

УДК 004.891.2

Н.Г. Ярушкина, А.С. Желепов

## ПРОТОТИП СИСТЕМЫ ПОИСКА И ВЫБОРА «СФОРМИРОВАННЫХ» КОМАНД ИТ-СПЕЦИАЛИСТОВ НА ОСНОВЕ ДАННЫХ ПРОЕКТНЫХ РЕПОЗИТОРИЕВ

**Ярушкина Надежда Глебовна**, доктор технических наук, профессор, окончила Ульяновский политехнический институт. Исполняющая обязанности ректора Ульяновского государственного технического университета, заведующая кафедрой «Информационные системы» УлГТУ. Имеет более 300 работ в области мягких вычислений, нечеткой логики, гибридных систем. [e-mail: jng@ulstu.ru].

**Желепов Алексей Сергеевич**, аспирант кафедры «Информационные системы» УлГТУ, окончил факультет информационных систем и технологий УлГТУ. Победитель молодежного форума iVolga-2014, организатор ИТ-конференции «Стачка». Имеет работы в области интеллектуального анализа данных. [e-mail: a.zhelepov@gmail.com].

### Аннотация

Рост сферы информационных технологий приводит к серьезной нехватке квалифицированных кадров. Из-за этого многие компании меняют устоявшиеся модели рабочих процессов, предоставляя сотрудникам возможность удаленной работы, чтобы иметь возможность поиска новых сотрудников на глобальном рынке.

Для компаний, сосредоточенных на разработке продуктовых решений, наблюдается тренд поиска уже «сформированных» проектных команд, специалистов, которые в течение длительного времени успешно работали вместе. Однако поиск таких команд накладывает определенный отпечаток на деятельность HR-отдела компании: появляется необходимость анализировать деятельность, разработки не отдельных сотрудников, а потенциальной команды в целом.

Представленная статья описывает прототип автоматизированной системы, основной задачей которой является поиск и подбор команд специалистов на основе данных открытых репозиторий исходного кода и связанных артефактов. В статье подробно рассматриваются состав архитектуры системы, алгоритм выбора основной команды проекта, выявленные в ходе исследования метрики деятельности группы, формулы расчетов значений метрик, а также их применение при решении задачи анализа проектного репозитория.

Ключевые слова: проектный репозиторий, удаленная команда, метрики командной работы, поиск, фильтрация.

doi: 10.35752/1991-2927-2020-1-5-97-105

## A PROTOTYPE OF A SYSTEM FOR SEARCHING AND SELECTING THE 'STAFFED' TEAMS OF IT-PROFESSIONALS BASED ON DATA FROM PROJECT REPOSITORIES

**Nadezhda Glebovna Yarushkina**, Doctor of Sciences in Engineering, Professor; graduated from Ulyanovsk Polytechnic Institute; Interim Rector of Ulyanovsk State Technical University, Head of the Department of Information Systems at UISTU; an author of more than 300 scientific papers in the field of soft computing, fuzzy logic, and hybrid systems. e-mail: jng@ulstu.ru.

**Aleksei Sergeevich Zhelepov**, Postgraduate Student at the Department of Information Systems of UISTU; graduated from the Faculty of Information Systems and Technologies of UISTU; a winner of the iVolga-2014 Youth Forum, a coordinator of the "Stachka" IT-Conference, an author of articles in the field of data mining. e-mail: a.zhelepov@gmail.com.

### Abstract

The growth of information technology sphere leads to a serious shortage of qualified personnel. Thus, many companies are changing well-established workflow models, e.g. giving employees the ability to work remotely. That happens due to the opportunity of hiring globally.

For companies focused on the development of product solutions, there is a trend of searching for "played" project teams, specialists who have successfully worked together for a long time. However, the search for such teams leaves a definite

imprint on the activities of the HR department of the company: it becomes necessary to analyze the activities and technical developments of not individual employees, but of the potential team as a whole.

The article describes a prototype of an automated system which main task is to search and select teams of specialists based on data from open source code repositories and related artifacts. The article details: the composition of the system architecture, the selection algorithm for the main project team, identified during the study of the group activity metrics, formulas for calculating the values of the metrics, as well as their application in solving the problem of analyzing the project repository.

Key words: repository, remote development team, teamwork metrics, search, filtering.

## ВВЕДЕНИЕ

На фоне глобального роста сфера информационных технологий в России испытывает недостаток квалифицированных кадров [1]. Исследование сообщества компаний-разработчиков программного обеспечения «РУС-СОФТ» показало, что проблема особенно актуальна для региональных ИТ-компаний [2].

При выборе нового места работы высококвалифицированный специалист оперирует следующими факторами: HR-бренд компании, условия труда, взаимодействие с руководителями, личный интерес участия в проектах организации. В результате наблюдается серьезный отток специалистов в столичные регионы, где располагается большинство крупных компаний-игроков отечественного ИТ-рынка.

Из-за нехватки квалифицированных кадров компании меняют модель работы, предоставляя сотрудникам возможность трудиться удаленно, не находясь в офисе. Переход обуславливается возможностью найма сотрудников компанией не только на локальном, но и на глобальном кадровом рынке.

