

An outsourcing IT project management model has been developed. The proposed model features taking into account the specifics of project management processes at outsourcing IT companies in terms of the uncertainty of the external and internal environment of their operation. The model is based on the stage-gate project management framework with fuzzy logic tools. The proposed modification of the fuzzy inference mechanism makes it possible to refuse to save the intermediate results which reduce the load on the database and create the possibility of using semantic networks. The technology of expert consultations was demonstrated by the example of decision-making regarding the assessment of the current status of the IT projects accepted by the outsourcing company for development.

Dynamic nature and cyclical management of the portfolio of IT projects involves constant monitoring of the results of implementation with an appropriate regular portfolio reforming. The model was developed to improve the efficiency of the software development sub-process and minimize the negative consequences of financial dependence on the customer.

The application software developed on the basis of the model of management of outsourcing IT projects and modification of the fuzzy inference mechanism has found practical application and was implemented in the computational practice of HYS Enterprise B.V. outsourcing IT company. Testing of the program shell has shown positive results in the course of solving the tasks peculiar to concrete stages of IT project management.

The proposed structure and composition of the fuzzy knowledgebase of the expert shell are quite typical in terms of IT outsourcing problems. It is expedient to use the developed model at outsourcing IT companies in the process of project portfolio management

Keywords: IT outsourcing, project management, fuzzy logic, inference mechanism, semantic network, expert system

UDC 004.94
DOI: 10.15587/1729-4061.2021.224529

DEVISING A TECHNOLOGY FOR MANAGING OUTSOURCING IT-PROJECTS WITH THE APPLICATION OF FUZZY LOGIC

Zoia Sokolovska

Doctor of Economic Sciences, Professor,
Head of Department*

E-mail: nadin_zs@te.net.ua

Oleksii Dudnyk

Postgraduate Student*

E-mail: frightempire@gmail.com

*Department of Economic Cybernetics and
Information Technologies

Odessa National Polytechnic University
Shevchenka ave., 1, Odessa, Ukraine, 65044

Received date 08.02.2021

Accepted date 09.03.2021

Published date 20.04.2021

How to Cite: Sokolovska, Z., Dudnyk, O. (2021). Devising a technology for managing outsourcing IT-projects with the application of fuzzy logic. *Eastern-European Journal of Enterprise Technologies*, 2 (3 (110)), 52–65. doi: <https://doi.org/10.15587/1729-4061.2021.224529>

1. Introduction

There are a large number of outsourcing IT companies carrying out simultaneous development of a number of projects within the project portfolio. The project portfolio management process allows organizations to run multiple related projects in parallel and obtain significant benefits. A group of projects that do not have joint control will most likely not be able to achieve desired results. To avoid this problem, present-day technologies and methods are used to simplify and improve the management process. However, the numerous management methods just change the way the decisions are made and do not simplify the decision making process. Although the project management practice has accumulated an abundance of knowledge and elaborated a great number of methodologies offering numerous tools, the need to manage an increasing number of various projects entail significant difficulties for IT companies. Because of the specifics of activities of IT companies, these difficulties can differ sharply from the general problems in the field of project management. It should also be borne in mind that project managers are often forced to monitor the project progress in a constantly changing environment. That is why researchers of the project management process try to generalize the processes involved in each project first of all. Most methodologies distinguish between the following “triple constraints” [1]: efficiency, schedule, and budget. However,

the IT projects are not subject to restrictions in the same way as the projects for manufacturing physical products, e.g. construction projects. Software is still governed by real constraints, although they are more abstract as usual. Therefore, they are hard to understand and discuss. However, both the customer and the contractor often forget or simply do not understand the IT limitations. This leads to unrealistic expectations and overly ambitious projects [2]. At least in part, this is explained by the wrong idea that software is largely unlimited and therefore has unlimited potential.

Also, the situation is not improved by the fact that conventional models and methods of project management do not recognize the reality of present-day companies and misinterpret priorities in the workplace, and do not affect the potential benefits as well. The project portfolio management process balances key portfolio constraints and provides a tool for decision-making throughout the portfolio performance cycle based on performance indicators. This allows the project portfolio managers to control and manage the full scope of work and the range of actions required to launch various numbers of projects adhering to the budget, schedule, and specifications of each project.

The defining characteristics of the project portfolio are combined to create a work environment in which financial and social performance is so high that the prediction and analysis of future consequences of achieving a concrete goal are understandable. The literature on this issue coincides

significantly with the idea that we are talking about risks and uncertainty [3, 4]. Therefore, decision-makers need to pay close attention to the methods and models intended to identify, assess, and manage risks and uncertainties. The main problem there is determining the mathematical apparatus needed to select models and methods to predict and solve project portfolio management problems. It should be borne in mind that the decision on the portfolio structure is based on a preliminary assessment of data and taking into account the specifics of its implementation in conditions of uncertainty caused by unknown future consequences. In addition, the enormous complexity of financial markets makes the stochastic approach less appropriate. Therefore, it is important to find other approaches to solving the problems of information uncertainty in the process of project portfolio management. Expert systems with fuzzy logic are a potential tool for solving this task, i.e., the use of fuzzy numbers and fuzzy sets to describe indeterminate phenomena and application of fuzzy logic in the data processing. An integrated approach that uses fuzzy logic systems to manage the portfolio will be a management process based entirely on fuzzy subphases:

1. Fuzzy initial data. Bringing the portfolio and environment data to a state acceptable for the fuzzy logic systems.
2. Fuzzy information processing. It uses fuzzy logic and fuzzy math.
3. Transformation of fuzzy information. It is a step to transforming information in a usable form that will lead to decision-making and management actions regarding the project portfolio.

Taking into account the abovementioned, involvement of the fuzzy set theory for its application in expert systems is an important area of further studies in the field of outsourcing IT project management.

2. Literature review and problem statement

Modification of the adaptive neuro-fuzzy inference system (ANFIS) was proposed in [5] as a tool. The proposal deserves special attention due to the introduction at the early stages of a model of an additional control element based on fuzzy logic. The introduction of fuzzy logic at early calculation stages improves the accuracy of input data for the next stages. Disadvantages of this modification include the inability to change the level of the input data refinement without changing the basic set of rules.

The model developed and presented in [6] can eliminate the fuzziness of input information for individual projects due to fuzzy logic. Disadvantages of this model include incompleteness of the data fact base to support project decisions.

A problem of portfolio optimization was considered in [7]. The whole process is divided into three stages: project portfolio analysis, market scenario forecasting, and portfolio rebalancing. Based on information on the stage of forecasting the market behavior, fuzzy sets of the second type are used at the last stage to find the best option for the portfolio rebalancing. Disadvantages of this decision include its atypicality and focusing mainly on the process of the portfolio rebalancing.

An unconventional solution to the problem of the project risk assessment was found in [8]. The authors have proposed risk assessment metrics based on fuzzy logic which increased the calculation accuracy. It has resulted in an opportunity to

offer a substantiated diversification of design solutions. As before, the disadvantage of this result consists in atypicality and focusing on the risk assessment process.

Budget investments were the object of study in [9]. The problem relates to investment optimization in conditions of uncertainty of managers' risk tolerance. In order to reflect the decision-making dynamics caused by the ambiguous risk tolerance, the authors developed a two-stage adaptive optimization model. Budget distribution is the result of the first stage where the actual level of each manager's tolerance is found. Project choice made by the manager is the result of the second stage which is adapted to the manager's risk tolerance. A concept of a neutral risk-free budget threshold is introduced. It is modeled with the help of fuzzy logic. On this basis, an ambiguous risk tolerance curve is constructed which can actually establish the attitude of managers to the risks. The disadvantage of this result consists in focusing on the risk assessment process as well.

A fuzzy methodology was proposed in [10] in the course of assessment of the project time and cost to reduce the uncertainty impact on the results obtained. The project evaluation and review technique (PERT) is combined in this study with fuzzy logic and peer review to reduce the impact of existing uncertainty on the results. Accordingly, estimates of project time and cost are more appropriate than the classic PERT. The disadvantage of this methodology consists in that the data sets for estimates may contain unusual or abnormal observations. In its turn, this reduces the overall accuracy of the methodology.

A model of a compromise between expenses and time by means of a genetic algorithm and the theory of fuzzy sets in the conditions of uncertainty was developed and presented in [11]. Fuzzy sets are used there to model uncertainties, and a genetic algorithm is used to obtain the minimum project cost and duration. Thus, optimal timetable results are obtained by establishing a fuzzy model of timetable optimization according to the different risk levels identified by decision-makers. Disadvantage of this model consists in focusing on cost and time estimation which corresponds to a small part of the software product life cycle.

