

УДК 658.012.32

А.А. Соколов<sup>1</sup>, Н.В. Доценко<sup>1</sup>, И.В. Чумаченко<sup>2</sup>

<sup>1</sup> Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков

<sup>2</sup> Харьковский национальный университет городского хозяйства имени А.Н. Бекетова, Харьков

## ПРИМЕНЕНИЕ МЕТОДА ДЕРЕВЬЕВ РЕШЕНИЙ ДЛЯ ОПРЕДЕЛЕНИЯ УРОВНЯ ПОДГОТОВКИ МУЛЬТИПРОЕКТНОЙ КОМАНДЫ

*В данной статье рассматриваются вопросы определения уровня компетенций у команды проекта мультипроектной организации, вопросы развития команды проекта и применяется метод деревьев решений.*

**Ключевые слова:** управление человеческими ресурсами; развитие команды проектов; мультипроектные организации; деревья решений.

### Введение

Современные IT компании, разрабатывающие параллельно большое количество проектов, используют различные модели бизнес-процессов и разработки. Наиболее распространенные методы разработки – Scrum [1], Agile [2, 3], Waterfall model [4].

Agile подразумевает тесное взаимодействие всех участников проекта, а так же прямые коммуникации всех членов команды в одном помещении. Scrum предполагает итерационный подход разработки крупного проекта, waterfall model (каскадная модель) – поток работ, последовательно проходящий все фазы разработки.

Не все компании ведут разработку методом Agile в связи с тем, что различные отделы и участники проекта находятся в разных офисах, городах и даже странах. В связи с этим сохранение и развитие команды проекта является одной из важнейших задач управления проектами.

Процесс формирования команды мультипроекта носит итерационный характер и предусматривает возможность перераспределения ресурсов в рамках управления мультипроектом. Использование компетентностного подхода при формировании команды проекта позволяет учитывать компетенции претендентов в команду.

Существующие методы учитывают наличие или отсутствие компетенции, однако не учитывают ее уровень [5].

Наличие компетенции подразумевает владение компетенцией на минимально необходимом уровне, что не обеспечивает заданные показатели эффективности функционирования (качества работы) команды мультипроекта.

Поскольку в рамках мультипроекта необходимая степень владения компетенциями может быть различной, возникает необходимость разработки

эффективных методов формирования мультипроектных команд на основе учета уровня владения компетенциями. В случаях, когда команда проекта уже полностью сформирована, но у некоторых специалистов недостаточно компетенции по определенным областям знаний, существует два пути решения проблемы: найм нового сотрудника и повышение квалификации текущего. В большинстве случаев на поиск и отбор нового сотрудника требуются значительные затраты времени и средств. В таком случае более эффективно развивать навыки текущих сотрудников.

Имея данные об уровнях компетенций текущих сотрудников, оценки руководителей, а также данные об опыте работы на занимаемой должности, руководитель может указать какой именно навык и у какого именно сотрудника нужно развивать, для достижения цели с минимальными временными и материальными затратами.

Поэтому задача классификации на основе известной базы данных является актуальной. Известно множество методов построения классификационных правил – от эвристических правил руководящих менеджеров, до применения методов интеллектуального анализа данных, как деревья решений, байесовские правила, нейронных сетей.

### Формулировка задач исследования

**Целью данной работы** является разработка рекомендаций по применению байесовского подхода и построенного на нем метода деревьев решений для определения степени подготовки участника проекта и определение его роли в проекте.

При решении задачу формирования команды проекта в мультипроектной среде необходимо учитывать следующие аспекты:

1) подбор нового сотрудника и определение наиболее подходящей роли в проектах компании;

2) реформирование имеющейся команды проекта при создании нового проекта.

При возникновении потребности в решении данной задачи формирования команды входными данными является резюме или CV (curriculum vitae), которое анализируется кадровым отделом и формализуется в виде табличных данных.

Так же во внимание принимается участие кандидата в уже завершенных проектах, их изучение и анализ его действий в проекте. На основе вышесеречисленных данных руководство принимает субъективное бинарное решение, характеризующее сотрудника как «хороший», или «плохой».

Для определения возможности включения претендента в команду проекта на основе данных о самом проекте предлагается составить тест, в котором будут определены основные направления требований к претендентам для участия в проекте.

Рассмотрим пример формирования команды проекта реконструкции адаптивного web-сайта, созданного на системе drupal. В команду проекта входят дизайнер, верстальщик и программист.

