4. *Гладышев А.И., Аборкина Е.С.* Вопросы применения существующих методов оценки сложности информационных систем // Вестник Российского нового университета. Серия «Сложные системы: модели, анализ, управление». 2016. Вып. 1–2. С. 114–118.
5. ГОСТ Р 51275-2006. Объект информатизации. Факторы, воздействующие на информацию. Общие положения. М., 2007. 7 с.
6. Доктрина информационной безопасности Российской Федерации от 5 декабря 2016 г. № Пр-646. М., 2016. 16 с.
7. *Платонов В.* Программно-аппаратные средства защиты информации: учебник. М.: Academia, 2014. 336 с.
8. *Родичев Ю.А.* Нормативная база и стандарты в области информационной безопасности: учеб. пособие. СПб.: Питер, 2017. 256 с.
9. *Семененко В.А.* Информационная безопасность: учеб. пособие. 4-е изд., стер. М.: МГИУ, 2010. 277 с.

Literatura

1. *Adamenko M.A.* Osnovy klassicheskoy kriptologii. Sekrety shifrov i kodov. M.: DMK Press, 2012. 256 s.
2. *Biryukov A.A.* Informatsionnaya bezopasnost': zashchita i napadenie. M.: DMK Press, 2012. 474 s.
3. *Gashkov S.B., Primenko E.A., Cherepnev M.A.* Kriptograficheskie metody zashity informatsii. M.: Akademia, 2010. 304 s.
4. *Gladyshev A.I., Aborkina E.S.* Voprosy primeneniya sushchestvuyushchikh metodov otsenki slozhnosti informatsionnykh sistem // Vestnik Rossiyskogo novogo universiteta. Seriya "Slozhnye sistemy: modeli, analiz, upravlenie". 2016. Vyp. 1–2. S. 114–118.
5. GOST R 51275-2006. Ob"ekt informatizatsii. Faktory, vozdeystvuyushchie na informatsiyu. Obshchie polozheniya. M., 2007. 7 s.
6. Doktrina informatsionnoy bezopasnosti Rossiyskoy Federatsii ot 5 dekabrya 2016 g. № Pr-646. M., 2016. 16 s.
7. *Platonov V.* Programmno-apparatnye sredstva zashity informatsii: uchebnik. M.: Academia, 2014. 336 s.
8. *Rodichev Yu.A.* Normativnaya baza i standarty v oblasti informatsionnoy bezopasnosti: ucheb. posobie. SPb.: Piter, 2017. 256 s.
9. *Semenenko V.A.* Informatsionnaya bezopasnost': ucheb. posobie. 4-e izd., ster. M.: MGIU, 2010. 277 s.

DOI: 10.25586/RNU.V9187.19.02.P.123 УДК 681.518:339.13

С.В. Хаджиева

ПРИМЕНЕНИЕ МЕТОДА ЖАДНОГО АЛГОРИТМА ДЛЯ ФОРМИРОВАНИЯ КОМАНД В IT-КОМПАНИИ В УСЛОВИЯХ AGILE-ТРАНСФОРМАЦИИ

Рассматривается возможность применения эвристического метода жадного алгоритма для решения класса прикладных задач, ориентированных на формирование IT-команд по функциональным ролям из имеющегося состава сотрудников в условиях Agile-трансформации.

Ключевые слова: IT-команда, IT-компания, Agile-трансформация, формирование команд, функциональная роль, жадный алгоритм, эвристический алгоритм, оптимизационная задача, задача о разбиении множества.

© Хаджиева С.В., 2019

S.V. Hadzhieva

USING THE GREEDY ALGORITHM FOR TEAM BUILDING IN AN IT COMPANY IN THE CONTEXT OF AGILE TRANSFORMATION

The article discusses the heuristic method of the greedy algorithm for solving a class of applied problems focused on the formation of IT teams by functional roles from the existing staff in the context of Agile-transformation.
Keywords: IT team, IT company, Agile transformation, team building, functional role, greedy algorithm, heuristic algorithm, optimization problem, set splitting problem

Введение

«Стратегия развития отрасли информационных технологий в Российской Федерации на 2014–2020 годы и на перспективу до 2025 года» содержит цель – достижение уровня развития сферы информационных технологий до полноценной отрасли российской экономики, создающей высокопроизводительные рабочие места и обеспечивающей выпуск высокотехнологичной и конкурентоспособной продукции. Для ее достижения предусмотрено решение, в частности, таких задач, как улучшение институциональных условий и снижение административных барьеров. Вовлечение IT-компаний в систему целенаправленной трансформации отрасли информационных технологий и перевода их на инновационный путь развития объясняет популярность внедрения командного подхода к управлению – Agile. По сравнению с административным управлением командный метод работы в IT-сфере является более эффективным, однако переход на него – процесс нелегкий, он требует перестройки мышления, поведения и способов принятия решений. Определяющую роль в успехе играет этап формирования IT-команд, выступая неотъемлемым этапом Agile-трансформации. В этих условиях в IT-компаниях возникает потребность в методах формирования команд, которые должны быть способны самостоятельно принимать решения по способу выполнения работ и отвечать за результат.

