

Моделювання в економіці, організація виробництва та управління проектами

УДК 510.6: (004.413.4:004.415.2)

Л.А. Волобуєва, Л.В. Мандрикова

Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков

АЛГОРИТМИЗАЦИЯ ПОДДЕРЖКИ РЕШЕНИЙ ПРИ УПРАВЛЕНИИ РИСКАМИ ПРОГРАММНЫХ ПРОЕКТОВ НА ОСНОВЕ НЕЧЕТКОЙ ЛОГИКИ

В статье рассмотрены вопросы автоматизации поддержки принятия решений при управлении рисками программных проектов с использованием нечеткой логики. Рассмотрено современное состояние указанной проблемы, описана специфика и проанализированы основные этапы процесса управления рисками программных проектов, построены нечетко-множественные модели для описания и решения слабоформулированных задач процесса. Предложены алгоритмы для автоматизации поддержки принятия решений при управлении рисками программных проектов в рамках эталонной модели классификации рисков.

Ключевые слова: управление программным проектом, оценка риска, нечеткая логика, жизненный цикл проекта.

Введение

Управление программным проектом (ПП) имеет свою специфику, заключающуюся в нематериальной природе ПП, процесс его реализации поддается мониторингу и контролю только посредством анализа документации [1 – 4], при этом наиболее сложные проекты часто требуют уникальных решений.

Повышение сложности программных проектов на фоне быстрой смены используемых технологий и необходимость их оперативного освоения и применения при разработке с целью отсрочки морального устаревания проекта, а также превалирование творческой составляющей при разработке ставят успешность реализации проекта в зависимость от квалификации и стабильности коллектива разработчиков. Это усугубляет проблему прогнозирования результатов выполнения проекта.

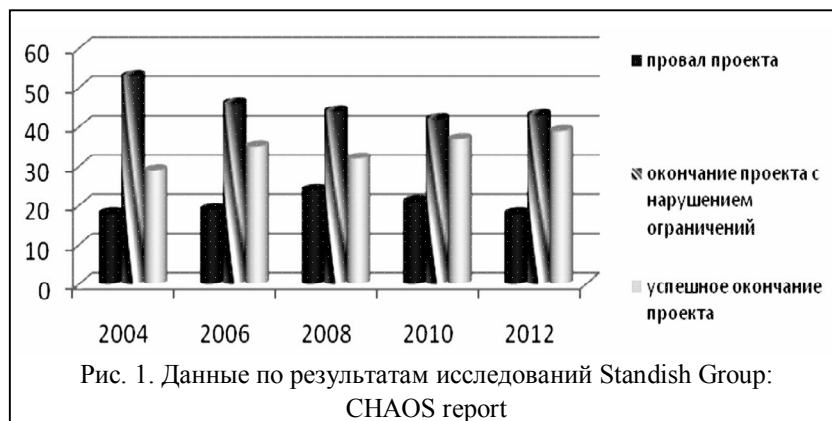
По данным компании Standish Group, опубликованным в отчетах CHAOS report [5], ежегодно

большинство программных проектов либо завершаются с нарушениями указанных в договоре ограничений, либо прекращаются досрочно в связи с провалом проекта (рис. 1). Указанная специфика процесса разработки программных проектов обосновывает необходимость использования специальных методов и средств управления рисками. Разработан и используется ряд стандартов по управлению рисками программным проектом.

Стандарт по управлению рисками программного обеспечения IEEE 1540: Standard for Software Risk Management, подготовленный в Институте управления проектами (PMI), содержит подробное описание всех этапов итеративного процесса управления рисками ПП.

Стандарт PMBOK [1] описывает процедуры управления рисками, в числе которых идентификация, качественная оценка и планирование реагирования на риски – это процедуры, которые могут быть автоматизированы с применением методов и средств инженерии знаний. Все эти процедуры взаимодействуют друг с другом, а также с другими процедурами. Каждая процедура выполняется, по крайней мере, один раз в отдельно взятом проекте.

