UDC 004.415.2

A.Stenin, V.Pasko, O. Lisovychenko, O.Polshakova, V. Soldatov

DEVELOPMENT OF SUBJECT-ORIENTED INNOVATIVE SOFTWARE PRODUCTS BASED ON ARTIFICIAL INTELLIGENCE SYSTEMS

Abstract: This article analyzes existing models of the development process of innovative software products for a specific subject area. Taking into account the multi-alternative nature of solutions at each stage of innovative software products development and compliance with constantly changing requirements, this article proposes an information and logical model of innovative software products development management process based on artificial intelligence systems. The essence of the proposed approach is that the development of an innovative software products is interpreted as an information object that changes in content and structure in the process of its creation. The presence of uncertainty and a large amount of information requires the use of intelligent decision support systems at each stage of ISP development.

*Keywords***:** innovative software product, multi-alternativeness, uncertainty, information model, decision making, intelligent decision support system

Introduction

The software development industry is undergoing a major transformation, and artificial intelligence (AI) is at the forefront of this change. With the ability to automate repetitive tasks, analyze massive amounts of data, and make predictions with a high degree of accuracy, artificial intelligence is revolutionizing the way software is developed, tested, and deployed. An important benefit of AI in software development is its ability to improve communication and teamwork. Using AI-based tools and platforms, developers can work with each other more effectively and efficiently, streamlining communication and reducing the time and effort required to complete innovative software development projects [1-3]. One of the defining functions of innovative software products (ISP) developer is making innovative decisions during the ISP development process. Today, decision-making involves a “field of knowledge” that is uncertain not only in volume, but also in dimension. In the process of developing a solution, dynamically changing economic, technological, organizational, natural and other aspects are taken into account. Sources of information can be the knowledge of experienced specialists, literature, regulatory materials of scientific and design organizations, Internet resources [4-6]. Hence, the need to use intelligent decision support systems (IDSS) is beyond doubt. Taking into account the above, this article proposes an infological model for managing the process of developing an ISP and a structural diagram of the IDSS for making innovative decisions at each stage of developing an ISP.

Statement of the problem

The human factor and the presence of multiple alternatives and a large amount of information in the process of developing an ISP introduce significant uncertainty and complexity into the adoption of innovative decisions. In addition, the process of developing an ISP requires extracting knowledge of a given subject area from various sources of information, including from Internet resources. In this regard, there is a need to create an informational model for managing the process of developing an ISP and a structural diagram of the IDSS for making innovative decisions at each stage of developing an ISP. The process of developing an ISP in this case is interpreted as an information object that changes in content and structure in the process of its creation.

Analysis of existing decisions

The information structure of the process of managing the creation of ISP should provide for the possibility of replenishing the management system with new knowledge. Each observation can add new knowledge about the software product, not provided for by the regulations, but, in the long term, increasing management efficiency. Despite the high degree of uncertainty in the process of creating a software product, due to the presence of the human factor, the process of creating a software product itself is completely deterministic and consists of a number of stages that can be built according to different schemes. Let's consider the most well-known models for developing ISP [7,8,12,17].

Cascade model. The first model that has become widely known and truly structures the development process is cascade or waterfall (Fig. 1) [9,10]. This model divides the process of creating a software product into successive stages (it should be noted that it was already used by various developers at that time, but neither the number nor the content of the stages was unified).

However, the practical use of this model revealed many of its shortcomings, the main one of which was that it is more suitable for traditional types of engineering activities than for software development. In particular, one of the biggest problems turned out to be its “predisposition” to possible inconsistencies between the resulting product and the requirements that were placed on it. The main reason for this is that a fully formed product appears only in the later stages of development, but since work at different stages was usually carried out by different specialists and the project was transferred from one group to another, then, according to the principle of a broken phone, it turned out that the output was not quite what was initially expected. But it is rational to use the cascade methodology:

- if the requirements at the initial stage are clearly defined and recorded;
- programmers with the required qualifications are available;
- the project is relatively small.

