УДК 004.052
DOI: 10.15827/0236-235X.125.005-011

Дата подачи статьи: 06.09.18
2019. Т. 32. № 1. С. 005–011

Использование нечетко-множественного подхода при управлении заданиями ИТ-проекта

*А.Р. Диязитдинова*¹, к.т.н., доцент, dijazitdinova@mail.ru

*Н.И. Лиманова*¹, д.т.н., профессор, nataliya.i.limanova@gmail.com

¹ Поволжский государственный университет телекоммуникаций и информатики,
г. Самара, 443010, Россия

Распределение и назначение ресурсов относятся к сложным многокритериальным задачам. В связи с этим в управлении проектами по созданию программных продуктов актуальной представляется задача разработки эффективных и универсальных методов оптимального распределения работ между исполнителями. Одним из возможных инструментов повышения обоснованности решений, принимаемых руководителем проекта компаний, занимающихся разработкой программных продуктов, может выступить нечеткая логика, которая позволяет оперировать слабоструктурированной и неточной информацией с использованием естественного языка.

В статье предлагается модель нечеткой продукционной системы для управления заданиями ИТ-проекта, позволяющая оперировать естественно-языковыми категориями с целью повышения эффективности принятия решений в условиях неопределенности и снижения затрат при возникновении неблагоприятных ситуаций. Рассмотрены особенности проекта по созданию программного продукта, разработана типовая схема процесса управления заданиями в ИТ-проекте, показана целесообразность применения аппарата нечетких систем для управления заданиями. Использование математического аппарата нечеткой логики позволит руководителю проекта работать с переменными, выраженными в качественных категориях, без перехода к средним значениям, что будет способствовать повышению качества принимаемых решений при управлении проектом.

В рамках работы рассматривается задача оценки успешности выполнения задания (тикета) разработчиками. Выделены шесть входных лингвистических переменных и одна выходная, для каждой из которых разработаны терм-множества и функции принадлежности. Построена экспертная база правил, включающая 81 продукционное правило; разработана модель нечеткой продукционной системы для управления заданиями на базе пакета Fuzzy Logic Toolbox for MatLab. В качестве схемы нечеткого вывода использован алгоритм Мамдани. Приведены результаты функционирования модели, которые могут быть полезны руководителям ИТ-проектов на практике.

Ключевые слова: ИТ-проект, управление проектом по созданию программного продукта, управление заданиями, нечеткие системы (fuzzy-системы), лингвистические переменные, функции принадлежности, Fuzzy Logic Toolbox for MatLab.

Одним из эффективных инструментов, способствующих успешному достижению конечного результата в разных областях деятельности, является проектное управление. Управление проектом по созданию *программного продукта* (ПП) – сложная, трудоемкая и длительная работа, требующая высокой квалификации задействованных специалистов. Основное отличие ИТ-проекта от проекта, реализуемого в других отраслях (строительство, производство и пр.), в том, что проектное управление при создании ПП имеет дело с не вещественными, неосязаемыми результатами.

Процесс принятия решения при управлении ИТ-проектами осуществляется в условиях неполноты информации и нечетко сформулированных требований, что накладывает дополнительные ограничения на качество создаваемого ПП. Кроме того, руководителю проекта необходимо учитывать влияние человеческого фактора, поскольку качество ПП, его себестоимость и опосредованно прибыль ИТ-компании находятся в прямой зависимости от профессионализма и знаний разработчиков. Это обстоятельство вынуждает рассматривать задачу распределения заданий и назначения ресурсов как одну из наиболее важных в деятельности руководителя проекта.

Данная задача относится к классу сложных многокритериальных задач, критерии в которой не всегда за-

даны в количественной форме, поэтому использование академического математического аппарата может быть затруднено. Повышение эффективности процесса назначения задач конкретным исполнителям может быть обеспечено инструментарием, базирующимся на методах и моделях нечетких продукционных систем, способных адекватно взаимодействовать с ЛПР за счет предоставления возможности оперировать лингвистическими переменными. Таким образом, в управлении ИТ-проектами представляется актуальной разработка эффективных и универсальных методов решения задачи оптимального распределения работ между исполнителями.

