

УДК 658.012.32:658.012.23:658.011.3:004.4

И.Б. ТУРКИН, Л.В. МАНДРИКОВА, Л.А. ВОЛОБУЕВА

*Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ»***НЕЧЕТКО-МНОЖЕСТВЕННЫЕ МОДЕЛИ И МЕТОД ДЛЯ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ ПРИ УПРАВЛЕНИИ РИСКАМИ ПРОГРАММНЫХ ПРОЕКТОВ**

Рассмотрено современное состояние проблемы автоматизации управления рисками программных проектов. Синтезирована эталонная модель классификации рисков программного проекта (ПП) на основе известных классификаций. Предложен нечетко-множественный метод и формализованы лингвистические переменные для автоматизации поддержки принятия решений при управлении рисками ПП в рамках эталонной модели классификации. Приведен пример построения базы правил для реализации автоматизированной поддержки принятия решений на основе использования базы классификации рисков, предложенной Barry Boehm., представленной средствами эталонной модели.

Ключевые слова: управление рисками программных проектов, поддержка принятия решений, нечеткая логика, классификация рисков программных проектов.

Введение

Специфика программного проекта заключается в его нематериальной природе. Управление таким проектом [1,2] затруднено из-за сложности отслеживания процесса разработки программного продукта, большого разнообразия существующих методологий и уникальностью решений для крупномасштабных программных проектов [3]. Процесс разработки программного продукта требует индивидуального подхода в каждом конкретном случае с явным превалированием творческой составляющей. Творчество - это интеллектуальная деятельность человека, законы которой нам неизвестны [4]. В силу своей специфики успешность реализации программного проекта является очень зависимой от человеческого фактора. Неудачный подбор коллектива разработчиков может привести к провалу проекта, независимо от вложенных в него средств и затраченного времени.

Рост сложности программного обеспечения и быстрая смена технологий вынуждает менеджеров проекта использовать ранее не отработанные ими технологии и инструментарию для отсрочки морального устаревания программного продукта. Это усугубляет проблему прогнозирования результатов. Риски [5] являются неотъемлемой частью любого проекта, но в силу описанных выше особенностей управление рисками программных проектов принимает особую актуальность. Новаторское программное обеспечение может принести как сверхприбыли так и большие убытки вследствие провала проекта. По данным Standish Group, проанализировавшей работу более трех сотен американских корпораций и итоги выполнения нескольких десятков тысяч проектов, связанных с разработкой ПО, видна неутешительная тенденция, представленная на рис. 1.

Из года в год растет процентное соотношение проектов, которые полностью провалились и были аннулированы до завершения. Статистика за 2008 год

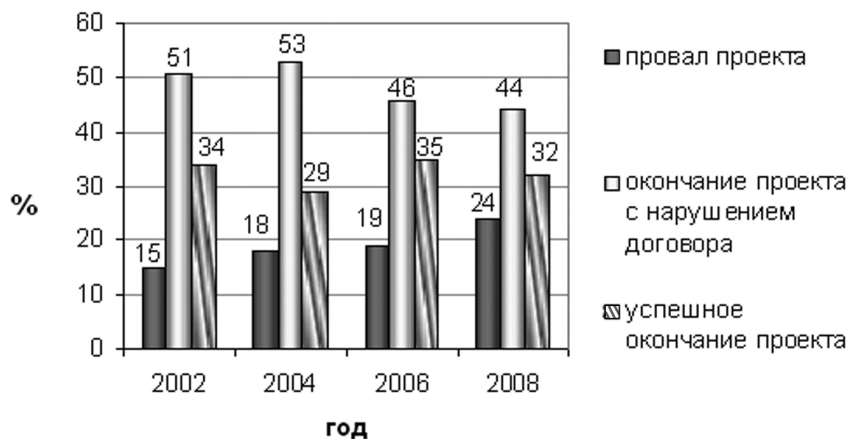


Рис. 1. Результаты исследований Standish Group: CHAOS reports

является худшей за прошедшие десять лет. Одной из основных причин, объясняющих наблюдаемую тенденцию, есть рост сложности разрабатываемого программного обеспечения.

