Application of generative artificial intelligence for risk management of software projects

Anna Dzheiranian
Department of Business Informatics
HSE University
Perm, Russian Federation
ORCID: 0009-0000-8916-2855

Mikhail Plaksin
Department of Business Informatics
HSE University
Department of Computer Science
Perm State University
Perm, Russian Federation
ORCID: 0000-0002-6288-8610

***Abstract* — The article highlights an innovative approach to risk management in software projects using generative artificial intelligence. It describes a methodology that involves the use of publicly available chatbots to identify, analyze, and prioritize risks. The Crawford method is used as a basis for risk identification. The authors propose specific formulations of requests to chatbots (instructs, prompts) that facilitate obtaining the necessary information. The effectiveness of the methodology has been demonstrated on five small software projects and a dozen of economic and organizational projects of significantly different scales, from small to federal. This confirms its applicability and practical value.**

***Keywords* — generative artificial intelligence, chatbot, risk management, risk identification, Crawford cards, software projects**

I. Introduction

The object of the proposed research is a generative model of artificial intelligence (GAI). The subject is the use of GAI for risk management of software projects.

The purpose of the research is to study the possibility of using GAI to manage risks arising during the implementation of software projects.

The study has the following results:

1) A conclusion on the possibility of using GAI in risk management of software projects has been provided.

2) A methodology for using GAI has been developed, and a seguence of prompts that can be used for this aim has been proposed.

3) A comparison of several publicly available GAI chatbots has been conducted.

The relevance of the research is determined by two factors:

1) In the modern world, neural networks are becoming an integral part of everyday life. Generative artificial intelligence stands out as a powerful tool capable of significantly simplifying and accelerating the solution of many tasks, the scope of which is yet to be defined. At the same time, the use of GAI tools fundamentally differs from the use of all other software systems. All other – "traditional" – software systems are initially created to solve some specific tasks. A "traditional" computing system behaves like an automaton, controlled by a set of commands known in advance (although it can be quite complex). There are instructions that describe how exactly to control such an automaton. The use of these software tools is stable in the sense that the same human actions cause the same reaction and lead to the same results. In contrast to "traditional" software systems, the behavior of GAI is extremely unstable. Repeating the same human actions can lead to very different consequences. There is no instruction that would define in advance what exactly should be done with GAI systems to achieve this or that result. Humanity has invented a new entity that behaves in some independent way and communication with which is yet to be learned.

2) Risk management is an important part of information system development management. To assess the relevance of improving risk management methods in informatics, it is enough to look at the results of research on the success of software projects [1]–[5]. The success statistics of software development projects can be roughly represented as a (slightly skewed) normal distribution curve. At one end will be successful projects, i.e., those that were completed on time, stayed within the planned budget, and implemented all the declared capabilities. The share of such projects according to the results of various studies fluctuates around 25-35%. On the other side will be projects that failed and were never completed. There are about 20% of such projects. In the middle are "controversial" projects that were completed, but either exceeded the planned deadlines, did not fit into the budget, or did not implement all the planned capabilities. The share of such projects is about half. This picture changes somewhat in studies of different years. Especially if the study takes into account the size of the project. But overall, it remains quite stable.

The low level of success indicates that the management of most projects is not able to identify all the risks threatening the project and react correctly to them. Risk management remains more of an art than a craft. The identification and analysis of risks are largely the result of using expert methods, such as the Crawford method [6], affinity diagrams [7], Ishikawa diagrams [8], fail stories, diversion analysis [9], rough models (Fermi method) [10], etc. In computer science, there are no yet accepted standardized methods, similar to the FMEA method [11], which has proven itself in mechanical engineering and several other industries. Therefore, any research aimed at turning risk management of software projects from art into technology is considered useful.

II. The MSF Risk Management Process

This study uses a risk management methodology that is a part of the Microsoft Solution Framework for Agile (MSF) technology [12].