A fuzzy system of human resource assessment for project management in key areas where professional services are appropriate for the project success was presented in [12]. This model determines the level of work experience, core competencies, impact, and amount of knowledge. Also, an algorithm for selecting individual members of the project management team was developed based on the proposed fuzzy system. Disadvantages of this system include the inability to change the basic set of rules.

The use of the presented achievements in the project management process has given positive results. However, the issues related to incompleteness and unreliability of the data fact base supporting the project decision making as well as the presence of a large number of intuitive management procedures remained unresolved. The use of expert judgment may be an option to overcome these difficulties [13]. Although this prediction technique can be quite effective in some cases, it also suffers from the problems of subjectivity, uniqueness, inconsistency, and vulnerability to knowledge loss if managers as bearers of knowledge leave the organization. It is also not uncommon for software management data sets to contain unusual or anomalous observations thus reducing the overall accuracy of any model derived empiri-

cally from these data [14]. Another problem is the inability to change the level of detail at the system inputs and outputs without changing the basic set of rules [15].

All this suggests that it is appropriate to conduct a study to find a solution for a balanced way of integrating expert and formal modeling using fuzzy logic. It is also advisable to conduct a study on the use of more formalized modeling methods that limit the consideration of subjective knowledge.

3. The aim and objectives of the study

The study objective implied elaborating a procedure for managing the process of project implementation in outsourcing IT companies using a mathematical apparatus of fuzzy logic.

To achieve this objective, the following tasks were set:

- develop a model of IT project management taking into account peculiarities of project management in outsourcing IT companies and using fuzzy logic;
- develop the architecture of a fuzzy expert system that would effectively adapt to the needs and problems of outsourcing IT project management;
- test the developed fuzzy model of outsourcing IT project management on an example of the task of assessing the status of a set of IT projects.

4. Development of a model of managing outsourcing IT projects using fuzzy logic

It was proposed to develop a model of IT project management based on the stage-gate framework as the most suitable for adaptation and use taking into account the IT outsourcing specifics. The classic stage-gate framework was integrated with the agile methodology and a fuzzy decision-making system was built into the overall outline of the project management process when moving between the framework stages.

The stage-gate framework is both a conceptual and an operational model for the maintenance of a new product from idea to launching [16]. This is the principle of managing the process of developing new products aimed at improving overall efficiency and achieving the strategic goals of the company. The stage-gate framework is not a software product, however, its conceptual simplicity and openness do not prevent the creation of concrete software implementation. It seems from the outside that the idea of dividing the process into smaller subprocesses is not new in general. But the idea of a framework consists not only in distribution. A gate between each stage is introduced for decision making in the interstage transition which qualitatively distinguishes the process by the formal presence of a decision-making point. Also, an additional positive effect in the introduction of the gate consists in the ability to bring calculations directly from the stage to the gate. More clear distribution allows each stage to focus on information acquisition and delegating its processing to the gate.

The stage-gate framework can be used as a model for managing the project development process. The generalized structure of the stage-gate framework is presented in Fig. 1.

The framework divides the process into a predetermined set of stages (consisting of a group of prescribed, related, and often parallel measures with a clearly defined gate as a point

of the stage-to-stage transition. It can be mentioned that gates control the process to some extent. Each gate is characterized by a set of results from the previous stage and a set of criteria for transition to the next stage. Input data are the results that the project manager delivers to the gate. Transition criteria are the points at which the success of the previous project stage will be assessed. Based on these criteria, a decision is made on the transition to the next project, completion of the project, postponement of the project, or its processing in favor of other projects. The basic structure of the framework helps eliminate some of the limitations imposed on the management process. Those limitations that cannot be removed immediately are eliminated due to the ability to adapt the framework to its own needs including the needs of IT project management. This eliminates the problems considered earlier.

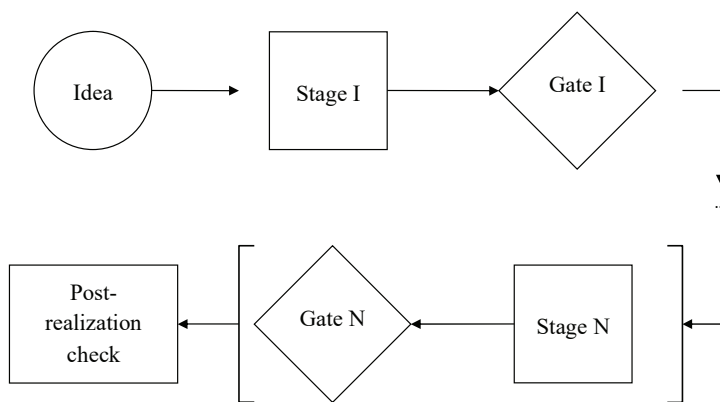


Fig. 1. The general structure of the stage-gate framework

One of the problems is that this framework was originally developed for equipment and consumer product manufacturing processes that are not relevant to the IT project development process. Although the basic structure of such processes remains useful, it needs to be adapted to the specifics of the IT project development processes so that it works as efficiently as in the development of other types of products. The next problem, one of the biggest, is that the software development process is often presented as a single stage. Certainly, newer framework versions have reservations about the possibility of returning to some previous stages in the framework, however, this does not take into account the cyclicity of this process in full. The operations performed in the software development process take from 75 % to 90 % of the total workflow. It would be more appropriate to take into account the cyclicity of this process or have several intermediate points at this stage which will enable decision making and conducting reassessment more often. Another standard problem consists in that the stages of requirements analysis and planning are often combined in one stage. Actually, this means that the distribution of the calendar plan for the early project stages is within 15 to 35 % of the project calendar time. Non-technical project participants often must explain what software development operations need to be completed to get a meaningful view of the project before moving on to the software development stage. A sufficiently wide range of calendar time distribution before this causes a high probability of transition to the stage of software development without all necessary information or the previously specified deadline [17].

Thus, a number of problems arise during the adaptation of the stage-gate framework to the software development process. Such problems were less noticeable when using successive stages of software development. However, the advent of a set of flexible software development methodologies has minimized the practice of using successive stages of development. This, in turn, has complicated the process of adapting the stage-gate framework to modern agile processes of software development.

Agile processes of software development are iterative development where requirements change according to the customer needs. It helps in adaptive planning and iterative development. The agile process corresponds to the software development life cycle including requirements collection, analysis, design, development, testing, and implementation of partially implemented software with the expectation of the customer's feedback [18]. The agile process is generally similar to the stage-gate framework except for the well-defined gates at which decisions are made to continue the project. This process requires less planning and divides tasks into smaller subtasks. From the beginning, it was designed for short-term projects taking into account specifics of teamwork which corresponds to the life cycle of software development. However, over time, hereditary methodologies have begun to evolve that extend the process to larger projects and teams. Customer involvement in the management of software development processes reduces the risks associated with later stages of software development, such as market introduction and testing on the customer's side. This is achieved due to the fact that it is a process in which you can make changes dynamically in accordance with the satisfaction of the customer's needs. Common features of the agile methodologies are as follows:

- adaptability to changing environments. With smaller planning cycles, it is easy to make changes in any period of the project's life. There is always an opportunity to improve and rethink the product inventory allowing the teams to make changes to the project in one or more iterations;

- ultimate goal may be unknown. The agile methodology is very useful for projects where the ultimate goal is not clearly defined. As the project progresses, the goals become clearer and development can easily adapt to the corresponding changes;

- speed and quality. Dividing a project into iterations allows the team to focus on high-quality development and testing. Testing during each iteration means that errors were identified and eliminated earlier which results in faster implementation of high-quality software;

- cooperation with the customer. Customers have many opportunities to see the progress of work and really influence the final product. They can gain a sense of ownership by working so closely with the project team. Agile projects also encourage feedback from users and team members throughout the project, so the lesson learning is used to improve future iterations.

The main difference between the agile and stage-gate methodologies consists in their different purposes. The stage-gate methodology is comprehensive in launching new projects through macro-planning and the agile methodology is for managing the project development process through micro-planning [19]. A detailed comparison of the two methodologies is given in Table 1.

The stage-gate methodology is multifunctional, i.e. it involves marketing specialists and technical staff and has several stages covering the entire process from elaborat-

ing the idea of a new product to launching this product. Solutions in the stage-gate methodology correspond to the investment decision model. The decision to move through the gates affects the redistribution of resources for projects as their potential manifests itself. Thus, the stage-gate methodology gives instructions on what projects have to be developed and then what to do within each project. On the other hand, the agile methodology is designed specifically to help the product developers quickly create working software with ongoing validation from the customer. Once the development project has been approved and its initial requirements have been outlined, the agile methodology focuses on the software development process. Each iteration of the software development process may not produce enough functionality to guarantee a market entry but to have a product that is potentially ready for release at the end of each iteration is its initial goal.