Профессиональные знания менеджеров программных проектов играют решающую роль в реализации проекта и являются наиболее ценным ресурсом предприятия - его интеллектуальным капиталом.

Сохранение, обновление и пополнение знаний ведущих специалистов на основе создания систем поддержки принятия решений - одна из важных проблем современного IT-предприятия.

Целью этой статьи является разработка средств инженерии знаний для поддержки принятия решений в условиях специфической неопределенности, которые смогут быть основой для разработки интеллектуальных систем, способных приобретать, хранить и совершенствовать знания опытных менеджеров проектов и разработчиков ПО для оперативной поддержки принятия решений при управлении рисками в процессе реализации проекта программного продукта.

1. Синтез эталонной модели классификации рисков программного проекта

Управление рисками включает в себя следующие процессы [1]: Планирование управления рисками (Risk Management Planning), Идентификация рисков (Risk Identification), Качественный анализ рисков (Qualitative Risk Analysis), Количественный анализ рисков (Quantitative Risk Analysis), Планирование реагирования на риски (Risk Response Planning), Мониторинг и контроль рисков (Risk Monitoring and Control).

Планирование управления рисками – принятие решения о подходах и планировании действий по управлению рисками проекта. Первым этапом управления рисками является выработка членами проектной группы концепции управления рисками, которая позволит описать ключевую часть формального документа, называемого Планом управления рисками. Определяющими положениями здесь являются: распределение ролей и ответственности, указание доступных ресурсов, определение временных ограничений для мероприятий по управлению рисками, определение применяемых инструментария и методик, разработка процедуры мониторинга прогресса программного проекта, утверждение анализируемых категорий риска.

Существует множество классификаций рисков проектов по разработке ПО. Хорошо известны работы Barry Boehm, Capel Jones, и SEI Software Risk Taxonomy, описывающие источники таких рисков.

Barry Boehm [4] предложил список десяти наиболее распространенных рисков программного проекта. SEI Software Risk Taxonomy [2] разделяет риски разработки ПО на три класса (Продукт, Среда разработки, Проектные ограничения), каждый из которых содержит соответствующие элементы. Классификация рисков в рамках MSF [6] разделяет источники рисков на четыре класса (Люди, Процессы, Технологии, Внешние условия), содержащие соответствующие элементы.

В рассмотренных классификациях риска большинство факторов риска пересекаются, хоть и отнесены к различным категориям или классам. Например, в SEI Software Risk Taxonomy в классе «продукт» в категории «требования» есть такие факторы как «полнота» и «понятность», в работе [7] в категории «состав проекта» есть фактор «полнота и четкость требований», обобщающий указанные два фактора Software Risk Taxonomy. Для обеспечения учета всех факторов риска, выделенных в различных классификациях, построим эталонную модель классификации рисков программного проекта. Для ее описания используем теоретико-множественное представление.

Для описания эталонной модели выделим следующие множества:

- классификаторов $X = \{x_1, x_2, \dots, x_n\}$;
- категорий $Y = \{y_1, y_2, \dots, y_m\}$;
- факторов $F = \{f_1, f_2, \dots, f_p\}$;
- составных факторов $S = \{s_1, s_2, \dots, s_k\}$;
- критериев $Z = \{z_1, z_2, \dots, z_w\}$.

Опишем отношения между этими множествами:

$R_1 \subseteq X \times Y$, «Категория y принадлежит классу x »; $R_2 \subseteq Y \times F$, «Фактор f принадлежит категории y »; $R_3 \subseteq Y \times S$, «Составной фактор s принадлежит категории y »; $R_4 \subseteq S \times F$, «Составной фактор s содержит фактор f »; $R_5 \subseteq F \times Z$ «Фактор f имеет критерий z ».