Особенности проекта по созданию ПП

В управлении ИТ-проектом можно выделить существенные особенности, которые обуславливают специфику процесса создания любого ПП [1, 2].

1. ПП как конечный результат ИТ-проекта не имеет вещественной формы, и его нельзя измерить в общепринятых единицах. В силу этого планирование работ ИТ-проекта должно осуществляться как можно детальнее. Каждый ИТ-проект является уникальным, поэтому в отличие от проектов в строительстве или

производстве у таких проектов нет нормативов затрат для типовых операций, как, собственно, нет и самого понятия «типовая операция». Планировать длительность работ можно лишь примерно, основываясь на опыте решения аналогичных задач в похожих проектах.

2. В течение жизненного цикла проекта требования к конечному ПП часто меняются. В результате возникает необходимость сложной процедуры изменения и согласования требований.

3. Следствием предыдущей особенности является неуправляемость большинства процессов разработки, поскольку представления заказчика о конечном результате зачастую весьма расплывчаты, а сам процесс получения качественного ПП не поддается формализации. В итоге невозможно осуществить детальное планирование необходимых работ.

4. Успешность ИТ-проекта на 100 % зависит от профессионализма задействованных сотрудников при том, что на рынке труда недостаточно высококвалифицированных специалистов.

5. Существует большое разнообразие используемых инструментов и сред разработки. Как следствие, чем новее и мощнее используемый инструмент разработки, тем меньше профессионалов, владеющих им.

6. Сегодня существует значительное количество разнообразных легких и тяжелых моделей разработки ПО: ГОСТ 34-й серии, SW-CMM, Rational Unified Process, Microsoft Solutions Framework, Extreme Programming, Scrum и др. Каждая модель обладает достоинствами и недостатками, поэтому компаниям методом проб и ошибок приходится адаптировать лучшие практики под собственные нужды [3]. Попытки предложить формальную детализованную методологию разработки ПО оказываются безуспешными, поскольку сам процесс разработки не поддается детализации и формализации. Слепое следование методологиям, предполагающим управляемость и предсказуемость процессов разработки проекта, приводит к непредсказуемым результатам.

Руководители ИТ-проекта должны учитывать факторы, определяющие сложность проекта, оказывать управляющее воздействие на них, применяя специфичные для данного вида проектов инструменты управления. Окончательный набор факторов, их вес и критичность зависят от специфики каждого конкретного проекта.

Помимо особенностей, касающихся непосредственно хода выполнения самого проекта, следует отметить и ряд особенностей ИТ-отрасли в целом, что косвенным образом влияет и на успешность того или иного проекта.

Во-первых, большая текучесть кадров. Для предотвращения утечки знаний, наработанных опытными сотрудниками, и для сокращения времени вхождения новичка в рабочий проект необходимо создание системы управления знаниями.

Во-вторых, отсутствие механизмов мотивации труда сотрудников. В ИТ-отрасли уровень зарплаты довольно высок, но количество профессионалов недостаточное, что обуславливает текучесть кадров, так как зачастую проще переманить специалиста у конкурента, чем повышать квалификацию имеющихся сотрудников.

В-третьих, наличие таких внешних факторов, негативно влияющих на деятельность ИТ-компаний, как высокий уровень киберугроз безопасности, отсутствие по сравнению с зарубежьем налоговых льгот для ИТ, несовершенное законодательство, отсутствие четко работающей системы обучения будущих разработчиков ПО.

Моделирование процесса управления заданиями ИТ-проекта

Согласно PMBoK, одним из важных процессов в области знаний является управление человеческими ресурсами компании. Фактически речь идет об эффективном управлении командой проекта. В зависимости от степени вовлеченности в процесс создания ПП участников проекта можно разделить на три группы: основная команда, расширенная команда, заинтересованные лица. Типовая ролевая модель процесса разработки ПП приведена в таблице.

