

Научная статья
УДК 004.021
<https://doi.org/10.24143/2072-9502-2025-3-102-111>
EDN DWCLNW

Метод классификации средств разработки программного обеспечения на основе данных опроса Stack Overflow

Алла Григорьевна Кравец[✉], Иван Витальевич Казьмин, Артём Геннадьевич Бондаренко

*Волгоградский государственный технический университет,
Волгоград, Россия, allagkravets@yandex.ru[✉]*

Аннотация. Исследование посвящено разработке и апробации нового подхода к классификации средств разработки программного обеспечения (ПО), основанного на анализе набора данных (датасета) опроса разработчиков Stack Overflow 2024 г., с целью решения актуальной проблемы выбора оптимального технологического стека в условиях разнообразия технологий. Целью исследования является выявление востребованных инструментов разработки ПО, их классификация по областям применения и актуальности для формирования базы знаний, которая может быть использована при создании рекомендательных систем. В качестве примера для демонстрации метода были использованы языки программирования (ЯП). Процесс включает подготовку данных из датасета Stack Overflow, а именно отбор признаков, таких как тип разработчика и используемые им инструменты; очистку, обработку пропусков, сокращение типов разработчиков и применение количественного метода классификации. Разработанный метод основан на расчете и нормализации частоты использования каждого языка в разрезе различных типов разработчиков для устранения дисбаланса в выборке. Для каждого ЯП вычисляются два ключевых показателя: максимальное нормализованное значение частоты использования (C_{\max}) и коэффициент вариации (CV), отражающий равномерность его использования. Классификация на 3 категории – «общего назначения», «отраслевые» и «нишевые» – производится путем сравнения индивидуальных показателей C_{\max} и CV языка с их медианными значениями по всему набору ЯП. Результаты, представленные в виде гистограммы, наглядно демонстрируют разделение языков: общего назначения (например, Python, JavaScript), отраслевые, востребованные в конкретных сферах (например, Swift, Kotlin, R), и нишевые (например, Crystal и Delphi). Предложенный метод формирует структурированное, основанное на данных представление о технологическом ландшафте, что может быть полезно разработчикам при выборе инструментов и закладывает основу для создания рекомендательных систем.

Ключевые слова: технологический стек, подбор датасета, программное обеспечение, анализ данных, языки программирования, коэффициент вариации

Для цитирования: Кравец А. Г., Казьмин И. В., Бондаренко А. Г. Метод классификации средств разработки программного обеспечения на основе данных опроса Stack Overflow // Вестник Астраханского государственного технического университета. Серия: Управление, вычислительная техника и информатика. 2025. № 3. С. 102–111. <https://doi.org/10.24143/2072-9502-2025-3-102-111>. EDN DWCLNW.

Original article

Method for classifying software development tools based on Stack Overflow survey data

Alla G. Kravets[✉], Ivan V. Kazmin, Artem G. Bondarenko

*Volgograd State Technical University,
Volgograd, Russia, allagkravets@yandex.ru[✉]*

Abstract. This study is devoted to the development and testing of a new approach to the classification of software development tools (ST), based on the analysis of the dataset from the Stack Overflow 2024 developer survey, in order to solve the urgent problem of choosing the optimal technology stack in the context of a variety of technologies. The purpose of the research is to identify the required software development tools, classify them by application areas and relevance to form a knowledge base that can be used to create recommendation systems. Programming languages (PL) were used as an example to demonstrate the method. The process includes the preparation of data from the Stack Overflow dataset, namely, the selection of features such as the type of developer and the tools used by him; cleaning, processing gaps, reducing the types of developers and applying a quantitative classification method. The developed

method is based on calculating and normalizing the frequency of use of each language in the context of different types of developers to eliminate the imbalance in the sample. Two key indicators are calculated for each PL: the maximum normalized frequency of use (C_{\max}) and the coefficient of variation (CV), reflecting the uniformity of its use. Classification into 3 categories – “general purpose”, “industry” and “niche” – is performed by comparing individual indicators of C_{\max} and CV of the language with their median values for the entire set of languages. The results, presented as a histogram, clearly demonstrate the division of languages: general-purpose (for example, Python, JavaScript), industry-specific, in-demand in specific areas (for example, Swift, Kotlin, R), and niche (for example, Crystal and Delphi). The proposed method forms a structured, data-based view of the technological landscape, which can be useful to developers when choosing tools and lays the foundation for creating recommendation systems.

Keywords: technology stack, dataset selection, software, data analysis, programming language, coefficient of variation

For citation: Kravets A. G., Kazmin I. V., Bondarenko A. G. Method for classifying software development tools based on Stack Overflow survey data. *Vestnik of Astrakhan State Technical University. Series: Management, computer science and informatics. 2025;3:102-111.* (In Russ.). <https://doi.org/10.24143/2072-9502-2025-3-102-111>. EDN DWCLNW.

