



УДК 004.42; 519.85

АВТОМАТИЗАЦИЯ УПРАВЛЕНИЯ РИСКАМИ ПРОГРАММНЫХ ПРОЕКТОВ НА ОСНОВЕ НЕЧЕТКОГО ЛОГИЧЕСКОГО ВЫВОДА

Т.М. Зубкова^a, Е.Н. Ишакова^{a,b}^a Оренбургский государственный университет, Оренбург, 460018, Российская Федерация^b Академия при Президенте Российской Федерации, РАНХиГС, Оренбург, 460000, Российская Федерация

Адрес для переписки: en_ischa@mail.ru

Информация о статье

Поступила в редакцию 25.06.15, принята к печати 23.07.15

doi:10.17586/2226-1494-2015-15-5-877-885

Язык статьи – русский

Ссылка для цитирования: Зубкова Т.М., Ишакова Е.Н. Автоматизация управления рисками программных проектов на основе нечеткого логического вывода // Научно-технический вестник информационных технологий, механики и оптики. 2015. Т. 15. № 5. С. 877–885.

Аннотация

Разработана методика управления рисками при проектировании программного обеспечения с использованием элементов искусственного интеллекта. На основе обзора существующих алгоритмов нечеткого вывода в области решения прикладных задач показана целесообразность применения одного из интеллектуальных методов для управления рисками программных проектов. Проанализированы источники информации при управлении программными проектами, выделены главные и второстепенные риски. Определены важнейшие параметры, по которым можно оценить возникновение неблагоприятной ситуации (длительность проекта, частота смены требований заказчика, соблюдение сроков работ, опыт участия разработчиков в подобных проектах и др.). Для анализа данных параметров разработан метод качественного нечеткого описания, использующий нечеткую логику. Оценка возможных ситуаций и формирование базы знаний опираются на опрос экспертов. Выявлены основные ограничения существующих автоматизированных систем в отношении их применимости к управлению рисками при проектировании программного обеспечения. На основании теоретических исследований разработана программная система, позволяющая автоматизировать процесс управления рисками программных проектов. Разработанная программная система автоматизирует процесс нечеткого вывода по следующим этапам: формирование базы правил систем нечеткого вывода, фазификация входных переменных, агрегирование подусловий, активизация подзаключений и аккумуляция заключений нечетких правил продукций, дефазификация переменных. Результатом автоматизации процесса управления рисками при проектировании программного обеспечения являются их количественная и качественная оценка и экспертные рекомендации по их минимизации. Практическая значимость работы заключается в том, что внедрение разработанной автоматизированной системы позволяет повысить результативность программных проектов.

Ключевые слова

управление рисками, программный проект, программная система, нечеткий логический вывод, нечеткая логика, лингвистические переменные, база знаний, экспертные оценки.

Благодарности

Работа выполнена при финансовой поддержке РФФИ и Правительства Оренбургской области (грант № 14-08-97031).

RISK MANAGEMENT AUTOMATION OF SOFTWARE PROJECTS BASED ON FUZZY INFERENCE

Т.М. Zubkova^a, Е.Н. Ishakova^{a,b}^a Orenburg State University, Orenburg, 460018, Russian Federation^b Russian Presidential Academy of National Economy and Public Administration, Orenburg, 46000, Russian Federation

Corresponding author: en_ischa@mail.ru

Article info

Received 25.06.15, accepted 23.07.15

doi:10.17586/2226-1494-2015-15-5-877-885

Article in Russian

For citation: Zubkova T.M., Ishakova E.N. Risk management automation of software projects based on fuzzy inference. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2015, vol. 15, no. 5, pp. 877–885.

Abstract

Application suitability for one of the intelligent methods for risk management of software projects has been shown based on the review of existing algorithms for fuzzy inference in the field of applied problems. Information sources in the management of software projects are analyzed; major and minor risks are highlighted. The most critical parameters have been singled out

giving the possibility to estimate the occurrence of an adverse situations (project duration, the frequency of customer's requirements changing, work deadlines, experience of developers' participation in such projects and others.). The method of qualitative fuzzy description based on fuzzy logic has been developed for analysis of these parameters. Evaluation of possible situations and knowledge base formation rely on a survey of experts. The main limitations of existing automated systems have been identified in relation to their applicability to risk management in the software design. Theoretical research set the stage for software system that makes it possible to automate the risk management process for software projects. The developed software system automates the process of fuzzy inference in the following stages: rule base formation of the fuzzy inference systems, fuzzification of input variables, aggregation of sub-conditions, activation and accumulation of conclusions for fuzzy production rules, variables defuzzification. The result of risk management automation process in the software design is their quantitative and qualitative assessment and expert advice for their minimization. Practical significance of the work lies in the fact that implementation of the developed automated system gives the possibility for performance improvement of software projects.