У каждого участника команды есть две роли: функциональная и командная. Функциональные роли относятся к должностным обязанностям и охватывают навыки, умения, знания и опыт. Командные роли отражают врожденные и приобретенные личностные качества. Между тем, если распределение командных ролей исследуется в социологии и психологии [2; 8; 11], особенности функционирования команд и качества, которыми должна обладать сформированная команда, описываются в методологиях управления проектами [1; 4; 5; 7], то вопросы о распределении сотрудников на команды по функциональным ролям в интересах теории управления в настоящее время являются недостаточно проработанными.

Применительно к сфере IT под командой понимают группу размером от 5 до 10 человек, которая наделена полномочиями и способна спроектировать, разработать, протестировать инкремент ценности решения за обозначенный временной интервал и, как правило, включает следующие основные функциональные роли: аналитик, разработчик и тестировщик.

Задачу о распределении сотрудников на равноценные IT-команды можно рассмотреть как задачу о разбиении множества чисел. Это задача определения, можно ли данное мультимножество N положительных целых чисел разбить на t подмножеств, таких, что сумма чисел $N_1 \ldots N_n$ будет равна. При $t = 2$ существует решение данной задачи методом псевдополиномиального времени динамического программирования. Однако в случаях, когда требуется разбить множество N на $t > 2$ подмножеств, данный метод становится слабо эффективным и неприменимым на практике. Для случаев $t > 2$ множеств применяют эвристические алгоритмы решения, которые позволяют найти решение оптимальное, либо приближенное [6]. Цель данной статьи – рассмотреть возможность применения эвристического метода жадного алгоритма к задаче разбиения сотрудников на $t > 2$ IT-команд.

Вышеизложенные положения обусловливают актуальность темы и необходимость исследования применимости эвристических алгоритмов для решения класса прикладных задач, ориентированных на формирование IT-команд по функциональным ролям из имеющегося состава сотрудников в условиях Agile-трансформации. Предполагается, что формирование IT-команд с использованием эвристических алгоритмов позволит получить оптимальное разбиение сотрудников на равноценные команды.

В [9; 10] была представлена блок-схема алгоритма распределения сотрудников на IT-команды по функциональным ролям. В данной статье рассматривается математическая формализация поставленной задачи, последовательность шагов жадного алгоритма для решения задачи распределения сотрудников на IT-команды по функциональным ролям и результаты проведенных экспериментов.

Постановка задачи

Дано N множество сотрудников, распределяемых на t подмножеств, то есть IT-команд. В предлагаемой модели каждый сотрудник характеризуется следующими параметрами: роль сотрудника – аналитик, разработчик, тестировщик; грейд сотрудника – совокупная числовая оценка навыков, умений и знаний – G {7, 8, 9, 10, 11, 12}; опыт O в месяцах. В соответствии с грейдом и опытом задается матрица коэффициентов эффективности сотрудников K. Лицо, принимающее решение, задает желаемое количество человек h в команде.

Итак, входными параметрами системы являются:

n – количество сотрудников N $\{1 \ldots n\}$;

R – роль сотрудника $\{D, A, Q\}$;

d – количество разработчиков $D = [1 \ldots d]$;

a – количество аналитиков $A = [1 \ldots a]$;

q – количество тестировщиков $Q = [1 \ldots q]$;

h – количество человек в команде;

t – количество команд $T\{1 \ldots t\}$;

G – множество, элементами которого являются оценки грейда сотрудников $G\{7, 8, 9, 10, 11, 12\}$;

O – массив, элементами которого являются оценки опыта сотрудников; $O = [o_i]$, $i \in [1, n]$;

K– массив, элементами которого являются коэффициенты эффективности разработчиков $K_D = [k_i]$, $i \in [1, d]$, аналитиков $K_A = [k_i]$, $i \in [1, a]$, тестировщиков $K_Q = [k_i]$, $i \in [1, q]$, присвоенные в соответствии с грейдом и опытом сотрудников.