According to MSF, the continuous risk management process consists of six stages: (1) identifying risks, (2) analyzing and prioritizing risks, (3) planning risks (assigning prevention and response plans for risks and comparing different plans for the same risk), (4) monitoring risks, (5) implementing risk responses (adjusting the project in response to realized risks), and (6) learning lessons (learning about risks).

We will be interested in the first three stages: risks identification, the analysis and prioritization of risks, and developing plans.

MSF defines risk as “any event or condition that can have a positive or negative impact on the project outcome”. Risk is characterized by the probability of a risk event occurring and the impact that the fact of this event will have on the project. To combine these two characteristics together, the concept of risk magnitude is introduced. The risk magnitude is calculated as the product of the risk probability and its impact.

MSF technology provides for the construction of two plans (a risk prevention plan and a risk response plan):

- The prevention plan includes actions aimed at reducing the likelihood of a risk occurring and reducing its potential threat to an acceptable level. These actions must be performed in advance.
- The mitigation plan includes responsive actions that need to be taken if the risk could not be prevented and the risk event occurred. This plan is put into action if a certain predetermined condition (risk trigger) is met.

III. Requirements and Project Selection

It was necessary to select software projects to conduct the research. There were only two requirements for the projects: they must have a brief description and they must be noticeably different from each other. The brevity of the description was determined by the quantitative limitations of the listed chatbots. The need for noticeable differences in projects was dictated by the fact that we were interested in the extent to which the GAI chatbots would consider the features of a particular project in their recommendations.

Thus, the study examined five small software projects: (1) “Designing data visualization tools based on a language-oriented approach”, (2) “Trade and information software system for managing vending machines”, (3) “System for printing photographs and magnets from company Lomobil”, (4) “Mobile application “Virtual parking”, and (5) “Searching for optimal advertising for business according to specified parameters”.

The selection of projects for the study was conducted almost by chance (“they happened to be at hand”). Two descriptions were compiled for the projects: a brief one and a more detailed one. It turned out that the level of detail of the project description plays a role. In the case of more detailed descriptions, the chatbots used information about the project details in their responses. As a result, chatbot responses were more exact and meaningful.

IV. Methodology of Using GAI for Risk Management

A. *Overview of the Crawford’s method*

The Crawford method [6] is used to identify risks. This is a modification of brainstorming in which special measures for avoiding the anchoring effect are taken. In real life it looks like this.

A group of experts (7-10 people) is formed. Each of them receives a pack of numbered cards. On the first card, each expert writes down the risk that he considers the most important for the project being analyzed. The presenter collects the completed cards.

After this, on the second card, each expert writes down the most important of the remaining risks (the second most important risk). The completed cards are collected again by the presenter. And so on for a specified number of times.

After this, risks are discussed and grouped. Different experts may describe the same risk in different words. These facts are revealed during the discussion. Same and similar risks are grouped. Physically, this is expressed in the formation of so-called affinity diagrams (the grouped cards are fastened with tape into a vertical strip) [7]. The length of the tape clearly demonstrates the importance of this risk from the point of view of the expert community. First, all risks of the first rank are discussed, then all risks of the second rank, etc. As a result, a set of risks is formed and sorted by importance. The importance of a risk is determined by the number of times that risk is mentioned by different experts.

B. *Assigning a chatbot a role*

The GAI chatbot acts as the manager of project that needs to be analyzed. The first instruction looks like: “You are an experienced project manager. You have been entrusted with the leadership of the next project…” The following is a description of the project.

C. *Providing a descriprion of the project*

The project description can be given at different levels of detail. The level of detail plays a role. The more detailed the project is, the more details the “experts” will be able to use when evaluating it.

D. *Formation of a group of experts*

Experts are generated by the chatbot as members of a team, which it manages as a project manager.

For each expert, his specialization is indicated. The composition of the team of experts is determined by the subject area to which the project relates. For software projects, these were business analysts, programmers, testers (quality assurance specialists), “human-computer” interaction specialists, and logisticians (specialists in the deployment of software systems). The specifics of the project may require the involvement of particular experts from other fields.