Keywords

risk management, software design, program system, fuzzy inference, fuzzy logic, linguistic variables, knowledge base, expert evaluations.

Acknowledgements

The work was financially supported by the Russian Federal Property Fund and the Government of the Orenburg region (grant № 14-08-97031).

Введение

Программные проекты не всегда достигают поставленной цели и воплощаются в конечный продукт, оправдывающий ожидания заказчика. В связи с этим особую актуальность приобретают проблемы управления рисками, которые неизбежно возникают при реализации программных проектов в силу большой неопределенности внутренней и особенно внешней среды предприятий и организаций [1].

Неграмотное управление рисками программных проектов может привести к увеличению труда программистов, растратам и множеству других проблем, вплоть до провала проекта. Непосредственным участником процесса управления рисками является заказчик, так как от него зависит понятная исполнителю цель разработки и условия реализации проекта (задание сроков, выделение средств, коммуникация с исполнителем и т.п.).

Процесс управления рисками программных проектов рассмотрен в работах С. Архипенкова, В.В. Липаева, А.Н. Терехова, А.А. Терехова, Т. ДеМарко, Т. Листера. В научных исследованиях выявлены проблемы управления рисками в проектах разработки программного обеспечения [2], разработаны инструменты и методы управления рисками [3], описаны факторы рисков [4] и методики профилактического управления рисками на ранних стадиях программных проектов [5].

При построении систем управления объектами достаточно успешно применяется методология нечеткого логического вывода [6], в частности, при разработке компонентов программного обеспечения [7, 8] и минимизации рисков проектов [9, 10].

Несмотря на достаточную разработанность методов нечеткого логического вывода, остаются открытыми вопросы их прикладной привязки к решению проблем конкретной предметной области, в частности, к управлению рисками программных проектов.

Постановка задачи

Риск будем рассматривать как возможность возникновения потерь, вытекающую из специфики тех или иных видов человеческой деятельности; вероятность принятия неверных или непринятия нужных управленческих решений; вероятность получения незапланированных результатов при осуществлении той или иной деятельности [11].

Для каждого риска, способного оказать негативное воздействие на достижение цели программного проекта, необходимо выбрать стратегию или комбинацию стратегий реагирования, представляющую наиболее эффективной: уклонение от риска, передачу риска, снижение риска или принятие риска.

Источниками информации при выявлении рисков будем считать различные доступные контрольные списки рисков программных проектов, которые следует проанализировать на применимость к данному конкретному проекту. Риски в новых проектах – это, как правило, проблемы завершенных и выполняемых проектов.

Главными рисками, которые затрагивают большинство проектов, являются нереалистичные сроки и бюджет, текучесть кадров, раздувание требований, низкая производительность [12]. Помимо главных рисков, необходимо учитывать и второстепенные, свойственные проектам с определенной спецификой задач.

Исследуемые этапы управления рисками программных проектов представим в виде схемы информационных потоков в нотации DFD (Data Flow Diagram) (рис. 1). Каждый объект на рис. 1 (функция, хранилище данных, внешняя сущность) имеет уникальный номер, не зависящий от его расположения.

Важнейшим этапом управления рисками является их идентификация и оценка, т.е. выявление рисков, способных повлиять на проект, и определение их характеристик (вероятность наступления, возможные неблагоприятные последствия, степень опасности, причина возникновения).