Существует ряд классификаций рисков программных проектов, частично повторяющие друг друга, но в тоже время каждая из них имеет свои достоинства. Хорошо известны работы



Barry Boehm, Caper Jones, и SEI Software Risk Taxonomy, описывающие источники таких рисков. Для обеспечения учета всех факторов риска, выделенных в выше названных классификациях, синтезирована эталонная модель рисков [2]. Эталонная модель содержит составные (s) и одинарные (f) риски, сгруппированные по категориям (y), категории отнесены к соответствующим классам (x).

Целью данной статьи является построение алгоритмической базы, работающей на основе использования эталонной модели классификации рисков ПП, в рамках информационной технологии поддержки принятия решений при управлении рисками ПП средствами нечеткой логики.

1. Использование нечетких множеств при управлении рисками

Одним из наиболее распространенных методов, применяемых для оценки влияния факторов риска, является метод расчета выходных значений факторов, основанный на нечетко-множественном представлении входных и выходных параметров. В частности на приведении к виду нечетких множеств как факторов, измеряемых в количественных шкалах, так и качественных [6].

Формализация входной информации о параметрах факторов рисков в нечетких узлах осуществляется в два этапа [2, 6]:

1. Описание каждого входного фактора риска в виде лингвистической переменной, заданной набором

$$\langle l, T, X, G, M \rangle,$$

где l – название лингвистической переменной;

T – терминальное множество, элементы которого являются именами нечетких переменных, определенных на множестве X;

G – синтаксическая процедура;

M – семантическая процедура, преобразующая в нечеткие переменные новые термы, полученные процедурой G.

2. Представление каждого терма как нечеткой переменной.

Знания экспертов отражают взаимосвязь между факторами в виде высказываний, связывающих входные факторы с выходным.

В качестве примера в табл. 1 приведена формализация фактора риска «Стабильность требований» в виде описания одноименной лингвистической переменной и ее термов как нечетких переменных, определенных на множестве X $(-\infty, 1]$. Риски предусмотренных классификаций представляются в соответствии с эталонной моделью (англ. reference model) классификации рисков ПП [2].

Функция принадлежности для лингвистической переменной f_1 «Стабильность требований» приведена на рис. 2.

Таблица 1

Фрагмент описания лингвистических переменных для формализации факторов риска ПП

Фактор риска	Лингвистическая переменная
Стабильность требований степень ущерба - серьезная; критерий - Requirements Stability Index $RSI = \frac{R_T - CR_C}{R_T}$ R_T – количество требований; CR_C – совокупное количество изменений.	f_1 – стабильность требований; l – стабильность требований; T – {высокая, средняя, низкая}; X – индекс стабильности требований в диапазоне $(-\infty, 1]$; G – синтаксическая процедура; M – семантическая процедура задания на X $(-\infty, 1]$ нечетких подмножеств $A_1^1 = \text{«высокая»}$, $A_2^1 = \text{«средняя»}$, $A_3^1 = \text{«низкая»}$, а также, преобразующая полученные процедурой G новые термы в нечеткие переменные.

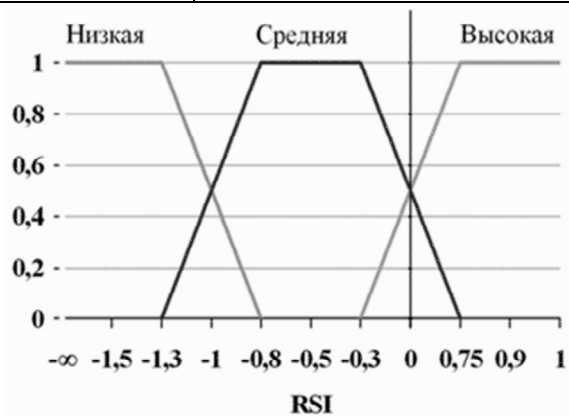


Рис. 2. Функция принадлежности лингвистической переменной «Стабильность требований»

В соответствии с эталонной моделью классификации необходимо описать классы, категории, составные факторы и одинарные факторы рисков. Перед тем как приступить к идентификации рисков программного проекта требуется выбрать и утвердить классификацию рисков. Это позволит определить набор одинарных и составных факторов, с которыми и будут выполняться дальнейшие действия.