Для продуктовых компаний наблюдается тенденция поиска не только отдельных специалистов, но уже «сформированных» команд разработчиков [3, 4, 5]. Понятие «сформированная» подразумевает, что группа специалистов имеет опыт совместной работы над проектами, соответственно процессы разработки уже выстроены, а взаимоотношения между членами команды налажены. Поиск «готовых» команд компаниями обуславливается особенностями современной разработки: необходимостью быстрой проверки гипотез и прототипирования, разработкой MVP и т. д.

Компания Skyeng, являющаяся одним из лидеров продуктовой разработки в России, провела собственное исследование, в ходе которого провела сравнительный анализ стандартной офисной модели работы сотрудников компании и удаленной. В данном виде организации труда HR-специалисты и технический руководитель компании выделили следующие преимущества:

- HR-отдел имеет возможность осуществлять поиск новых сотрудников независимо от их географического расположения;
- удаленные сотрудники более ответственно подходят к организации своего рабочего времени. Благодаря

этому повышается общая эффективность труда над проектом;

- удерживать высококвалифицированного сотрудника проще, так как большинство из них работают из регионов, при этом получая все преимущества работы в столичной компании.

Слаженная командная работа – залог успеха программного обеспечения как продукта. В результате на мировом ИТ-рынке возник феномен групп разработчиков, специализирующихся на разработке стартапов, предлагающих свои услуги по сопровождению уже состоявшихся продуктов. Например, известная группа разработчиков программного обеспечения Evil Martians.

Создание системы поиска проектных команд актуально в части автоматизации HR-процессов организаций, специализирующихся на продуктовой разработке. Процедура поиска команды отличается от поиска конкретного специалиста – требуется анализ разработок и портфолио команды для определения уровня ее компетентности. Данный вид анализа реализуется системой в виде расчета специализированных метрик и построения отчета деятельности команды.

Статья является отправной точкой диссертационного исследования. В ней описываются источники данных для организации глобального поиска проектных команд, базовые компоненты архитектуры разрабатываемой системы, выявленные в ходе исследования метрики, характерные для «сформированной» проектной команды, результаты проведенных вычислительных экспериментов для алгоритма фильтрации членов проектной команды.

Практическая ценность исследования заключается в применении представленной методики в работе руководителей ИТ-компаний и кадровых рекрутеров для поиска и подбора не только отдельных разработчиков, но и целых проектных команд.

## АРХИТЕКТУРА СИСТЕМЫ ПОИСКА ПРОЕКТНЫХ КОМАНД

Основным источником данных для тестирования прототипа системы поиска проектных команд и разрабатываемых алгоритмов был выбран сервис GitHub – «социальная сеть» разработчиков программного обеспечения, в которой зарегистрировано более 37 миллионов ИТ-специалистов и размещено более 100 миллионов проектных репозиториях.

На рисунке 1 представлена архитектура системы поиска проектных команд.

Представленная система состоит из 4 модулей:

- модуль формирования запроса  $Q(q_1, q_2, \dots, q_n)$  обеспечивает обмен данными между сервисом GitHub и системой поиска проектных команд;  $q_1, q_2, \dots, q_n$  – параметры запроса;
- модуль поиска основной команды проекта. GitHub-репозитории чаще всего специфичны из-за того, что находятся в открытом доступе. Это дает возможность участия в проекте сообществу разработчиков, не входящих в основную команду проекта. Функция данного модуля – фильтрация и выборка членов основной команды;
- модуль BI-аналитики позволяет произвести расчеты метрик командной слаженности, предлагая пользователю системы возможность наглядно оценить основную команду разработчиков проекта. В последующих статьях, посвященных исследованию, будут представ-

лены метрики и их связь с работой команды над проектами. В данной редакции BI-модуль упоминается лишь как необходимая часть системы для оценки деятельности команды в совокупности с системой управления проектом (в случае расчета характеристик для существующей команды) или выделенной среди множества участников открытого проекта основной группы разработчиков;

- модуль формирования рекомендаций предлагает оценку команды, рассчитывает индекс эффективности ее работы, рекомендации по улучшению проектной деятельности.

На момент написания статьи реализованы первые три модуля системы из представленного списка. Планы по созданию и развитию рекомендательного модуля представлены в заключительной части статьи.

Модуль формирования запроса  $Q(q_1, q_2, \dots, q_n)$  имеет ряд входных параметров для поиска на GitHub:

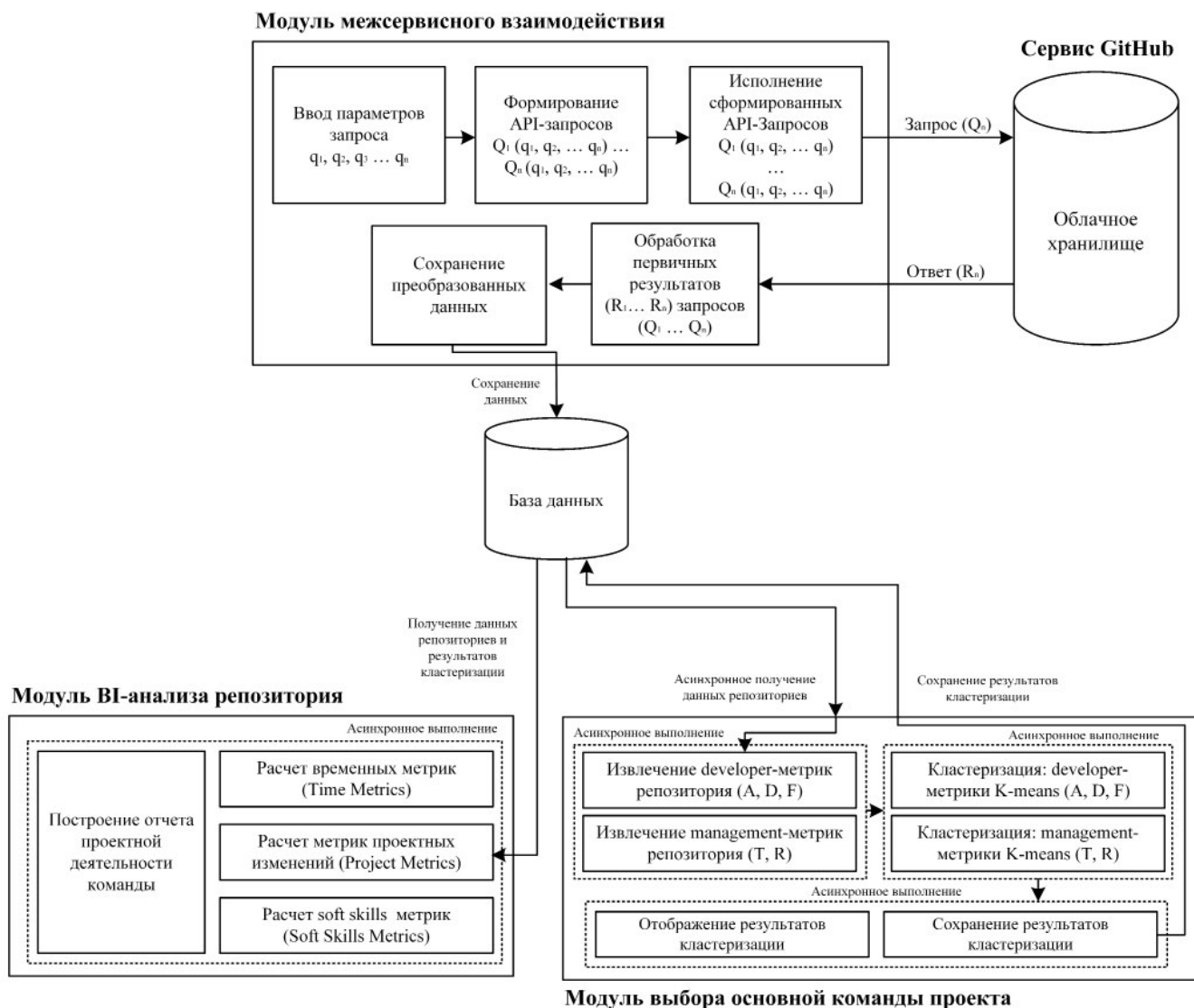


Рис. 1. Архитектура системы поиска проектных команд

- $q_1$  – технологии, используемые при разработке проекта. Данный параметр позволяет сегментировать репозитории по технологическим «стекам», что в первую очередь важно для HR-специалиста при подборе потенциальных кандидатов «под проект». Параметр описывается множеством строковых значений, обозначающих технологии программирования и администрирования:  $q_1 \subseteq \{'.NET', 'Ruby on Rails', 'Kubernetes' \dots 'n'\}$ ;

- $q_2$  – «индекс копирования» репозитория другими разработчиками / проектными командами. Параметр определяет степень использования проекта другими разработчиками. Диапазон возможных значений:  $q_2 \in [0; \infty]$ ;

- $q_3$  – «индекс популярности», социальная метрика репозитория: в системе GitHub разработчик имеет возможность оставить отзыв о проекте. Значение  $q_3$  – это множество  $[0; \infty]$ , которое отражает число положительных отзывов сторонних разработчиков;

- $q_4$  – наименование проектного репозитория. Строковая переменная;

- $q_5$  – наименование команды разработчиков. Строковая переменная;

- $q_6$  – число участников проекта, учитываются и не постоянные члены команды, где  $q_6 \in [0; \infty]$ ;

В ходе дальнейшего исследования планируется расширить возможности поиска, добавив ряд дополнительных параметров. Поиск реализован на языке программирования Python, взаимодействие с GitHub и получение данных построено на основе открытого API-сервиса.

### МОДЕЛИ ПРОЕКТНОЙ КОМАНДЫ И РЕПОЗИТОРИЯ ПРОЕКТА

Главной особенностью проектной команды, которая занимается разработкой проекта в виде конечного репозитория, является четкое разделение ролей [6]:

- Аналитики ( $A$ ) – члены команды, проводящие анализ потребностей пользователей и участников сообщества других разработчиков для понимания вектора дальнейшего развития продукта;

- Разработчики ( $D$ ) – члены команды, непосредственно выполняющие проектные задачи;

- Специалисты по качеству ( $Q$ ) – члены команды, отвечающие за правильность функциональной части продукта;

- Лидеры команды ( $T$ ) – члены команды, руководящие разработкой продукта, распределяющие задачи между участниками проектной команды.