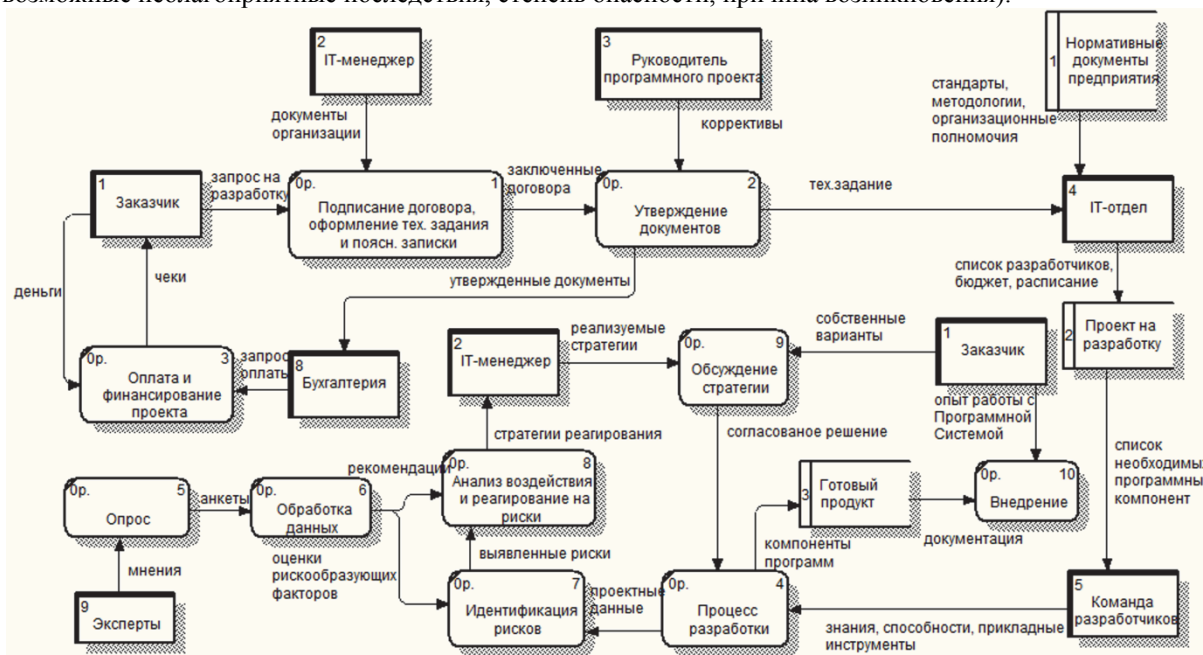


Рис. 1. Схема информационных потоков процесса управления рисками программных проектов

Входные данные для задачи управления рисками программных проектов, как правило, характеризуются той или иной степенью неопределенности, обусловленной неполнотой, внутренней противоречивостью, неоднозначностью, и представляют собой приближенные количественные или качественные оценки параметров процессов проектирования и управления проектированием [13].

Таким образом, так как исходные данные задачи трудно формализуемы, целесообразно применить один из интеллектуальных методов, основанный на нечеткой логике. Нечеткие алгоритмы, оперирующие лингвистическими переменными, значения которых задаются нечеткими множествами, удобны для описания слабо формализуемых процессов. Такие алгоритмы интуитивно более понятны. Основное внимание при применении методов нечеткой логики следует обратить на адекватное построение функций принадлежности по знаниям эксперта.

Реализация поставленной задачи

Для решения проблемы идентификации рисков программных проектов были выявлены риск-факторы, которые размещены в 4 базах знаний, насчитывающих 16 лингвистических переменных, среди которых 12 входных (факторы риска) и 4 выходных (степень риска). Список лингвистических переменных представлен в таблице.

Системы нечеткого вывода преобразуют значения входных переменных процесса управления в выходные переменные на основе использования нечетких правил.

Процесс нечеткого вывода выполняем по следующим этапам.

1. Формирование базы правил систем нечеткого вывода.
2. Фаззификация входных переменных (введение в нечеткость), которая включает не только отдельный этап выполнения нечеткого вывода, но и процедуру нахождения значений функций принадлежности нечетких множеств (термов) на основе обычных исходных данных.

За основу взята кусочно-линейная функция принадлежности в форме трапеции:

$$f_T(x; a; b; c; d) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ 1, & b \leq x \leq c \\ \frac{d-x}{d-c}, & c \leq x \leq d \\ 0, & d \leq x \end{cases}, \quad (1)$$

где a, b, c, d – параметры трапеции, определяемые в ходе опроса экспертов; x – значение

лингвистической переменной.

3. Агрегирование подусловий в нечетких правилах продукций.

Определяем степень истинности условий по каждому из правил системы нечеткого вывода. Для этого используем одну из возможных операций – логическую дизъюнкцию или конъюнкцию нечетких высказываний A и B :

$$T(A \vee B) = \max \{T(A), T(B)\}, \quad (2)$$

$$(A \wedge B) = \min \{T(A), T(B)\}, \quad (3)$$

где A, B – нечеткие высказывания о факторах риска программного проекта; $T(A), T(B), T(A \vee B)$ – значение истинности соответствующего нечеткого высказывания о факторах риска программного проекта.

