020, No. 46, pp. 569–583.
8. H. Dymova, O. Larchenko, Development of a computer program for solving network optimization problems “Computer-integrated technologies: education, science, production” 2020, No. 41, pp. 143–151

9. O. Seidykh, V. Chobanu, Optomozation of the network graphics of the complex of works, “Modern engineering and innovative technologi” 2018, No. 1(3), pp. 61–67

10. S. Velykodniy, Z. Burlachenko, S. Zaitseva-Velykodna, Architecture development of software for managing network planning of software project reengineering, “Innovative technologies and scientific solutions for industries” 2019, No. 2(8), pp. 25–35

11. Y. Kordunova et al., Expert Decision Support System Modeling in Lifecycle Management of Specialized Software, “Lecture Notes in Data Engineering, Computational Intelligence, and Decision Making” 2023, No. 149, pp. 367 – 383

12. Y. Kordunova et al., Analysis of the traditional and flexible approaches to creating software in dynamic conditions, “Management of Development of Complex Systems” 2021, pp. 71–77.

13. M. Belkasmi et al., Global IT project management: an agile planning assistance, “Advances in Smart Technologies Applications and Case Studies” 2020, pp. 575–582.

14. S. Liaskovska et al., Information technology of process modeling in the multiparameter systems “IEEE Second International Conference on Data Stream Mining & Processing (DSMP)” 2018, pp. 177–182.

15. Y. Martyn et al., Software for shelter’s fire safety and comfort levels evaluation, “Communications in Computer and Information Science” 2020, pp. 457–469.

16. K. Schwaber, J. Sutherland, The scrum guide “Scrum Alliance” 2020, p. 13.

17. V. Sachdeva, Requirements prioritization in agile: Use of planning poker for maximizing return on investment. “Advances in Intelligent Systems and Computing” 2017, pp. 403–409.

18. M. Bertling et al., The minimal viable production system (mvps) – an approach for agile (automotive) factory planning in a disruptive environment, “Advances in Production Research” 2019, pp. 24–33.

19. U. Jansen, W. Schulz, Flowchart tool for decision making in interdisciplinary research cooperation, “International Conference on Digital Human Modeling and Applications in Health, Safety, Ergonomics and Risk Management” 2017, pp. 259– 269.