 ISSN 1560-8956

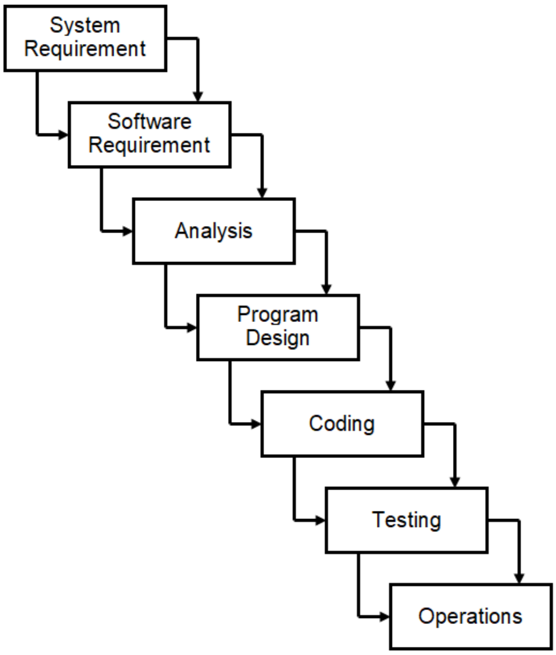

Fig. 1. Cascade model of the design process

V-shaped model. It was proposed precisely in order to eliminate the shortcomings of the cascade model, and the name – V-shaped, or hinged – appeared because of its specific graphical representation (Fig.2) [10,11].

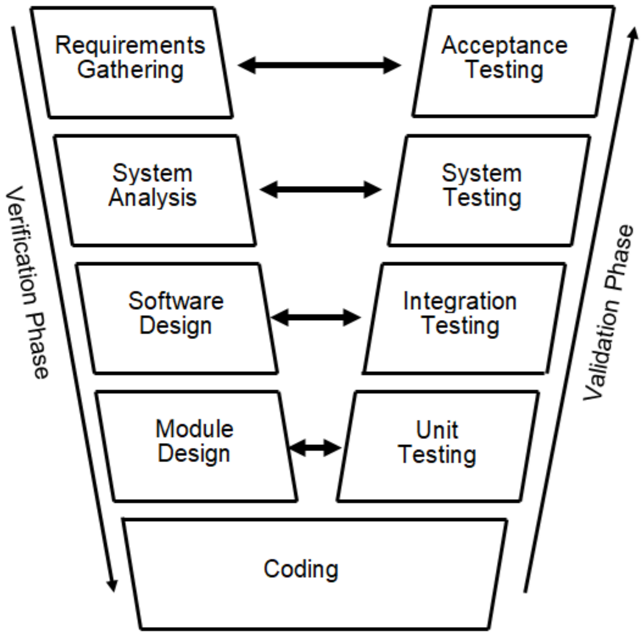

Fig. 2. V-shaped model of the design process

The V-shaped model made it possible to significantly improve the quality of software due to its focus on testing, and also largely resolved the problem of compliance of the created product with the put forward requirements thanks to verification and certification procedures in the early stages of development (the dotted lines in the figure indicate the dependence of the planning/problem setting stages and testing/acceptance [10].

However, if the development process takes a long time (sometimes up to several years), then the resulting product may actually be unnecessary to the customer, since his needs have changed significantly. But it is advisable to use the V-shaped methodology:

— if necessary, thorough testing and verification of the software product;

— for small and medium-sized projects whose requirements are defined and recorded;

— subject to the availability of qualified testers.

Incremental model. It involves dividing the overall package of requirements into various assemblies. These elements make up several development cycles, which are further combined into a “multi-waterfall”. At the first stage, a product with a basic set of functions is released, then they are gradually added until a full-fledged one is created product (Fig. 3) [17].

It is advisable to use the incremental methodology:

— if the requirements are clearly defined, but some elements can be refined during the work process;

— early launch of a product with basic functionality is required;

— there are a number of unpredictable features or purposes.

