МОДЕЛИРОВАНИЕ ДИНАМИКИ РОЛЕЙ РАЗРАБОТЧИКОВ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ПО ДАННЫМ РЕПОЗИТОРИЕВ ПРОГРАММНОГО КОДА

Тарасов Е.А., Ли Ч., Ковальчук С.В., канд.техн.наук
Университет ИТМО, Санкт-Петербург

Современные проекты разработки программного обеспечения характеризуются распределённостью участников и сложной структурой их взаимодействий. Платформы совместной разработки, такие как GitHub и GitLab, формируют богатую среду артефактов — коммитов, операций с ветками, задач, комментариев и др., которые отражают деятельность и различные роли пользователей. Анализ вышеперечисленных артефактов позволяет автоматически выделять пользовательские роли и изучать динамику их взаимодействий и трансформаций в проекте.

Определение ролей по данным репозиториев способствует пониманию структуры сообщества и оптимизации распределения задач. Однако существующие подходы преимущественно основаны на статистическом анализе и не учитывают динамический характер взаимодействий между участниками разработки.

Перспективным направлением имитации взаимодействия разработчиков как агентов с различными ролями и поведением является использование агентного моделирования (agent-based modeling, ABM). Такой подход обеспечивает возможность моделирования и прогнозирования изменений в проекте при варьировании структуры ролей или интенсивности коммуникаций.

Сложность взаимодействий участников и масштабность проектов требуют внедрения методов, одновременно учитывающих многие параметры, в том числе, временной контекст активности (например, последовательности коммитов, код-ревью, участие в обсуждениях задач). Дополнительно необходимо обеспечить воспроизводимость анализа и возможность сравнения результатов между проектами. Для соблюдения описанных требований необходим инструментарий, объединяющий возможность извлечения артефактов из репозиториев и моделирования поведения команд.

Одним из вариантов отработки навыков выделения ролей и реконструкции сетей взаимодействий является использование готовых датасетов, предоставляющих структурированные данные о репозиториях, коммитах, авторах и взаимодействиях, например, датасет, размещенный на платформе HarvardDataverse [1]. Использование готовых датасетов ускоряет воспроизводимость исследований и позволяет сравнивать результаты между проектами, поскольку структура данных в таких наборах стандартизирована и документирована.

Ключевые этапы идентификации ролей строятся на анализе данных репозитория и включают несколько подходов: кластеризацию, машинное обучение, сетевой анализ и процессный майнинг. В литературе встречаются различные наборы ролей и методы их определения. В научной литературе описаны методы

(в том числе, комбинированные) кластеризации, машинного обучения, сетевого анализа и интеллектуального анализа процессов [2–6].

Существующие методы выделения пользовательских ролей различаются по целям анализа, требованиям к данным и степени интерпретируемости получаемых результатов. Сравнительные характеристики ряда методов, используемых для выделения ролей представлены в Таблице 1.

Таблица 1 − Обзор методов выделения пользовательских ролей

Метод	**Роли**	**Артефакты/ Данные**	**Метрики/ Признаки**	**Масштаб исследования**	**Оценка/ Валидация**	**Источник**
Машинное обучение (Supervised)	Backend, Frontend, Full-stack, Mobile, DevOps, Data Science	Профили GitHub, проекты, языки программирования, биографии	Языки программирования, текстовые описания	2284 разработчика, 6 ролей	Precision: 0.88, AUC: 0.89	[5]
Архитектурный анализ + кластеризация	Эксперты по компонентам, различные типы контрибьюторов	Git коммиты, модификации файлов, структура репозитория	LOC+/-, FUNC+/-, комментарии, модификации файлов	Apache Tomcat, Kafka, Django и др.	Качественный анализ распределения вклада	[2]
Факторный анализ + иерархическая кластеризация	4 активные роли, 5 поддерживающих ролей	GitHub actions: коммиты, issues, PR, комментарии	19 метрик: code contribution, opinion contribution, network, administration	29 проектов, 20,838 контрибьюторов	Силуэтный анализ, качественная интерпретация	[4]
Интеллектуальный анализ процессов (process mining)	Project leader, Core member, Active developer, Peripheral developer, Bug fixer, Bug reporter	Version control, bug trackers, mail archives	Последовательности событий, временные метки	aMSN: 1725 разработчиков, GCC: 42373 bug reports	Визуальный анализ (Dotted Chart), сравнение с теоретическими моделями	[7]
Моделирование организационных рутин	HumanActor, DigitalActor, Mediator, Intermediary	GitHub issues, события, временные штампы	Последовательности действий, временные параметры	scikit-learn: 25,799 issues, 617,440 events	Марковские модели, матрицы переходов	[8]