Table 1

Comparison of software development project management methodologies

Characteristic	Stage-gate	Agile
Type	Macro-planning	Micro-planning, project management
Scope	From idea to launching	Development and testing can be extended to the pre-development phase
Organization	Multifunctional team (R&D, marketing, selling, operations)	Technical team (software developers, engineers)
Solution model	Investment: a leading management group can be involved in critical decision making	Tactical: decisions concerning the actions on the next sprint are mainly made by a self-managed team

Thus, the first feature of the agile methodology lies in that it is used mainly at the project design and testing stages. That is, it covers two of the five or six stages that are part of the typical stage-gate process of the framework. The latter is mainly used by technicians who are actually involved in software development.

The second conceptual difference between the two methodologies is the different views of main variables in the project management: workload, budget, and fulfillment duration. In the stage-gate framework, the amount of work is fixed and the budget and time are variable. In the agile methodology, fulfillment duration and budget are fixed in each sprint and the work scope can vary.

In addition, it should be borne in mind that the development of outsourcing IT projects has its own features. General stages of the outsourcing IT projects are as follows:

1. Project search. Even before the start of the software development process, the outsourcing company faces the problem of finding a project for development. This problem is multifactorial as it may include finding a customer, assessing the risks of working with a concrete existing or potential customer, and many other factors. It may also include dis-

cussions on timing, technologies, and the necessary financial and human resources.

2. Preliminary technical analysis of the project. After the final consolidation of the relationship between the outsourcing company and the customer by signing a commercial agreement, comes the stage of analysis of the project requirements. Unlike food companies, outsourcing companies do not need to build a business case and marketing strategies for the project. This makes the analysis process technical. The terms of development and the necessary resources for this have already been precisely discussed at this stage. The main purpose of this stage is to develop a preliminary plan for the software development process and recruit staff from available or non-available resources.

3. Agile sprints of software development. This stage is iterative and includes the main processes related to software development. Current agile methodologies for managing IT projects can be quite different in terms of the processes implemented in them. However, the main skeleton of the processes such as reevaluation and analysis of sprints, sprint planning, software development, functionality testing can be identified from almost all of them.

4. Project support. This stage is optional because it may not be provided under a contractual agreement with the customer. If it is provided, the deadlines can be quite significant in time. This can be support associated with the initial launch of the product or support with a pre-planned long term.

5. Retrospective analysis of the project. In order to further improve the process of developing and allocating resources between projects, a post-project analysis phase is carried out. Based on the information from the previous two stages and historical data (if any), information on the implementation of the current project is formed in order to update the historical data.

Summarizing the previous information, the stage-gate framework was modified which combined the classic stage-gate framework taking into account the features of outsourcing IT project management, Fig. 2.

It can be seen from Fig. 2 that there is one essential feature in the way of arranging the gates between stages. Some of the gates are not one-way. For example, the gate after the project search process is unidirectional because the project search process itself is not iterative. On the other hand, the gate before the agile sprint process is two-way because there is a process of assessing the results at the same gate after each sprint.

It also makes sense to consider using fuzzy logic in the stage-gate framework. The development of IT projects is characterized by the absence and fuzziness of information as well as the lack of a complete picture of the final result. The result is refined quite often when moving from one stage to another and thus each subsequent stage works with more accurate information. In this regard, one of the modifications of the stage-gate framework demonstrated in this work is

the construction of a fuzzy decision-making system at each gate (Fig. 3).

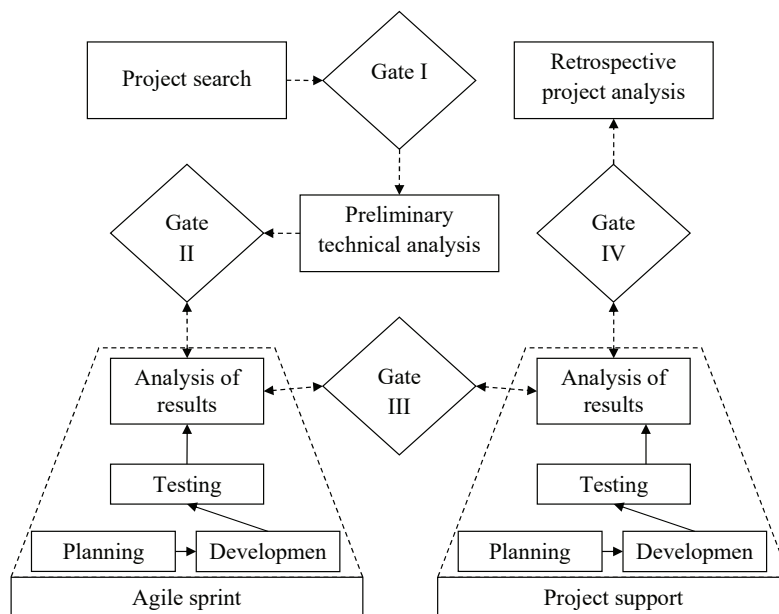


Fig. 2. Diagram of modification of the agile stage-gate framework taking into account features of outsourcing IT-projects

This has advantages over the classic stage-gate framework:

1. An opportunity is created to describe rules of decision-making and rules of stage-to-stage transition in a lexical format.
2. The need for large amounts of accurate numerical data is reduced in favor of lexical data.

A general structure of a modification of the agile stage-gate framework was developed on the basis of a fuzzy-logical approach taking into account features of the outsourcing IT projects (Fig. 4).

The modification features the use of fuzzy logic systems in the transition from the gates to the next stages. Also, in some cases, there is a need for the iterative application of the fuzzy logic system depending on the iterative nature of the stage. These results not only justify the need and relevance of the use of fuzzy logic in the process of managing the outsourcing IT projects but also demonstrate the benefits of this approach.

5. Development of a fuzzy expert system architecture adapted to the management of outsourcing IT projects

The architecture of a typical fuzzy expert system shell usually contains the following components: knowledgebase, initial data, fuzzy inference kernel, and an interface for working with the shell. The knowledgebase and the core of fuzzy inference are the most important components that make sense to study.

The proposed structure of the knowledgebase contains two components: a database of fuzzy rules and a database of linguistic variables used in the database of rules.

The human intellectual process is too complex to be represented as a deterministic algorithm. However, most professionals are able to form their knowledge in the form

of rules. The application of these rules gives an opportunity to solve problems in the subject area.

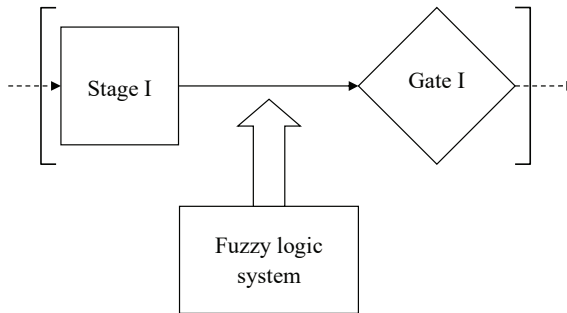


Fig. 3. The stage-gate framework using fuzzy logic

A rule can be defined as an IF-THEN structure that links information or facts in the IF part to certain actions or information in the THEN part. The rules are relatively easy to create and understand. In the terminology of expert systems, the IF-THEN structures are inference rules. Knowledgebases are created on their basis with mechanisms of a direct/inverse deduction. This makes it possible to find solutions to domain problems by reducing the total volume of calculations.

Thus, the database of fuzzy rules can be composed of an unlimited number of rules of the form:

$$\text{IF } (X=A) \text{ THEN } (Y=C),$$

where X, Y are linguistic variables that are set in the database of linguistic variables; A, C are the fuzzy linguistic equivalent of some well-defined meaning associated with a corresponding linguistic variable.

The database of linguistic variables consists of an unlimited number of variables of the form:

$$X : \text{Initial} : \left[\begin{array}{l} A : \text{Trapezoidal} : (A1, A2, A3, A4) \\ = B : \text{Trapezoidal} : (B1, B2, B3, B4) \end{array} \right],$$

where X is a linguistic variable; *Initial/Derivative* is a linguistic variable the value of which will be taken, for example, from a database or is derived in the process of working with the knowledgebase, respectively; $A : \text{Trapezoidal} : (A1, A2, A3, A4)$ and $B : \text{Trapezoidal} : (B1, B2, B3, B4)$ are the functions of belonging to a fuzzy set; A, B is a set of values for the linguistic variable X ; Trapezoidal is a trapezoidal type of the membership function which is used to describe values of a linguistic variable; $(A1, A2, A3, A4), (B1, B2, B3, B4)$ are clear values that lie behind the fuzzy values of A and B , respectively.