Лингвистическая переменная	Универсум	Единица измерения	Множество термов
<i>Входные переменные</i>			
«смена требований»	0–30	раз	{очень низкая, низкая, средняя, высокая, очень высокая}
«участие в подобных проектах»	0–100	проект	{очень малое, малое, среднее, большое, очень большое}
«несоблюдение сроков»	0–30	неделя	{слишком незначимое, незначимое, умеренное, значимое, слишком значимое}
«временной интервал»	0–180	день	{незначительный, умеренный, значительный, небольшой, большой}
«число сущностей базы данных»	0–60	штук	{низкое, среднее, высокое}
«число строк кода»	0–1000	тысяча строк	{низкое, среднее, высокое}
«число специалистов»	0–60	человек	{низкое, среднее, высокое}
«число модулей»	0–500	штук	{низкое, среднее, высокое}
«число ошибок тестирования»	0–10000	штук	{очень низкое, низкое, среднее, высокое, очень высокое}
«длительность командной работы»	0–120	месяц	{очень малая, малая, средняя, большая, очень большая}
«число несущественных задач»	0–500	штук	{малое, среднее, большое}
«последняя коммуникация с заказчиком»	0–365	дней	{недавно, не так давно, давно, слишком давно}
<i>Выходные переменные</i>			
«риск срыва проекта»	0–10	балл	{нет, игнорируемый, умеренный, критический, катастрофический}
«риск масштаба»	0–10	балл	{нет, игнорируемый, умеренный, критический, катастрофический}
«риск срыва сроков»	0–10	балл	{нет, игнорируемый, умеренный, критический, катастрофический}
«риск нарушения спецификаций»	0–10	балл	{нет, игнорируемый, умеренный, критический, катастрофический}

Таблица. Список лингвистических переменных и множество их термов

4. Активизация или композиция подзаключений в нечетких правилах продукций.

По весовым коэффициентам для всех правил нечетких продукций находим степень истинности каждого из подзаключений. Для этой цели используем модификацию метода нечеткой композиции – Min-активизацию:

$$\mu'(y) = \min \{c_i, \mu(y)\}, \quad (4)$$

где $\mu(y)$ – функция принадлежности терма, который является значением некоторой выходной переменной $\omega(y)$, заданной на универсуме Y ; c_i – степень истинности подзаключений для каждого из правил, входящих в рассматриваемую базу правил системы нечеткого вывода; $\mu'(y)$ – активизированная функция принадлежности заключения.

По умолчанию весовые коэффициенты всех правил приняты за единицу, каждое правило в рассматриваемой методике имеет одно заключение.

5. Аккумуляция заключений нечетких правил продукций.

Для каждой из выходных лингвистических переменных множества $W = \{\omega_1, \omega_2, \omega_3, \omega_4\}$ находим функции принадлежности. Целью данного этапа является объединение или аккумуляция всех степеней истинности заключений для получения функции принадлежности каждой из выходных переменных по формуле

$$\mu_V(x) = \max\{\mu_A(x), \mu_B(x)\}, \quad (5)$$

где V – аккумулярованное множество нечетких высказываний A и B о факторах риска программного проекта; $\mu_A(x), \mu_B(x), \mu_V(x)$ – значение функции принадлежности фактора риска x универсума X соответствующему нечеткому множеству A, B, V .

6. Дефазификация для получения конкретного числового значения результата.

Находим обычное значение для каждой из выходных лингвистических переменных множества $W = \{\omega_1, \omega_2, \omega_3, \omega_4\}$. Для дефазификации используем метод центра тяжести (алгоритм Мамдани) [14]:

$$y = \frac{\max_{\min} \int x \cdot \mu(x) dx}{\int_{\min}^{\max} \mu(x) dx}, \quad (6)$$

где y – степень риска в диапазоне от 0 до 10 (результат дефазификации); x – переменная, соответствующая выходной лингвистической переменной ω ; $\mu(x)$ – функция принадлежности нечеткого множества, соответствующая выходной переменной ω после этапа аккумуляции; \min, \max – левая и правая точки интервала носителя нечеткого множества рассматриваемой выходной переменной ω (универсум риска в интервале от 0 до 10).

Результаты применения разработанной методики

Процесс управления рисками в силу его трудоемкости целесообразно автоматизировать. В настоящее время на российском рынке присутствует более десятка систем данного класса. Среди них можно выделить многофункциональные системы @Risk Professional for Project, Dekker TRAKKER, Enterprise project, ER Project 1000, Intelligent Planner, Mesa/Vista Risk Manager, Risk Trak, Open Plan.

Основными ограничениями данных систем в отношении их применимости для управления рисками программных проектов являются:

- ориентация на риски информационной безопасности;
- необходимость постоянного доступа к сети Интернет для повседневного управления рисками;
- компиляция различных методик оценки рисков (СТО БР, PCI DSS, ISO 27001/27005, OCTAVE, NIST и др.), дающая специфично-экспериментальный результат.

В связи с этим авторами было принято решение о разработке собственной программной системы «Управление рисками разработки прикладных программных изделий» [15].

Программная система включает Windows-приложение, реализующее основной алгоритм работы, и файл базы данных FuzzyLogic.sqlite.