Требуется подобрать оптимальное распределение разработчиков, аналитиков и тестировщиков для каждой команды. Оптимальность решения определяется выполнением следующих условий (1.1–1.4)

$$\sum_{i=1}^{d} k_i \in T_1 \approx \sum_{i=1}^{d} k_i \in T_2 \ldots \sum_{i=1}^{d} k_i \in T_t, \tag{1.1}$$

$$\sum_{i=1}^{a} k_i \in T_1 \approx \sum_{i=1}^{a} k_i \in T_2 \ldots \sum_{i=1}^{a} k_i \in T_t, \tag{1.2}$$

$$\sum_{i=1}^{q} k_i \in T_1 \approx \sum_{i=1}^{q} k_i \in T_2 \ldots \sum_{i=1}^{q} k_i \in T_t, \tag{1.3}$$

$$\sum_{i=1}^{h} k_i \in T_1 \approx \sum_{i=1}^{h} k_i \in T_2 \ldots \sum_{i=1}^{h} k_i \in T_t. \tag{1.4}$$

Условие (1.1) выражает требование к равенству коэффициентов эффективности роли «разработчик» между всеми командами. Условие 1.2 выражает требование к равенству коэффициентов эффективности роли «аналитик» между всеми командами. Условие (1.3) выражает требование к равенству коэффициентов эффективности роли «тестировщик» между всеми командами. Условие (1.4) говорит о том, что суммарные коэффициенты эффективности между командами должны быть приближенно равны.

Выходными параметрами являются подмножества из IT-команд $T\{1...t\}$, представленные в виде матрицы $M = \left[m_{ij} \right]$, $i = t;\ j = 1...n$, элементами которой являются сотрудники N с ролями $\{D, A, Q\}$ из $N\{1...n\}$ при стремлении суммы коэффициентов эффективности каждой команды к среднему значению коэффициента эффективности, рассчитанному как частное от деления суммы коэффициентов эффективности всех сотрудников N на количество команд t по формуле:

$$\bar{k} = \frac{\sum_{i=1}^{n} k_i}{t}. \tag{2}$$

Тогда целевая функция задачи поиска множеств IT-команд может быть выражена как:

$$\sum_{i=1}^{d} k_i + \sum_{i=1}^{a} k_i + \sum_{i=1}^{q} k_i \to \bar{k}. \tag{3}$$

При ограничениях:

$$\begin{cases} d, a, q \geq 1 \\ t \leq \min \sum_{n=1}^{d,a,q} n_i \end{cases}. \tag{4}$$

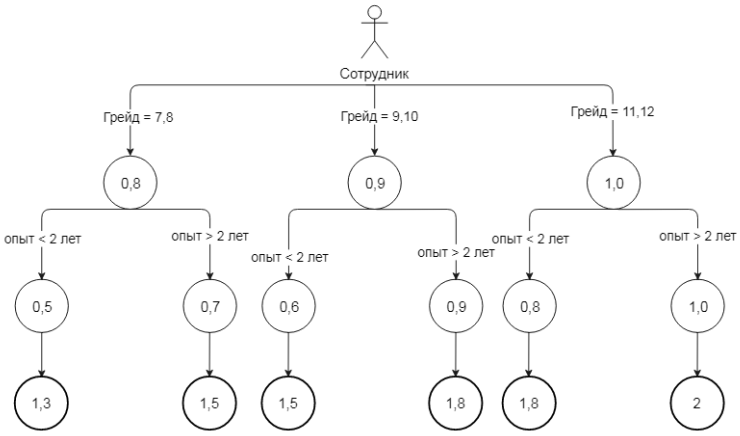
Метод жадного алгоритма при решении оптимизационных задач основан на том, что процесс принятия решения можно разбить на элементарные шаги, на каждом из которых принимается отдельное решение. Решение, принимаемое на каждом шаге, должно быть оптимальным только на текущем шаге и должно приниматься без учета предыдущих или последующих решений [3].

Алгоритм распределения сотрудников на IT-команды

Шаг 1. Количество команд рассчитывается по формуле (5), при условии, что количество команд t не может быть больше, чем количество сотрудников с ролью, в которой наименьшее количество человек:

$$t = \frac{n}{h}. \tag{5}$$

Шаг 2. Согласно рисунку рассчитать коэффициенты эффективности в соответствии с опытом и грейдом для всех сотрудников. На основании данных о грейде и опыте сотрудникам присваиваются коэффициенты эффективности. Коэффициенты рассчитываются экспертным путем с учетом того, что сотрудник с меньшим грейдом, но большим опытом оценивается выше, чем сотрудник с меньшим грейдом. Как правило, новым сотрудникам требуется минимум два года, чтобы «прочувствовать» и понять всю специфику работы в компании.



