№ 6(47) 2025
Наука і техніка сьогодні
серії: право, економіка, педагогіка, техніка, фізико-математичні науки

UDC 004.738.5:005.591

https://doi.org/10.52058/2786-6025-2025-6(47)-889-903

Liakh Ihor Mykhailovych Doctor of Technical Sciences, Associated Professor, Professor of the Department of Informatics and Physical-Mathematical Disciplines, Uzhhorod National University, Uzhhorod, tel.: (050)372-78-57, https://orcid.org/0000-0001-5417-9403

Kish Yurii Viktorovych assistant of the Department of Informative and Operating Systems and Technologies, Uzhhorod National University, Uzhhorod, tel.: (050)693-22-27, https://orcid.org/0009-0000-6167-0129

MODELS AND METHODS OF DECISION SUPPORT FOR SOFTWARE DEVELOPMENT RISK ASSESSMENT

Abstract. In the modern software development environment, risk management is a critical success factor for projects, especially under conditions of Global Software Development (GSD), where time zone differences, cultural diversity, and communication barriers exist. Traditional risk management approaches, focused on expert assessments or simple probabilistic models, which are insufficient to fully account for the dynamics and complexity of contemporary projects. In response to these challenges, this study proposes the ways of an adaptive Decision Support System (DSS) that integrates modern Artificial Intelligence (AI) approaches, fuzzy logic, machine learning, Bayesian networks, and other methods for both qualitative and quantitative risk analysis.

The study is based on an analysis of recent publications and examines the advantages and limitations of separate risk management models, including risk matrices, Bayesian networks, Monte Carlo simulation, fuzzy logic models, and machine learning methods, with a focus on their applicability in different project management contexts.

The proposed Decision Support System (DSS), which incorporates a knowledge base, a fuzzy inference engine, and a graphical user interface, has enabled the identification of directions for further improvement to support managerial decision-making in complex project environments.

The result of the study is the architecture of a Decision Support System (DSS), which is capable of effectively identifying, assessing, and mitigating risks within complex, dynamic, and distributed teams. The proposed approach is intended to enhance the accuracy of risk prediction, improve the justification of managerial decisions, and contribute to the development of more resilient and productive practices in the field of software development. Suggested architecture provides support for decision-making regarding the formation of an effective team configuration, taking into account risks that impact the execution of a project.

Keywords: decision support system, risks, machine learning, Bayesian networks, Monte Carlo simulation

Лях Ігор Михайлович доктор технічних наук, доцент, професор кафедри інформатики та фізико-математичних дисциплін, Державний вищий навчальний заклад «Ужгородський національний університет», м. Ужгород, тел.: (050)372-78-57, https://orcid.org/0000-0001-5417-9403

Кіш Юрій Вікторович асистент кафедри інформаційних управляючих систем та технологій, Державний вищий навчальний заклад «Ужгородський національний університет», м. Ужгород, тел.: (050)693-22-27, https://orcid.org/0009-0000-6167-0129

МОДЕЛІ ТА МЕТОДИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ДЛЯ УПРАВЛІННЯ РИЗИКАМИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Анотація. У сучасному середовищі розробки програмного забезпечення (ПЗ) управління ризиками є критичним чинником успіху проєктів, особливо в умовах глобальної розподіленої розробки (GSD), де існують часові, культурні та комунікаційні бар'єри. Традиційні підходи до управління ризиками, зосереджені на експертних оцінках або простих ймовірнісних моделях, які не здатні повною мірою врахувати динаміку та складність сучасних проєктів. У відповідь на ці виклики, у дослідженні запропоновано підходи для створення адаптивної системи підтримки прийняття рішень (СППР), яка поєднує сучасні підходи штучного інтелекту (ШІ), нечітку логіку, машинне навчання, байєсівські мережі та інші методи для якісного й кількісного аналізу ризиків.

Дослідження базується на аналізі сучасних публікацій і містить огляд переваг та обмежень окремих моделей управління ризиками, зокрема матриці ризиків, байєсівських мереж, симуляції Монте-Карло, моделей нечіткої логіки та методів машинного навчання, акцентуючи увагу на їх застосовності в різних контекстах управління проєктами. Запропонована СППР, яка включає базу знань, механізм нечіткого виводу та графічного інтерфейсу користувача дозволила визначити напрями подальшого вдосконалення для підтримки управлінських рішень у складних проєктних середовищах.

Результатом дослідження стала архітектура СППР, яка здатна ефективно ідентифікувати, оцінювати й мінімізувати ризики в складних, динамічних і розподілених командах. Запропонований підхід покликаний підвищити точність прогнозування ризиків, покращити обґрунтованість управлінських рішень та сприяти формуванню більш стійких і продуктивних практик у сфері розробки ПЗ. Запропонована архітектура забезпечує підтримку прийняття рішень щодо

формування ефективної конфігурації команди, враховуючи ризики, що впливають на виконання проєкту.