Программная система представляет совокупность методических и программных средств решения следующих задач:

- авторизация пользователей;
- ведение справочников и учет данных (физических и юридических лиц, их ролей, проектов, договоров);
- формирование базы знаний, исходя из мнений экспертов (анкетирование экспертов);
- идентификация и оценка рисков программных проектов по алгоритму Мамдани;
- поддержка принятия решений по реагированию на риски на основе экспертных рекомендаций;
- формирование отчетов по выявленным рискам.

Декомпозиция функций разработанной программной системы представлена в нотации IDEF (Integrated DEFinition) (рис. 2). Все функции на рис. 2 помечены уникальными номерами от 1 до 4.

Входными данными программной системы являются: риск-предпосылки (требования заказчика программного обеспечения, сведения о разработчиках, история аналогичных программных проектов, информация об этапах проекта и др.), экспертные правила и рекомендации, техническое задание и перечень рисков программного проекта.

В качестве управляющего воздействия выступают распоряжения руководителя в ходе проекта, стандарт предприятия. Ресурсами и участниками, поддерживающими выполнение функций, являются руководитель проекта, заказчик программного обеспечения, программисты, аппаратное и программное

обеспечение предприятия.

На выходе программной системы формируются: отчет о текущем состоянии выполнения проекта, оценка выявленных рисков и рекомендации экспертов по их минимизации, стратегия управления рисками.

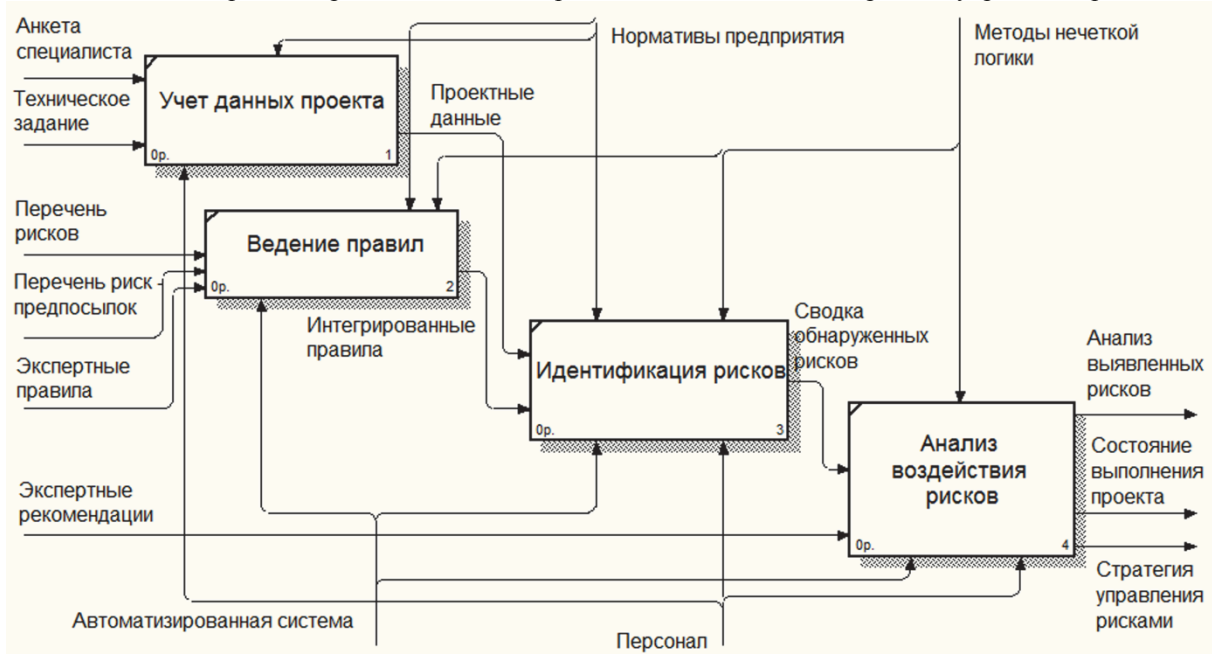


Рис. 2. Декомпозиция функций программной системы управления рисками программных проектов. АС – автоматизированная система

Для построения модели знаний в окне редактора структуры можно задать все лингвистические переменные.

Например, на рис. 3 задается лингвистическая переменная «смена требований». В ходе опроса эксперты оценили нижние и верхние пределы по каждому терму в диапазоне значений от 0 до 30. Далее рассчитаны параметры трапециевидной функции по формулам (7)–(10):

$$a = \min(a_j), \tag{7}$$

$$b = \min\{\max(a_j), \min(b_j)\}, \tag{8}$$

$$c = \max\{\max(a_j), \min(b_j)\}, \tag{9}$$

$$d = \max(b_j), \tag{10}$$

где a_j, b_j – оценка нижнего и верхнего предела термина j -го эксперта.