Пример представления классификации рисков ПП по Б. Бозму, приведен на рис. 3.

Выделены и описаны одинарные и составные факторы. К одинарным рискам относятся: (5) ненужная оптимизация и оттачивание деталей; (6) непрерывный поток изменений; (10) «разрыв» в квалификации специалистов разных областей знаний.

Непрерывный поток изменений представим: в виде фактора f_1 – стабильность требований; ненужную оптимизацию и оттачивание деталей, в виде фактора f_{22} – оптимизация; «разрывы» в квалификации специалистов разных областей знаний, в виде фактора f_{55} – подбор команды.

Составные факторы рисков: (s_1) дефицит специалистов, (s_2) нереалистичные сроки и бюджет, (s_3) реализация несоответствующей функциональности, (s_4)

разработка неправильного пользовательского интерфейса, (s_5) недостаток информации о внешних компонентах, определяющих окружение системы или вовлеченных в интеграцию, (s_6) недостатки в работах, выполняемых внешними (по отношению к проекту) ресурсами, (s_7) недостаточная производительность полученной системы.

Каждый из составных факторов содержит два и более одинарных факторов риска ПП. Для вывода

решений относительно значений составных факторов построены соответствующие базы решающих правил. Эта классификация содержит десять рисков программных проектов без разделения на классы и категории. Поэтому описан общий класс x_1 и категория y_1 . Далее будут описаны алгоритмы для автоматизации таких этапов процесса управления рисками ПП как: идентификация рисков ПП, качественная и количественная оценка рисков ПП.