Допускается, что один член команды может выполнять несколько ролей. Таким образом, модель проектной команды может быть представлена в виде множества:

$$M = \{A, D, Q, T\}.$$

Каждая из представленных ролей определенным образом взаимодействует с проектным репозиторием.

Роль «аналитика» проекта, представленного в виде открытого репозитория, часто выполняет сообщество заинтересованных в его разработке участников GitHub (или другой платформы).

Их взаимодействие с основной командой проекта отслеживается с помощью характеристик репозитория [7]:

- Задачи ( $T$ ), создаваемые не членами основной команды [8];

- Проектные изменения ( $PR$ ), предлагаемые участниками сообщества для исправления ошибок, направленные на улучшения проекта;

- Социальная активность участников, выраженная в комментариях к задачам ( $C$ ), признании значимости проекта в виде поставленных «звезд» ( $S$ ), а также копиях репозитория ( $F$ ).

Роль «разработчика» разделяется между основной командой и сообществом заинтересованных участников. Характеристиками репозитория, которые отражают вклад участников данной роли, являются:

- Коммиты ( $Cmt$ ) – вклад в проект в виде строк программного кода, описания функций, модулей проекта и т. д.;

- Ветки проекта ( $B$ ), отражающие активность выполнения поставленных проектных задач. Ветки создаются с целью упростить синхронизацию взаимодействия разработчиков проекта.

Роль «специалиста по качеству» также частично отводится сообществу заинтересованных IT-специалистов. Характеристиками [9], отражающими прогресс работы данной категории специалистов, являются:

- Задачи ( $T$ ), создаваемые с целью исправления ошибок в проекте;

- Проектные изменения ( $PR$ ), перед добавлением в основную ветку проектного репозитория, выполненная задача должна быть протестирована на наличие ошибок и соответствие требованиям задачи.

Роль «лидера» выполняется только членами основной команды проекта, заинтересованными в его развитии. Отличительными метриками являются:

- Выпускаемые версии продукта – релизы ( $R$ ). При создании релиза лидер команды объединяет множество веток других разработчиков, созданных ими при решении отдельных проектных задач;

- Распределенные задачи ( $T$ ) между участниками проекта.

Таким образом, модель проектного репозитория выражается через множество отношений проектных ролей и его артефактов:

$$Rep = \{A \rightarrow \{T, PR, C, S, F\}\};$$

$$D \rightarrow \{Cmt, B\}; Q \rightarrow \{T, PR\}; L \rightarrow \{R, T\}.$$

Представленная модель определяет связь взаимодействия отдельных проектных артефактов и ролей проектной команды. Основываясь на представленном описании модели, разработан и далее представлен алгоритм выделения основной проектной команды.

### РЕАЛИЗАЦИЯ АЛГОРИТМА ФИЛЬТРАЦИИ (ПОИСКА) ПРОЕКТНОЙ КОМАНДЫ

Алгоритм фильтрации предназначен для выбора основной группы разработчиков проектной команды на

основе данных каждого из найденных проектных репозиторий. В основе работы алгоритма применяется метод кластеризации *k*-means [10], который позволяет задать количество кластеров.

Опираясь на материал статьи [11], описывающей исследования групп разработчиков программного обеспечения, предполагалось применить анализ с учетом трех кластеров: J (Junior), M (Middle), S (Senior), обозначающих соответствующие уровни квалификации специалистов.

Однако данные виды кластеров не могут быть выбраны при исследовании команды на основе количественных данных проектного репозитория. Например, senior-специалисты не всегда вносят в проект много изменений, а больше координируют действия команды, собирают стабильные версии программного обеспечения (релизы). Поэтому анализ на основе количественных метрик при таком выборе кластеров может дать серьезное отклонение.

С учетом описанных выше особенностей были выбраны следующие кластеры:

- контрибьютор (К). Участник, вносящий относительно небольшой вклад в проект;
- участник (У). Специалист, который периодически принимает участие в улучшении проекта;
- предполагаемый член команды проекта (ПЧКП). Разработчик, активно развивающий проект;
- основной разработчик (ОР). Специалист, вносящий наибольший вклад в развитие проекта.

Процедура кластеризации проходит в два этапа: кластеризация на основе developer-метрик и management-метрик. Разделение позволяет учесть то, что опытные специалисты могут быть больше менеджерами проекта, чем непосредственно исполнителями.

В качестве developer-характеристик были выбраны:

- Additions and Deletions (A, D). Метрика описывает количество строк кода, которые были добавлены и удалены каждым из разработчиков проекта;
- Changed Files (CF). Количество файлов, измененных разработчиком за время работы над проектом.

В качестве management-метрик были выбраны:

- Task Count (TC). Количество созданных задач, метрика характеризует умение правильно понимать и определять вектор развития проекта;
- Release Count (RC). Количество выпущенных релизов разработчиком. Характеристика описывает умение «сводить» все предложенные другими разработчиками изменения, быть ответственным за очередную версию продукта с точки зрения его корректной работы.