This structure of the knowledgebase has several very important features. First, it is an opportunity to expand and reuse it. The knowledgebase can be expanded with new knowledge in the transition from specialist to specialist.

Second, it is a potential opportunity to combine a knowledgebase and a database in order to simplify the creation of a knowledgebase based on templates of inference rules and linguistic variables. In this case, it will be possible to use similar templates instead of the structures of inference rules and linguistic variables described earlier which will significantly speed up the process of creating a typical content of the knowledgebase.

The inference core is the basis of a typical software shell for working with fuzzy data. Next, describe the principles of its

work. Fig. 5 shows an example of a classical mechanism of fuzzy inference. Its main components include a database with clear information, a fact base for storing intermediate results, and a knowledgebase with information on how to deduce from the knowledgebase and the fact base.

The disadvantage of the classical inference mechanism consists in the constant need to access the database in order to obtain the information necessary to calculate and maintain the fact base in the current state. The disadvantage of obtaining information for calculations is quite significant, however, the use of fuzzy inference rules in comparison with the classical ones should get rid of this drawback.

The second drawback is not so important but it makes you think about the structure and ways to maintain the fact base. The fact base can physically be a part of the knowledgebase that is illogical or a part of a database that increases the load on the database. The problem of complicating the description of inference rules decreases with increasing the number of rules from which the knowledgebase is formed. This is achieved due to the fact that when the knowledgebase increases, a more detailed description allows the user to understand and work with it fast enough.

However, with an increase in the knowledgebase size, i.e. the number of rules in the knowledgebase, other problems appear. These problems arise from the process of working with this knowledgebase, namely, from the classical direct inference algorithm.

One can get rid of both drawbacks by reviewing how to work with the knowledgebase and the ways to maintain it. Instead of using clear inference rules, the previously described combination of fuzzy inference rules and simplified linguistic variables can be used. This structure of the knowledgebase makes it possible to get rid of the base of intermediate facts which was closely related to the need to store intermediate information for permanent access to the database (Fig. 6). This, in turn, makes it possible to use a semantic network that implements access to the database once to obtain the initial data.

Next, consider how a semantic network can be used for fuzzy inference. Semantic networks have been developed to present knowledge of an intelligent system that uses natural language. It was decided to use the semantic network processing system (SNePS) for the problem of fuzzy inference [20]. This system is a tagged directional graph in which nodes represent concepts and arcs represent binary relationships between concepts. Propositional graphs from the SNePS family are graphs in which each unique expression from the knowledgebase is represented by a node in the graph. The inference rule can be represented in a graph through the nodes of the rule itself, the formulas of the input and output arguments, as well as the arcs passing from the nodes of the rule to the nodes of the arguments. The arcs also indicate the roles that the argument plays in the rule explaining the connection. It is necessary to remember the connection of rules with each other through the inclusion of linguistic variables from the right or left part of one rule in the right or left part of another rule. If the left part of one rule occurs in the right part of the second rule, the second rule is called the predecessor, otherwise, a follower. Let us consider the following example and a semantic net for this set (Fig. 7):

- R1: IF (A1) THEN (B1).
- R2: IF (A2 AND A3) THEN (B2).
- R3: IF (B1 AND B2) THEN (C1).

Rule R1 is an equivalence rule which means the following. If the predecessor is TRUE, then the successor also takes the value of TRUE. Rules R2 and R3 are i-infer, which means that

each predecessor must be TRUE in order for the followers to be TRUE.

The inference graphs were developed and proposed by Schlegel and Shapiro [21] as extensions of propositional graphs. An inference graph is a graph of reasoning that it is capable of inverse, direct, and bidirectional inference. It can support parallel processing for reasoning using inference logic. The inference graphs modify the propositional graphs by adding channels between nodes along possible paths of inference. Channels carry priority messages to transmit new information from one node to another. The message priorities affect the order in which tasks are performed so that the messages are executed closer to the creation of response and the inappropriate inference tasks are canceled. A rule node is capable of performing inference operations using a set of rules known as rule usage information (RUI) [22]. When a message arrives at a rule node, it is converted to an RUI. The RUIs contain information about which predecessors are true or false as well as a set that explains how these values were derived. When a new RUI is created, it is combined with a set of existing ones. The resulting combination is used to determine whether the rule node of inference rules can be applied again. All RUIs created on the node are cached. This prevents re-withdrawal and shortening of the message flow cycles in the graph ignoring the already received RUIs in the cache.

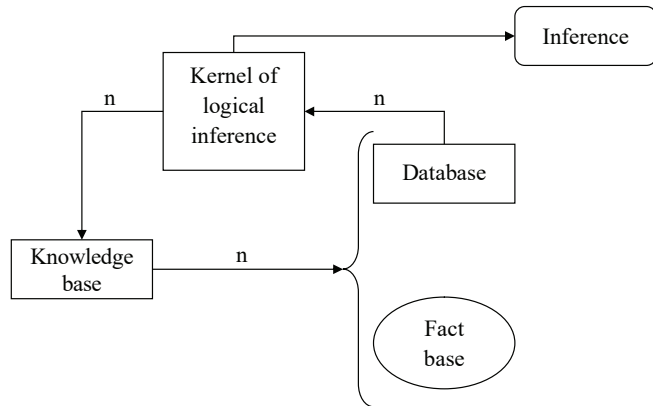


Fig. 5. The classic mechanism of fuzzy inference

Only direct inference in the system is used for fuzzy inference. Therefore, the structure of the inference graphs proposed by Schlegel and Shapiro can be greatly simplified:

1. Only two types of messages are required: i-infer and u-infer. The i-infer messages originate from new nodes that were received as TRUE or FALSE. They are then sent to the rule nodes in which the current node is the predecessor. The u-infer messages occur in the rule nodes that have just received all the necessary information about their predecessors. They inform the target node of its new status.

2. There is no need for parallel reasoning, so there is no need for message priorities.

3. Instead of RUI, it is possible to use the simple status of the rule which is updated when a new message is received. The status contains the result of counting the statuses of all predecessors of this rule.

Fig. 8 presents a graph (Fig. 7) with corresponding channels.

The channels allow predecessors to report rule nodes when they were calculated and also allow the rule nodes to report that they have been calculated. The dependence on initial data in the database is reversed in this option of using the semantic network. The core of the inference mechanism, i.e. the semantic network, no longer needs to constantly request for data. It only remains to count on the already available initial data and the potentiality of their expansion from the outside in the inference process.

Thus, this leads to a qualitative acceleration of the process of fuzzy inference. Also, it should be borne in mind that the features of the core of fuzzy inference of the shell leave the opportunity to work with knowledgebases of applied areas in which there is a need to simplify and accelerate the decision-making process. Such areas include project management of IT companies of various types (outsourcing and food companies, the companies developing integrated IT); project management of the companies outside the IT sector; management related to the management of material and immaterial resources regardless of the enterprise type, etc.