В рамках управления проектом значительное место в работе его руководителя занимают постановка задач перед командой в целом и перед каждым членом команды в частности и последующий мониторинг исполнения планов. Типовая схема процесса управления заданиями ИТ-проекта приведена на рисунке 1.

Одним из распространенных инструментов распределения полномочий и ответственности между участниками команды проекта является построение используемой для определения ролей и обязанностей разработчиков матрицы ответственности RACI (Responsible, Accountable, Consult before doing, Inform after doing). В результате кодирования формируется таблица, характеризующая участие той или иной роли в процессе выполнения задач. К сожалению, рекомендации к построению матрицы не содержат сведений о том, из чего должен исходить руководитель проекта, возлагая задачу на конкретного исполнителя.

Система управления заданиями представляет собой набор инструментов для постановки задач и мониторинга их исполнения, имеющих ряд функциональных возможностей: создание новой задачи, назначение ответственных, разграничение прав доступа, фиксирование трудоемкости исполнения задачи.

Алгоритм работы системы управления заданиями типовой и включает следующие шаги:

— постановка задачи (тикета), то есть определение ее сути и критичности, автора, исполнителя и контролера;

- определение сроков выполнения (трудоемкости) задачи: автор может установить фиксированный срок выполнения задачи, а может предоставить эту возможность исполнителю;
- исполнение задачи: пользователь самостоятельно фиксирует факт завершения;
- контроль исполнения: контролер осуществляет оценку полноты и корректности выполненной задачи, в зависимости от чего задача может быть завершена или отправлена на доработку;
- информирование автора о закрытии задачи.

Ролевая модель процесса разработки ПП

The role model of the software development process

Роль	Зона ответственности
Руководитель проекта	Формирование команды проекта Формирование коммуникационной среды Формирование планов, в том числе <ul style="list-style-type: none"> – оценка длительности и трудоемкости задач в процессе планирования; – распределение работ внутри команды Контроль выполнения планов, в том числе <ul style="list-style-type: none"> – мониторинг выполнения планов; – проверка на соответствие деятельности команды бизнес-процессу разработки; – организационная работа (в том числе и с заказчиком); – часть аналитической работы
Аналитик	Сбор требований заказчика Разработка технического проекта Доработка спецификации Разработка планов тестирования Концептуальное тестирование функциональности
Технический лидер	Разработка концептуальной архитектуры решения Разработка рабочего проекта Контроль качества кода и соответствие его проектным решениям по архитектуре Участие в формировании планов и оценке сложности и длительности задач Участие в комплексном тестировании
Разработчики	Разработка функциональности Проверка качества кода Исправление ошибок в коде Проведение первичного тестирования кода Участие в комплексном тестировании кода
Тестировщик	Тестирование функциональности Написание unit-тестов Участие в разработке планов тестирования
Технический писатель	Разработка пользовательской документации

Системы управления заданиями достаточно распространены на рынке. Существует большое количество как коммерческих, так и распространяемых по свободной лицензии продуктов. К наиболее популярным следует отнести пакеты Asana, Basecamp, JIRA,

Redmine, Битрикс24, Trello, Мегатлан, MS Project и др. Как показал анализ, ПП управления заданиями ориентированы именно на аспекты мониторинга исполнения задачи. Сам вопрос, почему та или иная задача возложена на конкретного исполнителя, остается открытым и решается субъективно руководителем проекта на основании его опыта.

С технической точки зрения вопрос автоматизации процесса управления заданиями достаточно прост. Основная сложность заключается в необходимости адекватно отобразить модель и стиль управления, принятые в компании. Поэтому ПО управления заданиями должно быть индивидуальным и отражать все особенности политики управления персоналом в конкретной компании. Следует отметить, что наличие системы управления заданиями автоматически не повышает эффективность работы: основные проблемы, связанные с налаживанием процессов коммуникации и контроля, не исчезают.