Введение

В настоящее время наблюдается значительное разнообразие технологий разработки программного обеспечения (ПО). Важной задачей является определение оптимального технологического стека для решения конкретных задач разработки. Для эффективного решения данной проблемы необходим всесторонний анализ современной индустрии разработки ПО, основанный на новых данных.

Актуальность данного исследования обусловлена возрастающей потребностью в систематизации и анализе быстро развивающегося технологического ландшафта в области разработки ПО. Настоящая работа предлагает структурированный подход к анализу новых технологических тенденций на основе обработки данных, полученных из опросов сообщества разработчиков. Ключевым аспектом исследования является классификация языков программирования (ЯП) по области применения и актуальности. В отличие от существующих рейтингов, таких как индекс TIOBE, который оценивает общую популярность языков на основе количества поисковых запросов, данный метод фокусируется на анализе реального использования технологий различными категориями разработчиков. Такой подход позволяет не просто констатировать популярность, а выявлять специализацию инструментов. Результаты исследования позволяют сформировать базу знаний для разработки рекомендательных систем для выбора технологического стека и способствующих повышению эффективности разработки ПО.

Целью настоящего исследования является классификация востребованных в данный момент инструментов разработки ПО, включающих ЯП, и их классификация по областям применения. Результаты анализа предполагается использовать для алгоритма подбора оптимальных технологических стеков.

В качестве источника данных был выбран набор данных (датасет) ежегодного опроса пользователей веб-ресурса Stack Overflow (SO). Он выступает ключевым ресурсом для разработчиков, оказывая значительное влияние на выбор и применение технологических стеков в процессе разработки ПО. Многочисленные исследования подтверждают роль

этого сервиса как обширной базы знаний, используемой для решения конкретных задач, обучения и обмена опытом [1, 2], состоящей преимущественно из текстовых данных, для анализа и категоризации которых применяются современные технологии классификации текстов [3]. В частности, SO предоставляет ответы на технические вопросы, код-снимпеты и примеры использования API, что значительно ускоряет процесс разработки и позволяет обходить «лексический разрыв» между запросом и решением [4, 5].

Анализ данных сервиса позволяет выявить актуальные технологические тренды и предпочтения разработчиков, что важно для определения оптимального технологического стека для решения конкретных задач [2, 6]. Исследования показывают, что SO вносит значительный вклад в различные области разработки ПО, включая системы рекомендаций [7].

Однако важно учитывать ограничения, связанные с использованием SO. Исследования показывают, что качество информации на платформе может варьироваться, и требуется валидация предложенных решений [8]. Кроме того, при выборе технологического стека важно учитывать не только популярность технологий на сервисе, но и особенности проекта, доступность специалистов и долгосрочные перспективы. Разработчики часто прибегают к повторному использованию кода из форума, что поднимает вопросы лицензирования и качества кода [9]. Также, несмотря на полезность сервиса для поиска контекстно-зависимой информации и предложений по решению проблем [10], следует учитывать социальные взаимодействия на платформе, поскольку они могут влиять на принятие ответов и, следовательно, на восприятие разработчиками определенных решений [11].

Новый метод классификации

Опираясь на проведенный анализ, подтверждающий значимость SO как ключевого ресурса, отражающего актуальные технологические тренды, предпочтения и опыт сообщества разработчиков, и учитывая необходимость структурированного подхода для извлечения ценных знаний из большо-

го и разнородного массива данных опроса, было решено разработать новый метод классификации средств разработки. Данный метод предназначен для систематизации анализа информации, содержащейся в датасете SO, с целью объективной оценки востребованности и областей применения различ-

ных инструментов. Общая блок-схема предложенного метода, иллюстрирующая последовательность этапов – от загрузки и предобработки данных до группировки, анализа актуальности и визуализации результатов, – представлена на рис. 1.



Рис. 1. Блок-схема метода классификации средств разработки ПО

Fig. 1. Flow diagram of the ST development classification method

Подготовка данных для классификации

Датасет представлен в виде файла формата csv и имеет название survey_results_public. Объем файла составляет 155 787 килобайт.

Для анализа данных, пример которых приведен

в табл. 1, и выявления возможных проблем с их качеством был разработан модуль подготовки данных с применением специализированных библиотек: Pandas, Numpy, Seaborn и Matplotlib.

Таблица 1

Table 1

Фрагмент содержания исходного датасета

Source dataset content fragment

Responded	MainBranch	Age	Employment	RemoteWork	Check	CodingActivities	EdLevel	...
1	I am a developer by profession	Under 18 years old	Employed, full-time	Remote	Apples	Hobby	Primary/ elementary school	...
2	I am a developer by profession	35-44 years old	Employed, full-time	Remote	Apples	Hobby; Contribute to open-source proj..	Bachelor's degree (B.A., B.S., B.Eng)	...
...
65436	I am learning to code	18-24 years old	Student, full-time	NaN	Apples	NaN	Some college/ university study without earning...	...
65437	I am a developer by profession	18-24 years old	Student, full-time	NaN	Apples	NaN	Secondary school (e.g. American high school)	...