Ключові слова: система підтримки прийняття рішень, ризики, машинне навчання, байєсівські мережі, симуляція Монте-Карло

Problem Statement. Software development is a complex and dynamic process that involves the interaction of numerous stakeholders, constantly evolving requirements, and a wide range of uncertainties. One of the key challenges associated with software development is risk management, as risks may arise at various stages of the software product lifecycle. These include budget overruns, schedule delays, reduced quality, technological challenges, and resource constraints. If these risks are not properly identified, assessed, and mitigated, they may lead to the violation of the project implementation schedule, significant financial losses, and reputational damage.

Successful software project management requires a systematic approach to risk assessment. Traditional methods largely rely on expert judgment, which is often insufficient in the context of newest complex software environments. With the adoption of agile methodologies, cloud computing, and AI-driven development, new risk factors continually emerge and evolve [19]. This is especially relevant in the context of Global Software Development (GSD), where additional challenges arise from time zone differences, cultural barriers, distributed teams, and heterogeneous infrastructure.

In this context, the study of models and methods for decision support in software development risk assessment becomes critically important. Decision Support Systems (DSS) enable the integration of both qualitative and quantitative approaches for identifying, evaluating, and mitigating risks. Among the most notable methods worth mentioning are statistical modeling, machine learning algorithms, fuzzy logic, Bayesian networks, and multi-criteria decision-making techniques. The integration of these tools allows organizations to more accurately predict potential risks, assess their impact, and implement appropriate mitigation strategies.

A particular difficulty lies in developing a DSS model that would be simultaneously adaptive to a changing environment, sensitive to uncertainties, and suitable for scaling. Given that traditional methodologies can no longer fully address the entire spectrum of risks, there is a growing need to implement innovative approaches that combine deep analytics with technological advancements in the field of AI.

The goal of this study is to develop and implement an effective risk prediction model for software development using modern artificial intelligence tools, taking into account the specific characteristics of GSD and providing highly accurate decision support. The expected outcomes of the research include improved risk management practices, enhanced reliability and efficiency of software projects, and a deeper understanding of the capabilities and limitations of AI applications in complex and dynamic software development environments.

Analysis of recent research and publications. One of the pressing issues in modern risk management within global software development is the lack of unified approaches to automated decision support.

In the work by Gupta and Muni, the use of neural networks for dynamic risk monitoring is examined [1]. While the proposed model demonstrates accuracy in identifying potential project delays, it does not take into account the impact of the human factor, which remains critically important in GSD environments.

In the study by Singh and Lee, a DSS-based system using multi-criteria analysis was proposed to assess risks associated with the geographical distribution of teams [2]. The authors claim that the application of the model reduced the risk of misunderstandings in a group of people by 23%; however, the results were obtained in a controlled environment without considering external integration tools (Slack, GitHub, etc.), which are widely used in real-world development teams.

The use of graphical models specifically Bayesian networks for risk assessment is demonstrated in the work by Lo and et alii [3]. A model was constructed to represent the relationships between risk types and technological factors. The authors prove that this approach enables the identification of nodal risks 15–20% earlier than traditional methods. However, the study did not account for shifting priorities throughout the project lifecycle, which are typical in Agile methodologies [7].

To test the flexibility of DSS approaches, Chow et alii. developed a simulation environment based on a simple scripting framework in which various fuzzy logic–based models were compared [4].

Despite the high adaptability of the proposed algorithm, its performance was significantly lower when handling large volumes of input data, as confirmed by testing on more than 1,000 cases.

Thus, despite active research in the field of DSS for GSD, most solutions have limitations–ranging from narrow applicability to insufficient empirical validation. This creates a need for models, which are capable of flexible integration into real software development processes, dynamically responding to changes, and scaling according to team structure and technical stack.

The goal of the article. The ability to effectively identify, assess, and mitigate risks that play a critical role in the success of software projects. However, current risk management practices often based on simplified or static models that do not account for the dynamic nature of software engineering. Most existing approaches depend on expert judgments or basic probabilistic models, which limits the accuracy of risk assessment in real-world conditions.

This issue becomes particularly relevant in the context of agile methodologies, DevOps practices, and distributed development, where risks are multidimensional, interdependent, and constantly changing. The lack of standardized DSS that integrate both quantitative and qualitative risk factors reduces the effectiveness of managerial decisions.

As a result, risk management is often carried out using certain strategies that quickly lose relevance amid rapid changes in technology, market conditions, and regulatory frameworks.

To ensure greater resilience of software systems to risks, it is necessary to develop models that combine real-time analytics, adaptive learning algorithms, and predictive analysis capabilities [5]. This will enable the creation of DSS, which is capable of supporting well-founded decisions in conditions of uncertainty and complexity within the project environment.

To achieve this goal, the following tasks are solved:

1. Analyze current approaches to risk assessment and management in software development, including both traditional and intelligent methods.

2. Investigate the application of fuzzy logic, Bayesian networks, Monte Carlo simulation, machine learning, and the Analytic Hierarchy Process (AHP) for risk modeling.

3. Develop a prototype architecture that includes a knowledge base, a fuzzy inference engine, a visualization interface, and integration with project data sources.