Процесс управления заданиями представляет собой сложную многокритериальную задачу распределения ресурсов по видам деятельности и по конкретным исполнителям с учетом их занятости, квалификации, специализации и т.д. [4]. Классическая теория предлагает использовать методы теории игр либо сводить к условиям задачи о назначении (для решения с помощью венгерского алгоритма). Но применение венгерского алгоритма требует количественного задания условий, что в реальной управленческой практике не всегда возможно. Классические методы теории игр в уникальных условиях разработки и реализации ИТ-проектов также не всегда оказываются неприменимыми, поскольку отсутствует возможность сбора статистических данных по вариантам решения каждой отдельной задачи.

Задачи, связанные с управлением проектом, характеризуются высоким уровнем ответственности ЛПР, изменяющейся структурой проекта как системы управления, слабоструктурированной, неопределенной и противоречивой информацией, на основании которой принимаются решения. Команде проекта приходится решать слабоструктурируемые и неструктурируемые проблемы, характеризующиеся невозможностью использования методов и моделей, основанных на точном описании проблем [5]. В подобных условиях одним из возможных инструментов может выступить нечеткая логика.

Нечеткая модель системы управления заданиями

Основой для разработки нечетких систем (fuzzy-system) является технология нечеткого моделирования, базирующаяся на средствах формализации и анализа слабоструктурированной и неполной информации, возникающей вследствие присущей объектам и процессам сложности [6, 7]. Особенность нечетких си-

стем заключается в том, что для описания поведения моделируемой системы используется лингвистическая аппроксимация, основанная или на знаниях экспертов предметной области, или на предварительном анализе статистической информации.

В 1994 г. Коско доказал теорему о нечеткой аппроксимации [8], в соответствии с которой любая математическая система может быть аппроксимирована системой на нечеткой логике. Следовательно, с помощью продукций вида «ЕСЛИ..., ТО...» при последующей их формализации средствами нечеткой логики можно точно отразить произвольную взаимосвязь «вход–выход» без использования сложного аппарата дифференциальных и интегральных вычислений, традиционно применяемого в управлении. В настоящее время нечеткая логика представляет собой достаточно детально разработанный и хорошо зарекомендовавший себя инструмент решения задач управления и принятия решений.

Отличительные достоинства fuzzy-систем [7, 9]:

- возможность оперировать нечеткими входными данными, например, непрерывно изменяющимися во времени значениями (динамические задачи) или значениями, которые невозможно задать однозначно в численном выражении;
- возможность отказа от сложных систем управления, основанных на решении дифференциальных уравнений, в случае, если это позволяет требуемая точность вычислений;
- описание процесса принятия решений на естественном языке с использованием субъективных и привычных для человека качественных оценок и привязка этих оценок к строгому математическому аппарату;
- возможность проведения качественных оценок входных и результирующих данных.

В рамках решения задачи управления заданиями рассматривается задача оценки успешности выполне-

ния задания разработчиками. Другие участники проекта не рассматриваются, поскольку в каждом проекте таким ролям, как руководитель проекта, аналитик, технический писатель, тестировщик, обычно соответствует один сотрудник. Поэтому при появлении задачи конкретной направленности (написание документации, аналитическая задача, тестирование и пр.) она может быть назначена только на сотрудника с требующейся специализацией. В рамках проекта заданий, ориентированных на разработку ПО, всегда в несколько раз больше, чем разработчиков, поэтому одной из важных задач, стоящих перед руководителем проекта, является корректное назначение им заданий.

Были выделены следующие входные лингвистические переменные и их терм-множества:

- complexity {low, average, higt} – сложность задачи {низкая, средняя, высокая};
- laborintensity {low, average, higt} – трудоемкость задачи {низкая, средняя, высокая};
- novelty {new, rare, typical} – новизна задачи {новая, редко встречающаяся, типовая};
- priority {non-critical, urgent, critical} – приоритет задачи {неприоритетная, срочная, критичная};
- developer {Junior, Middle, Senior} – профессионализм разработчика {младший, средний, старший};
- employment {low, average, high, very high} – занятость на проекте {низкая, средняя, высокая, очень высокая}.