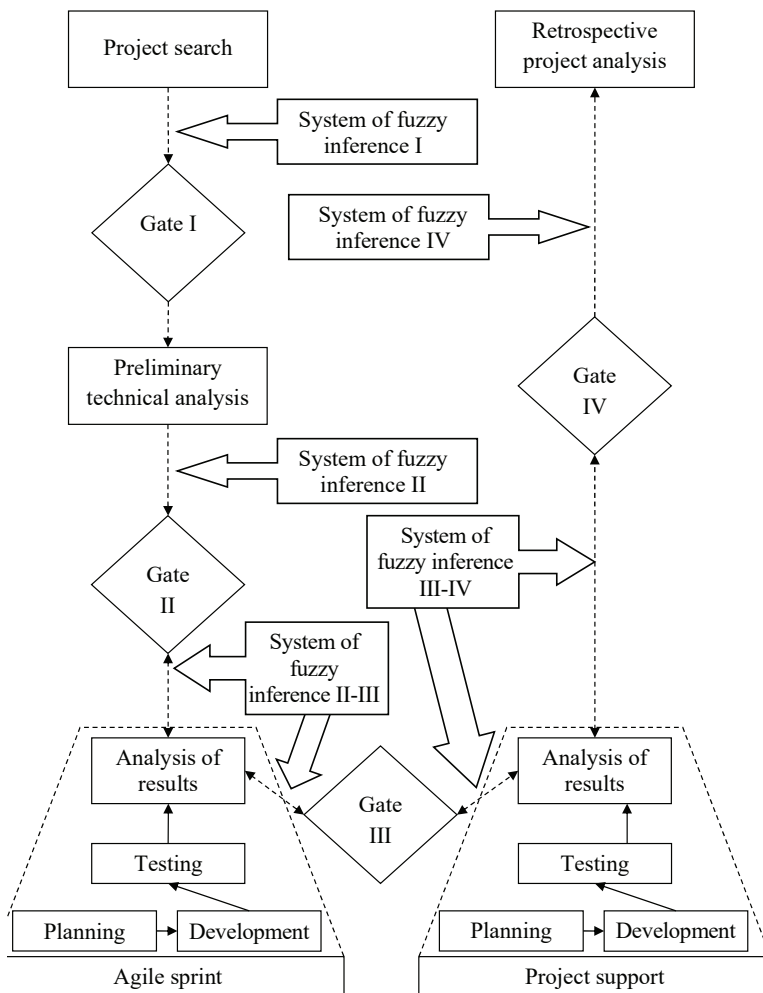


Fig. 4. Diagram of modification of the agile stage-gate framework taking into account features of outsourcing IT projects using fuzzy logic

There are fuzzy inference tools on the market, such as the MATLAB Fuzzy Logic Toolbox [23]. The package makes it possible to describe the knowledgebase in lexical format and can be used to model the project management process within outsourcing IT companies using the developed management model. However, the status of the platform as a modeling tool is a significant drawback. This makes it impossible to change the fuzzy inference algorithm allowing only filling the knowledgebase and observation of the final results. In contrast, a fuzzy shell based on a modification of the fuzzy inference system architecture and a fuzzy inference algorithm using SnePS graphs was proposed. This makes it typical and flexible enough to handle a variety of tasks.

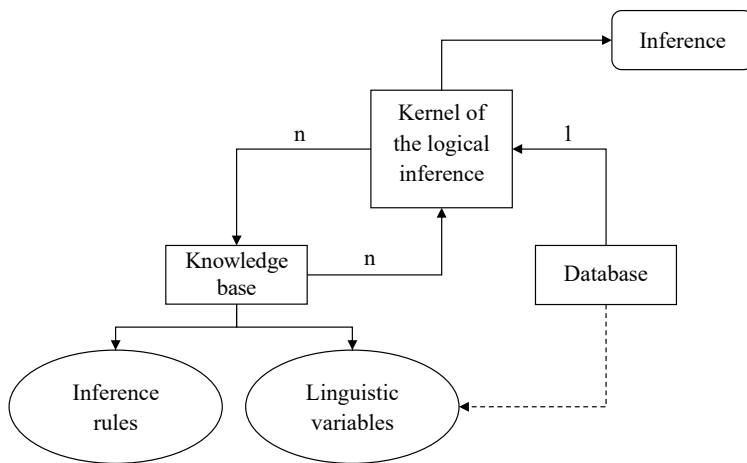


Fig. 6. Modified mechanism of the fuzzy inference

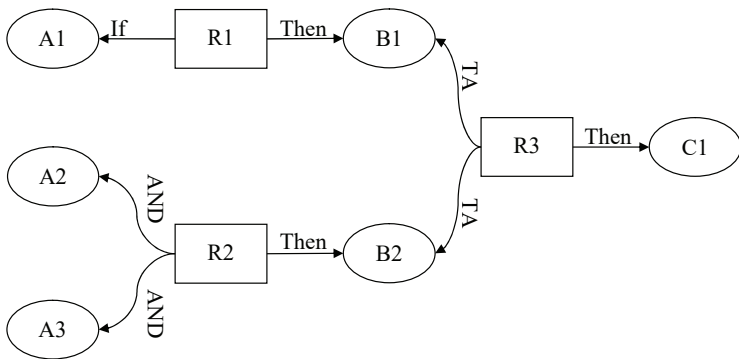


Fig. 7. Semantic net for a set of rules

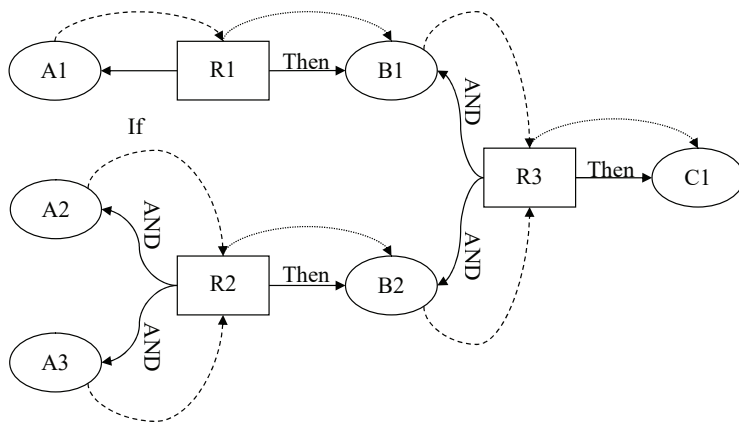


Fig. 8. Semantic network for a set of rules with channels

6. Testing a fuzzy model of IT project management: the task of assessing the project status

The introduction of a typical shell to solve the project management problems was tested on near-real data and made retrospectively. Data on the project implementation process and relevant economic indicators were generated on the basis of historical data remaining after the completion of projects.

To demonstrate the inference process using fuzzy inference rules, a task of re-evaluating the current state of the outsourcing IT company's project portfolio was chosen. This task is one of the most important tasks in the software development cycle as it is iterative and usually the longest. An outsourcing company always has the task of monitoring the current state of the project in order to respond in a timely manner to adverse changes in the project implementation process. There are many reasons that can lead to undesirable results and require prompt response to avoid financial problems associated with the project. Given the information described above about the problems encountered in the process of managing the IT projects, the most likely reasons for outsourcing the IT companies are as follows:

1. Overspending on team members.
2. Team lag behind in inventory.
3. Incorrectly defined stock priority.

From the point of view of an outsourcing IT company to address adverse changes, one of the most promising solutions is as follows:

1. Redistribution of the available personnel to compensate for economic indicators that lead to adverse changes.

2. Starting negotiations with the customer to expand the budget in order to expand the staff, the number of teams, or to compensate for economic indicators.

3. Starting negotiations with the customer to extend the project implementation time.

The decision made in solving this problem is usually multifactorial as it relates to and affects a project portfolio that consists of several projects within a single product frame. Now, let us form a fuzzy set of rules for this kind of problem. Taking into account the simulated situation and the above definitions, a fuzzy set of rules was built from the following templates:

1. IF [Project X does not have enough budget to support the team AND Other Projects have the ability to shuffle the team] THEN [Negotiations on shuffling of the Project X team have to be started].

2. IF [Project X does not have a budget to support the team AND Other Projects do not have the ability to shuffle the team] THEN [Negotiations to expand the budget of Project X have to be started].

3. IF [Project X excess of work is below a certain level AND Project X product backlog is below a certain level] THEN [Project X is executed ahead of schedule].

4. IF [Project X excess of work is above a certain level AND Project X product backlog is below a certain level] THEN [Project X is executed on time].

5. IF [Project X excess of work is above a certain level AND Project X product backlog is above a certain level] THEN [Project X is executed with a delay].

6. IF [(Project X is delayed OR Professional level of Project X team is below a certain level) OR [Project X is executed on time AND the priority of Project X is not low]] THEN (Project X does not have the opportunity to hand over staff).

7. IF [(Project X is executed ahead of time AND Average level of Project X team is above a certain level) OR (Project X is executed on time AND the priority of Project X is low)] THEN [Project X has the opportunity to hand over staff].

8. IF [Project X has a need to shuffle teams AND Other Projects have the ability to hand over staff] THEN [Negotiations on shuffling of Project X team have to be started].

9. IF [Project X has a need to shuffle teams AND Other Projects do not have the ability to hand over staff] THEN [Negotiations to expand Project X team need to be started].

10. IF [Project X is delayed AND Project X priority is low] THEN [No need to shuffle commands for Project X AND no need to extend Project X team and no need to extend Project X time].

11. IF [Project X is executed with a delay AND the priority of Project X is not low] THEN [There is a need to shuffle teams for Project X OR there is a need to expand the Project X team OR there is a need to extend Project X time].

The fuzzy knowledgebase built using the above templates was applied for a fuzzy inference on reevaluation of the current state of the project portfolio at different stages of product development. The available input data (Tables 2–5) which roughly describe the data of the project portfolio at the time of the three stages of development, i.e., initial (sprint 1), middle (sprint 5), late (sprint 9).