Распределение коэффициентов эффективности сотрудников

Шаг 3. Отсортировать по убыванию коэффициенты эффективности, отдельно для каждой группы сотрудников с ролью «разработчик», «аналитик», «тестировщик»:

$$D = \left[\max(n_i)\ldots\min(n_i)\right],\ i = d;$$
$$A = \left[\max(n_i)\ldots\min(n_i)\right],\ i = a;$$
$$Q = \left[\max(n_i)\ldots\min(n_i)\right],\ i = q.$$

Шаг 4. Распределить разработчиков по командам. Получить матрицу $D = \left[d_{ij}\right],\ i = t;$ j = целое от $d / t + 1$, где по строкам – множества команд T, по столбцам – коэффициенты эффективности, распределенные по следующим правилам.

4.1. Из $D = \left[\max(n_i)\ldots\min(n_i)\right],\ i = d$ выбрать t наибольших элементов, распределить их по столбцам T множеств в порядке убывания. Затем взять следующие t наибольших элементов, распределить их по T множествам в порядке, противоположном предыдущему распределению (по возрастанию). Чередуя порядок сортировки, продолжать распределять элементы по множествам T до тех пор, пока элементы не закончатся.

4.2. Подсчитать сумму коэффициентов эффективности разработчиков по строкам в каждом T множестве.

Шаг 5. Распределить аналитиков по командам. Получить матрицу $A = \left[a_{ij}\right]$, $i = t$; j = целое от $d / t + 1$, где по строкам – множества команд T, по столбцам – коэффициенты эффективности, распределенные по следующим правилам.

5.1. Отсортировать по возрастанию полученную на шаге 4.2 сумму коэффициентов эффективности разработчиков.

5.2. Из $A = \left[\max(n_i)...\min(n_i)\right]$, $i = a$ выбрать t наибольших элементов, распределить их по столбцам T множеств в порядке убывания. Затем взять следующие t наибольших элементов, распределить их по T множествам в порядке, противоположном предыдущему распределению (по возрастанию). Чередуя порядок сортировки, продолжать распределять элементы по множествам T до тех пор, пока они не закончатся.

5.3. Подсчитать сумму коэффициентов эффективности аналитиков и разработчиков по строкам в каждом T множестве.

Шаг 6. Распределить тестировщиков по командам. Получить матрицу $Q = \left[q_{ij}\right]$, $i = t$; j = целое от $d / t + 1$, где по строкам – множества команд T, по столбцам – коэффициенты эффективности, распределенные по по следующим правилам.

6.1. Отсортировать по возрастанию полученную на шаге 5.2 сумму коэффициентов эффективности разработчиков.

6.2. Из $Q = \left[\max(n_i)...\min(n_i)\right]$, $i = q$ выбрать t наибольших элементов, распределить их по столбцам T множеств в порядке убывания. Затем взять следующие t наибольших элементов, распределить их по T множествам в порядке, противоположном предыдущему распределению (по возрастанию). Чередуя порядок сортировки, продолжать распределять элементы по множествам T до тех пор, пока они не закончатся;

6.3. Подсчитать итоговую сумму коэффициентов эффективности в каждом t множестве для всех ролей.

Шаг 7. Вывести данные в матрицу $M = \left[d_{ij}\right]$, $i = t$; $j = h + 1$, где последняя строка – это итоговая сумма коэффициентов эффективности сформированных команд.

Тестирование алгоритма распределения сотрудников на IT-команды

Для тестирования данного алгоритма в качестве входных данных использовался набор данных о показателях эффективности 25 сотрудников с ролями «разработчик», «аналитик», «тестировщик».

Шаг 1. Количество команд t равно 5 (t = min = q = 5);

Шаг 2. Расчет коэффициентов эффективности в соответствии с опытом и грейдом для всех сотрудников представлен в таблице 1.

Таблица 1

Входные данные о сотрудниках

Сотрудник	**Роль**	**Грейд**	***k* по грейду**	**Опыт, мес.**	***k* по опыту**	$\sum_{i=1}^{n} k_i$
Сотрудник 1	D	12	1	30	1	2
Сотрудник 2	D	11	1	25	1	2
Сотрудник 3	D	11	1	13	0,8	1,8