Лингвистическая переменная [8, 9] задается 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 термины, полученные процедурой G .

Для реализации рассмотренных ранее классификаций на основе эталонной модели выделено 82 фактора риска. Фрагмент описания лингвистических переменных для формализации факторов риска f_1, f_{67}, f_{68} показан в табл. 1.

Таблица 1

Формализация факторов риска программного проекта

№	Фактор риска	Лингвистическая переменная	Функция принадлежности
1	<p>Стабильность требований</p> <p>степень ущерба - серьезная;</p> <p>критерий - Requirements Stability Index</p> $RSI = \frac{R_T - CR_C}{R_T}$ <p>R_T – количество требований; CR_C – совокупное количество изменений.</p>	<p>f_1 – Стабильность требований:</p> <p>l – стабильность требований; T – {высокая, средняя, низкая}; X – индекс стабильности требований в диапазоне $(-\infty, 1]$;</p> <p>G – синтаксическая процедура; M – семантическая процедура задания на $X = (-\infty, 1]$ нечетких подмножеств</p> <p>A_1^1 = "высокая", A_2^1 = "средняя", A_3^1 = "низкая", а также, преобразующая полученные процедурой G новые термины в нечеткие переменные.</p>	
67	<p>Выполнимость графика разработки</p> <p>степень ущерба - серьезная;</p> <p>критерий - Очевидность риска</p>	<p>f_{67} – Выполнимость графика разработки:</p> <p>l – выполнимость графика разработки; T – {высокая, средняя, низкая}; X – очевидность риска $[0, 1]$;</p> <p>G – синтаксическая процедура; M – семантическая процедура задания на $X = [0, 1]$ нечетких подмножеств</p> <p>A_1^{67} = "высокая", A_2^{67} = "средняя", A_3^{67} = "низкая", а также, преобразующая полученные процедурой G новые термины в нечеткие переменные.</p>	
68	<p>Объем бюджета</p> <p>степень ущерба - катастрофическая;</p> <p>критерий - Очевидность риска</p>	<p>f_{68} – Объем бюджета:</p> <p>l – объем бюджета; T – {достаточный, спорный, сомнительный}; X – очевидность риска $[0, 1]$;</p> <p>G – синтаксическая процедура; M – семантическая процедура задания на $X = [0, 1]$ нечетких подмножеств</p> <p>A_1^{68} = "достаточный", A_2^{68} = "спорный", A_3^{68} = "сомнительный", а также, преобразующая полученные процедурой G новые термины в нечеткие переменные.</p>	

2. Построение нечетко-множественных моделей для автоматизации поддержки принятия решений при управлении рисками программных проектов

Для этапов идентификации, качественного анализа и планирования реагирования характерно получение результата путем принятия решений на основании как качественных так и количественных характеристик, измеренных в различных шкалах. В целях автоматизации соответствующих задач поддержки принятия решений [8, 9] целесообразно

использовать аппарат нечетких множеств, позволяющий учесть «размытый» характер информации, поступающей в качестве входных данных для принятия решений.

Опишем теперь классификацию, предложенную в работе Варту Воеhm, через эталонную модель. Эта классификация содержит десять рисков программных проектов без разделения на классы и категории. Поэтому опишем общий класс x_1 и категорию Y_1 . R_1 будет содержать пару (x_1, y_1) . Из десяти рисков три являются одинарными и семь – составными.