Экспериментальная часть настоящего исследования посвящена исследованию систематических изменений в неявных нормах, регулирующих сроки выполнения обязательств участников, и является пробным этапом для подготовки к будущему изучению сложных закономерностей между естественно возникающими поведенческими нормами и формированием, стабилизацией и трансформацией ролей в рамках взаимодействия разработчиков в сообществах по разработке программного обеспечения с открытым исходным кодом. В рамках эксперимента рассматривается эволюция межпроектных норм в рамках отдельных команд или адаптации норм новыми членами команды.

Используемый набор данных. В проведенном анализе использовался массивный набор данных GitHub [1], включающий 10 649 574 записей пользователей, идентифицирующих 6176 организаций и 9437 активных репозиториев. Из отобранных записей были извлечены 666 375 коммитов из 9474 уникальных репозиториев, при этом 48 069 записей содержали активные вклады в репозитории.

В рамках исследования разработана комплексная структура извлечения временных характеристик, стандартизирующая все временные метки в формате UTC для географического сравнения. Для каждого коммита были выделены пять важных временных характеристик: час дня (0-23), день недели (0-6), классификация рабочего времени (9:00-18:00 UTC), индикатор выходных дней и квартальный период. Такое временное представление позволило провести детальный анализ моделей вклада в различных контекстах сообщества.

Показатели количественной оценки норм. В рамках исследования путем анализа распределения временных поведений были выделены ключевые показатели - соотношение рабочих часов (доля коммитов в течение стандартного рабочего времени), соотношение выходных дней и 24-мерные профили распределения по часам, характеризующие временную сигнатуру каждого сообщества. Были рассчитаны показатели разницы ролей (для основных и периферийных разработчиков) и индексы временной эволюции с помощью корреляции Пирсона между притоком новых членов и соотношением рабочих часов для репозиториев с достаточным количеством продольных данных (>4 квартала).

Геопространственная и временная интеграция. С целью учета географических факторов, были нормализованы и сгруппированы по географическим регионам местоположения участников по крупным городам/странам. Для каждого региона с достаточным количеством коммитов (>10) были рассчитаны коэффициенты рабочих часов, пиковые часы активности и модели участия в выходные дни. Отклонение региональных норм от глобальных базовых показателей репозитория позволило количественно оценить географическое влияние на временное поведение, и показать, как культурные и региональные факторы формируют модели сотрудничества в распределенных сообществах разработчиков.

Анализ особенностей. Использованный подход сочетал визуальную аналитику со статистическим моделированием. Визуализации включали сравнительные почасовые распределения по ролям участников, графики временной эволюции по квартальным периодам и географические тепловые карты моделей работы. При проведении статистической валидации были использованы методы

корреляционного анализа (корреляция Пирсона), описательная статистика по разным группам ролей, временные ряды для отслеживания динамики метрик и географическая кластеризация объектов с использованием распределений частоты коммитов.

Результаты и обсуждение. Анализ эволюции временных норм в четырех крупных репозиториях GitHub позволил выявить значимые закономерности в формировании и трансформации неявных норм сотрудничества:

1) microsoft/vscode демонстрирует устойчивость в зрелом сообществе, с долей вклада в рабочие часы (09:00–18:00 UTC) 49,8% и долей вклада в выходные дни 22,6%. Приверженность основных участников к соблюдению рабочего времени составляет 50,6% по сравнению с 0,0% для новых членов, что указывает на строгое соблюдение временных границ. Зрелое сообщество демонстрирует хорошую устойчивость норм с минимальным влиянием новых сотрудников на устоявшиеся закономерности.