Алгоритм выделения команды из множества разработчиков, работающих над проектом, подразумевает проведение следующих шагов:

- Предобработка и подготовка входных данных;
- Проведение процедур кластеризации участников проекта;
- Определение принадлежности каждого из участников к определенному кластеру и визуальное представление результата.

Перед проведением кластеризации участников производится предварительная обработка множества проектных данных: группировка разработчиков по количеству произведенных коммитов в репозиторий, внесенных изменений в файлы, созданных релизов и проектных задач.

Преобразованные данные разбиваются на три множества: численных значений developer- и management-характеристик и подписей. Структуры данных, которые описывают элементы множеств, представлены в листинге 1.

```
struct label {
    user_id: string
}

struct management {
    task_count: int
    release_count: int
}

struct development {
    additions_count: int
    deletions_count: int
    changed_files_count: int
}
```

Листинг 1. Структуры входных данных

Алгоритм кластеризации по методу *k*-средних предполагает заранее известное количество кластеров, в случае решаемой задачи – это количество равно 4, по числу представленных выше кластеров: «К», «У», «ПЧКП» и «ОР». Другим параметром алгоритма кластеризации является максимальное количество итераций – берется значение, равное 10 000 (получено опытным путем – при проведении вычислительных экспериментов, начиная примерно с 1000 итерации, центры кластеров изменяются незначительно).

Выходные данные алгоритма кластеризации представлены в виде множества соответствий «label-<название кластера>» или «label-<название кластера>». Благодаря чему возможно построение графического отображения результатов кластеризации (примеры представлены в части с вычислительными экспериментами).

### ВЫЧИСЛИТЕЛЬНЫЙ ЭКСПЕРИМЕНТ

Для тестирования алгоритма фильтрации и выделения основных участников команды были выбраны 10 проектных репозиторий, среди которых такие известные проекты, как:

- ClickHouse. Колоночная система управления базами данных (СУБД), разрабатываемая компанией Яндекс. Проект насчитывает в своей истории более 30 000 коммитов, 350 участников на GitHub, в том числе

и основную команду проекта. Особенность: основная команда проекта не распределена.

- Yii 2 Framework. Популярный среди веб-программистов PHP-фреймворк. Количество коммитов: около 20 000, число участников: 950. Особенность: основная команда проекта распределена по всему миру.

- Alumentations и Catalyst. Современные фреймворки, которыми пользуются инженеры по машинному обучению. Проекты появились относительно недавно. Alumentations активно разрабатывается при участии компании X5 – основная команда не распределена, Catalyst – инициативный проект нескольких разработчиков.

- 6 проектов международной группы разработчиков Evil Martians: PostCss, BrowsersList, AutoPrefixer, Nanold, Gon, ImgProху. Проекты широко применяются разработчиками программного обеспечения при создании веб-приложений.

Такой выбор репозитория обусловлен тем, что разработчики репозитория принимали участие в международной ИТ-конференции «Стачка», соорганизатором которой является соавтор данной статьи. Благодаря этому стало возможным практически проверить работу алгоритма на предмет совпадения участников проектной команды, выделенных алгоритмом, и реальных ее участников.

На основе developer- и management-метрик каждого из выбранных репозитория был проведен кластерный анализ участников проектов, результаты которого представлены в таблице 1.

Вывод на основе результатов, представленных в таблице 1: распределение участников проекта по кластерам в результате анализа management-метрик значительно меньше. Это объясняется тем, что лишь небольшое количество специалистов отвечает

Таблица 1

Результаты кластерного анализа участников выбранных проектных репозитория

№	Наименование репозитория	Число участников	Распределение участников в результате анализа на основе developer-метрик	Распределение участников в результате анализа на основе management-метрик
1	Alumentations	46	К: 26 У: 16 ПЧКП: 2 ОР: 2	К: 1 У: 1 ПЧКП: 2 ОР: 0
2	PostCss	288	К: 208 У: 77 ПЧКП: 2 ОР: 1	К: 1 У: 0 ПЧКП: 0 ОР: 0
3	Yii2	983	К: 875 У: 101 ПЧКП: 4 ОР: 2	К: 1 У: 1 ПЧКП: 1 ОР: 1
4	ClickHouse	350	К: 311 У: 32 ПЧКП: 6 ОР: 1	К: 14 У: 8 ПЧКП: 1 ОР: 0
5	Catalyst	36	К: 27 У: 7 ПЧКП: 1 ОР: 1	К: 1 У: 1 ПЧКП: 0 ОР: 0
6	BrowsersList	100	К: 89 У: 8 ПЧКП: 2 ОР: 1	К: 1 У: 1 ПЧКП: 0 ОР: 0
7	AutoPrefixer	155	К: 140 У: 11 ПЧКП: 3 ОР: 1	К: 1 У: 1 ПЧКП: 0 ОР: 0
8	Nanold	55	К: 41 У: 13 ПЧКП: 1 ОР: 0	К: 1 У: 0 ПЧКП: 0 ОР: 0

Продолжение табл. 1

№	Наименование репозитория	Число участников	Распределение участников в результате анализа на основе developer-метрик	Распределение участников в результате анализа на основе management-метрик
9	Gon	59	К: 52 У: 4 ПЧКП: 2 ОР: 1	К: 1 У: 0 ПЧКП: 0 ОР: 0
10	Imgproxу	31	К: 22 У: 7 ПЧКП: 1 ОР: 1	К: 1 У: 0 ПЧКП: 0 ОР: 0

за составление задач и подготовку стабильных версий программного обеспечения, в случае GitHub – это наиболее заинтересованные в развитии проекта разработчики или члены основной команды.