К одинарным рискам относятся: (5) Ненужная оптимизация и оттачивание деталей, (6) Непрерывающийся поток изменений, (10) "Разрыв" в квалификации специалистов разных областей знаний. Их описание покажем на примере (5) и (6). Например, Непрерывающийся поток изменений представим в виде фактора f_1 - Стабильность требований, Ненужную оптимизация и оттачивание деталей - в виде фактора f_{22} - Оптимизация, а "Разрыв" в квалификации специалистов разных областей знаний - в виде фактора f_{55} - Подбор команды. Остальные семь рисков относятся к составным. Это: (1) Дефицит специалистов, (2) Нереалистичные сроки и бюджет, (3) Реализация несоответствующей функциональности, (4) Разработка неправильного пользовательского интерфейса, (7) Нехватка информации о внешних компонентах, определяющих окружение системы или вовлеченных в интеграцию, (8) Недостатки в работах, выполняемых внешними (по отношению к проекту) ресурсами, (9) Недостаточная производительность получаемой системы.

Определение составных факторов покажем на примере риска (2). Нереалистичные сроки и бюджет представим в виде составного фактора S_2 . Для его описания пары $(s_2, f_{67}), (s_2, f_{68})$ должны принадлежать R4. Формализация составного фактора S_2 в виде лингвистической переменной «Нереалистичные сроки и бюджет» представлена в табл. 2. Функция принадлежности в табл. 2. представлена в общем виде и используется для всех нечетких множеств составных факторов риска S_j : $N_i = A_i^j, i = \overline{1,3}, j = \overline{1,7}$.

R2 содержит такие пары:

$$(y_1, f_{22}), (y_1, f_1), (y_1, f_{55}).$$

В R3 входят следующие пары:

$$(y_1, s_1), (y_1, s_2), (y_1, s_3),$$

$$(y_1, s_4), (y_1, s_5), (y_1, s_6), (y_1, s_7).$$

Реализация нечеткого вывода выполнена на основе использования механизма Мамдани (Mamdani) [8]. Это наиболее распространенный способ логического вывода в нечетких системах. В нем используется минимаксная композиция нечетких множеств. Формально задача поддержки принятия решения при идентификации риска (2) «Нереалистичные сроки и бюджет» ставится так. Заданы наблюдения относительно факторов риска f_{67}, f_{68} и таблица нечетких правил. Требуется выполнить фазификацию этих значений, вывести решение относительно S_2 на основе нечетких правил и полученное решение дефазифицировать для принятия решений относительно целевого признака – «Нереалистичные сроки и бюджет».

Входными данными для процедуры нечеткого вывода являются фазифицированные значения факторов f_{67} – «Выполнимость графика разработки» и f_{68} – «Объем бюджета». Для решения поставленной задачи опишем связь между входом и выходом с помощью в таблицы нечетких правил. Правила, используемые для идентификации составного фактора «Нереалистичные сроки и бюджет» приведены в табл. 3. Дефазификация выполняется центроидный методом. Геометрический смысл такого значения – центр тяжести для кривой итоговой функции принадлежности.

Таблица 2

Формализация составного фактора S_2

№	фактор риска (степень ущерба); критерий	лингвистическая переменная	функция принадлежности
2	Нереалистичные сроки и бюджет (катастрофическая); Очевидность риска	S_2 – Нереалистичные сроки и бюджет: I – Нереалистичные сроки и бюджет; T – {реалистичные, спорные, сомнительные}; X – очевидность риска [0, 1]; G – синтаксическая процедура; M – семантическая процедура задания на $X = [0, 1]$ нечетких подмножеств $A_1^2 =$ "реалистичные", $A_2^2 =$ "спорные", $A_3^2 =$ "сомнительные", а также, преобразующая полученные процедурой G новые термины в нечеткие переменные.	

Таблица 3

Правила, используемые для идентификации составного фактора «Нереалистичные сроки и бюджет»

№	f_{67} – выполнимость графика	f_{68} – объем бюджета	S_2 – нереалистичные сроки и бюджет
1	высокая	достаточный	реалистичные
2	высокая	спорный	спорные
3	высокая	сомнительный	сомнительные
4	средняя	достаточный	реалистичные
5	средняя	спорный	спорные
6	средняя	сомнительный	сомнительные
7	низкая	достаточный	спорные
8	низкая	спорный	сомнительные
9	низкая	сомнительный	сомнительные

3. Пример поддержки принятия решений при управлении рисками помощью СППР «Риск-менеджер»

Заданы значения факторов риска по выбранной классификации (Barry Boehm). Требуется идентифицировать риски, выполнить качественный и количественный анализ рисков.