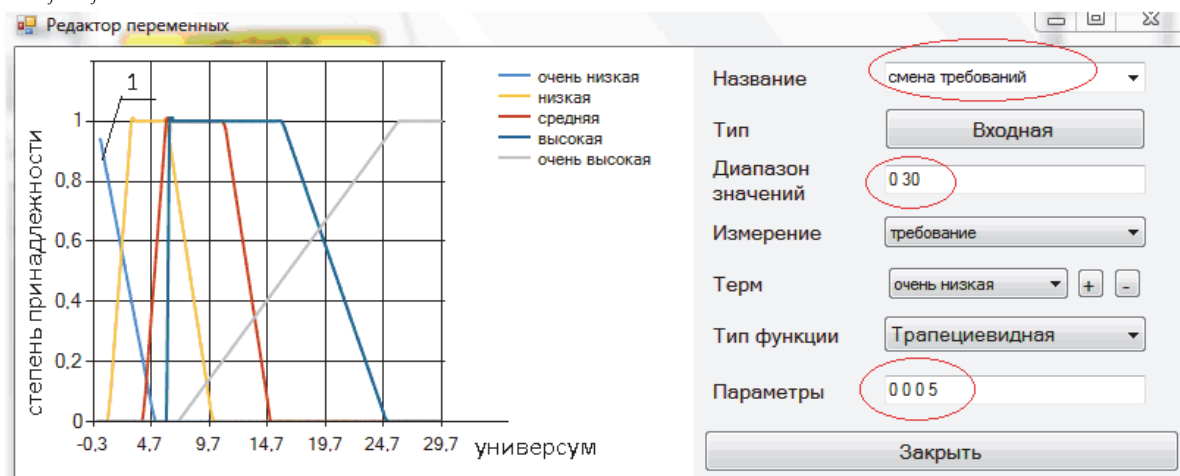


Рис. 3. Описание переменных в редакторе переменных

На рис. 3 параметры трапециевидной функции определены соответственно как «0 0 0 5». При подстановке данных параметров в формулу (1) получены графики функций принадлежности. Например, если x принадлежит интервалу от 0 до 30, $a = b = c = 0, d = 5$, то функция принадлежности термина «очень

низкая» будет соответствовать графику 1 (голубая линия) на рис. 3. По остальным лингвистическим переменным, приведенным в таблице, проводятся аналогичные действия.

Заполненная структура показана на примере базы знаний «Оценка риска срыва проекта» (рис. 4).

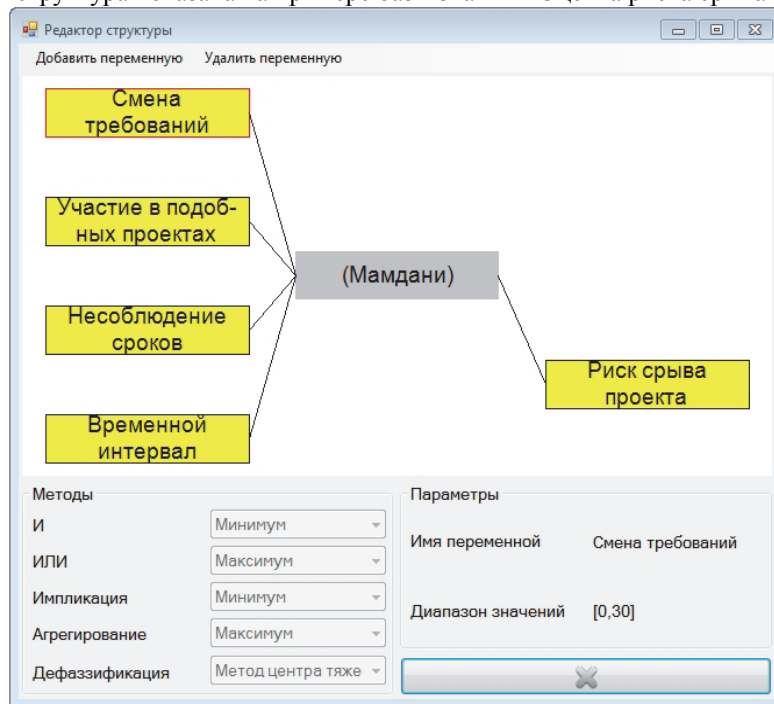


Рис. 4. Структура базы знаний «Оценка риска срыва проекта»

Для формирования правил вывода проводится опрос мнений экспертов и их согласование как параметров термов лингвистических переменных (риск-факторов) (рис. 5).