На рисунке 2 представлен пример иллюстрации кластерного анализа на основе developer- и management-метрик для проекта ClickHouse.

Практическая значимость работы алгоритма подтверждена участниками проектных команд ClickHouse, Yii 2, Catalyst, Albuementations и проектов Evil Martians. Результаты пересечения множеств участников ( $P_1$ ), выделенных алгоритмом, и множества действительных участников ( $P_2$ ), определенных экспертами, приведены в таблицах 2, 3.

Разделение проектов по таблицам 2 и 3 неслучайно: таблица 2 содержит проекты, в которые большой вклад вносит несколько разработчиков ввиду их сложности и широкого использования, таблица 3 – программные библиотеки, которые скорее упрощают процесс разработки и не являются полноценными фреймворками или СУБД. В дальнейшем для учета социальной составляющей работы над проектом планируется добавить параметр: soft skills. Его численная метрика будет выражена в виде количества сообщений и связанных подсообщений для каждого из участников проекта.

На основании экспертной оценки авторов проектов некоторые из действительных членов проектной команды не попали в кластеры «ПЧКП» и «ОР» ввиду того, что

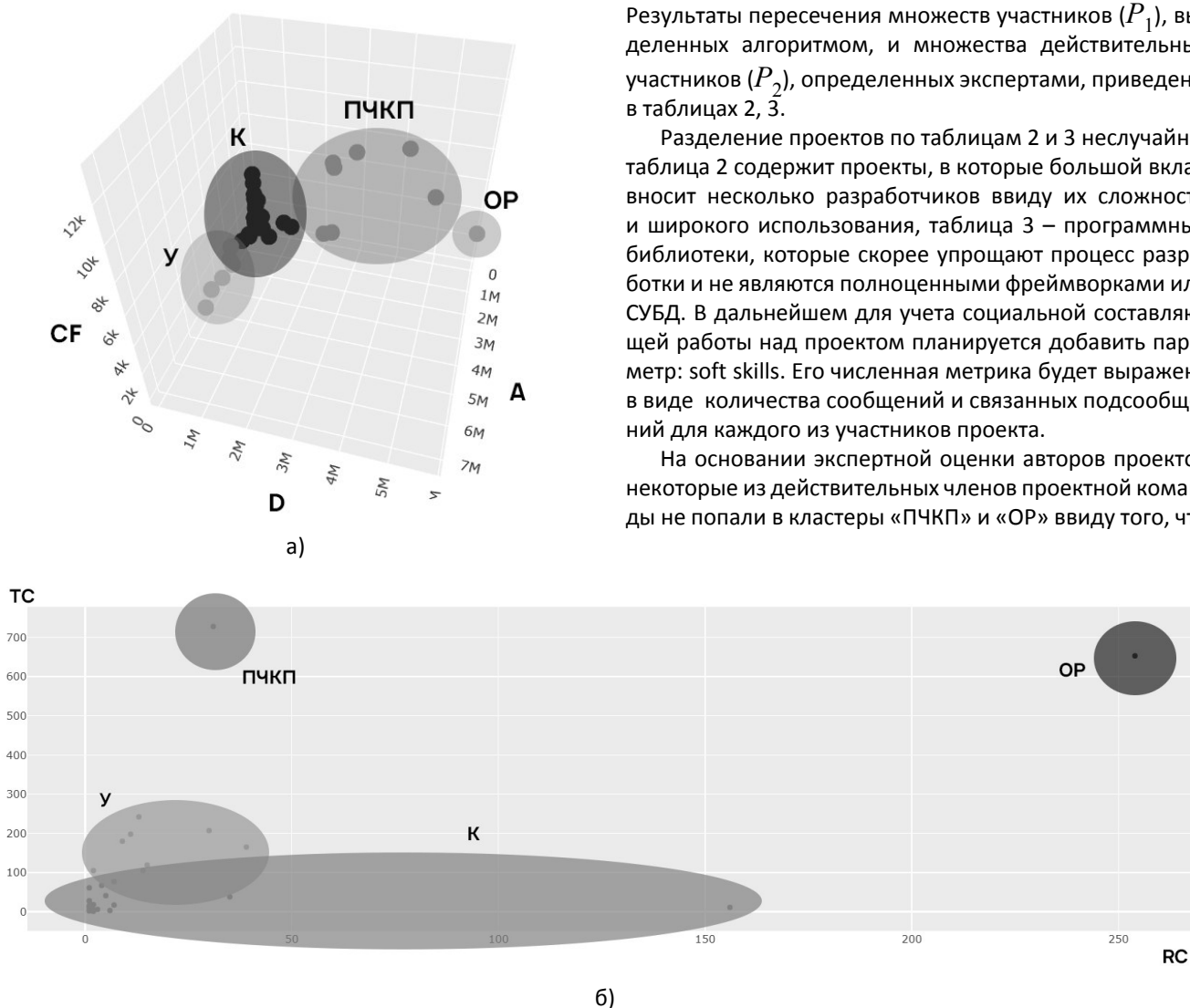


Рис. 2. Результаты кластерного анализа участников проекта ClickHouse: а) developer-метрики; б) management-метрики