Функции принадлежности входных переменных приведены на рисунке 2.

В качестве выходной переменной использована переменная result {unsatisfactory, satisfactory, successful} – успешность выполнения задачи {неудовлето-

вительно, удовлетворительно, успешно}, функция принадлежности которой приведена на рисунке 3.

Рис. 3. Функции принадлежности выходной переменной

Fig. 3. Membership functions of the output variable

В результате анализа экспертами было сформулировано 81 наиболее значимое правило. Приведем некоторые из них:

- 1) ЕСЛИ **complexity** average И **laborintensity** low И **novelty** typical И **priority** non-critical И **developer** Junior И **employment** low ТО **result** successful
- 2) ЕСЛИ **complexity** average И **laborintensity** average И **novelty** new И **priority** non-critical И **developer** Middle И **employment** average ТО **result** successful
- 3) ЕСЛИ **complexity** average И **laborintensity** average И **novelty** new И **priority** non-critical И **developer** Middle И **employment** higt ТО **result** satisfactory
- 4) ЕСЛИ **complexity** average И **laborintensity** average И **novelty** new И **priority** urgent И **developer** Senior И **employment** higt ТО **result** satisfactory
- 5) ЕСЛИ **complexity** average И **laborintensity** average И **novelty** new И **priority** urgent И **developer** Senior И **employment** very higt ТО **result** satisfactory

Рис. 2. Функции принадлежности переменной: а) сложность задачи, б) трудоемкость задачи, в) новизна задачи, г) приоритет задачи, д) профессионализм разработчика, е) занятость в проекте

Fig. 2. The variable membership functions: a) task complexity, б) task labour intensity, в) task novelty, г) task priority, д) developer's competencies, е) project employment

6) ЕСЛИ *complexity* average И *laborintensity* average И *novelty* new И *priority* critical И *developer* Middle И *employment* average TO *result* satisfactory

7) ЕСЛИ *complexity* average И *laborintensity* average И *novelty* new И *priority* critical И *developer* Middle И *employment* higt TO *result* unsatisfactory

8) ЕСЛИ *complexity* average И *laborintensity* average И *novelty* new И *priority* critical И *developer* Senior И *employment* higt TO *result* satisfactory

9) ЕСЛИ *complexity* average И *laborintensity* average И *novelty* rare И *priority* non-critical И *developer* Middle И *employment* low TO *result* successful

10) ЕСЛИ *complexity* average И *laborintensity* average И *novelty* rare И *priority* non-critical И *developer* Middle И *employment* higt TO *result* satisfactory

Для построения системы нечеткого вывода использован пакет Fuzzy Logic Toolbox системы Matlab. В составе Matlab присутствуют пять основных средств графического интерфейса пользователя: редакторы системы нечеткого вывода, функции принадлежности, правила вывода, а также средства просмотра правил и поверхности вывода. Эти средства связаны между собой динамически, и производимые изменения в одном из них влекут изменения в других [10, 11]. С помощью утилиты нечеткого вывода FIS Editor заданы следующие параметры системы нечеткого вывода: тип – Мамдани, метод агрегирования, метод аккумуляции, метод дефазификации, все входы и выходы нечеткой продукционной системы.

Функции принадлежности переменных были заданы в соответствующих окнах. Например, для переменной «Сложность задачи» установлена трапецевидная функция принадлежности и указан вид функции для каждого члена терм-множества. Аналогичным образом определены и другие переменные.

Заключительным этапом построения системы нечеткого вывода является определение набора правил, которые задают связь входных переменных с выходными (см. <http://www.swsys.ru/uploaded/image/2019-1/2019-1-dop/11.jpg>).