Main Content. In the rapidly changing field of software development, risk assessment plays a key role in ensuring project success. Software development projects face numerous uncertainties, including budget constraints, evolving requirements, and technological challenges. Decision Support Systems (DSS) provide structured methodologies and computational tools for risk analysis.

Decision Support Systems (DSS) are complex computational tools designed to assist managers in decision-making by providing structured analytical models, deep data analysis, and predictive simulations. They facilitate informed decision-making by processing large volumes of information, identifying patterns, and offering practical recommendations. In the context of risk assessment in software development, DSS integrate multiple techniques, including statistical modeling, to identify, evaluate, and mitigate potential risks. The use of such approaches enables project managers and developers to anticipate issues, assess the probability and impact of various risk factors, and implement proactive mitigation strategies.

Moreover, DSS enhance adaptability in project management by enabling teams to model various scenarios and analyze potential outcomes before making critical decisions. This allows organizations to respond dynamically to changing project conditions, unexpected technical challenges, or evolving market demands [6].

Several models have been developed to support risk assessment in software engineering, helping to identify, quantitatively evaluate, and mitigate risks. One of the most common is the risk matrix model–a qualitative approach that classifies risks based on their possibility of occurrence and impact. Risks are represented in a matrix, enabling the prioritization of mitigation measures. This model is simple and intuitive but may lack sufficient accuracy in complex cases [8].

The risk matrix model is typically represented by two axes: probability of their appearing (ranging from low to high), which reflects the possibility of a risk occurring, and impact (also ranging from low to high), which determines the severity of consequences if the risk materializes. This approach helps allocate resources effectively by prioritizing the most critical risks first and ensuring systematic risk management as depicted in Fig. 1.

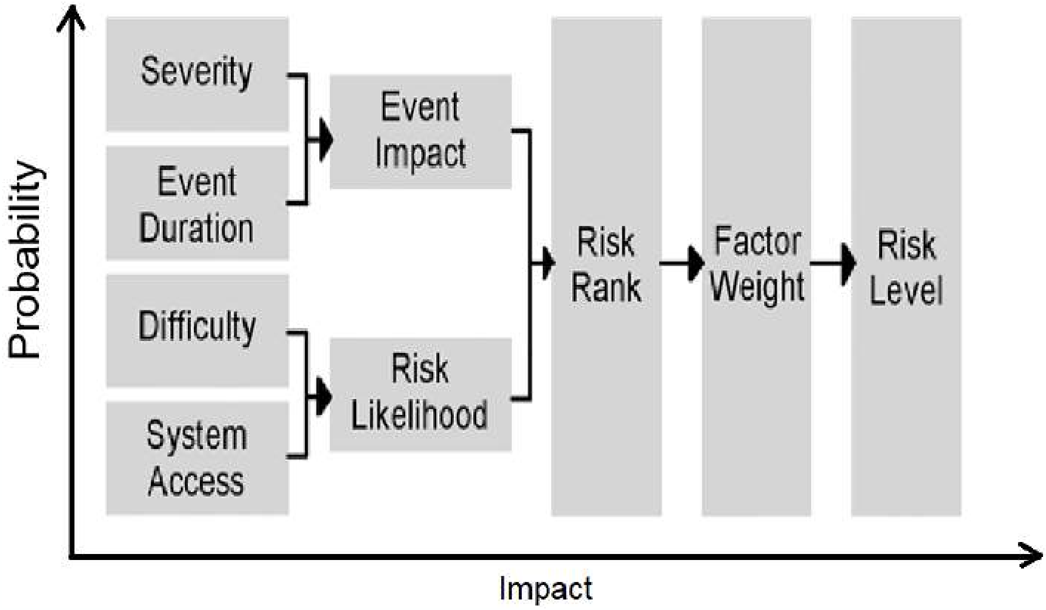

Fig. 1 Risk Matrix model

A second example is **Bayesian Networks (BNs)** – probabilistic graphical models that represent dependencies between variables. They allow for the dynamic updating of risk probabilities as new information becomes available, making them highly effective in uncertain environments. However, building accurate models requires comprehensive knowledge [8].

The **Bayes' formula** is expressed as:

$$P(A|B) = P(B|A) \times P(A)P(B) \tag{1}$$

Where P(A) is the **prior probability** of event A, P(B) is the **probability of observation** B, and P(A|B) is the **posterior probability** of event A given B. **Bayesian Networks** are widely recognized as a reliable method for **risk assessment**, **uncertainty analysis**, and **decision support** in dynamic systems [9].

Monte Carlo simulation is a powerful quantitative risk analysis technique that uses random sampling and probability distributions to model uncertainty across different scenarios. By running a large number of simulations, this method provides project managers with a comprehensive overview of potential outcomes, enabling them to assess risks more effectively. It is especially useful in complex projects where many variables influence the final result. Overall, Monte Carlo simulation is a valuable tool for risk analysis, offering data to support more informed decision-making and improved project planning. However, its application requires careful attention to data requirements and computational resources [10].