Таблица 2

Сравнение множеств  $P_1$  и  $P_2$  участников проектов ClickHouse, Yii 2, Catalyst, Alumentation

	ClickHouse	Yii 2	Catalyst	Alumentations
Множество участников $P_1$ (учитываются только кластеры «ПЧКП» и «ОР»)	BayoNet, Ivan Blinkov, Nikolai Kochetov, Vitaliy Zakaznikov, alexey-milovidov, chenxing-x, proller	samdark, NabiKAZ, SilverFire, 十巡洋艦, Carsten Brandt, Qiangxue	Tezikov Roman, Sergey Kolesnikov	Alexander Buslaev, Eugene Khvedchenya, Alex Parinov, Vladimir Igllovikov
Множество участников $P_2$ (экспертная оценка)	alesapin, bayoNet, blinkov, 4ertus2, KochetovNicolai den-crane, proller, alexey-milovidov	Qiangxue, samdark, SilverFire, cebe	Tezikov Roman, Sergey Kolesnikov	Alexander Buslaev, Eugene Khvedchenya, Alex Parinov, Vladimir Igllovikov
Результат пересечения множеств $P_1 \cap P_2$	5 из 8	3 из 4	2 из 2	4 из 4

Таблица 3

Сравнение множеств  $P_1$  и  $P_2$  участников проектов PostCss, BrowsersList, AutoPrefixer, Nanold, Gon, Imgproxy

	PostCss	BrowsersList	AutoPrefixer	Nanold	Gon	Imgproxy
Множество участников $P_1$ (учитываются только кластеры «ПЧКП» и «ОР»)	ai, ben-eb, jedmao	ai, AleshaOleg, akx, An-Tu	ai, bogdan0083, yepninja, Semigradsky	ai	gazay, torbjon, john-bai	DarthSim, koenpunt
Множество участников $P_2$ (экспертная оценка)	ai	ai, akx	ai, Semigradsky	ai	gazay	DarthSim
Результат пересечения множеств $P_1 \cap P_2$	1 из 3	2 из 4	2 из 4	1 из 1	1 из 3	1 из 2

вносят недостаточное количество изменений в проект. Это связано с рядом причин: участие в других проектах компании, выполнение функций поддержки продукта.

Таким образом, алгоритм планируется улучшить, добавив дополнительный параметр для кластерного анализа: количество закрытых задач участником проекта.

### ЗАКЛЮЧЕНИЕ

В данной работе были представлены: архитектура системы поиска проектных команд; модель проектной команды и взаимодействие ее отдельных ролей с артефактами проектного репозитория, алгоритм выбора участников основной проектной команды из множества специалистов, участвующих в процессе разработки открытого проекта.

Практическая значимость системы заключается в том, что она позволяет частично автоматизировать деятельность отдела компании, отвечающего за поиск специалистов и проектных команд. Формируя отчеты деятельности проектной команды на основе представ-

ленных метрик, HR-специалист сможет производить построение рейтинга проектных команд по интересующим технологиям для их последующего сравнения и принятия итогового решения о найме специалистов выделенной основной команды.

Дальнейшее развитие исследования будет связано с разработкой рекомендательного модуля системы, который планируется построить на базе парадигмы нечеткой логики [12]. Для получения действительно полезных для улучшения процессов разработки оценок и рекомендаций продолжится взаимодействие с авторами проектов, данные репозитория которых использовались для тестирования алгоритма фильтрации и поиска основной проектной команды.

### СПИСОК ЛИТЕРАТУРЫ

1. Исследование ИТ-кластера Ульяновской области / Н.Г. Ярушкина, Т.В. Афанасьева, О.В. Шиняева [и др.] ; отв. ред. Т.В. Афанасьева. Ульяновск : УлГТУ, 2013. 137 с.
2. Исследование РУССОФТ. Российской ИТ-инду-



стрии предвещают острый кадровый голод. URL: <https://www.it-world.ru/it-news/it/140881.html> (дата обращения: 22.11.2019).

3. Катаев. А. Управление большой и распределенной командой (выступление на конференции Teamlead Conf 2018). URL: <http://teamleadconf.ru/2018/abstracts/3205> (дата обращения: 22.11.2019).

4. An Approach to Similar Software Projects Searching and Architecture Analysis Based on Artificial Intelligence Methods / N. Yarushkina, G. Guskov, P. Dudarin, V. Stuchebnikov // Proceedings of the Third International Scientific Conference Intelligent Information Technologies for Industry (IITI'18). Advances in Intelligent Systems and Computing. Springer, Cham, 2018. Vol. 1. pp. 341–352.