Работа с системой начинается с создания проекта и утверждения классификации рисков, которая будет использоваться для поддержки принятия решений по управлению рисками ПП.

Выбранная классификация представляется в терминах эталонной модели классификации рисков.

В интерактивном режиме менеджером проекта задаются наблюдаемые значения всех необходимых для выбранной классификации факторов риска по соответствующему критерию.

Далее, для всех одинарных факторов выполняется фазификация значений. Для каждого составного фактора система фазифицирует все входящие в него одинарные факторы и на основе соответствующей базы нечетких правил выполняет логический вывод значений составного фактора.

Результаты идентификации факторов риска для тестового проекта с использованием классификации по (Barry Boehm), полученные системой Риск-менеджер, показаны на рис. 2.

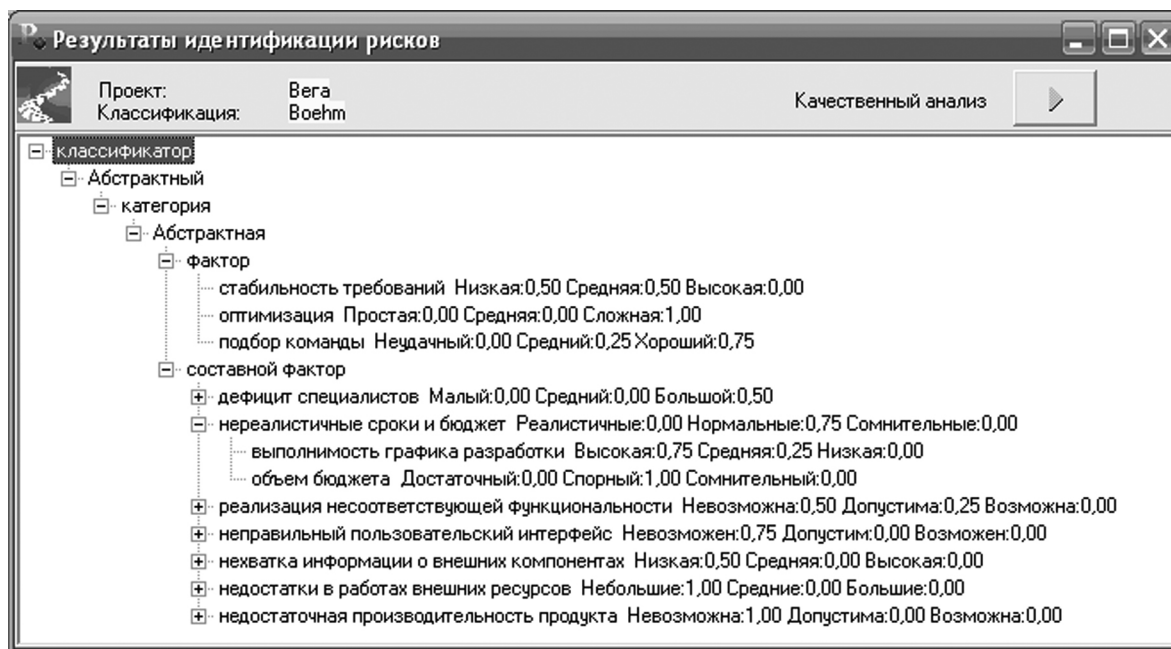


Рис. 2. Результаты идентификации рисков анализируемого программного проекта, полученные с помощью СППР Риск-менеджер

Ранг риска определяется на основании матрицы вероятностей и последствий [4]. Для всех десяти рисков проекта определяется ранг риска в зависимости от вероятности его наступления и степени ущерба от реализации риска.