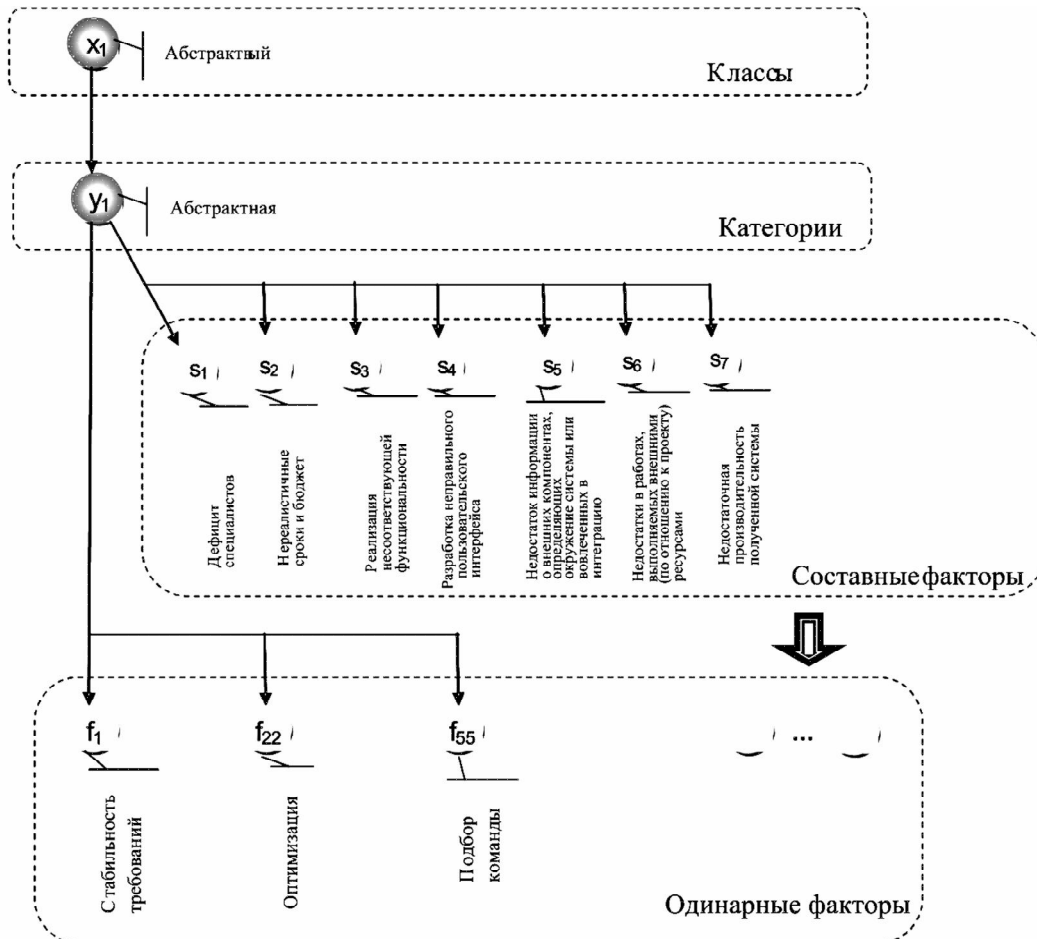


Рис. 3. Пример представления классификации рисков ПП по Б. Бозму в терминах эталонной модели

2. Алгоритмизация поддержки решений при управлении рисками ПП

Идентификация рисков [7] – это выявление рисков, способных повлиять на проект, и документальное оформление их характеристик. Это итеративный процесс, который периодически повторяется на всем протяжении проекта, так как в его жизненном цикле могут обнаруживаться новые риски.

Исходные данные для выявления и описания характеристик рисков могут браться из разных источников. На вход алгоритма подается результат выбора классификации рисков. Далее, работая в рамках выбранной классификации, запрашиваются точные значения всех самостоятельных и входящих в составные одинарных факторов риска согласно

соответствующим критериям оценки, выполняется их фаззификация.

Блок-схема алгоритма приведена на рис 4.

Результатом работы этого алгоритма являются вычисленные значения степеней принадлежности к термам соответствующих лингвистических переменных всех заданных одинарных и составных факторов.

При анализе каждого риска определяется вероятность его возникновения и учитываются последствия его реализации. В работе используется следующая градация степени ущерба [7,8]: незначительная (4), терпимая (3), серьезная (2), катастрофическая (1). Вероятность риска считается низкой, если ее значение попадает в диапазон $[0; 0,25)$; средней при значениях $[0,25; 0,5)$; высокой при значениях от 0,5. Результаты анализа рисков предоставляются в виде таблицы, упорядоченной по рангу риска.