Окончание табл. 1

Сотрудник	Роль	Грейд	k по грейду	Опыт, мес.	k по опыту	$\sum_{i=1}^{n} k_i$
Сотрудник 4	D	8	0,8	18	0,5	1,3
Сотрудник 5	D	9	0,9	14	0,6	1,5
Сотрудник 6	D	8	0,8	26	0,7	1,5
Сотрудник 7	D	7	0,8	26	0,7	1,5
Сотрудник 8	D	10	0,9	26	0,9	1,8
Сотрудник 9	D	7	0,8	18	0,5	1,3
Сотрудник 10	D	7	0,8	18	0,5	1,3
Сотрудник 11	A	12	1	34	1	2
Сотрудник 12	A	12	1	34	1	2
Сотрудник 13	A	11	1	13	0,8	1,8
Сотрудник 14	A	11	0,9	26	0,9	1,8
Сотрудник 15	A	10	0,9	26	0,9	1,8
Сотрудник 16	A	9	0,9	14	0,6	1,5
Сотрудник 17	A	8	0,8	26	0,7	1,5
Сотрудник 18	A	8	0,8	26	0,7	1,5
Сотрудник 19	A	7	0,8	18	0,5	1,3
Сотрудник 20	A	7	0,8	18	0,5	1,3
Сотрудник 21	Q	11	0,9	26	0,9	1,8
Сотрудник 22	Q	12	1	34	1	2
Сотрудник 23	Q	12	1	33	1	2
Сотрудник 24	Q	9	0,9	14	0,6	1,5
Сотрудник 25	Q	8	0,8	26	0,7	1,5

Шаг 3. Сортировка коэффициентов эффективности по убыванию для каждой группы сотрудников по ролям:

$$D = [2;\ 2;\ 1{,}8;\ 1{,}8;\ 1{,}5;\ 1{,}5;\ 1{,}5;\ 1{,}3;\ 1{,}3;\ 1{,}3];$$

$$A = [2;\ 2;\ 1{,}8;\ 1{,}8;\ 1{,}8;\ 1{,}5;\ 1{,}5;\ 1{,}5;\ 1{,}3;\ 1{,}3];$$

$$Q = [2;\ 2;\ 1{,}8;\ 1{,}5;\ 1{,}5].$$

Шаги 4–7. Результаты распределения сотрудников по командам для каждой роли представлены в таблицах 2–5.

Таблица 2

Распределение разработчиков по командам

Команды	**Сортировка K_D по убыванию**	**Сортировка K_D по возрастанию**	$\sum K_D$
T1	2	1,3	3,3
T2	2	1,3	3,3
T3	1,8	1,3	3,1
T4	1,8	1,5	3,3
T5	1,5	1,5	3

Таблица 3

Распределение аналитиков по командам

Команды	Сортировка $\sum K_D$ по возрастанию	Сортировка K_A по убыванию	Сортировка K_A по возрастанию	$\sum K_A$	$\sum K_A + \sum K_D$
T5	3	2	1,3	3,3	6,3
T3	3,1	2	1,3	3,3	6,4
T1	3,3	1,8	1,5	3,3	6,6
T2	3,3	1,8	1,5	3,3	6,6
T4	3,3	1,8	1,5	3,3	6,6

Таблица 4

Распределение тестировщиков по командам

Команды	Сортировка $\sum K_A + \sum K_D$ по возрастанию	$\sum K_Q$	$\sum K = \sum K_A + \sum K_D + \sum K_Q$
T5	6,3	2	8,3
T3	6,4	2	8,4
T1	6,6	1,8	8,4
T2	6,6	1,5	8,1
T4	6,6	1,5	8,1

Таблица 5

Сформированные команды

Команда/Роль	T1	T2	T3	T4	T5
D	Сотрудник 1 или 2	Сотрудник 1 или 2	Сотрудник 3 или 8	Сотрудник 3 или 8	Сотрудник 5, 6 или 7
D	Сотрудник 4,9 или 10	Сотрудник 4,9 или 10	Сотрудник 4,9 или 10	Сотрудник 5, 6 или 7	Сотрудник 5, 6 или 7
A	Сотрудник 13, 14 или 15	Сотрудник 13, 14 или 15	Сотрудник 11 или 12	Сотрудник 13, 14 или 15	Сотрудник 11 или 12
A	Сотрудник 16, 17 или 18	Сотрудник 16, 17 или 18	Сотрудник 19 или 20	Сотрудник 16, 17 или 18	Сотрудник 19 или 20
Q	Сотрудник 24 или 25	Сотрудник 22 или 23	Сотрудник 24 или 25	Сотрудник 22 или 23	Сотрудник 21
$\sum K$	8,1	8,4	8,1	8,3	8,4

Анализ распределения сотрудников по командам, полученный с применением жадного алгоритма, показывает, что в рассматриваемом в данной работе примере удалось добиться оптимального распределения сотрудников. Из 25 сотрудников создано 5 IT-команд, укомплектованных по 5 человек (два разработчика, два аналитика и один тестировщик). Значение коэффициента эффективности команд составило от 8,1 до 8,4 при стремлении целевой функции к среднему коэффициенту эффективности, равному 8,26. Дисперсия коэффициентов эффективности команд составила 0,0184, что можно считать приемлемым результатом.