Из-за большого количества столбцов была разработана специальная процедура, позволяющая отобразить несколько релевантных для цели этого исследования атрибутов, содержащих, в частности, информацию о типе разработчика (DevType) и используемых ЯП (LanguageHaveWorkedWith). После определения списка необходимых столбцов DataFrame фильтруется, оставляя только указанные атрибуты, что способствует уменьшению объема данных и повышению эффективности дальнейшего анализа.

Далее полученный датасет был проанализирован на предмет наличия пропущенных значений. Количество пропущенных значений варьируется от колонки к колонке. Важными для дальнейшего анализа и классификации являются колонки DevType и LanguageWorkedWith. Количество пропущенных в них значений – 5 992 и 5 692 соответственно. Строки с пустыми значениями в этих столбцах были удалены.

Затем датасет был проверен на количество уникальных значений функцией df.nunique(). Большое количество уникальных значений обусловлено ти-

пом хранимых данных. Разработчики указывают сразу несколько инструментов, с которыми работали. В результате данные в колонке приобрели следующий вид: Bash/Shell (all shells); Go; HTML/CSS; Java; JavaScript; Python; TypeScript. Это касается всех колонок, кроме DevType.

Далее был проанализирован список уникальных значений колонки DevType с целью определить роли разработчиков и исключить неподходящие. Список довольно обширен и содержит записи как областей, непосредственно связанных с разработкой, так и косвенно относящихся к ней. Для исключения выбраны следующие типы: Developer Experience, Developer Advocate, Database administrator, Research & Development role, System administrator, Senior Executive (C-Suite, VP, etc.), Data or business analyst, Educator, Product manager, Scientist, Engineer, site reliability, Marketing or sales professional, Other (please specify). В результате количество уникальных записей сократилось с 34 до 20 (табл. 2), число разработчиков представлено на рис. 2.

Таблица 2

Table 2

Описание типов разработчиков

Description of developer types

Тип разработчика	Описание
Developer, full-stack	Веб-программист, который освоил полный стек технологий для реализации проекта
Student	Студент или обучающийся
Academic researcher	Академический исследователь или ученый-исследователь
Developer, back-end	Специалист, который занимается серверной частью веб-сайтов
Developer, front-end	Специалист, который создает интерфейсы веб-сайтов и приложений
Developer, desktop or enterprise applications	Специалист по разработке настольных или корпоративных приложений
Cloud infrastructure engineer	Инженер облачной инфраструктуры
Data scientist or machine learning specialist	Специалист, который использует статистические методы, машинное обучение, анализ данных и прогнозирующую аналитику для превращения необработанных данных в полезные выводы
Developer, embedded applications or devices	Специалист, который пишет управляющее ПО для различных устройств и гаджетов
DevOps specialist	Универсальный IT-специалист, который работает над запуском ПО
Engineering manager	Специалист, который планирует, организует и управляет инженерными проектами, обычно для создания продуктов или систем
Designer	Дизайнер
Security professional	Специалист в области компьютерной безопасности
Developer, mobile	Разработчик мобильных приложений
Developer, game or graphics	Разработчик игр
Developer, QA or test	Тестировщик
Developer, AI	Разработчик искусственного интеллекта
Blockchain	Специалист, который создает и поддерживает приложения на основе блокчейна
Hardware engineer	Инженер, который занимается разработкой физического оборудования, необходимого для компьютерной системы
Data engineer	Специалист, ответственный за разработку, поддержку и управление инфраструктурой данных в организации