Результаты количественного анализа в виде списка рисков в порядке убывания ранга риска представлены на рис. 3.

По результатам качественного анализа для всех рисков, ранг которых превысил заданное пороговое значение, равное четырем, в рамках количественного анализа определяется RE (фактор зависимости от риска). Результаты количественного анализа – перечень рисков с высоким рангом, который представлен на рис. 4. Риски перечислены в порядке убывания ранга.

№	фактор	ранг	степень ущерба	вероятность	классификация	категория
1	СФ дефицит специалистов	9	катастрофическая	очень вероятно	Абстрактный	Абстрактная
2	стабильность требований	6	серьезная	очень вероятно	Абстрактный	Абстрактная
3	СФ нереалистичные сроки и бюджет	6	катастрофическая	возможно	Абстрактный	Абстрактная
4	СФ реализация несоответствующей функциональности	3	катастрофическая	маловероятно	Абстрактный	Абстрактная
5	оптимизация	2	незначительная	очень вероятно	Абстрактный	Абстрактная
6	подбор команды	2	серьезная	маловероятно	Абстрактный	Абстрактная
7	СФ неправильный пользовательский интерфейс	2	серьезная	маловероятно	Абстрактный	Абстрактная
8	СФ нехватка информации о внешних компонентах	2	серьезная	маловероятно	Абстрактный	Абстрактная
9	СФ недостатки в работах внешних ресурсов	2	серьезная	маловероятно	Абстрактный	Абстрактная
10	СФ недостаточная производительность продукта	2	серьезная	маловероятно	Абстрактный	Абстрактная

Рис. 3. Результаты качественного анализа рисков

№	фактор	RE	ранг	степень ущерба	вероятность	классификатор	категория
1	СФ дефицит специалистов	0,85	9	катастрофическая	очень вероятно	Абстрактный	Абстрактная
2	стабильность требований	0,44	6	серьезная	очень вероятно	Абстрактный	Абстрактная
3	СФ нереалистичные сроки и бюджет	0,50	6	катастрофическая	возможно	Абстрактный	Абстрактная

Рис. 4. Результаты количественного анализа рисков

Идентифицированные риски фиксируются в Реестре рисков. По результатам качественного и количественного анализа Реестр рисков обновляется. Полученные данные используются в качестве входной информации на последующих этапах процесса управления рисками ПП – планировании реагирования на риски, а также мониторинга и контроля рисков.

Риски остальных классификаций выявляются посредством описания этих классификаций в терминах эталонной модели и последующего решения соответствующих задач поддержки принятия решений на основе построения аналогичных моделей.

Полученные результаты обеспечат сохранение интеллектуального капитала IT-предприятия в виде знаний ведущих менеджеров проекта и разработчиков ПО, сокращение сроков подготовки программного проекта, повышение качества проектных решений при проектировании программного обеспечения, повышение качества продукции.

Выводы

Описаны основные вопросы, возникающие при автоматизации управления рисками программных проектов. Синтезирована эталонная модель классифи-

кации рисков ПП. Показано построение нечетко-множественных моделей принятия решений на примере подзадач идентификации, качественного и количественного анализа рисков в рамках одной из возможных классификаций рисков ПП. Разработанные модели и средства являются основой интеллектуального ядра системы «Риск-менеджер». СППР «Риск-менеджер» обеспечивает автоматическое формирование реестра рисков и плана управления рисками, мониторинг рисков проекта, ранжирование рисков и поддержку принятия решений по смягчению рисков.