На начальном этапе определяем отличительные черты проекта, влияющие на состав команды проекта:

1. Для создания нового дизайна нет необходимости учитывать особенности cms drupal.

2. Для верстки адаптивного сайта необходим специалист, владеющий навыками кроссбраузерной адаптивной верстки с применением технологий (html 5, css3).

3. Для программирования сайта на cms drupal необходим опытный программист, имевший опыт успешной реализации аналогичных проектов. Поскольку в проекте предусмотрена реконструкция существующего сайта, то к особенностям cms добавляется необходимость опыта разбора «чужого» кода.

В результате анализа специфики проекта определяем систему требований:

Дизайнер:

1. Знание Adobe Photoshop: да/нет.  
2. Знание разрешений экранов современных топовых устройств: да/нет.

3. Разработка дизайна макетов проекта под различные разрешения (десктоп/мобильные устройства): да/нет.

Верстальщик:

1. Отличное знание HTML 5, CSS 3, Bootstrap: да/нет.

2. Знание PHP, JavaScript: да/нет.  
3. Владение Adobe Photoshop: да/нет.  
4. Знание MooTools, jQuery: да/нет.  
5. Верстка страниц любой сложности с соблюдением стандартов и кросс-браузерной совместимости: да/нет.

6. Английский (чтение документации): да/нет.

Программист:

1. Стаж (опыт работы): количество лет.  
2. Разработка веб-сайтов с нуля на cms drupal: да/нет.

3. Разработка веб-сайтов на других cms: да/нет.

4. Хорошее знание php: да/нет.

5. Работа с базами данных MySQL: да/нет.

6. Знание JavaScript: да/нет.

7. Английский (чтение документации): да/нет.

## Оценка компетентности на основе нечетких отношений

Ответы на вопросы к тестам формируем в виде таблицы (нечеткого отношения).

Результаты опроса претендентов приведены в табл. 1.

Таблица 1

Результаты опроса

	B <sub>1</sub>	B <sub>2</sub>	B <sub>3</sub>	B <sub>4</sub>
Иванов	0,4	1	0	0,5
Петров	1	0,6	0,5	0,6
Сидоров	0,3	0	0,9	0,4
Кузнецов	1	0,5	0,9	0,4

Связь между вопросами и проверяемыми умениями также представляем в виде таблицы (табл. 2) – нечеткого отношения, где

B – вопросы для проверки умений P;

P – умения (навыки).

Таблица 2

Таблица нечетких отношений

B/P	Программирование на PHP	Разработка типовых решений на CMS	HTML верстка	JAVA программирование	MySQL программирование
B <sub>1</sub>	1	0,2	0	0	0
B <sub>2</sub>	0	0,4	0,8	1	0
B <sub>3</sub>	0,5	1	0,3	0,4	0
B <sub>4</sub>	0,8	0,5	0,1	0	1

На следующем этапе с помощью максимных отношений, с учетом первичных данных о стаже работы и субъективных оценках формируем общую таблицу связей проверяемых кандидатов и умений (табл. 3). Например, для претендента Иванов значение 0,5 (Программирование на PHP) определяется следующим образом:

$$0.5 = \max(\min(1, 0.4), \min(1, 0), \min(0, 0.5), \min(0.5, 0.8)) .$$

Таким образом, подготовили данные для построения дерева решений. Данных в конкретном примере для построения показательного дерева решений недостаточно, в связи с этим для построения дерева будут применены случайно сгенерированные данные в большем количестве. Далее на основе данных таблицы применим метод деревьев решений.

Таблица 3

Общая таблица связей проверяемых и умений

	Программирование на PHP	Разработка типовых решений на CMS	HTML верстка	JAVA программирование	MySQL программирование
Иванов	0,5	0,5	0,8	1	0,5
Петров	1	0,5	0,6	0,6	0,6
Сидоров	0,5	0,9	0,3	0,4	0,4
Кузнецов	1	0,9	0,5	0,5	0,4

**Применение метода деревьев решений для классификации показателей компетенций сотрудников**

Деревья решений последовательно разбивают кандидатов на группы по одной из переменных, так чтобы эти группы насколько возможно отличались по величине риска некомпетентности. Процесс разбиения продолжается до тех пор, пока оставшиеся группы не становятся настолько малы, что следующее разбиение не приведёт к статистически значимому различию в уровне риска. Каждому листу дерева затем приписывается определённая категория кандидатов.