2) vuejs/awesome-vue демонстрирует эрозию норм, вызванную распределенностью сообщества, демонстрируя соотношение рабочего времени к общему времени в 44,2% и коэффициент вклада в выходные дни в 27,6%, что отражает гибкую временную культуру. Основные участники поддерживают 57,1% рабочего времени по сравнению с 42,9% у новых членов, при этом наблюдается сильная отрицательная корреляция (r = -0,717) (рис. 1а) между притоком новых членов и соблюдением рабочего времени. Полученный результат указывает на значительное размывание норм по мере глобализации сообщества (рис. 1б).

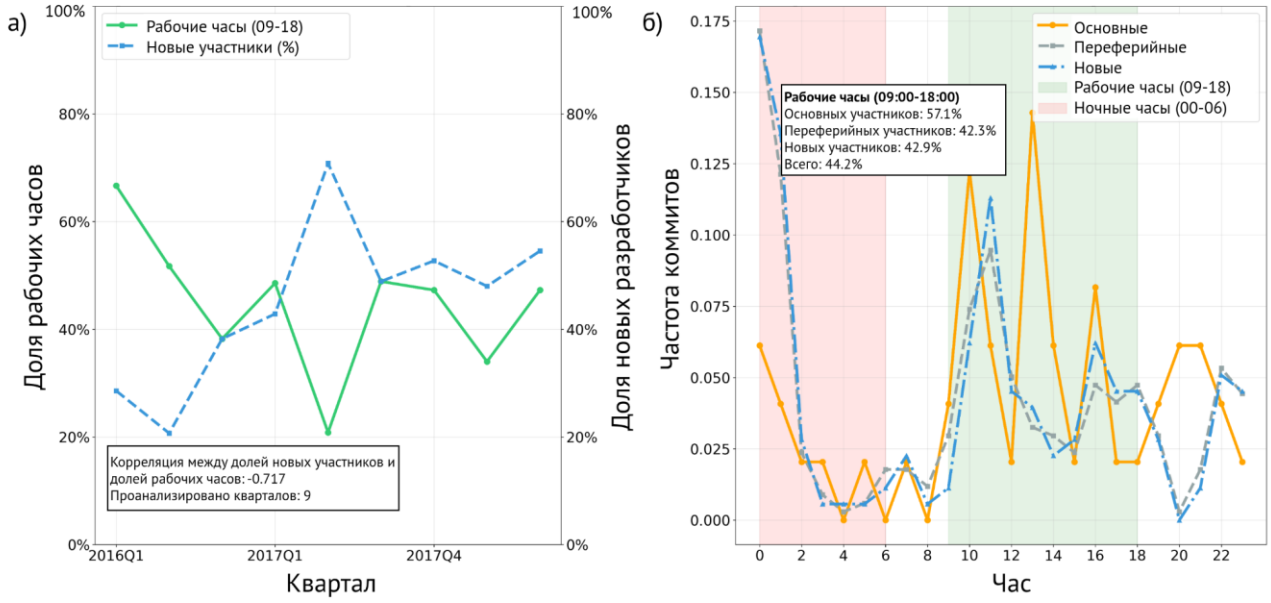


Рисунок 1 – Пример анализа репозитория для vuejs/awesome-vue: а) эволюция временных норм; б) распределение коммитов по часам

3) square/picasso представляет собой корпоративную инверсию норм, демонстрируя самый высокий показатель рабочего времени (71,4%) и самый низкий показатель выходных дней (20,0%) в нашей выборке. Происходит характер-

ная инверсия норм: новые участники демонстрируют более высокую приверженность рабочему времени (84,6%), чем основные участники (66,7%), что отражает влияние корпоративной политики в отношении рабочего времени на вклад в работу с открытым исходным кодом.

4) tencent/wepy демонстрирует культурные временные особенности, показывая 60,0% долю рабочего времени с исключительно низким показателем по выходным дням (14,0%), что отражает национальные особенности культуры труда в КНР. Наблюдается еще одно нарушение нормы: основные участники демонстрируют более низкую приверженность рабочему времени (36,4%), чем новые члены (52,9%), что указывает на культурное влияние на модели участия.