Рис. 5. Редактор правил базы знаний

Далее задаются конкретные значения входных переменных. Например, на рис. 6 значение переменной «смена требований» – 12 раз, «участие в подобных проектах» – 9 проектов, «несоблюдение сроков» – 29 недель, «временной интервал» – 146 дней.

Просматриваем всю базу правил (рис. 5), находим степени принадлежности значений входных параметров каждому терму входных и выходной переменных. Для этого на рис. 6 по каждому входному параметру восстановлен перпендикуляр до пересечения с соответствующим графиком функции принадлежности (позиция 1 – красная линия).

После агрегирования подусловий, активизации подзаключений и аккумуляирования заключений нечетких правил продукции по формулам (2)–(5) получаем фигуру на графике «Риск срыва проектов». Вычисляя центр тяжести фигуры по формуле (6), определяем численное значение риска срыва проекта

(перпендикуляр в позиции 2 – красная линия). В нашем случае количественное значение риска равно 8,26, согласно введенной классификации он интерпретируется как «катастрофический» и выдается рекомендация экспертов – «Детальнее согласовывайте требования с заказчиком».

Таким образом, проводится идентификация и анализ проектных рисков, их количественная и качественная оценка, а также даются экспертные рекомендации по их минимизации.

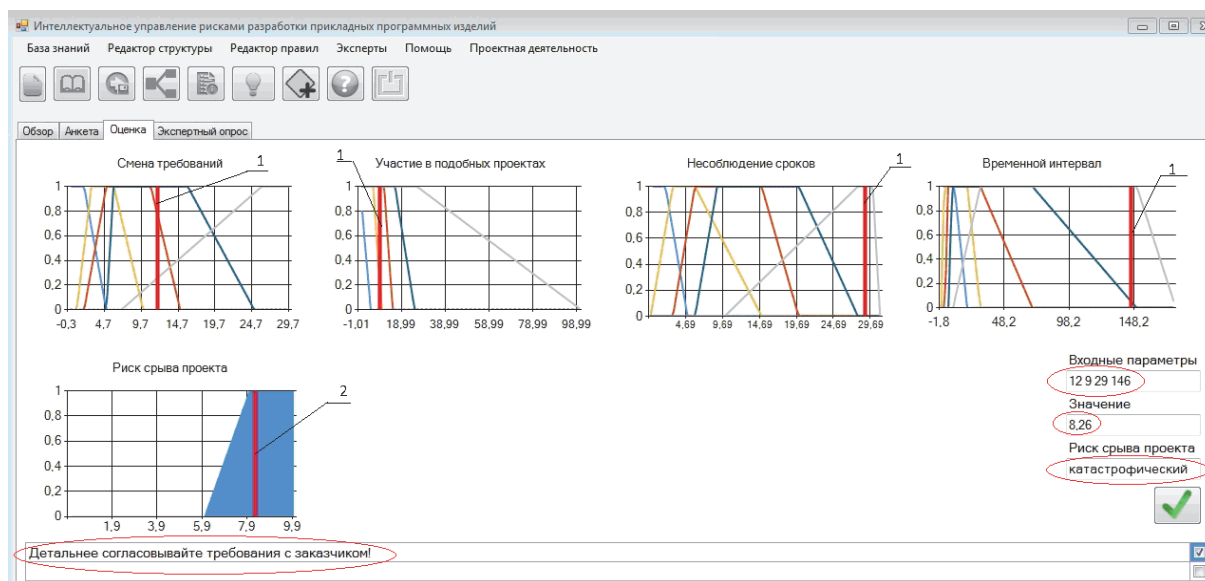


Рис. 6. Результат экспертной оценки риска

Заключение

Основным результатом работы является информационное, математическое и программное обеспечение для решения задач интеллектуального управления рисками средних и крупных программных проектов на ранних стадиях их разработки.

Решены следующие научно-практические задачи:

- выявлены основные параметры, по которым можно оценить вероятность возникновения неблагоприятных ситуаций в программном проекте;
- разработана методика управления рисками программных проектов с использованием элементов искусственного интеллекта;
- разработан метод качественного нечеткого описания факторов риска программных проектов.

В предложенной модели акцент делается на предотвращение риска, так как снижение рисков на поздних стадиях жизненного цикла разработки программного обеспечения приводит к дорогостоящим и неэффективным управляющим воздействиям. Данный подход отличается тем, что использует модель, основанную на профилактическом управлении рисками на ранних стадиях программных проектов на основе нечеткого логического вывода.

Разработанная методика позволила автоматизировать процесс интеллектуального управления рисками, что, в конечном итоге, повысило оперативность и объективность принимаемых управленческих решений.