Monte Carlo simulation is widely used in project management, finance, engineering, and other fields where uncertainty plays a key role. By utilizing random sampling and probability distributions, it models the potential variability and unpredictability of factors that may affect the course of a project. This method is especially valuable for projects with complex, interrelated variables, where traditional deterministic approaches cannot fully capture the entire range of possible outcomes.

Repeated simulations provide a comprehensive imagination of possible scenarios, helping to assess the probability of achieving specific outcomes.

Fuzzy Logic Models work with vague and ambiguous data, making them ideal for risk assessment in software engineering, where uncertainty is common. These models use linguistic variables and fuzzy rules to evaluate risks, providing flexibility in decision-making as depicted in Fig. 2 [10, 18].

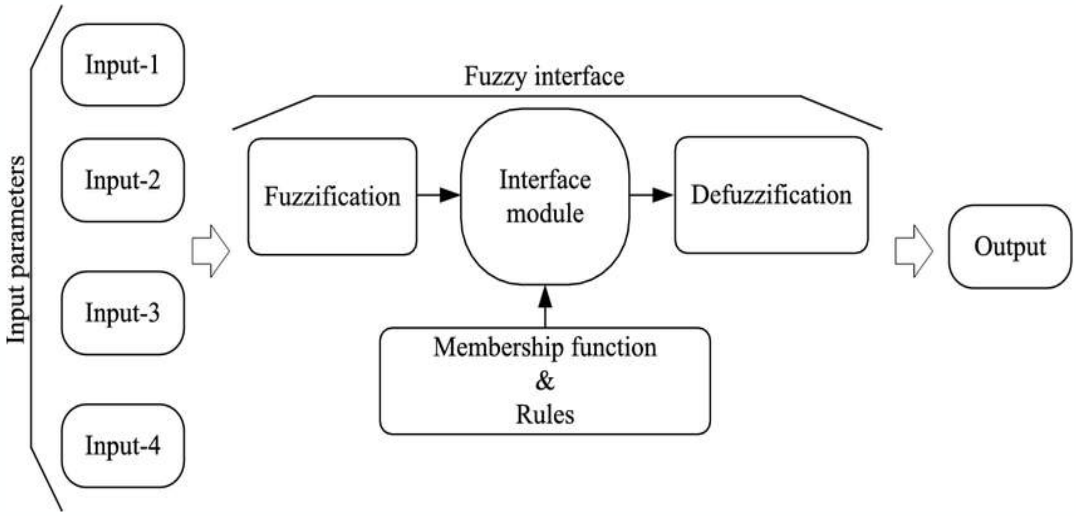

Fig. 2 A typical outline for a fuzzy logic model with four inputs and one output

A fuzzy logic model is a powerful tool for handling vague, ambiguous, or incomplete data, making it especially useful in situations dominated by uncertainty, such as software risk management. Unlike traditional binary logic, which operates with clear true or false values, fuzzy logic allows modeling of data that do not fit into assigned categories.

These models are based on linguistic variables–such as "high", "medium", and "low" – which allow managers to describe uncertain conditions more intuitively. For example, instead of assigning an exact probability of system failure, a fuzzy logic model can use terms like "likely", "unlikely", or "uncertain." Fuzzy rules built upon these variables establish relationships between different risk factors, enabling the system to assess and prioritize risks based on the available data.

The main advantage of fuzzy logic lies in its adaptability in decision-making. It can process heterogeneous input data, including incomplete or imprecise information, making it a valuable tool for risk management. By integrating expert knowledge and human experience through fuzzy rules, these models help make informed decisions even when precise data is lacking [11].

The accuracy and reliability of fuzzy logic models largely depend on the quality of the rules and the level of expertise involved in their creation. If the rules are too vague or oversimplified, the model may fail to provide a credible risk assessment, limiting its effectiveness in forecasting and mitigating potential threats. Expert knowledge plays a crucial role in developing and refining these rules to adequately reflect real-world conditions and the complexities of risk dynamics.

Machine learning (ML) approaches are increasingly being applied in risk assessment. These models utilize historical project data to identify potential risks and recommend appropriate mitigation strategies. As the volume of data grows, the accuracy of ML model predictions improves; however, their effectiveness depends heavily on the proper selection of relevant features and adequate training.

Machine learning (ML) has become an integral part of risk assessment across various fields such as project management, finance, healthcare, engineering etc. These models analyze historical project data to uncover hidden patterns, enabling more accurate predictions of potential risks. For example, decision trees identify the most critical factors for success or failure; neural networks learn complex relationships between variables for more effective forecasting; and support vector machines classify data to distinguish between high-risk and low-risk situations [12].

The main advantage of machine learning in risk assessment is its ability to analyze large volumes of data for forecasting. By examining historical data on timelines, costs, resource allocation, and performance metrics, ML models can predict risks, identify emerging issues, and suggest mitigation strategies based on past trends. This enables project managers to act proactively rather than waiting for problems to arise.

However, for effective performance, ML models require high-quality and well-structured input data. Important factors may include team efficiency, changes in project requirements, market conditions, and more. Additionally, ML models demand proper training on historical data, which is not always available or may be incomplete, affecting the models' ability to generalize information accurately.