5. Evil Martians. URL: <https://evilmartians.com/> (дата обращения: 22.11.2019).

6. Comparison of Software Complexity Metrics / Ali Athar Khan, Amjad Mahmood, Sajeda M. Amralla and Tahera H. Mirza // International Journal of Computing and Network Technology. 2016. Vol. 4, iss. 1. pp. 19–25.

7. Using Metrics to Evaluate Software System Maintainability / Coleman Don, Ash Dan, Lowther Bruce, Oman Paul // Computer. IEEE Computer Society Press, Los Alamitos, CA, USA. 1994. Vol. 27, iss. 8. pp. 44–49.

8. Подход к оценке трудоемкости задач в процессе разработки программного обеспечения на основе нейронных сетей / П.В. Дударин, В.Г. Тронин, К.В. Святов, В.А. Белов, Р.А. Шакуров // Автоматизация процессов управления. 2019. № 3 (57). С. 65–72.

9. Static and Dynamic Complexity Analysis of Software Metrics / Kamaljit Kaur, Kirti Minhas, Neha Mehan, Namita Kakkar // World Academy of Science, Engineering and Technology International Journal of Computer, Electrical, Automation, Control and Information Engineering. 2009. Vol. 3, no 8. pp. 1936–1938.

10. Youguo LiHaiyan Wu. A Clustering Method Based on K-Means Algorithm // International Conference on Solid Devices and Materials Science, 2012.

11. Afanasyeva T., Zheleпов A., Zagaichuk I. Framework for Accessing Professional Growth of Software Developers // ICCTA 2019, 5th International Conference on Computer and Technology Applications, 2019.

12. Ярушкина Н.Г. Методы нечетких экспертных систем в интеллектуальных САПР. Саратов, 1997. 106 с.

#### REFERENCES

1. Yarushkina N.G., Afanaseva T.V., Shiniaeva O.V. et al. *Issledovanie IT-klastera Ulyanovskoi oblasti. Otv. red. T.V. Afanaseva*. [Research of IT-Cluster of Ulyanovsk District. Edited by T. V. Afanaseva]. Ulyanovsk, UISTU Publ., 2013. 137 p.

2. *Issledovanie RUSSOFT. Rossiiskoi IT-industrii*

*predrekaiut ostryi kadrovyi golod* [RUSSOFT Research. Russian IT Industry is Going to Face Limit of Qualified Software Engineers]. Available at: <https://www.it-world.ru/it-news/it/140881.html> (accessed 22.11.2019).

3. Kataev A. *Upravlenie bolshoi i raspredelennoi komandoi (vystuplenie na konferentsii Teamlead Conf. 2018)* [Remote and Distributed Team Management (Report on Teamlead Conf. 2018)]. Available at: <http://teamleadconf.ru/2018/abstracts/3205> (accessed 22.11.2019).

4. Yarushkina N., Guskov G., Dudarin P., Stuchebnikov V. An Approach to Similar Software Projects Searching and Architecture Analysis Based on Artificial Intelligence Methods. *Proceedings of the Third International Scientific Conference Intelligent Information Technologies for Industry (IITI'18). Advances in Intelligent Systems and Computing*. Springer, Cham, 2018, vol. 1, pp. 341–352.

5. *Evil Martians*. Available at: <https://evilmartians.com/> (accessed 22.11.2019).

6. Ali Athar Khan, Amjad Mahmood, Sajeda M. Amralla and Tahera H. Mirza Comparison of Software Complexity Metrics. *International Journal of Computing and Network Technology*, 2016, vol. 4, iss. 1, pp. 19–25.

7. Coleman Don, Ash Dan, Lowther Bruce, Oman Paul Using Metrics to Evaluate Software System Maintainability. *Computer. IEEE Computer Society Press*, Los Alamitos, CA, USA, 1994, vol. 27, iss. 8, pp. 44–49.

8. Dudarin P.V., Tronin V.G., Sviatov K.V., Belov V.A., Shakurov R.A. Podkhod k otsenke trudoemkosti zadach v protsesse razrabotki programmnoo obespecheniia na osnove neironnykh setei [An Approach to Labor Intensity Evaluation in Software Development Process Based on Neural Networks]. *Avtomatizatsiia protsessov upravleniia* [Automation of Control Processes], 2019, no. 3 (57), pp. 65–72.

9. Kamaljit Kaur, Kirti Minhas, Neha Mehan, Namita Kakkar. Static and Dynamic Complexity Analysis of Software Metrics. *World Academy of Science, Engineering and Technology International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 2009, vol. 3, no 8, pp. 1936–1938.

10. Youguo LiHaiyan Wu. A Clustering Method Based on K-Means Algorithm. *International Conference on Solid Devices and Materials Science*. 2012.

11. Afanaseva T., Zheleпов A., Zagaichuk I. Framework for Accessing Professional Growth of Software Developers. *ICCTA 2019, 5th International Conference on Computer and Technology Applications*. 2019.

12. Yarushkina N.G. *Metody nechetkikh ekspertnykh sistem v intellektualnykh SAPR* [Methods for Fuzzy Expert Systems in Intellectual CAD-Systems]. Saratov, 1997. 106 p.