Средство просмотра правил вывода позволяет отобразить процесс нечеткого вывода и получить результат (см. <http://www.swsys.ru/uploaded/image/2019-1/2019-1-dop/12.jpg>). Главное окно средства просмотра состоит из нескольких графических окон, располагаемых в виде строк и столбцов. Количество строк соответствует числу правил нечеткого вывода, а количество столбцов – числу входных и выходных переменных, заданных в разрабатываемой нечеткой системе. В каждом окне отображаются соответствующая функция принадлежности, уровень ее среза (для входных переменных) и вклад отдельной функции принадлежности в общий результат (для выходных переменных) [11]. Для оценки успешности выполнения задачи разработчиками в окне Rule Viewer в нижней левой части экрана вводятся значения входных переменных,

а результат оценки отображается в правом верхнем углу окна. Также система нечеткого вывода предоставляет возможность визуализации поверхности зависимости степени успешности выполнения задачи от двух входных параметров при фиксированных значениях остальных параметров.

Оценка успешности выполнения каждого конкретного задания (тикета) индивидуальна. Система нечеткого логического вывода поможет руководителю проекта оценить позитивный и негативный результаты решения задачи в случае назначения тикета на конкретного исполнителя и скорректировать свой выбор.

Заключение

Исследование процесса управления ИТ-проектом в общем и процесса управления заданиями в частности показало наличие слабоструктурированной, неформализуемой и субъективной информации, что препятствует успешному достижению конечного результата. Предлагаемая нечеткая модель системы управления заданиями является открытой, то есть руководители проекта могут добавлять в нее новые входные переменные, присущие конкретному ИТ-проекту, что приведет только к изменению количества правил при сохранении логики модели. Использование математического аппарата нечеткой логики даст возможность руководителю проекта работать с переменными, выраженными в качественных категориях, без перехода к средним значениям, что в итоге снизит субъективность оценки принимаемых решений.

Литература

1. Ройс У. Управление проектами по созданию программного обеспечения; [пер. с англ.]. М.: Лори, 2002. 424 с.
2. Кожина А.В. Особенности управления ИТ-проектами // Общественные науки и общество: социально-гуманитарные науки. 2016. № 4. С. 84–88.
3. Архипенков С.Я. Руководство командой разработчиков программного обеспечения. М.: Самиздат, 2008. 80 с.
4. Диязитдинова А.Р., Кольцова В.А. Концепция ИСУП для совершенствования управления распределением трудовых ресурсов // Инфокоммуникационные технологии. 2014. № 4. С. 71–76.
5. Гречуха Е.И. Нечеткий ситуационный подход и управление проектами. Аналитический обзор // Научный вестн. междунар. гуманитарного ун-та. 2010. № 2. С. 79–82.
6. Ехлаков Ю.П., Пермякова Н.В. Нечеткая модель оценки рисков продвижения программных продуктов // Бизнес-информатика. 2014. № 3. С. 69–78.
7. Круглов В.В., Длин М.И. Интеллектуальные информационные системы: компьютерная поддержка систем нечеткой логики и нечеткого вывода. М.: Физматлит, 2002. 256 с.
8. Штовба С.Д. Проектирование нечетких систем средствами MATLAB. М.: Горячая линия–Телеком, 2007. 288 с.
9. Бочарников В.П. Fuzzy-технология: Математические основы. Практика моделирования в экономике. СПб: Наука РАН, 2001. 328 с.
10. Леоненков А.В. Нечеткое моделирование в среде MATLAB и fuzzyTECH. СПб: БХВ-Петербург, 2003. 719 с.
11. Дьяконов В.П., Круглов В.В. Математические пакеты расширения MATLAB: Специальный справочник. СПб: Питер, 2001. 480 с.