As data volumes increase, machine learning models improve their predictions by adapting to new patterns and trends. However, this can complicate their structure, requiring careful tuning and validation to avoid over-fitting. Continuous monitoring and adjustment are also necessary to maintain the reliability of risk assessments amid the project's changing conditions.

Thus, ML models have great potential to improve risk assessment due to their analytical and predictive capabilities. When applied correctly, with attention to data quality, feature selection, and training, they significantly enhance the ability of projects to identify, evaluate, and mitigate risks.

Other methods are also used to improve decision-making in software development.

AHP (Analytic Hierarchy Process) is a structured decision-making method that breaks down complex problems into a hierarchy of criteria and alternatives. It assigns weights to various risk factors based on expert judgments, helping managers effectively prioritize risks [13]. Developed by Thomas Saaty in the 1970s, AHP has become a powerful tool in decision science, transforming subjective opinions into measurable data through step-by-step comparisons, ensuring a logical and transparent choice [14].

The Delphi method collects expert opinions through iterative surveys to achieve consensus on risk assessment. This method is useful when empirical data are limited but relies heavily on the expertise of the participants [15].

Scenario analysis examines different risk variants by modeling project conditions and their possible outcomes [16]. This helps managers prepare action plans for various risk situations.

Cost-benefit analysis evaluates the financial consequences of risk mitigation strategies by comparing their costs with the expected benefits. This approach helps select economically viable risk management methods [17].

The dynamic nature of projects, technological changes, and the human factor create challenges that require continuous updating of risk assessment methodologies. Future research should focus on integrating artificial intelligence, big data analytics, and real-time monitoring into decision support systems to improve the accuracy of predictions and the effectiveness of management.

Decision support methods and models provide structured frameworks for the identification, analysis, and mitigation of risks.

While traditional approaches, such as risk matrices and Monte Carlo simulations, offer foundational insights, contemporary techniques–including machine learning and Bayesian networks–significantly improve the accuracy of risk predictions.

By integrating these advanced methodologies, software development teams can enhance risk management processes, optimize project outcomes, and secure long-term project success.

Decision Support Systems (DSS) provide a structured approach to risk assessment in software development and the formulation of mitigation strategies.

The aim is to create a model-oriented DSS for effective risk evaluation in development projects.

The proposed DSS includes the following components:

– **Risk Identification** – detection of potential risks based on historical data and expert assessments.

– **Risk Analysis** – application of mathematical models, including probability and impact matrices, as well as the Analytic Hierarchy Process (AHP).

– **Risk Prioritization** – assigning weights through pairwise comparisons.

– **Decision Support Mechanism** – using fuzzy inference systems to generate recommendations.

Calculation of Risk Exposure:

$$\Re = P(R) \times I(R) \tag{2}$$

where: **Risk Exposure** – the overall impact of the risk; **Probability of Risk Occurrence** – the likelihood that the risk will happen; **Consequences (Impact) of the Risk** – the severity of the outcomes if the risk materializes.

Calculation of the Weighted Risk Score:

$$WRS = \sum_{i=1}^{n} Wi \times Ri \tag{3}$$

where: *Weighted Risk Score*; *weight* of each risk factor; *score* of the corresponding risk assessment.

Risk Assessment Based on Fuzzy Logic:

$$F(x) = max\ (min(A, B, C), D) \tag{4}$$

Where A, B, C, D are Fuzzy values.

We designed a DSS prototype to assist project managers in risk assessment. The system consists of:

– Knowledge Base: stores risk factors and mitigation strategies.

– Inference Engine: applies AHP and fuzzy logic for risk prioritization

– **User Interface** – provides visual reports on risk levels, displaying the results of the analysis in the form of risk categories and their corresponding weight coefficients (Table 1).

Table 1

Risk Categories and Corresponding Weights

Risk Category	**Weight (%)**
Requirement Risks	30
Schedule Risks	25
Communication Risks	20
Technical Risks	15
External Risks	10

As depicted in Fig. 3, the architecture of a decision support system for risk assessment and risk management strategies consists of three main components: a knowledge base (KB), a decision support engine (DSE), and a graphical user interface (GUI).

The knowledge base contains relevant data, including:
– Dimension of Data Structure Definition.
– Aspects, Risks and Control strategies.
– Questions, answers and rules.

The Decision Support Engine processes information based on predefined rules provided by the user. The GUI interface provides user interaction with the DSS, provides access to key functions for managing risks in software development projects, and displays reports and visualizations in the form of charts and bar graphs.