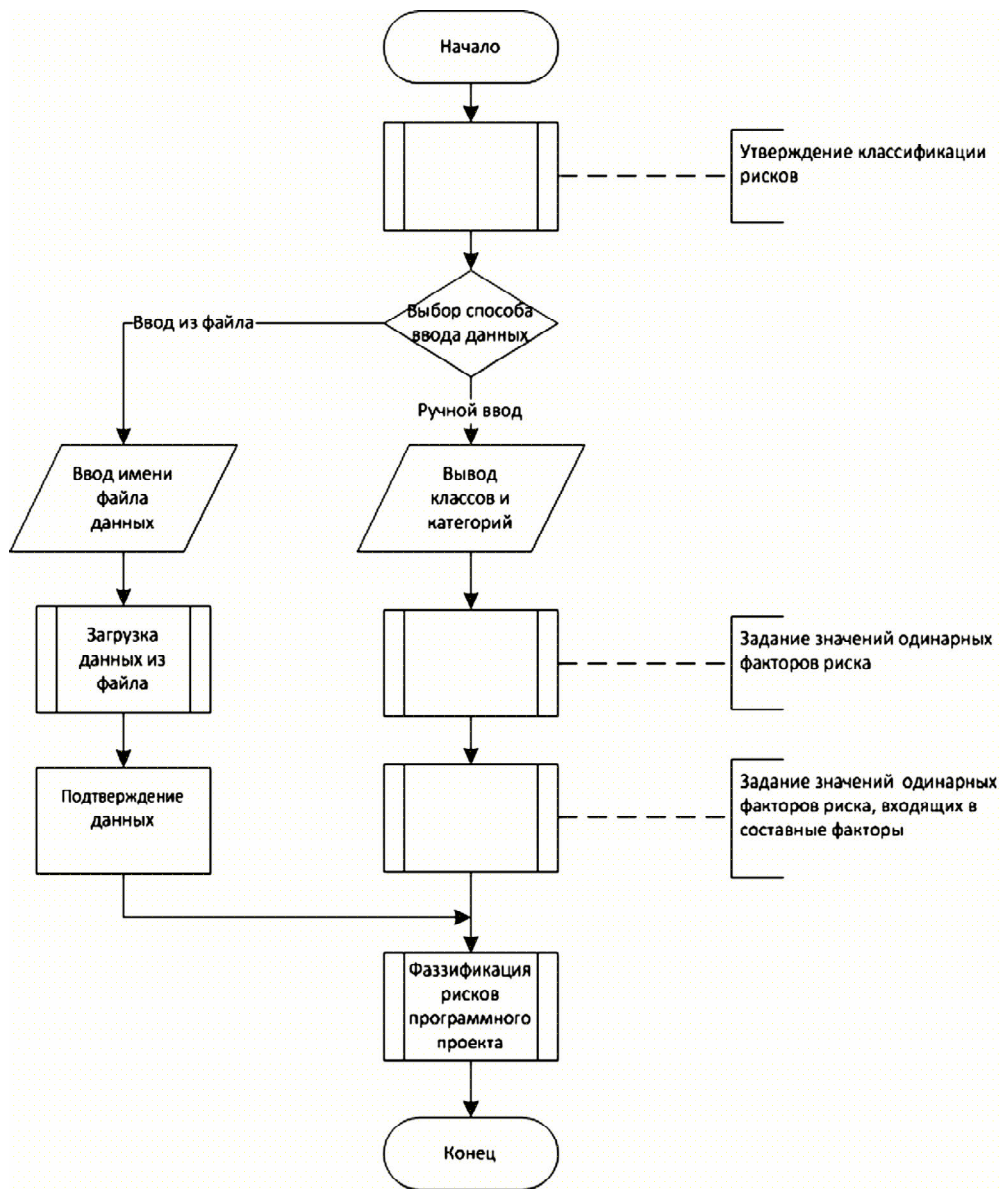


Рис. 4. Блок-схема алгоритма идентификации рисков программного проекта

Вероятность риска и возможный ущерб от его реализации пересматриваются при обновлении данных проекта, поэтому таблица также обновляется на каждой итерации процесса управления рисками ПП. В результате проведения анализа определяется набор наиболее весомых рисков, которые потом отслеживаются на всех этапах реализации программного проекта.

Для определения ранга риска используется матрица вероятностей и последствий [7] в адаптированном виде (табл. 2). В строках таблицы степень ущерба, в столбцах – вероятность возникновения риска. В ячейках на пересечении строк и столбцов – ранг риска. После определения ранга каждого идентифицированного риска риски ранжируются, и формируется результирующая таблица. Ранг риска: высокий (6 – 9), средний (3 – 4), низкий (1 – 2).

На вход алгоритма поступает перечень идентифицированных рисков программного проекта. На выходе – результирующая таблица рисков проекта.

В этой таблице идентифицированные риски упорядочены в порядке убывания ранга.