The qualifications of an expert can be described at different levels. One may limit oneself to phrases such as “experienced professional”, “has work experience of more than ten years” (or vice versa “less than three years”). Alternatively, one can provide a detailed listing of the expert’s knowledge, skills, certificates, etc.

NB! A more detailed description, as a rule, does not provide any benefit compared to a brief description such as “more than ten years of work experience”. There is a fundamental difference with the project description. There the detail of the description plays a role, here it does not. The reason is unknown.

To obtain a detailed description of the expert’s qualifications, one can use the same chatbot or another. It is enough to give it a request: “You are the project manager. You need to hire three business analysts: a junior, a middle and a senior. List what knowledge, skills and abilities each of them should have". But – we repeat once again – we did not find any sense in such detail.

An instruction for generating a team of experts may look, for example, like this: "Your team consists of experts: two business analysts, two programmers, two economists, two lawyers. The first expert in each pair has at least ten years of experience, the second – no more than three".

Experts can be given names: "Anna is an experienced business analyst", "Boris is an experienced programmer", etc. After that, one can call them by name: "Let Anna do this", "Let Boris do this", etc. This can be useful since one have to contact the same expert several times.

E. *Survey of experts*

After describing the experts, they are interviewed according to the following scheme:

1) "Let the expert name the characteristics of the project that seem significant to him".

2) "Let the expert name the risks of project implementation that are associated with these characteristics and justify his opinion".

3) "Let the expert evaluate the probability of the risk occurring on a given scale and the damage from the realization of the risk on a given scale" (at the same time, specific numbers in the chatbot's responses cannot be trusted).

4) "Let the expert name the means to reduce the likelihood of the risk occurring and to reduce the damage if it occurs".

5) "Let the expert name indicators by which it can be judged that the danger of this risk occurring is increasing" (it has not yet been possible to get a good answer to this question from the chatbot).

F. *Grouping of risks*

Since in Crawford's method the importance of a risk is determined by the number of times it was named by experts, one can ask the chatbot to classify risks by similarity of wording: "Group similar risks. Indicate which expert named each risk and how exactly it was formulated".

V. Interesting Points in Building a Methodology of Using GAI for Risk Management

Next, some interesting aspects of communicating with GAI chatbots are discussed.

A. *Ways to organize a collective survey of experts*

The Crawford card method is a method of collective examination. The question about how to organize the collective work of a group of experts using one GAI chatbot arose. One chatbot had to speak on behalf of several (in our case, first nine, and then ten) experts. A contradiction arose: it was necessary to communicate with one chatbot and at the same time communicate with ten experts. This contradiction could be resolved in three ways: (1) division in time, (2) division in space, and (3) division in structure. The time separation implied that the dialogue with the chatbot would be repeated several times. Moreover, each time the bot will be given the task of presenting itself as a new expert. The spatial separation meant that the dialogue would be run on several different computers with multiple instances of the bot. It was not clear whether these several instances would behave differently "on their own" or whether each of them would have to be given some kind of input. The division in the structure assumed that a certain structure would be built. That structure would allow to communicate with one interlocutor (assign only one role to the chatbot), but at the same time receive answers from several different experts.

The latter option was attractive due to its efficiency. It was the one chosen. We took advantage of the fact that the examination process is conducted collectively and this process has a leader. It was the presenter who was chosen as our interlocutor. He was asked to carry out the entire procedure "inside the chatbot" and immediately provide us with its results.