Приведение значений опыта и грейда к единой системе коэффициентов эффективности позволило получить числа одного порядка в множестве {1,3; 1,5; 1,8; 2}, что является важным условием для работы жадного алгоритма.

Что касается выполнения условий равенства коэффициентов эффективности между командами по ролям (1.1–1.3), то их выполнение возможно в случае, если имеется возможность разбить сотрудников одной роли поровну между командами и при этом сотрудников одной роли в команде больше одного человека. Так, в рассмотренном примере в каждую из пяти команд распределены по два аналитика и по два разработчика – условия (1.1, 1.2) выполнены. Полученные результаты подтверждает серия экспериментов, представленных в табл. 6. Эксперименты проведены на наборе данных о показателях эффективности сотрудников, сгенерированных случайным образом для формирования 5 IT-команд из 25 сотрудников (10 разработчиков, 10 аналитиков, 5 тестировщиков).

В противном случае, когда число сотрудников одной роли в каждой команде равно одному или их нельзя разбить поровну между командами, сбалансировать команды по эффективности для данной роли не получится. Так, в рассмотренном примере в командах по одному тестировщику условие (1.3) не выполнено. В табл. 7 представлены результаты серии экспериментов, формирования 5 IT-команд из 25 сотрудников для случаев, когда нельзя разбить поровну между командами сотрудников с ролью «разработчик», а число тестировщиков в каждой команде равно одному (13 разработчиков, 10 аналитиков, 5 тестировщиков).

Несмотря на выявленные ограничения при достижении условий (1.1–1.3), полученное решение можно считать оптимальным, так как конечная цель равноценности команд достигнута. Фактические показатели коэффициентов эффективности по командам равны плановым показателям коэффициентов эффективности с дисперсией от 0 до 0,5 (см. табл. 6–7).

Анализ полученных результатов позволяет говорить о том, что к рассмотренной задаче оптимизации применим принцип жадного алгоритма, так как последовательность локально оптимальных выборов дает глобально оптимальное решение. Например, на шаге 3 при распределении сотрудников первой группы (с ролью разработчик) жадный выбор не закрывает путь к оптимальному решению: для любого решения есть другое, согласованное с жадным выбором и не хуже первого. Подзадача, возникшая после жадного выбора на шаге 3, аналогична исходной. По индукции следует, что такая последовательность жадных выборов дает оптимальное решение.

Все вышесказанное доказывает, что обозначенная гипотеза подтвердилась. Метод жадного алгоритма применим для решения класса задач, направленных на формирование IT-команд в условиях перехода на командное управление с целью получения оптимального распределения сотрудников на равноценные команды.

Известно, что в процессе формирования команд немаловажное значение играет учет личностных характеристик сотрудников. Поэтому в качестве ограничения к применению предложенного способа необходимо отметить, что предложенный способ формирования команд по функциональным ролям будет наиболее эффективен для IT-компаний с развитой системой IT-рекрутмента. В таких компаниях уже на этапах подбора и найма персонала учитываются личностные характеристики сотрудников с целью комфортного взаимодействия и исключения риска конфликтности, что является базой для успешного командообразования.

Таблица 6

Эксперименты формирования IT-команд из 25 сотрудников (10 разработчиков, 10 аналитиков, 5 тестировщиков)

№	Плановые показатели коэффициентов эффективности				Фактические показатели коэффициентов эффективности по командам																					D
	$\sum_{i=1}^{d,a,q} k_i$			$\bar{k}$	$\sum_{i=1}^{d,a,q} k_i$															$\bar{k}$						
					D					A					Q											
	D	A	Q		T1	T2	T3	T4	T5	T1	T2	T3	T4	T5	T1	T2	T3	T4	T5	T1	T2	T3	T4	T5	
1	3,2	3,3	1,76	8,26	3,3	3,3	3,1	3,3	3	3,3	3,3	3,3	3,3	3,3	1,5	2	1,5	2	1,8	8,1	8,4	8,1	8,3	8,4	0,0184
2	3,28	3,4	1,58	8,26	3,2	3,3	3,4	3,3	3,2	3,4	3,4	3,5	3,4	3,3	1,6	1,5	1,4	1,5	1,9	8,2	8,2	8,3	8,2	8,4	0,0064
3	3,26	3,46	1,58	8,3	3,4	3,2	3,3	3,2	3,2	3,5	3,4	3,5	3,4	3,5	1,4	1,9	1,4	1,6	1,6	8,3	8,5	8,2	8,2	8,3	0,0120
4	3,34	3,42	1,62	8,38	3,4	3,3	3,2	3,4	3,4	3,5	3,4	3,3	3,4	3,5	1,5	1,7	2	1,5	1,4	8,4	8,4	8,5	8,3	8,3	0,0056
5	3,34	3,5	1,62	8,46	3,3	3,4	3,3	3,4	3,3	3,5	3,5	3,4	3,6	3,5	1,9	1,5	1,7	1,4	1,6	8,7	8,4	8,4	8,4	8,4	0,0144