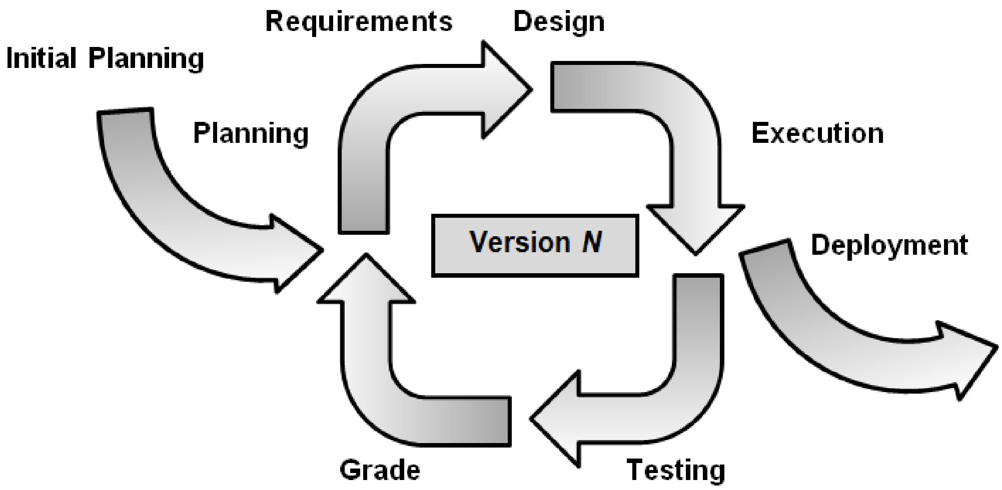

Fig. 3. Incremental model

Flexible model (Agile). A popular approach that focuses on flexibility, collaboration and process optimization to deliver a quality project. It is an iterative approach that prioritizes

 ISSN 1560-8956

feedback from the software owner and adaptation to changing requirements. The Agile software development cycle can be broken down into six phases: planning, design, development, testing, deployment, and maintenance [17].

It is advisable to use flexible methodology:

— when needs change dynamically;

— if frequent changes are necessary;

— for large and long-term projects that are constantly modified to suit market conditions.

Spiral model. Proposed by Barry Boehm in 1988, it became a significant breakthrough in understanding the nature of software development (Fig. 4) [5,9,11,12].

Boehm's spiral model focuses on design. In fact, software development occurs only at the last turn of the spiral according to the usual waterfall model, but this is preceded by several design iterations based on the creation of prototypes - and each iteration includes the stage of identifying and analyzing risks and the most difficult tasks. Since the spiral model mainly covers design, in its original form it was not widely used as a method for managing the entire life cycle of software creation. However, its main idea, which is that the process of working on a project can consist of cycles going through the same stages, served as the starting point for the development of the information and infological model for managing the process of developing an ISP, proposed below.