Таблица 2

Адаптированная матрица вероятностей и последствий

		Вероятность возникновения		
		низкая	средняя	высокая
Степень ущерба	1	4	6	9
	2	3	4	6
	3	2	3	4
	4	1	2	3

Каждая строка результирующей таблицы описывает идентифицированный риск: его порядковый номер, числовое значение ранга, степень ущерба, вероятность его возникновения, класс и категорию к которой относится этот риск в рамках используемой классификации рисков программных проектов. Блок-схема алгоритма приведена на рис. 5.

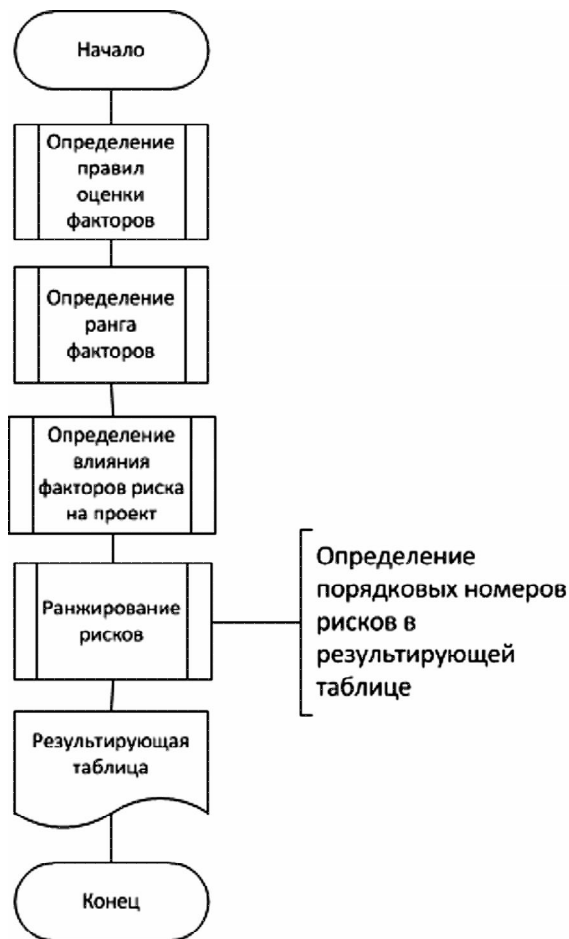


Рис. 5. Блок-схема алгоритма качественного анализа рисков ПП

Результаты качественного анализа используются для формирования реестра рисков ПП, а также для проведения количественного анализа рисков программного проекта.

Для количественной оценки составных факторов риска выполняется дефазификация значений методом Mamdani [6]. Суть этого метода заключается в нахождении центра тяжести объединенной области значения. Входными данными алгоритма являются значения степеней принадлежности составного фактора к термам соответствующей лингвистической переменной и таблица рисков программного проекта, полученная в результате качественного анализа рисков. На выходе алгоритма – доработанная таблица рисков, включающая в себя численные значения RE для каждого риска. Блок-схема алгоритма количественного анализа представлена на рис. 6.

Собственный риск RE оценивается в полуколичественной шкале следующим образом [8]:

$$RE_i = Q_i * w_i, \quad (i = \overline{1, NR}),$$

где Q – количественная оценка вероятности реализации рассматриваемого риска ПП, w – весовой коэффициент, характеризующий степень ущерба от реализации риска, NR – количество рассматриваемых рисков.

Для описания степени ущерба (Damage Degree, DD) используется такая шкала: (DD=1) катастрофическая, (DD=2) серьезная, (DD=3) терпимая, (DD=4) незначительная. Весовой коэффициент w_j вычисляется как

$$\frac{1}{DD_j}, \quad j = \overline{1, 4}.$$

По результатам количественного анализа обновляется реестр рисков, который в дальнейшем используется на следующем этапе процесса управления рисками программных проектов – этапе планирования реагирования.

Выводы

Рассмотрены вопросы поддержки принятия решений при управлении рисками программных проектов.