Литература

1. *Руководство к своду знаний по управлению проектами. (PMBOK®) [Текст]. Четвертое издание. - Project Management Institute, Inc., 2004. – 241 с.*
2. *Guide to Software Engineering Base of Knowledge (SWEBOOK) [Текст]. - IEEE Computer Society, 2004. – 200 p.*
3. *Сомервилл, И. Инженерия программного обеспечения, 6-е издание [Текст]. / И. Сомервилл – М.: Издательский дом «Вильямс», 2002. – 624 с.*

4. *Архипенков, С. Лекции по управлению программными проектами [Электронный ресурс] / С. Архипенков. – М., 2009. – Режим доступа: <http://www.arkhipenkov.ru>.*

5. *Вишняков, Я.Д. Общая теория рисков [Текст]: учеб. пособ. для студ. высш. учеб. заведений / Я.Д. Вишняков. – М.: Издательский центр «Академия», 2007. – 368 с.*

6. *Risk Management Process Whitepaper [Электронный ресурс]. - MSt Solutions Framework, 2001. – Режим доступа: <http://www.microsoft.com/msf>*

7. *Шафер, Д.Ф. Управление программными проектами: достижение оптимального качества при минимуме затрат [Текст]: пер. с англ. / Д.Ф. Шафер, Р.Т. Фатрел, Л.И. Шафер. – М.: Издательский дом «Вильямс», 2003. – 1136 с.*

8. *Люггер, Д.Ф. Искусственный интеллект: стратегии и методы решения сложных проблем, 4-е издание [Текст]: пер. с англ. / Д.Ф. Люггер. - М.: Издательский дом «Вильямс», 2003. – 864 с.*

9. *Гаврилова, Т.А. Базы знаний интеллектуальных систем [Текст] / Т.А. Гаврилова, В.Ф. Хорошевский. – СПб: Питер, 2000. – 384 с.*

Поступила в редакцию 14.04.2011

Рецензент: д-р техн. наук, проф., заведующий кафедрой компьютерных систем и сетей В.С. Харченко, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков.

НЕЧІТКО-МНОЖИННІ МОДЕЛІ І МЕТОД ДЛЯ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ПРИ УПРАВЛІННІ РИЗИКАМИ ПРОГРАМНИХ ПРОЄКТІВ

І.Б. Туркін, Л.В. Мандрікова, Л.О. Волобуєва

Розглянуто сучасний стан проблеми автоматизації управління ризиками програмних проєктів. Синтезована еталонна модель класифікації ризиків програмного проєкту на основі відомих класифікацій. Запропонований нечітко-множинний метод і формалізовані лінгвістичні змінні для автоматизації підтримки прийняття рішень при управлінні ризиками ПП в рамках еталонної моделі класифікації. Наведений приклад побудови бази правил для реалізації автоматизованої підтримки прийняття рішень на основі використання класифікації ризиків, запропонованої Barry Boehm, що представлена засобами еталонної моделі.

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

FUZZY SET MODELS AND METHOD FOR MAKING DECISION SUPPORT AT MANAGEMENT OF PROGRAMMATIC PROJECTS RISKS

I.B. Turkin, L.V. Mandrikova, L.A. Volobueva

The modern state of problem of programmatic projects risks management automation is considered. The standard model of programmatic project risks classification is synthesized on the basis of the known classifications. Offered fuzzy set method and linguistic variables are formalized for automation of making decision support at a management the PP risks within the framework of standard model of classification. The resulted example of rule base construction is represented for realization of the automated making decision support on the basis of the use of risks classification, offered Barry Boehm, that presented by facilities of standard model.

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

Туркин Игорь Борисович – д-р техн. наук, проф., заведующий кафедрой инженерии программного обеспечения, Национальный аэрокосмический университет, Харьков, Украина, e-mail: po_as_k603@d6.khai.edu.

Мандрікова Людмила Васильевна – ассистент кафедры инженерии программного обеспечения, Национальный аэрокосмический университет, Харьков, Украина, e-mail: po_as_k603@d6.khai.edu.

Волобуєва Лина Алексеевна – канд. техн. наук, доцент кафедры инженерии программного обеспечения, Национальный аэрокосмический университет, Харьков, Украина, e-mail: linavolobueva@gmail.com.