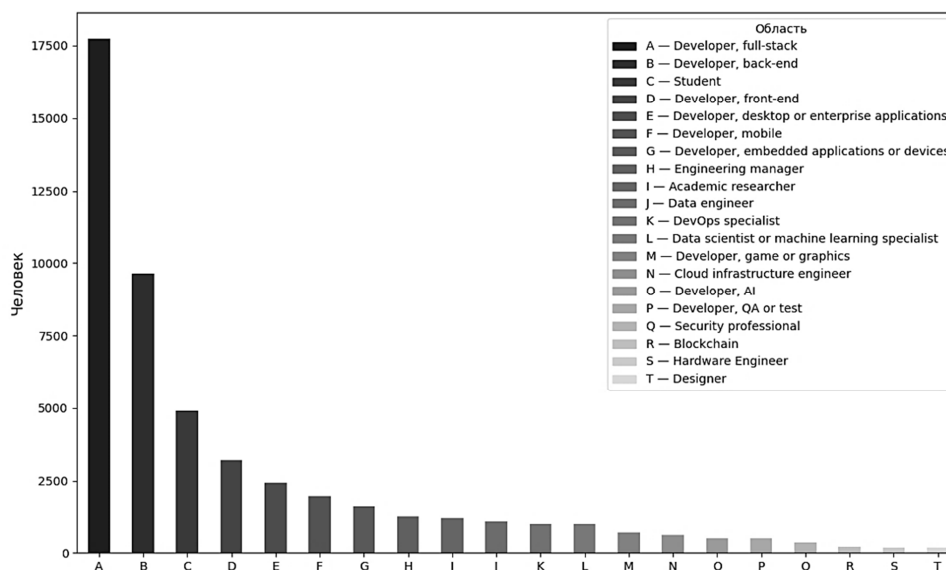


Рис. 2. Количество разработчиков по областям деятельности

Fig. 2. Number of developers by fields of activity

Чтобы определить разработчиков по типу, были отображены все уникальные значения колонки DevType. Этим значениям были присвоены определения, раскрывающие предметную область работы конкретного специалиста. Большинство респондентов заняты в области веб-разработки. С наибольшим показателем расположились full-stack разработчики – 17 500 человек. За ними следуют back-end специалисты с показателем менее 10 000. Далее, по убыванию, за ними следуют студенты и front-end-разработчики. Самыми малочисленными группами оказались дизайнеры, hardware engineer, блокчейн-разработчики и специалисты в области компьютерной безопасности.

В последующем разделе будет представлен анализ ЯП. Для каждой категории сформированы тепловые карты, отражающие следующие параметры: название инструмента, метрика Count (ча-

стота упоминания DevType) для соответствующих типов специалистов.

Классификация языков программирования

Приступая к классификации ЯП, необходимо выявить наиболее характерную область применения для конкретной технологии в текущем датасете, а также определить ее популярность по состоянию на 2024 г. Так как технологий в датасете несколько, в этом разделе будет подробно описан метод для категории ЯП.

Далее были осуществлены следующие изменения: преобразованы списки словарей колонки LanguageWorkedWith, созданы строки для каждого языка, сгруппировано и посчитано их количество, транспонирована таблица, чтобы DevType стали колонками (табл. 3).

Таблица 3

Table 3

Фрагмент таблицы языков программирования

Programming language table fragment

DevType	Language	Academic researcher	Blockchain	Cloud infrastructure engineer	Data engineer	Data or business analyst	...
0	Ada	25.0	17.0	10.0	6.0	14.0	...
1	Apex	8.0	7.0	10.0	11.0	11.0	...
2	Assembly	114.0	24.0	19.0	19.0	13.0	...
3	Bash/Shell	511.0	79.0	400.0	471.0	116.0	...
4	C	428.0	40.0	110.0	105.0	52.0	...
5	C#	122.0	35.0	123.0	172.0	82.0	...
6	C++	468.0	51.0	96.0	119.0	59.0	...
...

Каждая строка представляет собой данные о количестве разработчиков, использующих конкретный ЯП. Однако эти данные могут быть искажены преобладающей выборкой веб-разработчиков. Исходя из цели анализа, следует учитывать долю от количества разработчиков каждого типа в таблице напротив каждого из языков. Нормализуем данные, присвоив вместо количества разработчиков их долю от максимального количества:

$$M_{\text{norm}}(L, D) = M(L, D) / M_{\text{max}}(L, D), \quad (1)$$

где $M_{\text{norm}}(L, D)$ – нормализованное значение метрики для ЯП L и типа разработчика D ; $M(L, D)$ – исходное количество разработчиков, использующих язык L в области D ; $M_{\text{max}}(L, D)$ – максимальное количество разработчиков среди всех языков для данного типа разработчика D .

Формула (1) приводит значения к диапазону $[0; 1]$, позволяя сравнивать относительную популярность ЯП среди различных категорий разработчиков (табл. 4).

Таблица 4

Table 4

Фрагмент таблицы с нормализованными значениями

Table fragment with normalized values

DevType	Language	Academic researcher	Blockchain	Cloud infrastructure engineer	Data engineer	Data or business analyst	...
0	Ada	0.026261	0.133858	0.023256	0.006536	0.038781	...
1	Apex	0.008403	0.055118	0.023256	0.011983	0.030471	...
2	Assembly	0.119748	0.188976	0.044186	0.020697	0.036011	...
3	Bash/Shell	0.536765	0.622047	0.930233	0.513072	0.321330	...
4	C	0.449580	0.314961	0.255814	0.114379	0.144044	...
5	C#	0.128151	0.275591	0.286047	0.187364	0.227147	...
6	C++	0.491597	0.401575	0.223256	0.129630	0.163435	...
...

Далее была построена тепловая карта для каждого ЯП (рис. 3).