Fuzzy set approach for IT project task management

A.R. Diyazitdinova¹, Ph.D. (Engineering), Associate Professor, diyazitdinova@mail.ru
N.I. Limanova¹, Dr.Sc. (Engineering), Professor, nataliya.i.limanova@gmail.com

¹ Volga State University of Telecommunication and Informatics, Samara, 443010, Russian Federation

Abstract. Resource distribution and allocation problems are complex multi-criteria tasks. Therefore, the problem of development of effective and universal technologies of work assignment among performers turns out to be challenging in software project management. One of the possible solutions to increase the relevance of project management decision-making in software development companies might be fuzzy logic. It allows processing semi-structured and inaccurate information using a natural language.

The paper proposes a model of fuzzy production system to manage IT project tasks that allows operating natural language categories to improve the efficiency of decision making under uncertainty and cost cutout in the extreme. The authors consider software product development features; develop a typical logic of IT project tasks management process; prove fuzzy logic technology application reasons are for project management. Implementation of fuzzy logic mathematical tools technique allows a project manager to operate variables represented in quality categories without transferring to mean values that enables decision-making quality increase.

The paper considers a problem of task (ticket) development performance evaluation. There are derived six input linguistic variables and one output. There are developed term sets and membership functions for each of them. The built expert rule base includes 81 production rules. A model of fuzzy logic production system model for tasks management has been implemented using Fuzzy Logic Toolbox for MatLab. The Mamdani algorithm has been used for fuzzy inference. The provided results of the model functioning would be useful for IT project managers.

Keywords: IT project, software development project management, task management, fuzzy systems, linguistic variables, membership functions, MatLab fuzzy logic toolbox.

References

1. Royce W. *Software engineering project management*. Addison-Wesley Longman Publ., Boston, MA, USA, 1998 (Russ. ed.: Moscow, Lori Publ., 2002, 424 p.).
2. Kozhina A.V. IT project management features. *Education and Science without Borders: Social and Human Sciences*. 2016, no. 4, pp. 84–88 (in Russ.).
3. Arkhipenkov S.Ya. *Software development team leadership*. Moscow, Samizdat Publ., 2008, 80 p.
4. Diyazitdinova A.R., Koltsova V.A. The ERP concept to improve labor resources allocation management. *Infocommunication Technologies*. 2014, no. 4, pp. 71–76 (in Russ.).
5. Grechukha E.I. Fuzzy citation approach and project management. Analytical review. *Sci. Herald of the Intern. Humanitarian Univ.* 2010, no. 2, pp. 79–82 (in Ukr.).
6. Ekhlakov Yu.P., Permyakova N.V. Fuzzy risk assessment model for software promotion. *Business Informatics*. 2014, no. 3, pp. 69–78 (in Russ.).
7. Kruglov V.V. *Intelligent information systems: computer support for fuzzy logic and fuzzy inference systems*. Moscow, Fizmatlit Publ., 2002, 256 p.
8. Shtovba S.D. *Fuzzy System Design Using MATLAB*. Moscow, Goryachaya liniya–Telekom Publ., 2007, 288 p.
9. Bocharnikov V.P. *Fuzzy-technology: mathematical basics. simulation practice in economy*. St. Petersburg, Nauka RAN Publ., 2001, 328 p.
10. Leonenkov A.V. *Fuzzy simulation in MATLAB and fuzzyTECH*. St. Petersburg, BHV-Peterburg Publ., 2003, 719 p.
11. Dyakonov V.P., Kruglov V.V. *MATLAB mathematical expansion packs*. St. Petersburg, Piter Publ., 2001, 480 p.

Примеры библиографического описания статьи

1. Диязитдинова А.Р., Лиманова Н.И. Использование нечетко-множественного подхода при управлении заданиями ИТ-проекта // Программные продукты и системы. 2019. Т. 32. № 1. С. 5–11. DOI: 10.15827/0236-235X.125.005-011.
2. Diyazitdinova A.R., Limanova N.I. Fuzzy set approach for IT project task management. *Software & Systems*. 2019, vol. 32, no. 1, pp. 5–11 (in Russ.). DOI: 10.15827/0236-235X.125.005-011.