To implement the fuzzy inference, an application based on the inference core working on the principles of SNePS semantic network was developed. Taking into account the initial data from the tables and the knowledgebase created according to the rule templates described above, the results of fuzzy inference were obtained.

For the first sprint, the expert system does not recommend any action. This is logical as at the moment of development there are no grounds for recommendations on the extension of terms, budget, etc. as demonstrated by the expert system.

For the fifth sprint, the [Migration Start Negotiate Budget=Yes] node with a confidence factor of 0.9 was activated. Given the lack of a budget for the Migration team and the impossibility of relocating other teams to balance the budget, the expert system recommends starting negotiations to expand the budget for the group working with migration issues.

For the ninth sprint, the result becomes multifactorial. The [Services Start Negotiate Budget=Yes] node was activated with a confidence factor of 1. The [Migration Start Negotiate Team Exchange=Yes] node was also activated with a confidence factor of 0.8.

It is recommended in this example to begin negotiating the budget extension for the Services development team and begin the process of shuffling the teams for the budget equalization for the Migration team.

This decision was made because the Portal team had the ability to transfer developers to the Services team without risk for the entire team falling behind. All other results do not give recommendations and are not demonstrated.

Table 2

Projects in the portfolio: basic information

Project in the portfolio	Priority	Concise description
UI UX	High	The interface design project
Portal	High	The interface software development project
Migration	Low	The project of developing services of migration of current customer data for the new system
Services	Middle	The project of developing the business logic of the new system

Table 3

Projects in the portfolio: sprint 1

Project in the portfolio	Budget	Labor excess	Material stock of the product	Professional level of the team
UI UX	Adequate	None	Big	Middle
Portal	Adequate	None	Big	Senior
Migration	Adequate	None	Big	Junior
Services	Adequate	None	Big	Middle

Table 4

Projects in the portfolio: sprint 5

Project in the portfolio	Budget	Labor excess	Material stock of the product	Professional level of the team
UI UX	Adequate	None	Middle	Middle
Portal	Adequate	Low	Big	Middle
Migration	Inadequate	None	Average	Middle
Services	Adequate	Low	Low	Middle

Table 5

Projects in the portfolio: sprint 9

Project in the portfolio	Budget	Labor excess	Material stock of the product	Professional level of the team
UI UX	Adequate	Low	None	Middle
Portal	Adequate	None	Low	Middle
Migration	Inadequate	Low	Low	Middle
Services	Inadequate	Average	None	Senior

In order to conclude on the success of the project management using the proposed method and the presented results of using the software shell based on this method, it is necessary to measure the method's productivity.

Productivity measurement can be considered as a process of quantifying the actions that lead to the project completion where measurement is a process of quantifying the effectiveness and efficiency of these actions [24].

The criteria of cost, time, and quality were considered by Schonhar [25] and Baccarini [26] as internal measures of the project management efficiency. These three aspects are also called the “iron triangle”. Two of these three criteria, namely duration, and cost of the project can be considered absolute values. At the same time, smaller values correspond to more efficient project management. The project duration can be counted as the number of sprints from the date of the project start to the date of its completion.

To formalize the implicit assumptions underlying the calculation of the project implementation cost, efforts E_R are determined as a result of using a resource with a certain intensity R_I through the given duration Δt . We will use a general linear approximation to calculate the effort $E_R=R_I\Delta t$.

These efforts can be compared by converting them into costs using the cost norm C_R which can be obtained from the structure of resource allocation [28]. The units of these specific cost rates are the units of currency per unit of effort. The lower index R reminds us that the cost rate varies from resource to resource. As a result, we have the total value $C=C_RE_R$.

Analysis of the earned value combines three elements: budget, schedule, and volume when using the value as a common medium of exchange. Thus, the unit of the primary financial currency of the project (for example, dollars, pounds, euros) becomes a unit for all indicators of the earned value. Therefore, different dimensions can be compared because they have a common basis.

The earned value system includes volume and integrates it with costs and the schedule. First, it is necessary to determine the cost of a fully completed project in the context of the costs that were foreseen and agreed upon in the project plans. The project earns money only when a certain task is completed. The amount of this cost is determined by the costs that were budgeted. Next, the cost analysis includes a graph in this general basis of comparison asking how much costs should have occurred according to the project schedule. There are basically three terms that identify the earned value:

1. Scheduled cost (C_S) is the approved budget allocated for planned works.

2. Actual cost (C_A) is the actual cost of the work performed: the sum of direct and indirect costs incurred for the performance of work on this activity during the project execution.

3. Earned value (budgeted cost (C_B) is the cost of the performed work provided in the budget. It relates the initial planned cost of the project and the speed at which the team completes the project.

The difference between the budget cost and the other two costs determines whether the project is ongoing, ahead, or behind the budget or schedule. The difference in costs is calculated as $\Delta C_A=C_B-C_A$. When $\Delta C_A<0$, the project exceeds the budget. The difference in the schedule is calculated as $\Delta C_S=C_B-C_S$. When $\Delta C_S<0$, the project lags behind the schedule.

Next, standard performance indicators of earned value are formed by comparing the earned value with each of the other two costs in the triplet. The first indicator is the cost ratio $F_A=C_A/C_B=1-\Delta C_A/C_B$. The second indicator is

the decomposition coefficient $F_S=C_S/C_B=1-\Delta C_S/C_B$. The project is executed according to the schedule and budget when $F_A=F_S=1$. The project is ahead of the budget or schedule when the coefficients are less than one. The coefficient of execution of the cost schedule can be obtained from the following coefficients:

$$F = F_A + F_S - 1 + \sqrt{\frac{1}{2}(F_A - F_S)^2}. \quad (1)$$

Subtraction of 1 allows the combined coefficient to be equal to 1 if the project is executed both according to the schedule and the budget. The square root of the squared difference between the factors is added to the sum so that the effect of one high individual factor cannot be hidden by a low number in another factor.

Here are the relative coefficients that can be calculated in the process of the problem solution. The success of project management focuses on the project process and, in particular, on the successful implementation of cost, time, and quality objectives. Projects are formed to achieve goals and success is measured by how well these goals have been achieved. Time R_T can be measured in terms of adherence to the schedule. Time, as an indicator of success, can be measured in terms of exceeding/non-compliance relative to the schedule as a percentage of the original plan [29]. Costs R_C can be measured in terms of budget execution. Costs as an indicator of success can be measured as excess/non-compliance in a percentage of the initial budget [29]. Quality R_Q can be measured in terms of compliance with functional and technical specifications. The success of technical indicators depends on the extent to which the technical requirements specified at the beginning of the phase implementation have been achieved [30].

Having presented a way of carrying out calculations, we will pass directly to calculations based on the considered example of the introduction of the proposed method. Let us start with the calculations on the historical data left after the project. The available historical data on the implementation of this project are based on the team distribution and duration of the project.

Regarding the project duration, the project was planned for 5 months (10 sprints) and completed in 6 months (12 sprints) of which the last month was spent on finalizing the project of the Migration team. The available approximate data on team distribution are shown as follows:

1. Monthly costs of specialists by levels: USD1,000 for the Junior, USD2,000 for the Middle, USD3,000 for the Senior.

2. Distribution of professional level of specialists by teams. For Services: 2 Juniors, 1 Middle, 1 Senior. For UI|UX: 2 Middles, 2 Seniors. For Portal: 2 Middles, 2 Seniors. For Migration: 2 Juniors, 1 Middle.

The scheduled cost C_S is the budget for the selected specialists during the planned term of the project, namely 5 months:

$$C_S = C_S(\text{Migration}) + C_S(\text{Services}) + C_S(\text{Portal}) + C_S(\text{UI|UX}) = 155,000 \$.$$

The actual cost C_A is the actual cost of the specialists' work during the entire project implementation period, namely 6 months. Also, given the initial data, we will keep in mind

that the staff number was expanded during the development process. Namely, after sprint 5, one Middle specialist was added to the Migration team. After sprint 9, one Middle specialist was also added to the Services team. Calculate based on this information:

$$C_A = C_A(\text{Migration}) + C_A(\text{Services}) + C_A(\text{Portal}) + C_A(\text{UI|UX}) = 167,000 \$.$$

Budgeted cost C_B is the cost of timely developed projects. That is, for the considered data, these are the Services, UI|UX, and Portal projects. The cost of an unfinished Migration project for this indicator is not taken into account: $C_A = C_A - C_A(\text{Migration}) = \text{USD}136,000$. All data necessary for the calculation of the coefficient of execution of the schedule of expenses are calculated using (1): $\Phi = 1.416$. The value higher than one indicates the lag of the considered set of projects.