Метод деревьев решений основывается на Байесовом подходе, представленном в примере:

Имеем базу данных кандидатов в проект или на должность, отнесенных к классам.

Кандидаты 1..12 – имеют типы А, Б, В (точнее, их объективные показатели).

Классы – Хороший, Плохой.

База данных известных кандидатов представлена в табл. 4.

Таблица 4

База данных известных кандидатов

1	А	Хороший
2	А	Хороший
3	А	Хороший
4	Б	Хороший
5	А	Плохой
6	Б	Плохой
7	Б	Плохой
8	В	Хороший
9	В	Хороший
10	В	Плохой
11	В	Плохой
12	В	Плохой

В базе данных 12 кандидатов: 6 «Хороших», 6 «Плохих».

Вероятность класса «Хороший»

$$P(\text{Хороший}) = \frac{6}{12} = 0.5 .$$

Вероятность класса Плохой

$$P(\text{Плохой}) = \frac{6}{12} = 0.5 .$$

В классе Хороших кандидатов 3 клиента относятся к типу А.

Вероятность клиента А в классе Хороших кандидатов

$$P(A / \text{Хороший}) = \frac{3}{6} .$$

В классе Плохих клиентов 1 клиент1 типа А.

Вероятность клиента А в классе Плохих клиентов

$$P(A / \text{Плохой}) = \frac{3}{6} .$$

Тогда вероятность того, что *новый кандидат* типа А (которого еще нет в базе данных) будет отнесен к классу Хороших кандидатов, равна

$$P(\text{Хороший}/A) = \frac{P(\text{Хороший}) * P(A/\text{Хороший})}{P(\text{Хороший}) * P(A/\text{Хороший}) + P(\text{Плохой}) * P(A/\text{Плохой})} = 0.75 .$$

Тогда вероятность того, что *новый кандидат* типа А (которого еще нет в базе данных) будет отнесен к классу Плохих кандидатов, равна

$$P(\text{Плохой}/A) = \frac{P(\text{Плохой}) * P(A/\text{Плохой})}{P(\text{Хороший}) * P(A/\text{Хороший}) + P(\text{Плохой}) * P(A/\text{Плохой})} = 0.25 .$$

Данного нового кандидата А следует отнести к типу Хороший, поскольку вероятность такого отнесения 0,75.

Если база данных будет меняться со временем, вероятности также будут меняться, и отнесение новых кандидатов к классам также может быть другим.

Аналогично можно рассчитать вероятности отнесения к соответствующему классу новых кандидатов типа Б и В.

Дерево решений или классификаций представляет процесс принятия решений, где на каждом этапе принимается бинарное решение. Дерево состоит из узлов и ветвей, где узлы обозначают разветвления решений.

Внутренними узлами являются те, которые имеют двух наследников, в то время конечные узлы не имеют наследников. Конечный узел имеет классметку, связанную с ним таким образом, что данные, которые соответствуют условию узла, присваиваются к этому классу.

Для определения дерева решений воспользуемся стандартной функцией пакета MatLab, и случайно сгенерированными исходными данными (табл. 5):

t = classregtree(T(:,1:7),T(:,8),'names',{'P1' 'P2' 'P3' 'P4' 'P5' 'P6' 'P7' },'minparent',1).

Таблиця 5

Исходные данные  
для построения дерева решений

2.5000	0.6000	0.8000	0.9000	0.7000	0.9000	1.0000	1.0000
2.0000	0.1000	0.8000	0.8000	0.2000	0.2000	0.7000	1.0000
5.0000	0.7000	0.1000	0.9000	1.0000	0.7000	0.8000	1.0000
4.0000	0.7000	0.1000	0.3000	1.0000	0.7000	0.8000	0
3.5000	0.7000	0.1000	0.9000	1.0000	0.1000	0.1000	1.0000
1.5000	0.7000	0.1000	0.9000	1.0000	0.7000	0.8000	0
4.0000	0.3000	0.8000	0.9000	1.0000	0.1000	0.8000	1.0000
1.0000	1.0000	0.7000	0.2000	0.9000	0.7000	0.1000	0