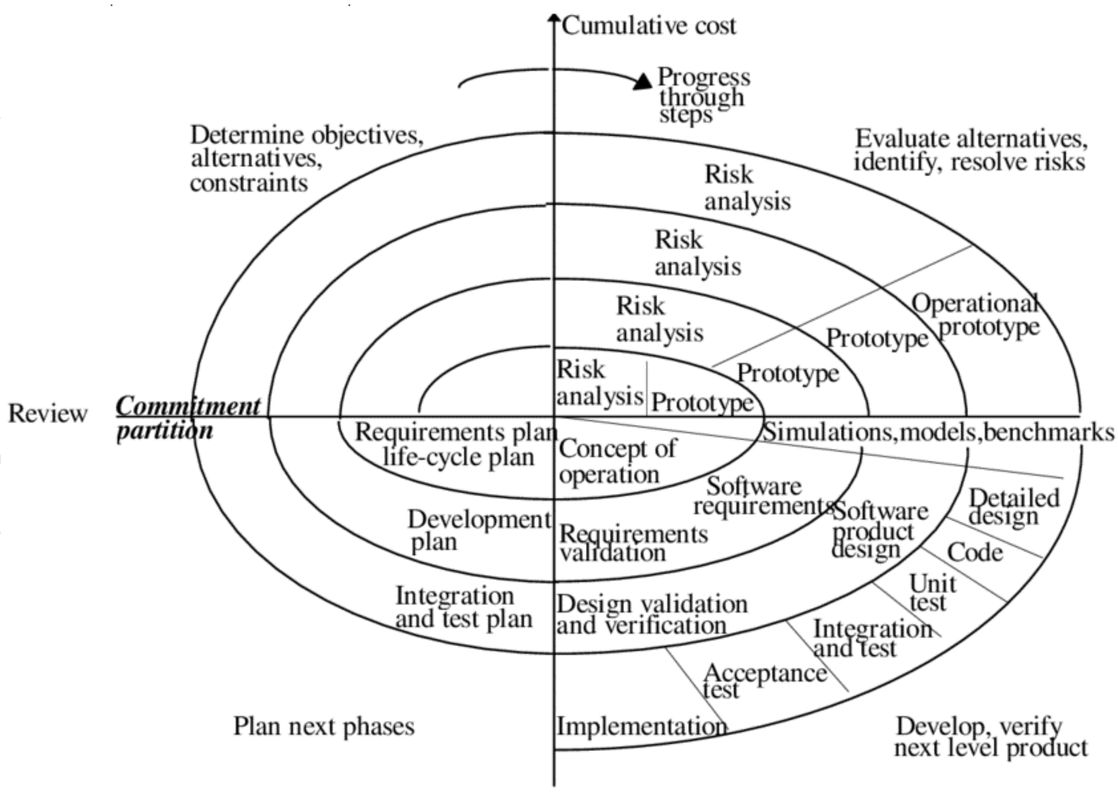

Figure 4. Spiral model by Barry Boehm

Infological model for managing the ISP development process

The essence of the proposed approach is that the process of developing an ISP is interpreted as an information object that changes in content and structure in the process of its creation [12]. Consequently, the process of developing an ISP can be described by an ordered sequence of states of the developed ISP, the last of which represents the finished software product, i.e.

$$S_o \rightarrow S_1 \rightarrow S_2 \rightarrow \ldots \rightarrow S_i \rightarrow \ldots S_n. , \qquad (1)$$

Here, for each state S_i, a certain set of $m_i^1\ m_i^2 \ldots m_i^m$, где $i=\overline{0,n}$ is determined, and characterizes the degree of completion of the ISP development.

We will further assume that each intermediate state S_i corresponds to two integral estimates P_i and Q_i, which exhaustively characterize the degree of completion of the ISP from the quantitative and qualitative sides. Obviously, the functions P and Q on the ordered set of states S_i ($i=\overline{0,n}$) must have an increasing character. For the sake of completeness in describing the process of developing an ISP, we will assume that the integral estimates P and Q are independent of each other.

Further, the ISP development process is divided into separate sub-processes (development steps) corresponding to the development stages adopted according to (1). Let us denote the increment of integral characteristics achieved at the i-th stage through ΔP_i and ΔQ_i.

The managing process of the development of ISP consists in the fact that at each step its control action $u(i)$ is set, which determines the values of ΔP_i and ΔQ_i, and transfers the degree of development of ISP from the state (P_{i-1} , Q_{i-1}) to the state (P_i,Q_i). Management $u(i)$ can be considered as a choice of one of the alternative ways to ensure innovativeness of the project being developed. In this case, the transfer to a new state is realized by performing a certain set of procedures.

Naturally, at each step i, a number of natural and artificial restrictions are imposed on the control action $u(i)$. Otherwise, $u(i)$ can take values from some set of possible control actions, i.e.

$$u(i) \in V(i). \qquad (2)$$

We will assume that for i=0 the values $P_0 = Q_0 = 0$.

The values of the integral characteristics in the subsequent steps will be determined by the formulas:

$$\begin{cases} P_i = \varphi(u(k), P_{i-1}); \\ Q_i = \gamma(u(k), Q_{i-1}); \\ (P_i, Q_i) = f(u(i), (P_{i-1}, Q_{i-1})); i = \overline{1,n} \end{cases} \qquad (3)$$

By (P_i ,Q_i) we mean the set of all states of the ISP development process, to which it can be transferred from the initial state in i steps, using the control action $u(k) \in V(k), k = \overline{1,\imath}$.