Таблица 7

Эксперименты формирования IT-команд из 25 сотрудников (13 разработчиков, 10 аналитиков, 5 тестировщиков)

№	Плановые показатели коэффициентов эффективности				Фактические показатели коэффициентов эффективности по командам																					D
	$\sum_{i=1}^{d,a,q} k_i$			$\bar{k}$	$\sum_{i=1}^{d,a,q} k_i$															$\bar{k}$						
					D					A					Q											
	D	A	Q		T1	T2	T3	T4	T5	T1	T2	T3	T4	T5	T1	T2	T3	T4	T5	T1	T2	T3	T4	T5	
1	4,18	3,46	1,66	9,3	4,7	4,6	4,8	3,4	3,4	3,4	3,4	3,4	3,6	3,4	1,6	1,6	1,3	1,8	2	9,7	9,6	9,5	8,8	8,9	0,1400
2	4,3	3,48	1,62	9,4	5	4,8	4,7	3,4	3,6	3,6	3,5	3,4	3,5	3,4	1,3	1,6	1,7	1,8	1,7	9,9	9,9	9,8	8,7	8,7	0,3280
3	4,5	3,24	1,68	9,42	5,1	5	5,1	3,7	3,6	3,3	3,2	3,3	3,2	3,2	1,5	1,7	1,5	1,7	2	9,9	9,9	9,9	8,6	8,5	0,4384
4	4,3	3,38	1,66	9,34	4,9	4,7	4,8	3,5	3,6	3,5	3,4	3,3	3,4	3,3	1,4	1,7	1,5	1,9	1,8	9,8	9,8	9,6	8,8	8,7	0,2384
5	4,16	3,26	1,72	9,14	4,8	4,8	4,5	3,3	3,4	3,2	3,2	3,3	3,3	3,3	1,7	1,5	1,7	1,9	1,8	9,7	9,5	9,5	8,5	8,5	0,2784

Практическая ценность применения метода жадного алгоритма к задаче формирования команд обусловлена потребностью IT-компаний в способах решения задач получения оптимально возможного распределения сотрудников в процессе Agile-трансформации. Рассмотренные методы и полученные алгоритмы командообразования могут быть применены к конкретному рабочему процессу в IT-компании.

Перспективы дальнейшего исследования проблемы распределения сотрудников на команды в условиях Agile-трансформации направлены на решение таких задач, как формирование команд из специалистов, обладающих навыками и компетенциями смежных функциональных ролей, так называемых T-shaped-специалистов. Для оценки результатов и выбора наилучшего решения будет построена компьютерная модель в среде имитационного моделирования.

Заключение

Распределение сотрудников на команды должно выполняться последовательно для каждой функциональной роли. Сначала по командам должны быть распределены сотрудники с одной ролью таким образом, чтобы значения коэффициентов эффективности сотрудников заданной роли были приблизительно равны между командами. Затем к команде с самым высоким суммарным показателем эффективности необходимо добавить сотрудников другой роли с наименьшей суммой коэффициентов эффективности и продолжать укомплектование команд до тех пор, пока не будут распределены все сотрудники данной роли. Принцип сортировки и последовательного распределения позволяет добиться сбалансированности между командами. Подобным образом в команды могут быть распределены не только рассмотренные роли разработчика, аналитика, тестировщика, но и любые другие функциональные роли, например, дизайнеры, архитекторы и т.д.

В статье описан алгоритм, основанный на принципах жадного выбора, для формирования IT-команд по функциональным ролям из имеющегося состава сотрудников в условиях Agile-трансформации в IT-компании. Выполнены постановка задачи и ее математическая формализация. Проведена серия экспериментов, подтверждающих пригодность применения метода жадного алгоритма для решения задач формирования команд в IT-компаниях.