To complete the picture, relative coefficients were also calculated. The success of a set of projects in terms of time can be calculated as a ratio of time actually spent to the planned execution time of this set of projects (in months): $R_T = 1.05$.

The success of the set of projects in terms of costs can be calculated as the ratio of C_A to C_S : $R_C = 1.077$.

The success of a set of projects in terms of quality can be calculated as the ratio of the number of timely developed functionality to the planned amount of functionality. In this example, the developed functionality can be considered from the point of view of completion/non-completion of subprojects. Since three of the four projects were completed on time, $R_Q = 0.75$.

A value of R_T greater than one indicates a negative deviation from the success by the time factor. A value of R_Q less than one indicates a negative deviation from the success of the quality factor. A value of R_C greater than one indicates a negative deviation from success by the cost factor.

Similar coefficients were calculated taking into account the implementation of the recommendations proposed by the expert system. Since the data provided are historical and the project has not been re-implemented, the implementation of the expert system is according to the model. As in the calculations of historical data, we take into account that one Middle specialist was added to the Migration team after sprint 5 and one Middle specialist was added to the Services team after sprint 9. In addition, the expert system recommended starting the process of shuffling the specialists between teams with the aim of equalizing the team professional level of the Migration team. This was achieved due to the ability of the Portal team to give one of the specialists without the risk of lagging behind for the team.

Calculation of the planned value C_S does not differ from the similar calculation in historical data, i.e. $C_S = \text{USD}155,000$, so let us move on to the results of calculation of the actual value $C_A = \text{USD}161,000$. Taking into account the expert system recommendations, it was planned that all four subprojects will be completed on time. In this case, the value of the budget value C_B will not differ from the value of the already calculated actual value C_A , i.e. $C_B = \text{USD}161,000$. Having all the necessary data, calculate the coefficient of execution of the cost schedule, using (1):

$F = 0.988$. A value close to one indicates the timely completion of the considered set of projects without overspending.

Similar to the calculations of historical data, calculations of relative economic indicators were performed: $R_T = 1$; $R_C = 1.038$; $R_Q = 1$. A value of R_T equal to one indicates no deviation from success by the time factor. A value of R_C greater than one indicates a negative deviation from success by the cost factor. The value of R_Q equal to one indicates no deviation from success by the quality factor.

Comparison of the results of the evaluation of the relative success rates of historical data and the data as a result of the simulated implementation of the expert system shows that implementation of the expert system would most likely lead to timely completion of the set of projects that was confirmed by the coefficients of quality R_T for time and R_Q equal to one for quality. In its turn, these same coefficients for historical data indicate negative deviations. It is more interesting to observe the situation with the coefficient of quality for costs R_C . Negative deviations can be observed for both data sets. But in the case of the introduction of the expert system, we had a value of 1.038 which is slightly less than the historical data of 1.077. This indicates less deviation when using the expert system.

A similar comparison of the results of the assessment of the combined coefficient of execution of the cost schedule Φ indicates the uniqueness of the result. For historical data, the value of this coefficient 1.416 indicates negative deviations from the planned process and the lag of the set of projects from the implementation plan which is also confirmed by relative coefficients. Instead, the value of 0.988 obtained when using the expert system indicates, albeit small, but still positive deviations in the process and the lack of lag behind the implementation plan.

Finally, let us consider interesting results outside the relative and combined coefficients. The first result is that the expert system recommends starting the negotiation process with the customer to expand the budget just when it is most needed. Namely, after the expansion of the Migration team at the end of sprint 5 and the Services team after sprint 9. If this moment is missed, it is likely that this difference in the cost of the specialists' works, the outsourcing IT company will pay for themselves. At the same time, in the case of timely resolution of this issue, the company that ordered the product will suffer losses.

The second result is more obvious as it concerns the timely completion of the Migration project. Depending on the agreement between the outsourcing IT company and the customer company, one of them will incur losses in the amount of the cost of specialist labor for the time required to complete the Migration subproject. This is an undesirable result. In the case of using the expert system and timely shuffling of teams in order to "align" the professional level of the Migration team, such a result is not expected.

7. Discussion of the results obtained in the study of the project management model in outsourcing IT companies

The proposed model of IT project management is based on the modification of the classical architecture of the stage-gate framework (Fig. 1) due to its integration with the agile methodology. The results of comparative analysis

of the above methodologies of software development have allowed us to establish distinctive features relevant to the general outline of outsourcing IT project management (Table 1). The advantage of the presented combination of methodologies consists in a clear detailing and separation of the stages of project management and the boundaries between them which are the process of making subsequent management decisions. Arrangement of the gates between the stages, the possibility of taking into account the iterative nature of the processes further contribute to the clear structuring of the management circuit shown in Fig. 2. According to the complete absence or presence of fuzzy information at different stages of project management, it was proposed to use fuzzy logic in the built modification of the agile stage-gate framework (Fig. 3).

The applied embodiment of the given concept is reflected in the developed shell of the fuzzy expert system. Optimization of the structure of the fuzzy knowledgebase and modification of the fuzzy inference algorithm were proposed (Fig. 6). Suggestions for improving the semantic structure of the network and the search algorithm were presented and illustrated in Fig. 7, 8. The use of a fuzzy modification of semantic architecture of the knowledgebase in the process of inference in the reasoning column has allowed us to reverse dependence on initial data in the database. Flexibility and openness of the developed shell positively distinguish the shell from existing analogs [23, 31]. At the same time, it enables adaptation of the shell to solve the problems of managing outsourcing IT projects using fuzzy logic.

Application of the proposed management model using fuzzy logic in the practice of an outsourcing IT company was presented on the example of expert advice on establishing the status of a portfolio of IT projects. Basic information on portfolio projects is given in Tables 2–5: near-real model data provided by HYS Enterprise B.V outsourcing IT company (Odesa, Ukraine) were used. Improvement in management efficiency was evaluated on the background of the developed model using the coefficient of execution of the cost schedule according to the criteria of cost, quality, and time. The algorithm is represented by formula (1). The calculation results have proved the projected improvement in management efficiency by 1.43 times.

Main advantages of the proposed procedure:

1. Taking into account flexibility of specifics of managing outsourcing IT-projects.
2. Ability to describe knowledge of IT project management processes in a lexical format which simplifies the process of work with a fuzzy knowledgebase.
3. Dynamism of the fuzzy knowledgebase which makes it possible to supplement it in the transition from user to user.
4. Leveling of unusual or anomalous observations due to the possibility to describe them in the knowledgebase in a lexical format.
5. Flexibility of the knowledgebase structure contributes to its adaptation to other subject areas.

Limitations of the proposed method include:

1. Subjectivity of filling the knowledgebase can lead to a conflict of simultaneous filling by different users.
2. Lack of ability to combine fuzzy computations with clear ones.

Disadvantages of the proposed procedure include:

1. Manual formation of knowledgebase content. In the future, it will be possible to use rule templates and combine

them with original data to increase the speed of the knowledgebase filling.

2. Lack of historical data storage component. In the future, it will be possible to add this component to preserve the preliminary results of the expert assessment. This will increase the degree of smoothing unusual or abnormal observations. Further studies should be aimed at reducing the impact of the above shortcomings. It also makes sense to pay attention to combining fuzzy computations with clear ones. This will simplify the process of filling the knowledgebase and reduce the influence of subjectivity in its filling.

8. Conclusions

1. A fuzzy model of providing support in making managerial decisions regarding promotion of the process development in individual IT projects and project portfolios was proposed. The essence of this model is dividing the development process into stages and individual steps with the required level of detail. Diagnostication and decision-making concerning the further implementation of the project and/or project portfolio in terms of incomplete and/or unclear information are accomplished at the boundaries of these stages.

This model eliminates the shortcomings inherent in the known methodologies of IT project management:

- it provides the process detailing appropriate in specific circumstances;
- it provides an assessment of the project/portfolio functioning environment in conditions of uncertainty by means of processing the data of different quantitative and qualitative nature;
- it improves the validity of the decision-making criteria based on the establishment of relationships between individual project's elements and their environment.

2. Instrumental implementation of the model was realized as an integration of the modified state-gate framework, agile methodology, and the developed shell of the fuzzy expert system. Modification of direct inference algorithm within the shell based on the use of the SNePS semantic network as the fuzzy inference core and its development to support fuzzy rules was proposed. Expert knowledge is described in a simplified lexical format.