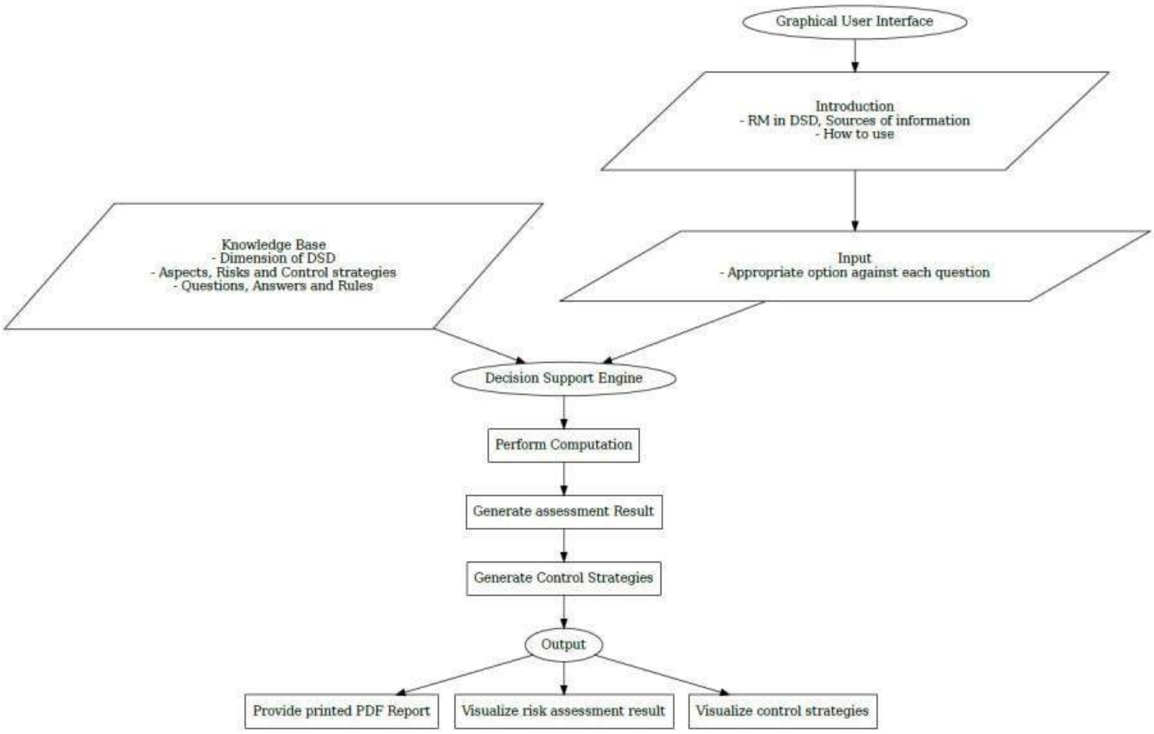

Fig. 3 Architecture of DSS

Most existing risk assessment models in software development rely on historical data and expert judgments. However, these approaches often lack flexibility and are unable to adapt to the rapidly changing conditions of modern projects. In the context of Agile methodologies, where risks evolve dynamically, traditional methods frequently prove to be ineffective.

This work suggests the approach of an adaptive decision support system that leverages Artificial Intelligence (AI) and Reinforcement Learning (RL) to continuously update and improve risk assessment models in real time.

Key Features of the Proposed Approach:

1. Intelligent Real-Time Risk Prediction:

The system applies deep learning models to analyze risks based on real-time project data. It collects and processes information from platforms such as Jira, Trello, GitHub, and

Slack, enabling it to evaluate code changes, team communication, and task progress dynamically.

2. Reinforcement Learning for Automated Model Updates:

By implementing algorithms like Q-learning or Proximal Policy Optimization (PPO), the system can continuously learn and adapt to new conditions without manual intervention.

3. Automated Risk Management Recommendations:

Natural Language Processing (NLP) techniques are used to analyze team communications (e.g., Slack, Microsoft Teams), predicting potential risks based on sentiment analysis and frequency of discussions. A conversational assistant, similar to ChatGPT, is integrated to provide real-time guidance and mitigation suggestions [20].

4. Real-Time Risk Visualization:

The user interface includes interactive heat maps, dependency graphs, and decision trees for intuitive risk assessment. Dashboards for project managers display risk probabilities and recommended actions, enhancing situational awareness and decision-making.

The expected outcomes of the study include improved risk assessment accuracy through self-learning models, as well as the ability to make rapid decisions without the need for manual data analysis.

This suggested approach combines cutting-edge artificial intelligence technologies with the real-world needs of risk management in software development.

The research has the potential to be groundbreaking, as currently, there are no decision support systems (DSS) that integrate artificial intelligence reinforcement learning for risk evaluation in this context.

Conclusions. Considering the dynamic nature of the software development industry, organizations must adopt systematic approaches to identifying, assessing, and mitigating risks at various stages of the software product life cycle. While traditional risk management methods remain useful, however they are often insufficient for effectively responding to the complexity and uncertainty inherent in modern projects.

The study analyzed a range of models and methods for risk assessment in software development, including statistical approaches, machine learning techniques, Bayesian networks, and fuzzy logic-based models. The application of these methods enables a shift from classical models to more accurate and adaptive mechanisms that account for the real-time state of a project, predict risks, and suggest effective mitigation strategies. This contributes to more precise risk identification, rational resource allocation, and improved project outcomes.

Future research in this area should focus on adapting decision support systems (DSS) to the evolving software development environment through the implementation of self-learning mechanisms that enhance risk prediction accuracy. Equally important is the integration of DSS with project management tools and team collaboration platforms to ensure seamless and real-time risk assessment and response.