We call such a set the reachability set (P_i,Q_i), which is defined using recursive relations (4) of the form

$$\begin{cases} (P_k, Q_k) = F[u(k), (P_{k-1}, Q_{k-1})] \\ u(k) \in V(k), k = \overline{1, \iota}, i = \overline{1, n} \end{cases} \quad (4)$$

The terms of reference for the development of an ISP must specify the requirements that an ISP must satisfy after its development is completed. Based on this, it is possible to determine the indicators P_n and Q_n, characterizing the final state of development, which should belong to a certain range of acceptable values ($P_n^`$, $Q_n^`$), i.e.

$$(P_n, Q_n) \in (P_n^`, Q_n^`) \quad (5)$$

Thus, the process of developing an ISP with control actions $V(u(i))$, will be admissible if $u(i)$ will transfer the ISP from the initial state to the final one, which will satisfy condition (5).

Based on this, in order to successfully achieve the goal - the development of ISP, it is necessary to fulfill the condition

$$(P_i, Q_i) \cap (P_i^`, Q_i^`) \neq \emptyset, i = \overline{1, n} \quad (6)$$

Condition (6) means that the set of all ISP development states must be in the set of admissible ISP states in accordance with the requirements. Otherwise, with a changed innovation forecast, it is necessary either to change the terms of reference for development, thereby changing $(P_i^`, Q_i^`)$, $i = \overline{1, n}$, or to expand the area of possible control actions $u(i)$, $i = \overline{1, n}$.

Let, as a result of the execution of (i-1) steps, the ISP development process enters the state (P_{i-1}, Q_{i-1}). Then the set of admissible control actions at the i-th step is determined as follows:

$$\begin{cases} V^`(i) = \{u(i): (P_i, Q_i) = f[u(i), (P_{i-1}, Q_{i-1})]\} \\ (P_i, Q_i) \in (P_i^`, Q_i^`), i = \overline{1, n} \end{cases} \quad (7)$$

As a result, the process of managing the development of ISP in its final form can be written as:

$$u(i) \in V(i) \cap V^`(i), i = \overline{1, n}. \quad (8)$$

Condition (8) means that from the point of view of the innovations that have arisen during the development of ISP, it is possible to change the control actions in acceptable values in accordance with changes in the current and final requirements. Condition (8) can be satisfied at each step by several control actions.

Implementation of decision making in IDSS

As mentioned above, the most effective tool for making a potentially better decision are *Intelligent Decision Support Systems* (IDSS) [13,14]. The greatest effect from the introduction of IDSS is achieved where, when making decisions, poorly formalized factors (economic, political, social) are taken into account along with economic indicators, which is inherent in the majority of objects and subjects for which ISP is developed. IDSS is a universal analytical tool that can be used in any application area where, as a result of certain analytical and logical procedures, it is necessary to make an innovative decision [13]. Fig. 5 shows a generalized structure of the IDSS for making innovative decisions in the process of developing ISP. This structure of the IDSS provides, in dialogue with experts, automated adjustment to the subject area under study by introducing into the system basic concepts, attributes, their possible meanings, connections between them, as well as types of possible situations characteristic of a given subject area. A knowledge base (KB) is defined as a knowledge representation model, which is often called simply a knowledge model (KM). This IDSS is based on the use of specialized information warehouses (Data Warehouse) and OLAP (On-Line Analytical Processing) technologies – dynamic multidimensional data analysis using an effective tool for data mining (Data Mining), modeling and forecasting [15,16]. Such IDSS can be customized to develop an ISP in a specific subject area.