Полученные результаты создают необходимую основу для практической реализации предложенного метода в системе поддержки принятия решений и совершенствования процесса командообразования в сфере IT.

Литература

1 *Вольфсон Б.И.* Гибкие методологии разработки. СПб.: Питер, 2017. 144 с.

2. *Галкина Т.П.* Социология управления: от группы к команде. М.: Финансы и статистика, 2004. 224 с.

3. *Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К.* Алгоритмы. Построение и анализ / под ред. И.В. Красикова. 2-е изд. М.: Вильямс, 2005. 1296 с.

4. *Сазерленд Д.* Scrum. Революционный метод управления проектами / пер. с англ. М. Гескиной. М.: Манн, Иванов и Фербер, 2017. 272 с.

5. *Селюк А.В., Денисова С.С.* Управление проектной командой. Тюмень: Изд-во Тюменского гос. ун-та, 2013. 216 с.

6. *Скиена С.* Алгоритмы. Руководство по разработке. 2-е изд. СПб.: БХВ-Петербург, 2011. 720 с.
7. *Стеллман Э., Грин Д.* Постигая Agile. Ценности, принципы, методологии / пер. с англ. М. Гескиной. М.: Манн, Иванов и Фербер, 2017. 448 с.
8. *Фоппель К.* Создание команды. Психологические игры и упражнения. М.: Генезис, 2003. 400 с.
9. *Хаджиева С.В.* Алгоритм формирования кросс-функциональных команд в IT-компании // Проблемы и перспективы внедрения инновационных телекоммуникационных технологий. Сборник материалов V Международной научно-практической очно-заочной конференции. Оренбург: Поволжский государственный университет телекоммуникаций и информатики, Оренбургский филиал, 2019. С. 255-260.
10. *Хаджиева С.В.* Разработка моделирующего алгоритма формирования команды IT-проекта // Информационные технологии в науке, бизнесе и образовании: сб. тр. X Междунар. науч.-практ. конф. студентов, аспирантов и молодых ученых. М.: Московский государственный лингвистический университет, 2018. С. 133–139.
11. *Чалдини Р.* Психология влияния. СПб.: Питер, 2001. 288 с.

Literatura

1. *Vol'fson B.I.* Gibkie metodologii razrabotki. SPb.: Piter, 2017. 144 s.
2. *Galkina T.P.* Sociologiya upravleniya: ot gruppy k komande. M.: Finansy i statistika, 2004. 224 s.
3. *Kormen T., Lejzerson Ch., Rivest R., Shtajn K.* Algoritmy. Postroenie i analiz / pod red. I.V. Krasikova. 2-e izd. M.: Vil'yams, 2005. 1296 s.
4. *Sazerlend D.* Scrum. Revolyucionnyj metod upravleniya proektami / per. s angl. M. Geskinoj. M.: Mann, Ivanov i Ferber, 2017. 272 s.
5. *Selyuk A.V., Denisova S.S.* Upravlenie proektnoj komandoj. Tyumen': Izd-vo Tyumenskogo gosudarstvennogo universiteta, 2013. 216 s.
6. *Skiena S.* Algoritmy. Rukovodstvo po razrabotke. 2-e izd. SPb.: BHV-Peterburg, 2011. 720 s.
7. *Stellman E., Grin D.* Postigaya Agile. Cennosti, principy, metodologii / per. s angl. M. Geskinoj. M.: Mann, Ivanov i Ferber, 2017. 448 s.
8. *Foppel' K.* Sozdanie komandy. Psihologicheskie igry i uprazhneniya. M.: Genezis, 2003. 400 s.
9. *Hadzhieva S.V.* Algoritm formirovaniya kross-funkcional'nyh komand v IT-kompanii // Problemy i perspektivy vnedreniya innovacionnyh telekommunikacionnyh tekhnologij. Sbornik materialov V Mezhdunarodnoj nauchno-prakticheskoj ochno-zaochnoj konferencii. Orenburg: Povolzhskij gosudarstvennyj universitet telekommunikacij i informatiki, Orenburgskij filial, 2019. S. 255-260.
10. *Hadzhieva S.V.* Razrabotka modeliruyushchego algoritma formirovaniya komandy IT-proekta // Informacionnye tekhnologii v nauke, biznese i obrazovanii: sb. tr. X Mezhdunar. nauch.-prakt. konf. studentov, aspirantov i molodyh uchenyh. M.: Moskovskij gosudarstvennyj lingvisticheskij universitet, 2018. S. 133–139.
11. *Chaldini R.* Psihologiya vliyaniya. SPb.: Piter, 2001. 288 s.