Ensuring the engagement of all stakeholders and fostering a culture of risk awareness are essential prerequisites for maximizing the effectiveness of implemented solutions. Such approaches will allow reduce the likelihood of project failure, increase software reliability, and create an adaptive, flexible environment capable of responding promptly to the challenges of the modern IT industry.

References:

1. Gupta A., Mooney K. Neural network-based approach to dynamic software risk detection in global teams *// Journal of Software Engineering and Applications*. 2020. Vol. 13. No. 4. P. 212–227. DOI: 10.4236/jsea.2020.134013.

2. Singh R., Lee J. H. Multi-criteria decision support system for distributed software development risk assessment // *International Journal of Information Technology & Decision Making*. 2021. Vol. 20. No. 5. P. 1183–1204. DOI: 10.1142/S0219622021500412.

3. Lo M., Chen Y., Farooq M. Bayesian risk modeling in distributed agile environments: A dependency-based approach // *Software Quality Journal*. 2019. Vol. 27. No. 3. P. 789–805. DOI: 10.1007/s11219-018-9440-7.

4. Chow K. W., Martinez L., Yang X. Fuzzy logic-based DSS evaluation for scalable risk management in software engineering // *Expert Systems with Applications*. 2022. Vol. 185. Article 115657. DOI: 10.1016/j.eswa.2021.115657.

5. Ibraigheeth M. Software project failure avoiding through risk analysis and management // *Journal of Intelligent Systems and Applied Data Science*. 2024. Vol. 2. No. 2. P. 7–17.

6. How can decision support systems improve risk management? *LinkedIn* URL: https://www.linkedin.com/advice/3/how-can-decision-support-systems-improve-risk-management

7. Maharao C. S. A study on impact of agile and devops practices on software project management success // *ShodhKosh: Journal of Visual and Performing Arts*. 2022. Vol. 3. No. 1. DOI: 10.29121/shodhkosh.v3.i1.2022.3397.

8. Lemmens S. M. P. et al. The risk matrix approach: a helpful tool weighing probability and impact when deciding on preventive and diagnostic interventions // *BMC Health Services Research*. 2022. Vol. 22. No. 1. DOI: 10.1186/s12913-022-07484-7.

9. Hosseini S., Ivanov D. Bayesian networks for supply chain risk, resilience and ripple effect analysis: A literature review // *Expert Systems with Applications*. 2020. Vol. 161. Article 113649. DOI: 10.1016/j.eswa.2020.113649.

10. Agarwal K. What Can The Monte Carlo Simulation Do For Your Portfolio? *Insvestopedia*. URL:https://www.investopedia.com/articles/investing/112514/monte-carlo-simulation-basics.asp

11. Wu H., Xu Z. Fuzzy Logic in Decision Support: Methods, Applications and Future Trends *// International Journal of Computers Communications & Control*. 2020. Vol. 16. No. 1. DOI: 10.15837/ijccc.2021.1.4044.

12. Mwangi M. The Role of Machine Learning in Enhancing Risk Management Strategies in Financial Institutions // *International Journal of Modern Risk Management*. 2024. Vol. 2. No. 1. P. 44–53. DOI: 10.47604/ijmrm.2643.

13. Canco I., Kruja D., Iancu T. AHP, a Reliable Method for Quality Decision Making: A Case Study in Business // *Sustainability*. 2021. Vol. 13. No. 24. Article 13932. DOI: 10.3390/su132413932.

14. Akhrouf M., Medjamia H., Bessou M. An AHP-based approach to selecting a priority public transportation mode for investment // *International Journal of the Analytic Hierarchy Process*. 2024. Vol. 15. No. 3. DOI: 10.13033/ijahp.v15i3.1150.

15. Shi C. et al. Using the Delphi Method to Identify Risk Factors Contributing to Adverse Events in Residential Aged Care Facilities // *Risk Management and Healthcare Policy*. 2020. Vol. 13. P. 523–537. DOI: 10.2147/RMHP.S243929.

16. Bullis J. Scenario planning: strategies, techniques, & examples. *Cube Software | FP&A Without the Friction*. 2024. URL: https://www.cubesoftware.com/blog/scenario-planning

17. Makka K., Kampova K. Use of the cost–benefit analysis method in the risk management process of SMEs // *SHS Web of Conferences*. 2021. Vol. 129. Article 03019. DOI: 10.1051/shsconf/202112903019.

18. Akbar M. A., Shameem M., Khan A. A., Nadeem M., Alsanad A., Gumaei A. A fuzzy analytical hierarchy process to prioritize the success factors of requirement change management in global software development // *Journal of Software: Evolution and Process*. 2021. Vol. 33. No. 2. DOI: 10.1002/smr.2292.

19. Taboada I., Daneshpajouh A., Toledo N., de Vass T. Artificial intelligence enabled project management: a systematic literature review // *Applied Sciences*. 2023. Vol. 13. No. 8. Article 5014.