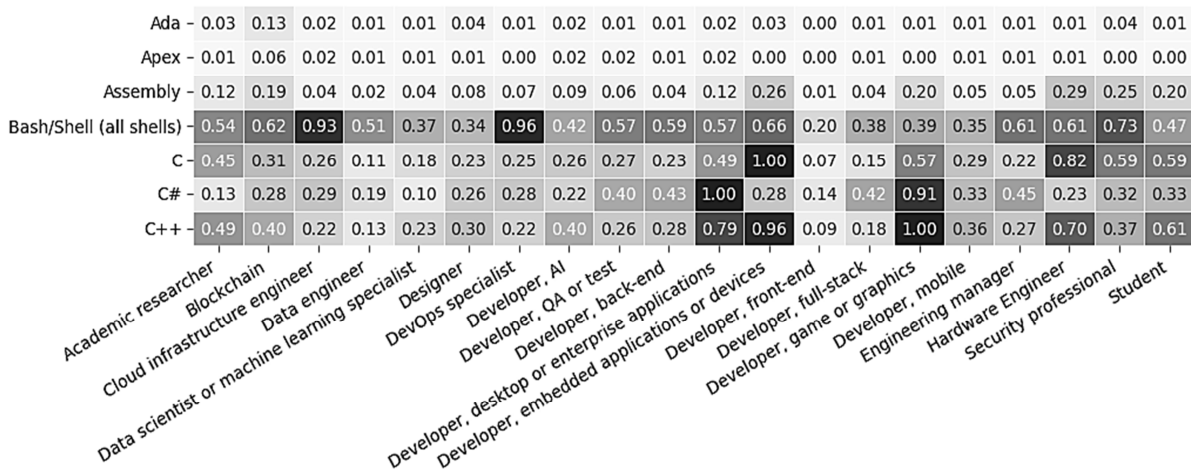


Рис. 3. Тепловая карта языков программирования (фрагмент)

Fig. 3. Heat map of programming languages (fragment)

Результаты показывают распределения значений для выбранного языка относительно группы разработчиков и позволяют оценить его востребованность.

На рис. 3 представлен фрагмент этой тепловой карты, демонстрирующий распределение для 6 язы-

ков. На этом фрагменте заметно, что некоторые языки, например язык C++ или Bash/Shell, показывают относительно высокие значения для нескольких групп разработчиков, в то время как другие, например Ada или Assembly, имеют в целом более

низкие показатели среди представленных на фрагменте типов специалистов. Это иллюстрирует общую тенденцию разной степени популярности и специализации ЯП.

Оценка «Актуальность» заключается в классификации инструментов по трем типам: общего назначения, нишевые и отраслевые. Чтобы определить, является ли язык нишевым, был построен график (рис. 4) распределения максимальных зна-

чений и найдено медианное:

$$C_{\text{mean}} = 1 / N \sum_{i=1}^N C_i, \quad (2)$$

где C_{mean} – среднее значение Count; N – количество типов специалистов; C_i – значение Count для i -го типа специалистов.

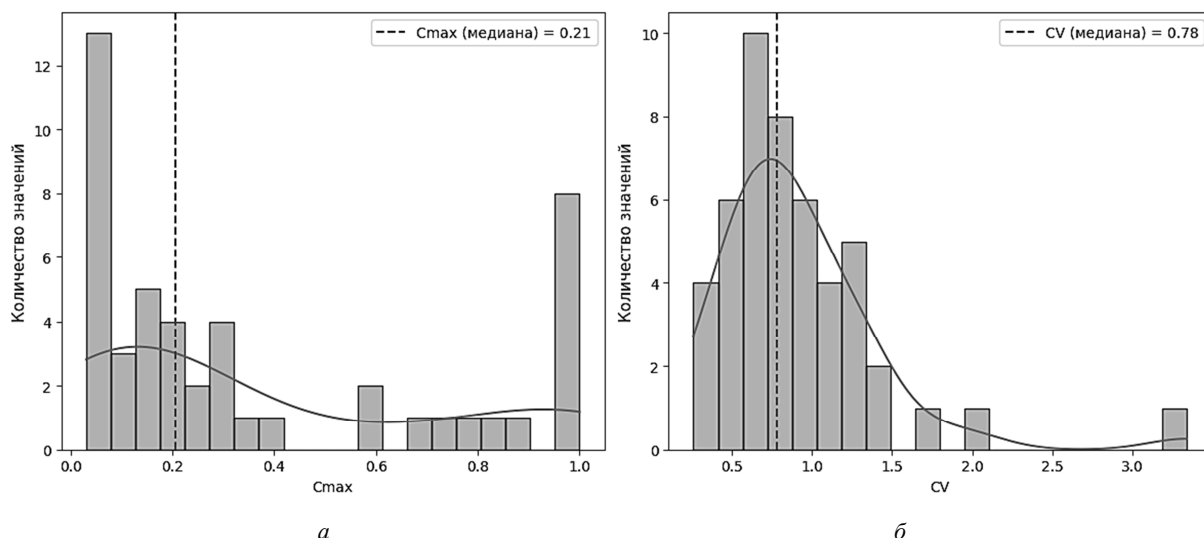


Рис. 4. Диаграмма распределения максимального значения Count (а) и коэффициента вариации для языков программирования (б)

Fig. 4. Distribution diagram of maximum Count (a) and coefficient of variation (b) for programming languages

Медианное значение для C_{max} ЯП равно 0,21 (см. рис. 4). Это означает, что почти половина языков используется редко или в очень ограниченных контекстах. Такой высокий процент языков с низкой максимальной популярностью логично выделяет категорию «нишевых». Возьмем Cobol. У него $C_{\text{max}} = 0,03$, что значительно ниже порога 0,21. Cobol известен как устаревший язык, применяемый преимущественно в старых банковских системах (legacy-системах).

