

МЕТОДЫ И СРЕДСТВА АНАЛИЗА МЕТРИК ПРОЦЕССА РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ОЦЕНКИ ЭФФЕКТИВНОСТИ КОМАНДЫ

Бабушкин А.М.

Научный руководитель – канд. техн. наук, доцент Болдырева Е.А.

Университет ИТМО

eaboldyreva@itmo.ru

Введение

Промышленная разработка программного обеспечения в современных реалиях невозможна без систем контроля версий, таких как GitHub, GitLab и т.п., без таск-треккеров и других инструментов, оптимизирующих командную разработку. Такие инструменты генерируют большой объем данных о ходе работы и активности отдельных разработчиков. На практике эти данные либо применяются лишь для сопровождения задач, либо и вовсе игнорируются. Из-за отсутствия формализованных методов анализа метрик процесса разработки усложняется процесс принятия эффективных управленческих решений, таких как распределение нагрузки и корректировка состава команды.

Основная часть

В работе рассматривается подход к автоматическому анализу метрик процесса разработки программного обеспечения на основе данных из систем контроля версий — в первую очередь Git-репозиториях. Основным источником информации служат коммиты, ветвления, запросы на слияние и связанные с ними артефакты: они отражают структуру вносимых изменений, интенсивность работы и характер взаимодействия внутри команды. Помимо этого, предусмотрена возможность интеграции с таск-трекерами для уточнения контекста решаемых задач и их текущего статуса, однако ключевые метрики формируются именно на основе VCS-данных.

Методика анализа включает несколько этапов. Сначала выполняется сбор и нормализация данных из системы контроля версий, идентификация разработчиков и, при наличии, сопоставление коммитов с задачами трекера. Затем рассчитываются базовые метрики: частота и объем коммитов, распределение изменений по файлам и модулям, доля автоматических и служебных изменений, соблюдение конвенций оформления сообщений, характеристики ветвления и слияния. На основе этих показателей формируются производные метрики эффективности команды, отражающие вклад отдельных участников в ключевые участки кода, баланс нагрузки, стабильность темпа разработки и устойчивость процесса к изменениям.

Итогом работы является программный комплекс, предоставляющий менеджерам и тимлидам набор метрик и наглядных визуализаций. Их совокупность позволяет выявлять узкие места в процессе разработки, обнаруживать перегруженных или, напротив, недостаточно задействованных участников команды, а также отслеживать, как изменения в практиках разработки — политиках ветвления, процедурах код-ревью, правилах оформления коммитов — сказываются на общей динамике работы. Предлагаемый подход рассчитан на регулярное применение как в промышленных, так и в учебных командах и может служить надежной основой для управленческих решений, опирающихся на объективные данные из системы контроля версий.

Выводы

В рамках работы был разработан подход к автоматическому анализу метрик процесса разработки, опирающийся на данные систем контроля версий и допускающий интеграцию с таск-трекерами. Предложенная методика формализует оценку эффективности команды через систему базовых и производных метрик, что дает возможность диагностировать дисбаланс нагрузки и оценивать влияние принятых практик на ход работы. В конечном счёте это обеспечивает руководителей и менеджеров объективной доказательной базой для принятия решений — будь то перераспределение задач или корректировка состава команды, — повышая прозрачность процесса разработки в проектах любого масштаба.

Литература

1. Gousios G., Spinellis D. GHTorrent: GitHub's data from a firehose // Proceedings of the 9th IEEE Working Conference on Mining Software Repositories. — IEEE, 2012. — URL: https://kblincoe.github.io/publications/2014_MSR_Promises_Perils.pdf (дата обращения: 15.02.2026)
2. Bird C., Nagappan N., Murphy B. et al. Don't touch my code! Examining the effects of ownership on software quality // Proceedings of the 19th ACM SIGSOFT Symposium on Foundations of Software Engineering (FSE). — ACM, 2011. — URL: <https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/bird2011dtm.pdf> (дата обращения: 27.01.2026)
3. Хабр статья “Как я статистику git парсил”, автор — bakhirev — URL: <https://habr.com/ru/articles/763342> (дата обращения: 09.02.2026)