20. Chou J. S., Chong P. L., Liu C. Y. Deep learning-based chatbot by natural language processing for supportive risk management in river dredging projects // *Engineering Applications of Artificial Intelligence*. 2024. Vol. 131. Article 107744.

Література:

1. Gupta A., Mooney K. Neural network-based approach to dynamic software risk detection in global teams // *Journal of Software Engineering and Applications*. 2020. Vol. 13. No. 4. P. 212–227. DOI: 10.4236/jsea.2020.134013.

2. Singh R., Lee J. H. Multi-criteria decision support system for distributed software development risk assessment // *International Journal of Information Technology & Decision Making*. 2021. Vol. 20. No. 5. P. 1183–1204. DOI: 10.1142/S0219622021500412.

3. Lo M., Chen Y., Farooq M. Bayesian risk modeling in distributed agile environments: A dependency-based approach // *Software Quality Journal*. 2019. Vol. 27. No. 3. P. 789–805. DOI: 10.1007/s11219-018-9440-7.

4. Chow K. W., Martinez L., Yang X. Fuzzy logic-based DSS evaluation for scalable risk management in software engineering // *Expert Systems with Applications*. 2022. Vol. 185. Article 115657. DOI: 10.1016/j.eswa.2021.115657.

5. Ibraigheeth M. Software project failure avoiding through risk analysis and management // *Journal of Intelligent Systems and Applied Data Science*. 2024. Vol. 2. No. 2. P. 7–17.

6. How can decision support systems improve risk management? *LinkedIn* URL: https://www.linkedin.com/advice/3/how-can-decision-support-systems-improve-risk-management

7. Maharao C. S. A study on impact of agile and devops practices on software project management success // *ShodhKosh: Journal of Visual and Performing Arts*. 2022. Vol. 3. No. 1. DOI: 10.29121/shodhkosh.v3.i1.2022.3397.

8. Lemmens S. M. P. et al. The risk matrix approach: a helpful tool weighing probability and impact when deciding on preventive and diagnostic interventions // *BMC Health Services Research*. 2022. Vol. 22. No. 1. DOI: 10.1186/s12913-022-07484-7.

9. Hosseini S., Ivanov D. Bayesian networks for supply chain risk, resilience and ripple effect analysis: A literature review // *Expert Systems with Applications*. 2020. Vol. 161. Article 113649. DOI: 10.1016/j.eswa.2020.113649.

10. Agarwal K. What Can The Monte Carlo Simulation Do For Your Portfolio? *Insvestopedia*. URL:https://www.investopedia.com/articles/investing/112514/ monte-carlo-simulation-basics.asp

11. Wu H., Xu Z. Fuzzy Logic in Decision Support: Methods, Applications and Future Trends

// *International Journal of Computers Communications & Control*. 2020. Vol. 16. No. 1. DOI: 10.15837/ijccc.2021.1.4044.

12. Mwangi M. The Role of Machine Learning in Enhancing Risk Management Strategies in Financial Institutions // *International Journal of Modern Risk Management*. 2024. Vol. 2. No. 1. P. 44–53. DOI: 10.47604/ijmrm.2643.

13. Canco I., Kruja D., Iancu T. AHP, a Reliable Method for Quality Decision Making: A Case Study in Business // *Sustainability*. 2021. Vol. 13. No. 24. Article 13932. DOI:10.3390/su132413932.

14. Akhrouf M., Medjamia H., Bessou M. An AHP-based approach to selecting a priority public transportation mode for investment // *International Journal of the Analytic Hierarchy Process*. 2024. Vol. 15. No. 3. DOI: 10.13033/ijahp.v15i3.1150.

15. Shi C. et al. Using the Delphi Method to Identify Risk Factors Contributing to Adverse Events in Residential Aged Care Facilities // *Risk Management and Healthcare Policy*. 2020. Vol. 13. P. 523–537. DOI: 10.2147/RMHP.S243929.

16. Bullis J. Scenario planning: strategies, techniques, & examples. *Cube Software | FP&A Without the Friction*. 2024. URL: https://www.cubesoftware.com/blog/scenario-planning

17. Makka K., Kampova K. Use of the cost–benefit analysis method in the risk management process of SMEs // *SHS Web of Conferences*. 2021. Vol. 129. Article 03019. DOI: 10.1051/shsconf/202112903019.

18. Akbar M. A., Shameem M., Khan A. A., Nadeem M., Alsanad A., Gumaei A. A fuzzy analytical hierarchy process to prioritize the success factors of requirement change management in global software development // *Journal of Software: Evolution and Process*. 2021. Vol. 33. No. 2. DOI: 10.1002/smr.2292.

19. Taboada I., Daneshpajouh A., Toledo N., de Vass T. Artificial intelligence enabled project management: a systematic literature review // *Applied Sciences*. 2023. Vol. 13. No. 8. Article 5014.

20. Chou J. S., Chong P. L., Liu C. Y. Deep learning-based chatbot by natural language processing for supportive risk management in river dredging projects // *Engineering Applications of Artificial Intelligence*. 2024. Vol. 131. Article 107744.