Далее были определены стандартное отклонение и коэффициент вариации, исходя из формул

$$C_{\text{std}} = \sqrt{1 / N \sum_{i=1}^N (x_i - C_{\text{mean}})^2}; \quad (3)$$

$$CV = C_{\text{std}} / C_{\text{mean}}, \quad (4)$$

где C_{std} – стандартное отклонение для каждого инструмента с долями использования $x_1, x_2, x_3, \dots, x_N$ по N ролям; CV – коэффициент вариации для ЯП.

Коэффициент вариации (см. рис. 4) показывает, насколько максимальная популярность языка в ка-

кой-то одной области превышает его среднюю популярность по всем областям. Если это значение больше медианного (в данном случае 0,78), то он, скорее всего, сосредоточен в узкой сфере. Если значение меньше медианы, язык используется равномерно в разных сферах.

Для оценки актуальности (relevance) были определены 3 класса актуальности {0, 1, 2}, которые интерпретируются на лингвистической шкале {«общего назначения», «нишевые» и «отраслевые»}.

Итоговая формула оценки актуальности выглядит следующим образом

$$\text{relevance} \sim \begin{cases} 0, & \text{if } C_{\text{max}} > C_{\text{max, med}} \text{ and } CV > CV_{\text{med}}, \\ 1, & \text{if } C_{\text{max}} < C_{\text{max, med}}, \\ 2, & \text{if } C_{\text{max}} > C_{\text{max, med}} \text{ and } CV < CV_{\text{med}}, \end{cases}$$

где C_{max} – максимальное значение Count для инструмента; $C_{\text{max, med}}$ – медианное максимальное значение Count для инструмента.

Результаты исследования актуальности инструментов представлены в виде гистограммы на

рис. 5: инструменты классифицированы и сгруппированы по трем категориям, обозначенным разными цветами в соответствии с легендой: универ-

сальные (черный цвет), узкоспециализированные (темно-серый) и непопулярные (светло-серый).

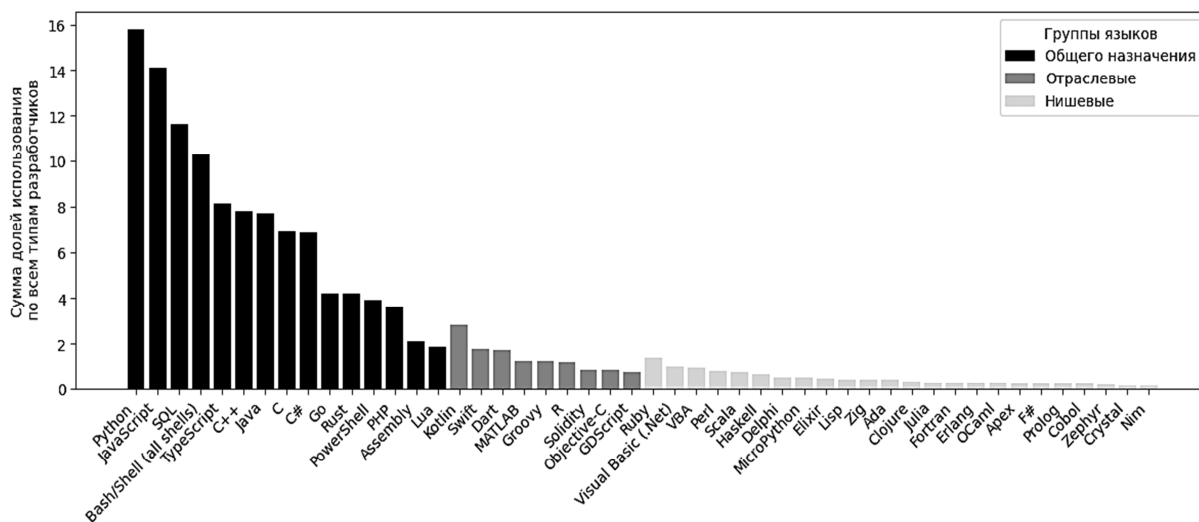


Рис. 5. Гистограмма актуальности языков программирования по группам

Fig. 5. Bar chart of programming language relevance by group

Высота столбца каждой технологии на диаграмме коррелирует с уровнем ее актуальности, рассчитанным на основе анализа данных о количестве упоминаний (Count) в различных источниках, как описано в формулах (2)–(5). Языки программирования на оси X отсортированы по убыванию этого показателя для каждой группы. Наиболее крупные, выделенные черным цветом, такие как Python, JavaScript и C++, представляют собой самые популярные языки общего назначения. Темно-серым цветом отмечены отраслевые языки, например R, Kotlin и Swift, востребованные в конкретных областях. Множество ЯП в последней группе, изображенных светло-серым цветом, демонстрируют значительно меньшую актуальность, например Crystal и Delphi. Таким образом, визуальное представление наглядно демонстрирует относительную популярность и востребованность каждого инструмента, а также его принадлежность к одной из трех выделенных групп, в рамках проведенного исследования.