B. *A way to achieve a variety of expert opinions*

After this, the question arose about the formation of an "expert group". Initially, the request simply referred to "experts" without any specification. But this approach did not provide a diversity of opinions. The answers given by different experts were often repeated (more precisely, the first three experts named nine different risks, and all the remaining experts repeated the same nine risks in different combinations). And the basis of the Crawford method is precisely that different experts will have different views on the same problem. An attempt was made to solve this problem head-on. The chatbot was asked to name thirty different risks (more precisely, the request was supplemented with the following requirement: "Consider the fact that each expert has different professional experience and knowledge. During brainstorming, they generate a variety of ideas. That is, risk formulations cannot be the same among experts, but may imply one and same risk"). Which is what the bot did. And thus again came into conflict with Crawfod's method. According to Crawford, the importance of a risk is determined by how many experts name it. That is, it is impossible to prohibit different experts from calling the same risk (possibly describing it in different words). But each expert had to name the risks in accordance with his expert vision.

The next solution was found. We abandoned the idea of "experts in general" and decided to clarify the specialization and experience of each expert. As a basis for determining the nomenclature of experts, a development team based on the Microsoft Solution Framework for Agile technology [12] was taken. For greater variety, the work experience of each expert has been added.

As a result, the beginning of the request to the chatbot took the form: "You are the manager of a software project that is planned to be implemented. It is necessary to identify the risks that exist during its implementation. For this you have a group of 10 experts. Two of them are business analysts, two are programmers, two are QA specialists, two are user experience specialists, two are software product deployment specialists. The first specialist in a pair is experienced (at least 10 years of work experience), the second is a beginner (up to three years of work experience). Let each expert name the characteristics of the project that seem important to him. The following is a description of the project".

C. *Instability of responses*

In ordinary consciousness, a computer is perceived as an automaton that operates according to a certain algorithm, converting input data into output data. Under these conditions, it is logical to expect that the same answer will be given to the same request (or – taking into account the experience of searching for information on the Internet – the answer received when repeating the request will be close to the answer to the first request).

It turned out that this is not the case for the GAI. The answer received when repeating the same request could differ significantly both in content (the same experts began to list other risks) and in form. For example, a couple of times the YandexGPT [13] responded to a request with the following phrase: "I'm not in the mood today. Come back another time!"

D. *Chatbots are confident in their knowledge*

GAI chatbots are sure in their know-it-all nature. In several dialogues, a following sentence was added to the request: "If any expert does not have enough information to make a decision, let him ask for the missing information". None of the chatbots responded to this proposal. There was always enough information for all the "chatbot experts".

A traditional expert system is able to understand that it is missing some information and request it. A search engine (a la Google) may indicate that during a search it did not find certain words from the query or did not find an answer to the question at all. GAI chatbots are confident that they have all the necessary knowledge to answer any request. With the exception of the boundaries of political correctness that are unacceptable from the point of view of the chatbots' authors (but this is another aspect of GAI).

E. *Loss of dialogue context*

Theoretically, each chatbot undertakes to remember a certain number of previous replicas and respond to the next replica, considering this context (e.g., one session of dialogue with a BING AI [14] can consist of no more than 30 requests). In practice this is not always the case. Such problems arose especially often when communicating with ChatGPT [15].

Related to the issue of loss of context is the issue of the size of the information provided by the chatbot. The output volume of all chatbots is limited. Some chatbots can request "continued issuance". Some of them must be asked to do this "manually". At this moment, a loss of context and a violation of the form of information delivery often occurs. To bypass these restrictions, it makes sense to change the request to the chatbot and split it into several replicas.

F. *The degree to which project specifics are reflected in risks*

Links that reflect the features of the project are quite rare. Here is an example for a project "Designing data visualization tools based on a language-oriented approach" (quoted from a conversation with BING AI, risks identified by a developer, links to project features are in italics):

1) "Lack of time to implement the project. (Rationale: The project is complex and takes a long time to develop, especially since *there is only one person on the team*)".

2) "Lack of experience in working *with domain-specific languages*. (Rationale: Working *with domain-specific languages* requires specialized skills and knowledge that may be in short supply)".

3) "Difficulty in *creating interactive chart customization*. (Rationale: *Creating interactive charts* is a complex task that requires a deep understanding of user needs and *data visualization technologies*)".