References

1. Titarenko B.P. *Upravlenie Riskami v Innovatsionnykh Proektakh* [Risk Management in Innovative Projects]. Moscow, MGSU Publ., 2011, 144 p.
2. Schmidt C., Dart P., Johnston L., Sterling L., Thorne P. Disincentives for communicating risk: a risk paradox. *Information and Software Technology*, 1999, vol. 41, no. 7, pp. 403–411. doi: 10.1016/S0950-5849(99)00011-7
3. Tah J.H.M., Carr V. Towards a framework for project risk knowledge management in the construction supply chain. *Advances in Engineering Software*, 2001, vol. 32, no. 10–11, pp. 835–846. doi: 10.1016/S0950-5849(01)00035-7
4. Drew Procaccino J., Verner J.M., Overmyer S.P., Darter M.E. Case study: factors for early prediction of software development success. *Information and Software Technology*, 2002, vol. 44, no. 1, pp. 53–62. doi: 10.1016/S0950-5849(01)00217-8
5. Adler T.R., Leonard J.G., Nordgren R.K. Improving risk management: moving from risk elimination to risk avoidance. *Information and Software Technology*, 1999, vol. 41, no. 1, pp. 29–34.

6. Kudinov Y.I. Synthesis of a fuzzy-logic control system. *Journal of Computer and Systems Sciences International*, 1999, vol. 38, no. 1, pp. 158–164.
7. Chen J., Rine D.C. Training fuzzy logic controller software components by combining adaptation algorithms. *Advances in Engineering Software*, 2003, vol. 34, no. 3, pp. 125–137. doi: 10.1016/S0965-9978(02)00140-0
8. Kandel A., Zhang Y.-Q., Henne M. On use of fuzzy logic technology in operating systems. *Fuzzy Sets and Systems*, 1998, vol. 99, no. 3, pp. 241–251.
9. Wang J. A fuzzy project scheduling approach to minimize schedule risk for product development. *Fuzzy Sets and Systems*, 2002, vol. 127, no. 2, pp. 99–116. doi: 10.1016/S0165-0114(01)00146-4
10. Carr V., Tah J.H.M. A fuzzy approach to construction project risk assessment and analysis: construction project risk management system. *Advances in Engineering Software*, 2001, vol. 32, no. 10–11, pp. 847–857. doi: 10.1016/S0965-9978(01)00036-9
11. Ishakova E.N., Zubkova T.M., Medvedev A.S. Program system of the assessment of risks in the higher education area with the use of productional and frame model. *Vestnik OSU*, 2014, no. 1, pp. 183–188. (In Russian)
12. DeMarco T., Lister T. *Waltzing with Bears: Managing Risk on Software Projects*. Dorset House, 2003, 144 p.
13. Gil'man D.V., Taganov A.I. *Metodologicheskie Osnovy Analiza i Attestatsii Urovnei Zrelosti Protsessov Programmnykh Proektov v Usloviyakh Nechetkosti* [Methodological Bases of the Analysis and Validation of Process Maturity Levels of Software Projects in Fuzziness Conditions]. Moscow, Goryachaya Liniya-Telekom Publ., 2014, 168 p.
14. Shtovba S.D. Obespechenie tochnosti i prozrachnosti nechetkoi modeli Mamdani pri obuchenii po eksperimental'nym dannym. *Problemy Upravleniya i Informatiki*, 2007, no. 4, pp. 102–114.
15. Zubkova T.M., Ishakova E.N., Mulyukov R.R. *Programmnyaya Sistema Intellektual'nogo Upravleniya Riskami Razrabotki Prikladnykh Programmnykh Izdelii*. Certificate of State Registration of Computer Programs, no. 2014662317, 2014.

Зубкова Татьяна Михайловна – доктор технических наук, профессор, профессор, Оренбургский государственный университет, Оренбург, 460018, Российская Федерация, bars87@mail.ru

Ишакова Елена Николаевна – кандидат педагогических наук, доцент, доцент, Оренбургский государственный университет, Оренбург, 460018, Российская Федерация; доцент, Академия при Президенте Российской Федерации, РАНХиГС, Оренбург, 460000, Российская Федерация, en_ischa@mail.ru

Tatiana M. Zubkova – D.Sc., Professor, Professor, Orenburg State University, Orenburg, 460018, Russian Federation, bars87@mail.ru

Elena N. Ishakova – PhD, Associate professor, Associate professor, Orenburg State University, Orenburg, 460018, Russian Federation; Associate professor, Russian Presidential Academy of National Economy and Public Administration, Orenburg, 46000, Russian Federation, en_ischa@mail.ru