Заключение

Проведен анализ востребованности технологических стеков, применяемых современными разработчиками. На основе датасета опроса «Stack Overflow 2024» рассмотрены языки программирования (ЯП). Анализ включает выявление наиболее популярных языков и оценку их «актуальности», которая определяется на основе распределения частоты упоминания конкретных ЯП среди различных

типов специалистов.

Научная новизна работы заключается в разработке и применении количественного метода классификации ЯП, основанного на анализе распределения их популярности среди различных типов специалистов. Метод использует данные опроса SO и статистические метрики, что позволяет объективно классифицировать языки как инструменты общего назначения, отраслевые или нишевые. Такой подход формирует структурированное представление об актуальном технологическом ландшафте и может использоваться для поддержки решений в области стратегического планирования разработки программного обеспечения.

Ключевым результатом работы является формирование комплексного представления о текущих технологических трендах и областях применения различных инструментов. Были выявлены как инструменты общего назначения, широко используемые специалистами разных профилей, так и отраслевые решения, востребованные в конкретных сферах разработки. Полученные данные позволяют получить структурированное представление о технологическом ландшафте и могут быть полезны как начинающим, так и опытным разработчикам для выбора оптимального набора инструментов.

Дальнейшие исследования будут направлены на разработку рекомендательных систем, позволяющих автоматически подбирать оптимальный стек технологий для решения конкретных задач.

Список источников

1. Yang Wenhua, Shen Chaochao. Understanding the Role of Stack Overflow in Supporting Software Development Tasks: A Research Perspective // *International Journal of Software Engineering and Knowledge Engineering*. 2023. V. 33 (07). P. 1119–1148. DOI: 10.1142/S0218194023500274.
2. Giwangkoro G. G., Nugroho Y. S. Unveiling Research Trends in Stack Overflow: A Comprehensive Analysis of General Discussion Theme // *2024 International Conference on Smart Computing, IoT and Machine Learning (SIML)*. Surakarta, Indonesia, 2024. P. 130–136. DOI: 10.1109/SIML61815.2024.10578280.
3. Kravets A., Semenochkin D. Text Classification Technologies in Document Categorization Systems. A Survey // *Advances in Systems Science and Applications*. 2024. V. 24. N. 2. P. 133–165.
4. Silva R., Roy Ch., Rahman M. M., Schneider K., Paixão K., Dantas C., Maia M. CROKAGE: effective solution recommendation for programming tasks by leveraging crowd knowledge // *Empirical Software Engineering*. 2020. V. 25. P. 1–52. DOI: 10.1007/s10664-020-09863-2.
5. Rubei R., Di Sipio C., Nguyen Ph., Rocco J., Di Ruscio D. PostFinder: Mining Stack Overflow posts to support software developers // *Information and Software Technology*. 2020. V. 127. P. 106367. DOI: 10.1016/j.infsof.2020.106367.
6. Dada O. A., Obaido G., Sanusi I. T., Aruleba K., Yunusa A. A. Hidden Gold for IT Professionals, Educators, and Students: Insights From Stack Overflow Survey // *IEEE Transactions on Computational Social Systems*. 2023. V. 10. N. 2. P. 795–806. DOI: 10.1109/TCSS.2022.3151130.
7. Minaoar Hossain Tanzil, Shaiful Chowdhury, Somayeh Modaberi, Gias Uddin, Hadi Hemmati. A systematic mapping study of crowd knowledge enhanced software engineering research using Stack Overflow // *Journal of Systems and Software*. 2025. V. 226. P. 112405. <https://doi.org/10.1016/j.jss.2025.112405>.
8. Meldrum S., Licorish Sh., Savarimuthu B. T. R. Exploring Research Interest in Stack Overflow -- A Systematic Mapping Study and Quality Evaluation // *arXiv:2010.12282*. URL: <https://arxiv.org/abs/2010.12282> (дата обращения: 17.02.2024).
9. Yuan Huang, Furen Xu, Haojie Zhou, Xiangping Chen, Xiacong Zhou, Tong Wang. Towards exploring the code reuse from stack overflow during software development // In *Proceedings of the 30th IEEE/ACM International Conference on Program Comprehension (ICPC '22)*. NY: Association for Computing Machinery, 2022. P. 548–559. <https://doi.org/10.1145/3524610.3527923>.
10. Vadlamani S. L., Baysal O. Studying Software Developer Expertise and Contributions in Stack Overflow and GitHub // *2020 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. Adelaide, SA, Australia, 2020. P. 312–323. DOI: 10.1109/ICSME46990.2020.00038.
11. Zhang Z., Mao X., Lu Y., Wang S., Lu J. An Empirical Study on the Influence of Social Interactions for the Acceptance of Answers in Stack Overflow // *2020 27th Asia-Pacific Software Engineering Conference (APSEC)*. Singapore, 2020. P. 425–434. DOI: 10.1109/APSEC51365.2020.00051.