Описана формализация рисков ПП на примере в терминах эталонной модели при использовании классификации по Ваггу Боehm.

Построена алгоритмическая база для поддержки принятия решений на этапах идентификации, качественного и количественного анализа рисков программных проектов.

Список литературы

1. Руководство к своду знаний по управлению проектами. (PMBOK®). Четвертое издание. [Текст]. – Project Management Institute, Inc., 2004. – 241 с.
2. Туркин И.Б. Нечетко-множественные модели и метод для поддержки принятия решений при управлении рисками программных проектов [Текст] / И.Б. Туркин, Л.В. Мандрикова, Л.А. Волобуева // Радиоэлектронні і комп'ютерні системи. – 2011. – №2(50). – С.112-118.
3. Сомервилл И. Инженерия программного обеспечения, 6-е издание: пер. с англ. [Текст] / И. Сомервилл – М.: Изд. дом «Вильямс», 2002. – 624 с.
4. Bourque P. Guide to the Software Engineering Body of Knowledge, Version 3.0 [Electronic Source] / P. Bourque, R.E. Fairley, eds. – IEEE Computer Society, 2014. – Режим доступа: www.swebok.org.
5. The Standish Group [Electronic Source]. – Режим доступа: <http://www.standishgroup.com>.
6. Люгер Д.Ф. Искусственный интеллект: стратегии и методы решения сложных проблем, 4-е издание: пер. с англ. [Текст] / Д.Ф. Люгер. – М.: Изд. дом «Вильямс», 2003. – 864 с.
7. Архипенков С. Лекции по управлению программными проектами [Электронный ресурс] / С. Архипенков. – Режим доступа: http://www.arkhipenkov.ru/resources/sw_project_management.pdf. – 2009.
8. Шафер Д.Ф. Управление программными проектами: достижение оптимального качества при минимуме затрат: пер. с англ. [Текст] / Д.Ф. Шафер, Р.Т. Фатрел, Л.И. Шафер. – М.: Изд. дом «Вильямс», 2003. – 1136 с.

Поступила в редколлегию 11.06.2015

Рецензент: д-р техн. наук, проф. И.В. Шостак, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков.