The advantage of the procedure consists in a balanced way of integrating expert and formal modeling of expert knowledge.

Gradual replenishment of the institutional memory of the expert system will make it possible to expand its application in the IT-sphere. The shell can be recommended as an independent platform for application in project management in multisubject areas.

3. The developed fuzzy model was tested at HYS Enterprise B.V. outsourcing IT company (Odesa, Ukraine). The company provided model data close to the real task of assessing the current status of the IT projects portfolio. The implementation results showed a planned increase in efficiency of the IT project portfolio management in terms of relative and absolute efficiency indicators. Taking into account the above, it is advisable to use the model in the context of the overall management of the project portfolio of IT companies in conditions of uncertainty and risk.

References

1. Rosenau, M. D., Githens, G. D. (2005). *Successful Project Management: A Step-by-Step Approach with Practical Examples*. John Wiley & Sons, 384.
2. Polak, J., Wojcik, P. (2015). Knowledge management in it outsourcing/offshoring projects. *PM World Journal*, 4 (8). Available at: <https://pmworldlibrary.net/wp-content/uploads/2015/08/pmwj37-Aug2015-Polak-Wojcik-knowledge-management-IT-outsourcing-second-edition.pdf>
3. Keynes, J. M.(1937). The General Theory of Employment. *The Quarterly Journal of Economics*, 51 (2), 209. doi: <https://doi.org/10.2307/1882087>
4. Atkinson, R., Crawford, L., Ward, S. (2006). Fundamental uncertainties in projects and the scope of project management. *International Journal of Project Management*, 24 (8), 687–698. doi: <https://doi.org/10.1016/j.ijproman.2006.09.011>
5. Kumar, C., Doja, M. N. (2018). A Novel Framework for Portfolio Selection Model Using Modified ANFIS and Fuzzy Sets. *Computers*, 7 (4), 57. doi: <https://doi.org/10.3390/computers7040057>
6. Hassanzadeh, F., Collan, M., Modarres, M. (2012). A Practical Approach to R&D Portfolio Selection Using the Fuzzy Pay-Off Method. *IEEE Transactions on Fuzzy Systems*, 20 (4), 615–622. doi: <https://doi.org/10.1109/tfuzz.2011.2180380>
7. Pai, G. A. V. (2017). Fuzzy Decision Theory Based Metaheuristic Portfolio Optimization and Active Rebalancing Using Interval Type-2 Fuzzy Sets. *IEEE Transactions on Fuzzy Systems*, 25 (2), 377–391. doi: <https://doi.org/10.1109/tfuzz.2016.2633972>
8. Nguyen, T. T., Gordon-Brown, L., Khosravi, A., Creighton, D., Nahavandi, S. (2015). Fuzzy Portfolio Allocation Models Through a New Risk Measure and Fuzzy Sharpe Ratio. *IEEE Transactions on Fuzzy Systems*, 23 (3), 656–676. doi: <https://doi.org/10.1109/tfuzz.2014.2321614>
9. Wang, S., Wang, B., Watada, J. (2017). Adaptive Budget-Portfolio Investment Optimization Under Risk Tolerance Ambiguity. *IEEE Transactions on Fuzzy Systems*, 25 (2), 363–376. doi: <https://doi.org/10.1109/tfuzz.2016.2582906>
10. Habibi, F., Taghipour Birgani, O., Koppelaar, H., Radenović, S. (2018). Using fuzzy logic to improve the project time and cost estimation based on Project Evaluation and Review Technique (PERT). *Journal of Project Management*, 3, 183–196. doi: <https://doi.org/10.5267/j.jpm.2018.4.002>
11. Acar Yildirim, H., Akcay, C. (2019). Time-cost optimization model proposal for construction projects with genetic algorithm and fuzzy logic approach. *Revista de La Construcción*, 18 (3), 554–567. doi: <https://doi.org/10.7764/rdlc.18.3.554>
12. Adeola, O. S., Ganiyu, A. R. (2020). A Fuzzy System for Evaluating Human Resources in Project Management. *International Journal of Technology Diffusion*, 11 (1), 66–95. doi: <https://doi.org/10.4018/ijtd.2020010105>
13. Hughes, R. T. (1996). Expert judgement as an estimating method. *Information and Software Technology*, 38 (2), 67–75. doi: [https://doi.org/10.1016/0950-5849\(95\)01045-9](https://doi.org/10.1016/0950-5849(95)01045-9)
14. Miyazaki, Y., Terakado, M., Ozaki, K., Nozaki, H. (1994). Robust regression for developing software estimation models. *Journal of Systems and Software*, 27 (1), 3–16. doi: [https://doi.org/10.1016/0164-1212\(94\)90110-4](https://doi.org/10.1016/0164-1212(94)90110-4)
15. Gray, A. R., MacDonell, S. G. (1999). Fuzzy logic for software metric models throughout the development life-cycle. 18th International Conference of the North American Fuzzy Information Processing Society - NAFIPS (Cat. No.99TH8397). doi: <https://doi.org/10.1109/nafigs.1999.781694>
16. Stage-gate model. Available at: <https://www.stage-gate.com/stage-gate-model>
17. McConnell, S. (2006). *Software Estimation: Demystifying the Black Art*. Microsoft Press, 308.
18. Sharma, S., Sarkar, D., Gupta, D. (2012). Agile processes and methodologies: A conceptual study. *International Journal on Computer Science and Engineering*, 4 (5), 892–898. Available at: https://www.yashada.org/yash/egovcii/static_pgs/TC/IJCSE12-04-05-186.pdf
19. Cooper, R., Kielgast, S., Vedsmand, T. (2016). Integrating Agile with Stage-Gate® – How New Agile-Scrum Methods Lead to Faster and Better Innovation. Available at: <https://innovationmanagement.se/2016/08/09/integrating-agile-with-stage-gate>
20. Shapiro, S. C., Rapaport, W. J. (1992). The SNePS family. *Computers & Mathematics with Applications*, 23 (2-5), 243–275. doi: [https://doi.org/10.1016/0898-1221\(92\)90143-6](https://doi.org/10.1016/0898-1221(92)90143-6)
21. Schlegel, D. R., Shapiro, S. C. (2014). Concurrent Reasoning with Inference Graphs. *Graph Structures for Knowledge Representation and Reasoning*, 138–164. doi: https://doi.org/10.1007/978-3-319-04534-4_10
22. Choi, J., Shapiro, S. C. (1992). Efficient implementation of non-standard connectives and quantifiers in deductive reasoning systems. *Proceedings of the Twenty-Fifth Hawaii International Conference on System Sciences*. doi: <https://doi.org/10.1109/hicss.1992.183186>
23. Fuzzy Logic Toolbox. MathWorks. Available at: <https://uk.mathworks.com/products/fuzzy-logic.html>
24. Neely, A., Gregory, M., Platts, K. (1995). Performance measurement system design: A literature review and research agenda. *International Journal of Operations & Production Management*, 15 (4), 80–116. doi: <https://doi.org/10.1108/01443579510083622>

25. Shenhar, A. J., Levy, O., Dvir, D. (1997). Mapping the dimensions of project success. *Project management journal*, 28 (2), 5–13. Available at: <http://www.reinventingprojectmanagement.com/material/other/7.%20Mapping%20dimensions%20of%20projects%20success%20PMJ%201997.pdf>
26. Baccarini, D. (1999). The Logical Framework Method for Defining Project Success. *Project Management Journal*, 30 (4), 25–32. doi: <https://doi.org/10.1177/875697289903000405>
27. Papke-Shields, K. E., Beise, C., Quan, J. (2010). Do project managers practice what they preach, and does it matter to project success? *International Journal of Project Management*, 28 (7), 650–662. doi: <https://doi.org/10.1016/j.ijproman.2009.11.002>
28. Rad, P. F., Cioffi, D. F. (2000). Work and resource breakdown structures for formalized bottom-up estimating. The George Washington University, Washington, D.C. Available at: <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.583.8756&rep=rep1&type=pdf>
29. Might, R. J., Fischer, W. A. (1985). The role of structural factors in determining project management success. *IEEE Transactions on Engineering Management*, EM-32 (2), 71–77. doi: <https://doi.org/10.1109/tem.1985.6447584>
30. Freeman, M., Beale, P. (1992). Measuring project success. *Project Management Journal*, 23 (1), 8–17.
31. Orchard, R. A. (1998). FuzzyCLIPS Version 6.04A User's Guide. Integrated Reasoning. Available at: <https://quentin.pradet.me/blog/media/FuzzyCLIPS/fzdocs.pdf>