References

1. Yang Wenhua, Shen Chaochao. Understanding the Role of Stack Overflow in Supporting Software Development Tasks: A Research Perspective. *International Journal of Software Engineering and Knowledge Engineering*, 2023, vol. 33 (07), pp. 1119-1148. DOI: 10.1142/S0218194023500274.
2. Giwangkoro G. G., Nugroho Y. S. Unveiling Research Trends in Stack Overflow: A Comprehensive Analysis of General Discussion Theme. *2024 International Conference on Smart Computing, IoT and Machine Learning (SIML)*. Surakarta, Indonesia, 2024. Pp. 130-136. DOI: 10.1109/SIML61815.2024.10578280.
3. Kravets A., Semenochkin D. Text Classification Technologies in Document Categorization Systems. A Survey. *Advances in Systems Science and Applications*, 2024, vol. 24, no. 2, pp. 133-165.
4. Silva R., Roy Ch., Rahman M. M., Schneider K., Paixão K., Dantas C., Maia M. CROKAGE: effective solution recommendation for programming tasks by leveraging crowd knowledge. *Empirical Software Engineering*, 2020, vol. 25, pp. 1-52. DOI: 10.1007/s10664-020-09863-2.
5. Rubei R., Di Sipio C., Nguyen Ph., Rocco J., Di Ruscio D. PostFinder: Mining Stack Overflow posts to support software developers. *Information and Software Technology*, 2020, vol. 127, p. 106367. DOI: 10.1016/j.infsof.2020.106367.
6. Dada O. A., Obaido G., Sanusi I. T., Aruleba K., Yunusa A. A. Hidden Gold for IT Professionals, Educators, and Students: Insights From Stack Overflow Survey. *IEEE Transactions on Computational Social Systems*, 2023, vol. 10, no. 2, pp. 795-806. DOI: 10.1109/TCSS.2022.3151130.
7. Minaoar Hossain Tanzil, Shaiful Chowdhury, Somayeh Modaberi, Gias Uddin, Hadi Hemmati. A systematic mapping study of crowd knowledge enhanced software engineering research using Stack Overflow. *Journal of Systems and Software*, 2025, vol. 226, p. 112405. <https://doi.org/10.1016/j.jss.2025.112405>.
8. Meldrum S., Licorish Sh., Savarimuthu B. T. R. Exploring Research Interest in Stack Overflow -- A Systematic Mapping Study and Quality Evaluation. *arXiv:2010.12282*. Available at: <https://arxiv.org/abs/2010.12282> (accessed: 17.02.2025).
9. Yuan Huang, Furen Xu, Haojie Zhou, Xiangping Chen, Xiacong Zhou, Tong Wang. Towards exploring the code reuse from stack overflow during software development. In *Proceedings of the 30th IEEE/ACM International Conference on Program Comprehension (ICPC '22)*. New York, Association for Computing Machinery, 2022. Pp. 548-559. <https://doi.org/10.1145/3524610.3527923>.
10. Vadlamani S. L., Baysal O. Studying Software Developer Expertise and Contributions in Stack Overflow and GitHub. *2020 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. Adelaide, SA, Australia, 2020. Pp. 312-323. DOI: 10.1109/ICSME46990.2020.00038.
11. Zhang Z., Mao X., Lu Y., Wang S., Lu J. An Empirical Study on the Influence of Social Interactions for the Acceptance of Answers in Stack Overflow. *2020 27th Asia-Pacific Software Engineering Conference (APSEC)*. Singapore, 2020. Pp. 425-434. DOI: 10.1109/APSEC51365.2020.00051.

Статья поступила в редакцию 19.04.2025; одобрена после рецензирования 15.06.2025; принята к публикации 23.07.2025
The article was submitted 19.04.2025; approved after reviewing 15.06.2025; accepted for publication 23.07.2025

Информация об авторах / Information about the authors

Алла Григорьевна Кравец – доктор технических наук, профессор; профессор кафедры систем автоматизированного проектирования и поискового конструирования; Волгоградский государственный технический университет; allagkravets@yandex.ru

Иван Витальевич Казьмин – аспирант кафедры систем автоматизированного проектирования и поискового конструирования; Волгоградский государственный технический университет; hoghor2@mail.ru

Артём Геннадьевич Бондаренко – инженер УЦ «Цифровая кафедра»; Волгоградский государственный технический университет; temdit01@yandex.ru

Alla G. Kravets – Doctor of Technical Sciences, Professor; Professor of the Department of Computer-Aided Design and Search Design Systems; Volgograd State Technical University; allagkravets@yandex.ru

Ivan V. Kazmin – Postgraduate Student of the Department of Computer-Aided Design and Search Design Systems; Volgograd State Technical University; hoghor2@mail.ru

Artem G. Bondarenko – Engineer of the Training Center «Digital Department»; Volgograd State Technical University; temdit01@yandex.ru