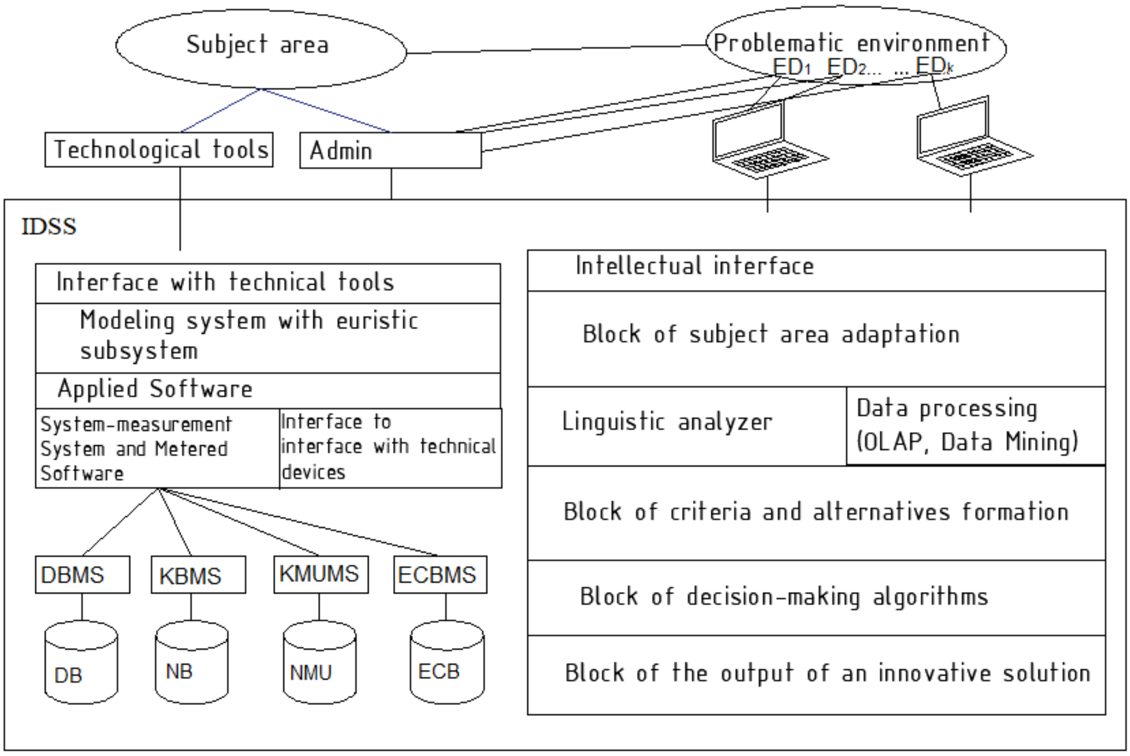

Fig. 5. Generalized structure of IDSS (DBMS, KBMS, KMUMS, ECBMS – management systems of database, knowledge base, knowledge model update, evaluation criteria base, respectively; ED_1, ED_2 ... ED_K – expert designers)

This paper proposes to build IDSS based on ontological analysis of the subject area and ISP development process for this subject area [18]. To structure domain knowledge, this IDSS uses three main categories of ontologies [19]:

— static ontology – describes the terms of the subject area, their properties and relationships;

— dynamic ontology – determines the states that arise during the functioning of the system, the method of transforming some states into others;

— epistemic ontology – describes knowledge at the level of a conceptual model that controls the process of transition of a system from one state to another.

One of the advantages of using ontologies as a research tool is a systematic approach to studying a subject area. In this case, a holistic view of the subject area is achieved; uniformity of the material, which facilitates the process of perceiving information; completeness of the display – the construction of an ontology allows you to cover, among other things, the missing logical connections between the concepts of the subject area.

In this case, the decision-making process at each stage of ISP development, according to the infological model for managing the ISP development process, can be represented by the following diagram (Fig. 6).