But from the point of view of "area fire" chatbots work well. As a criterion, we took the classic list of risks of software projects [16]. It turned out that the GAI "closes" this list by 70%.

In one of the experiments, risk identification was carried out first by people "manually", and then using a chatbot. The relationship turned out to be as follows. A total of 14 risks were identified: 9 were identified by both humans and the chatbot, 1 by only humans, 4 by only the chatbot.

G. *Correctness of risk grouping*

How correctly does the GAI group the responses of different experts? Differently. The association was purely formal, almost meaningless. There were collections of risks that were identical in meaning, but with different names. There also was a grouping of risks associated with one problem but understood by different experts from different points of view. This is another example of chatbot instability.

VI. CHATBOT OVERVIEW

In this study, four publicly available chatbots were used as GAI tools: YandexGPT 2 [13], BING AI (Copilot) [14], ChatGPT [15], and GigaChat [17]. All of these are capable of generating human-like text based on context and past conversations.

As a result of the analysis, no significant differences were found among the chatbots in terms of the tasks we solve. Subjectively, BING AI appeared to be more effective, followed by ChatGPT in second place, YandexGPT in third, and GigaChat in last place. Therefore, further research will be based mainly on the example of BING AI.

Nevertheless, the reader may wonder why the most popular ChatGPT is not ranked first. The reason is that it often lost context in our experiments. In contrast, BING AI showed more stability in its responses, possibly due to the use of the exact generation mode. However, it should be emphasized again that this rating is subjective. Under other conditions, the opinion could be different, as bots show variability, instability in their answers, which also leads to the impossibility of creating objective criteria.

VII. EVALUATION OF APPROACH EFFECTIVENESS

The main limitations that must be considered when applying GAI (both in risk management and in any other area):

1) GAI does not understand the meaning of the task proposed to it. Its answers are the result of some probabilistic algorithms. This is important to keep in mind, since during a dialogue with a chatbot it is easy to create an illusion of the reasonableness of some of its answers. The probabilistic nature of the GAI's responses leads to very high instability of its work. The same human actions can generate completely different reactions from the chatbot.

2) In particular, this means that the effectiveness of using any communication methods with GAI chatbots is not guaranteed.

3) GAI knowledge is very limited (despite trillions of parameters and a phenomenal number of texts loaded into the neural network). In reality, a person receives a huge amount of information in non-verbal form. This information is not available to neural networks.

4) GAI is not aware of the boundaries of its knowledge and ignorance. It will never understand that it doesn't know something. A chatbot is sure that it knows everything.

5) GAI is not able to rationally explain the answers it produces.

6) Moreover, GAI can give completely incorrect answers ("halucinate"), but categorically insist on their truth.

7) GAI has some information "in general", but does not have information about each specific project. This means that the neural network is not able to consider the specifics of each certain project. More precisely, this accounting depends on three factors:

a) The details of the project description that a person will provide to the chatbot.

b) The size of the context that the chatbot takes into account when preparing a response

c) The probabilistic (i.e., unpredictable) nature of developing an answer. Therefore, the chatbot's responses can be perceived in terms of "area fire", but not in terms of describing a specific situation.

The main advantages of GAI in terms of identifying and analyzing risks:

1) Extensive knowledge in a given subject area (taking into account the lack of understanding of the specifics of a particular project, a specific situation, and the fact that this knowledge still needs to be able to be extracted). Experience has shown that chatbots do a good job of identifying risks at a "fundamental level".

2) Extensive knowledge of subject areas related to the project. Opportunity to evaluate the project from different points of view.

3) The ability to quickly involve many experts of different specializations in the work and obtain group expertise of the project.

4) The methodology, originally developed for software projects, was adapted to apply it to economic and organizational projects of significantly different scales, from small to federal. To assess the latter, indicators of national security of the Russian Federation were used. The transfer was successful [18]. The technique works in other subject areas too. Naturally, each field requires its own set of experts.