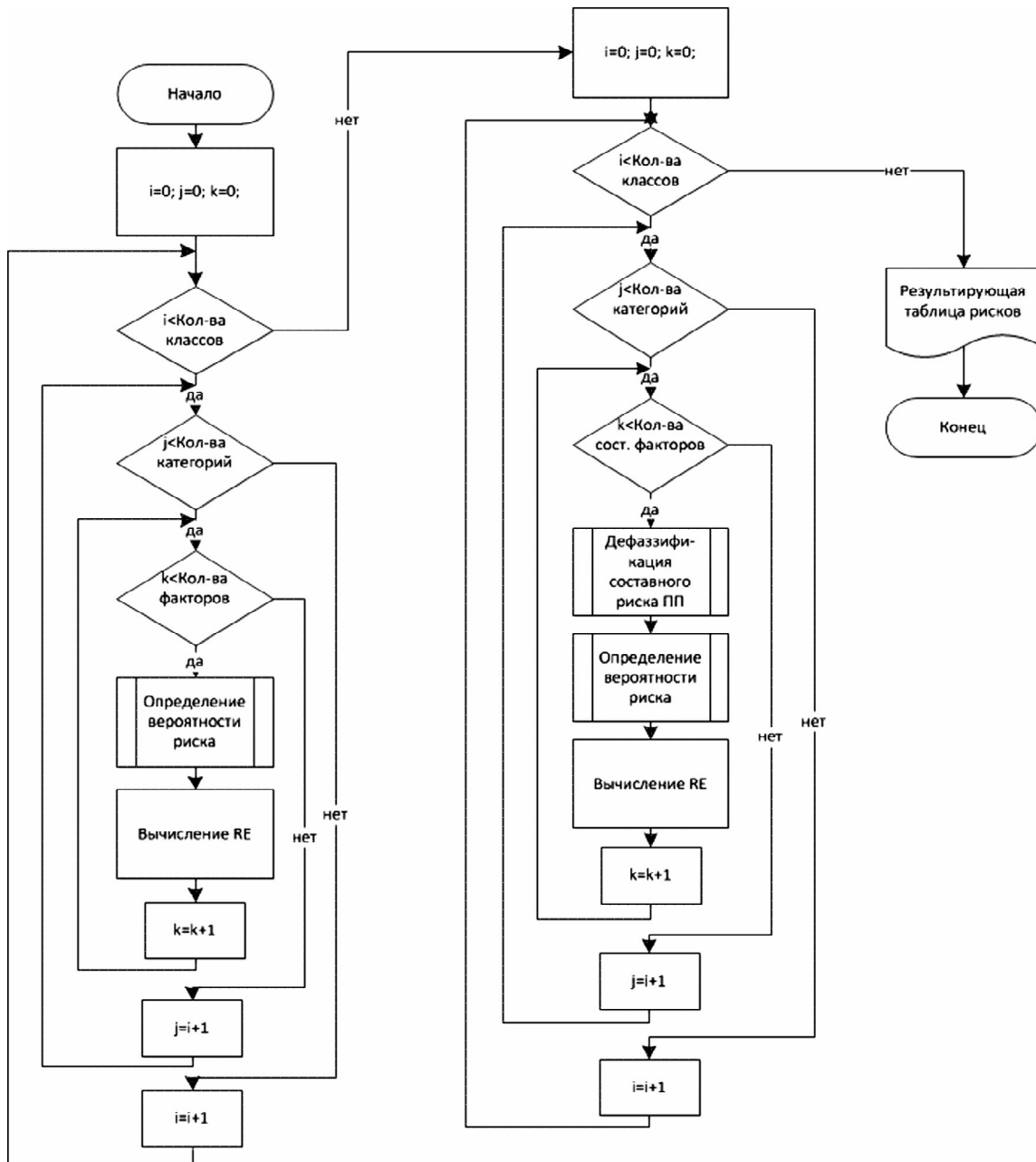


Рис. 6. Блок-схема алгоритма количественного анализа рисков ПП

АЛГОРИТМІЗАЦІЯ ПІДТРИМКИ РІШЕНЬ ПРИ УПРАВЛІННІ РИЗИКАМИ ПРОГРАМНИХ ПРОЄКТІВ НА ОСНОВІ НЕЧІТКОЇ ЛОГІКИ

Л.О. Волобуєва, Л.В. Мандрікова

В статті розглянуті питання автоматизації підтримки прийняття рішень при управлінні ризиками програмних проєктів з використанням нечіткої логіки. Розглянуто сучасний стан зазначеної проблеми, проаналізовано основні етапи процесу управління ризиками програмних проєктів, побудовані нечітко-множинні моделі для опису і рішення слабко-структурованих задач процесу. Запропоновано алгоритми для автоматизації підтримки прийняття рішень при управлінні ризиками програмних проєктів в рамках еталонної моделі класифікації ризиків.

Ключові слова: управління ризиками програмних проєктів, алгоритмізація підтримки прийняття рішень, нечітка логіка, класифікація ризиків програмних проєктів.

ALGORITHMIZATION OF THE DECISION SUPPORT BASED ON FUZZY LOGIC FOR RISK MANAGEMENT IN SOFTWARE PROJECTS

L.A. Volobuyeva, L.V. Mandrikova

The paper deals with the automation of decision support for risk management in software projects using fuzzy logic. The modern state of the problem is considered, the main steps in the process of software projects risk management are analyzed, fuzzy models are built to describe and solve semistructured problems of the process. The algorithms for automated decision support for risk management in software projects using reference model of risk classification are proposed.

Keywords: software projects management, making decision support automation, fuzzy logic, classification of software projects risks.