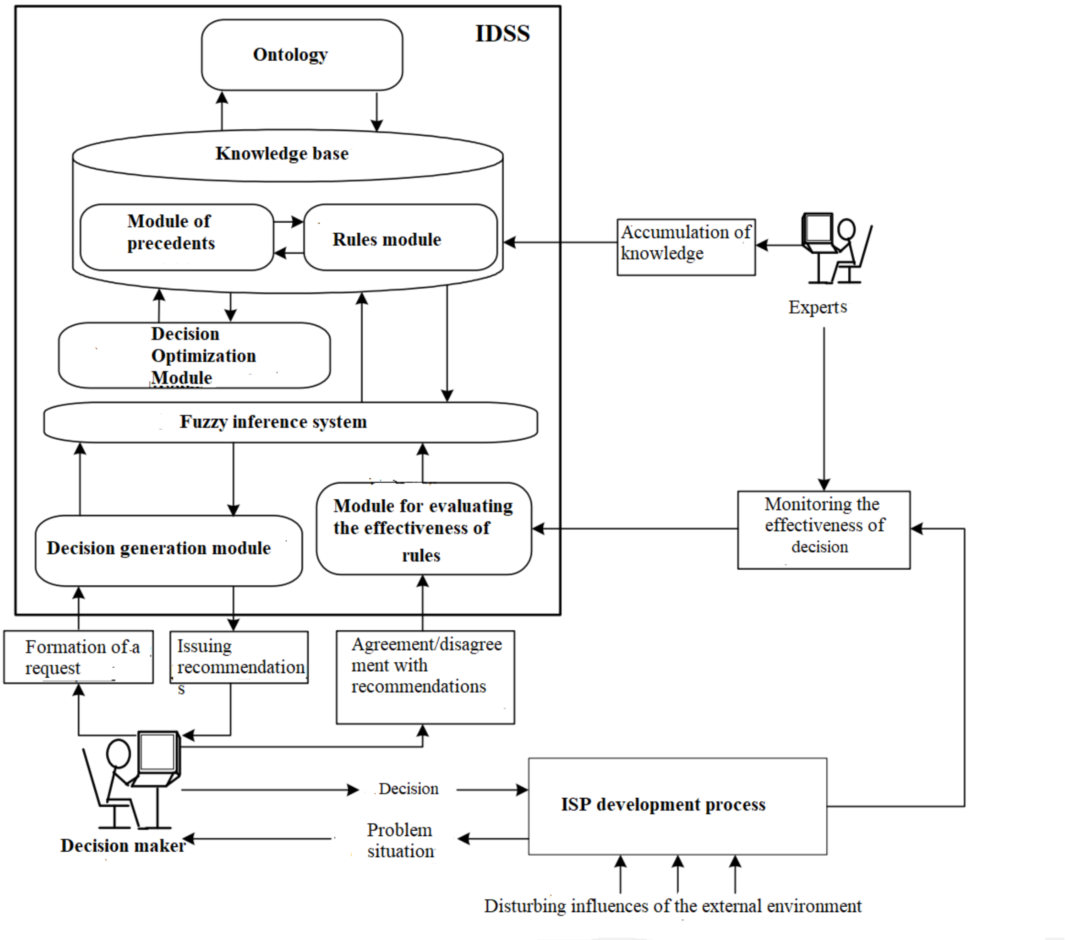

Fig. 6. Scheme for choosing a decision using IDSS

This IDSS performs the formation of decision alternatives (search for possible solution options) and evaluation of alternatives (comparison of options), as well as the presentation of recommendations to users. Adoption the final decision remains with the decision maker (DM).

The proposed IDSS provides for monitoring the effectiveness of decisions made as a result of the implementation of recommendations given by the system, which is based on an assessment of the effectiveness of the rules contained in the knowledge base.

Alternative decisions in this IDSS can be formed both quantitatively and verbally. In this case, to evaluate a set of alternatives, it is advisable to use intelligent methods, in particular, fuzzy logic methods, which allow automatic ranking of alternatives based on decision rules.

To ensure the possibility of evaluating alternatives constructed on the basis of linguistic variables, a scheme of the fuzzy inference procedure is used (Fig. 7) [20].

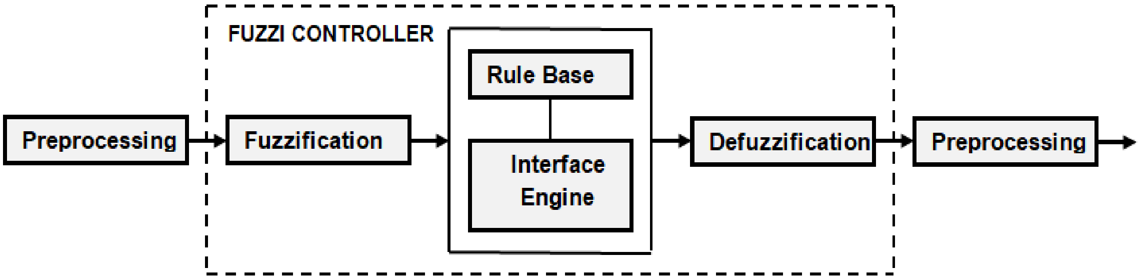

Fig. 7. Scheme of the fuzzy inference procedure

In today's technological world, AI-powered tools have become indispensable for software developers, UI/UX design, and quality assurance teams. These artificial intelligence tools can optimize processes, improve creativity, and improve the overall quality of software products.