VIII. Conclusion

During the study, the use of GAI in the process of risk management of software projects was tested. The study demonstrated the fundamental suitability of the GAI for this task, despite the disadvantage of operational instability. However, the authors have not yet ensured that the GAI takes the features of a particular software project seriously enough into account. It is more about "area fire". And the GAI copes well with this and is not inferior to other sources of risks information.

A methodology for organizing a group examination, simultaneous survey of a "group of GAI-experts" using the GAI chatbot was proposed. The article provided the texts of the relevant requests and examples of their execution. Their advantages and disadvantages were discussed.

In addition, the proposed methodology turned out to be suitable not only in relation to software projects, but also in other subject areas (e.g., risk assessment of economic projects).

Directions for further research have been identified:

- To ensure that the specifics of the projects are taken into account when identifying risks.
- To improve the virtual "community of experts".
- To ensure that the expert provides a verifiable justification for the likelihood and impact of the risk.
- To improve the quality of risk indicators proposed by the experts.
- To continue expanding the proposed methodology to other subject fields.

References

[1] S. Prabhakar and P. Seetharamaiah, "Organizational Strategies and Social Interaction Influence in Software Development Effort Estimation," IOSR Journal of Computer Engineering, 2014, vol. 16, no. 2, pp. 29–40.

[2] Y. Quan and L. Zi-quan, "Research on Risk Evaluation and Risk Optimization of IT Projects," 2008 International Conference on Information Management, Innovation Management and Industrial Engineering, Taipei, Taiwan, 2008, pp. 119–123.

[3] R. Stern and J. C. Arias, "Review of Risk Management Methods," Business Intelligence Journal, 2011, vol. 4, no. 1, pp. 51–78.

[4] S. J. Andriole, "Editorial: Where did IT go," International journal of technology management, 2022, vol. 89, no. 1/2, pp. 1–1.

[5] A. S. Khatavakhotan and S. H. Ow, "Improving IT Risk Management Process by an Embedded Dynamic Verifier Core: Towards Reducing IT Projects Failure," 2012 Third International Conference on Intelligent Systems Modelling and Simulation, Kota Kinabalu, Malaysia, 2012, pp. 684–687.

[6] "The Crawford Method." [Online]. Available: http://pmpro.ru/metod-krouforda.html

[7] "Affinity Diagram – Kawakita Jiro or KJ Method." [Online]. Available: https://project-management.com/affinity-diagram-kawakita-jiro-or-kj-method/

[8] K. Ishikawa, "What is Total Quality Control? The Japanese Way," Prentice Hall, London, 1985.

[9] "Anticipatory Failure Determination (AFD)." [Online]. Available: https://www.whereinnovationbegins.net/

[10] W. Poundstone, "Are You Smart Enough to Work at Google?" Little, Brown Spark, New York, 2012.

[11] R. J. Mikulak, R. E. McDermott and M. R. Beauregard, "The basics of FMEA," Productivity Press, New York, 1996.

[12] Microsoft Corporation, "Microsoft Solutions Framework MSF Risk Management Discipline," Seattle, 2002, vol. 1.1.

[13] Chatbot "YandexGPT 2." [Online]. Available: https://yandex.ru/project/alice/yagpt

[14] Chatbot "Bing AI." [Online]. Available: https://www.bing.com/

[15] Chatbot "ChatGPT." [Online]. Available: https://gpt-chatbot.ru/

[16] B. W. Boehm, "Software risk management: principles and practices," in IEEE Software, 1991, vol. 8, no. 1, pp. 32–41.

[17] Chatbot "GigaChat" [Online]. Available: https://developers.sber.ru/gigachat/login

[18] A. D. Dzheiranian and M. A. Plaksin, "Application of generative artificial intelligence to assess the risks and safety of federal projects," International Scientific and Methodological Conference, Minsk, Belarus, 2024, vol. 2, no. 2, pp. 294–298.