Сравнительная модель эволюции временных норм выявляет три различных паттерна: организационное влияние, при котором корпоративные проекты поддерживают более высокие показатели соотношения рабочего времени (49,8%-71,4%), чем проекты, реализуемые силами сообщества (44,2%); географические эффекты, при которых концентрированная географическая база коррелирует с более четкими границами между работой и отдыхом; и динамика ролевых норм, демонстрирующая паттерн стабильности (microsoft/vscode), где основные участники доминируют в сохранении норм; паттерн конфликта (vuejs/awesome-vue), где новички способствуют эрозии норм (r = -0,717); и паттерн трансформации (square/picasso, tencent/wepy), где организационные нормы изменяют практику сообщества. Культурные особенности очевидны в показателях работы в выходные дни: западные проекты (22,6%-27,6%) против проектов азиатского региона (tencent/wepy - 14,0%).

Анализ полученных результатов позволяет предположить ограниченность однонаправленной модели адаптации, выявляя многонаправленный процесс взаимной адаптации в коллективах разработчиков, в котором временные нормы возникают из организационных структур, географии и динамики ролей. Обнаруженные в корпоративных репозиториях закономерности инверсии норм демонстрируют, как организационные структуры могут достаточно быстро проникать в сообщества открытого исходного кода, меняя предполагаемое направление принятия норм. Сильная отрицательная корреляция (r = -0,717) в репозитории vuejs/awesome-vue предоставляет пример влияния глобализации на временные нормы.

Практические выводы предполагают, что сопровождающим проектов следует разрабатывать стратегии временных ожиданий, соответствующие структуре и географии их сообщества. Глобальные сообщества выигрывают от инструментов асинхронного рабочего процесса, позволяющих учитывать круглосуточные модели внесения вклада, в то время как корпоративные проекты должны балансировать организационные структуры с гибкостью сообщества. Низкий показатель работы в выходные дни (14,0%) в tencent/wepy предполагает, что ожидания работы в выходные дни могут создавать культурные трения.

Ограничения и направления дальнейших исследований включают в себя сосредоточение внимания на высокоактивных репозиториях, что потенциально может упускать из виду более мелкие сообщества. В будущих исследованиях

планируется также учитывать сложность внесения вклада, выходящую за рамки времени коммитов, и расширить исследование на дополнительные репозитории, чтобы охватить весь спектр типов проектов с открытым исходным кодом, а также расширить набор анализируемых артефактов репозитория и спектр применяемых инструментов. В перспективе планируется разработка универсального интеллектуального инструмента для анализа репозиториев и поддержки принятия решений по управлению проектом.

Исследование выполнено за счет гранта Российского научного фонда №24-11-00272, https://rscf.ru/project/24-11-00272/.

Список литературы

1. Gong Q. и др. A Representative User-centric Dataset of 10 Million GitHub Developers. Harvard Dataverse, 2018. С. 2147483648, 2147483648, 1772242944, 2510.
2. Shi K. Establishing Contributor Roles within Software Repositories by Mining Architectural Information: BSc Thesis. University of Groningen, 2021. 22 с.
3. Liang J. T., Zimmermann T., Ford D. Towards mining OSS skills from GitHub activity // Proceedings of the ACM/IEEE 44th International Conference on Software Engineering: New Ideas and Emerging Results. Pittsburgh Pennsylvania: ACM, 2022. С. 106–110.
4. Gao C. и др. Large language models empowered agent-based modeling and simulation: a survey and perspectives // Humanit. Soc. Sci. Commun. 2024. Т. 11, № 1. С. 1259.
5. Montandon J. E., Valente M. T., Silva L. L. Mining the Technical Roles of GitHub Users // Inf. Softw. Technol. 2021. Т. 131. С. 106485.
6. Семенько Д. С. Обзор исследований в области сетевого анализа сообществ на платформе GitHub // Социология Науки И Технологий. Социология науки и технологий, 2025. Т. 1. С. 174–192.
7. Poncin W., Serebrenik A., Brand M. V. D. Process Mining Software Repositories // 2011 15th European Conference on Software Maintenance and Reengineering. Oldenburg, Germany: IEEE, 2011. С. 5–14.
8. Namgay P., Johnson D. Towards Modelling and Simulation of Organisational Routines // Computational Science – ICCS 2024 / под ред. Franco L. и др. Cham: Springer Nature Switzerland, 2024. Т. 14838. С. 367–379.