Conclusion

The obvious fact is that software products have an important feature: knowledge about them has a relatively large dynamic component, which must be taken into account in operational management. This is due to the evolution of knowledge in the subject area. In addition, since the API development process is multi-alternative, the problem of multi-criteria arises, which, as a rule, requires the involvement of IDSS. The use of IDSS is also due to the fact that, firstly, the presence of the human factor in the process of developing ISP introduces a large share of uncertainty, and, secondly, it is necessary to look at the entire range of acceptable solutions in the field of using ISP, which requires the development of automated methods for extracting knowledge of a given subject. areas from various sources of information, including Internet resources. This article proposes an infological model of the process of managing the development of an innovative software product. The essence of the proposed approach is that the development of an innovative software product is treated as an information object that changes in content and structure during the process of its creation. The structure of an intelligent decision support system is proposed, which allows making innovative decisions at each stage of the process of developing an innovative software product.

 ISSN 1560-8956

REFERENCES

1. *Agustin Yague*. Analyzing software product innovation assessment by using a systematic literature review/ Hawaii International Conference on System Science, IEEE, 2014. -P.5049-5058. DOI 10.1109/HICSS.2014.620.

2. *Paul Quintas*. A product-process model of innovation in software development/Journal of Information Technology, volume 9, 1994. -P.3–17.

3. *Paul Quintas*. Programmed Innovation? Trajectories of Change in Software Development / Information Technology & People, Vol.7, No. 1, 1994. - P.25-47.

4. *Lipaev V.V. and others*. Technology of designing complexes of automated control systems programs. - M.: Radio and Communications, 1983. - 235 p.

5. *Boham B.W.* Software engineering design. - M.: Radio and Communications, 1985. - 512 p.

6. *Vendrov A.M.* CASE technologies. Modern methods and tools for designing information systems. - M.: Finance and Statistics, 2000. - 352p.

7. *Winston W.* Royce Managing the Development of Large Software Systems Proc. IEEE WESKON, 1970, №6. –pp. 1-9.

8. *Orlov S.A.* Software development technologies. – SPb.: Peter. -2002. – 464 p.

9. *Richard W. Selby*. Software Engineering: Barry W. Boehm's Lifetime Contributions to Software Development, Management, and Research. — John Wiley & Sons, 2007. - 832 p.

10. *Dustin E., Reshka D., Paul D.* Automated software testing. – M.: Lori. -2003. – 590 p.

11. *Solovyyov N. A., Chernoprudova E. N.* Automation systems for software development. – Orenburg: LLC IPK “University”, 2012. – 230 p.

12. *Stenin A., Melkumyan K., Soldatova M., Polshakova O.* Infological model development of innovative software product / Adaptive automatic control systems. No. 1 (38), 2021. - P. 18-27.

13. *Osipov V. P., Sivakova T.V., Sudakov V.A., Nechaev Yu.I.* Intellectual core of the decision support system / / Preprints of the IPM named after M. V. Keldysh. 2018. No. 205. 23p. DOI: 10.20948/prepr-2018-205 URL: http://library.keldysh.ru/preprint.asp?id=2018-205.

14. *Kaklauskas A.* Intelligent Decision Support Systems. In book: Biometric and Intelligent Decision Making Support.___Intelligent Systems Reference Library. Vilnius Gediminas Technical University, 2015. –P.31-85. DOI: 10.1007/978-3-319-13659-2_2.

15. *Barseghyan A.A., Kupriyanov M.S., Stepanenko V.V., Kholod I.I.* Data analysis technology. Data Mining, Visual Mining, Text Mining, OLAP. - St. Petersburg: Publishing house BHV, 2007. -384 p.

16. *Ostrovskaya K.Yu.* Further research into methods of intelligent data analysis for processing test results. System technologies, No. 4(129), 2020. -P.146-157. DOI 10.34185/1562-9945-4-129-2020-1.