Построенное дерево решений имеет вид

Decision tree for regression

- 1 if  $P2 < 0.85$  then node 2 elseif  $P2 \geq 0.85$  then node 3 else 0.6375
- 2 if  $P6 < 0.15$  then node 4 elseif  $P6 \geq 0.15$  then node 5 else 0.714286
- 3 fit = 0.1
- 4 if  $P2 < 0.5$  then node 6 elseif  $P2 \geq 0.5$  then node 7 else 0.45
- 5 if  $P6 < 0.8$  then node 8 elseif  $P6 \geq 0.8$  then node 9 else 0.82
- 6 fit = 0.8
- 7 fit = 0.1
- 8 if  $P2 < 0.4$  then node 10 elseif  $P2 \geq 0.4$  then node 11 else 0.775
- 9 fit = 1
- 10 fit = 0.7
- 11 fit = 0.8.

В результате анализа дерева решения определяются направления повышения компетенции.

Правила в дереве решений определяют, что если претендент имеет опыт по данному навыку ( $P2$ ) и на вопрос, проверяющий этот навык, ответил очень слабо (меньше 0,1 балла), то ему необходимо повысить знания по данному навыку, аналогично с последующими навыками, согласно дереву, представленному на рис. 1.

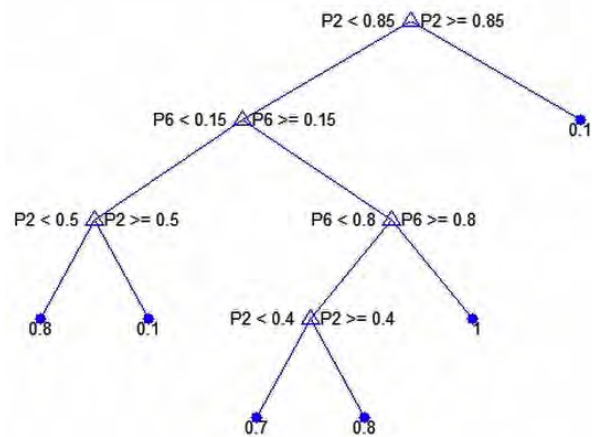


Рис. 1. Дерево решений

## Выводы

Данный метод позволяет давать рекомендации ответственному лицу по подбору кадров, какие из кандидатов заведомо сразу не подойдут на вакантную должность, а какие подойдут, но не имеют компетенции, которую нужно повысить.

## Список литературы

1. *Agile Project Management with Scrum (Microsoft Professional)*, 2004 by Ken Schwaber ISBN:073561993x.
2. Коваль Г.І. Удосконалення процесу розроблення сімейств програмних систем елементами гнучких методологій / Г.І. Коваль, А.Л. Колесник, К.М. Лавріщева, О.О. Слабоспицька // *Проблеми програмування*. – 2010. – № 2-3. – С. 261-270.
3. [Электронный ресурс]. – Режим доступа к ресурсу: <http://agilemanifesto.org/>.
4. Royce, Winston (1970), "Managing the Development of Large Software Systems".
5. 5th Ed 2013, *Project Management Body of Knowledge PMBOK Guide*.
6. *Matlab User's Guide, Chapter 9, Statistical Pattern Recognition*, 2002 by Chapman &Hall/CRC.

Поступила в редколлегию 3.03.2014

**Рецензент:** д-р техн. наук, доцент М.К. Сухонос, Харьковский национальный университет городского хозяйства имени А.Н. Бекетова, Харьков.

## ЗАСТОСУВАННЯ МЕТОДУ ДЕРЕВ РІШЕНЬ ДЛЯ ВИЗНАЧЕННЯ СТУПЕНЯ ПІДГОТОВКИ МУЛЬТИПРОЕКТНОЇ КОМАНДИ

О.О. Соколов, Н.В. Доценко, І.В. Чумаченко

У даній статті розглядаються питання визначення рівня компетенцій у команді проекту мультипроектної організації, питання розвитку команди проекту і застосовується методу дерев рішень.

**Ключові слова:** управління людськими ресурсами; розвиток команди проектів; мультипроектні організації; дерева рішень.

## APPLICATION OF THE METHOD OF TREES SOLUTIONS FOR DEFINITIONS OF POWERS TRAINING OF COMMANDS

O.O. Sokolov, N.V. Dotsenko, I.V. Chumachenko

In this article discussed the issues of determining the level of competence of the team project multiproject organization, development of the project team and the method of decision trees.

**Keywords:** management human capitals; development of command of projects; multiproject organizations; trees of decisions.