

УДК 004.9

ПРОЕКТИРОВАНИЕ ПОДХОДА ПО УПРАВЛЕНИЮ ТРЕБОВАНИЯМИ ПРОГРАММНЫХ ПРОЕКТОВ С ПРИМЕНЕНИЕМ ЯЗЫКОВЫХ МОДЕЛЕЙ

Кавеев Р.Р. (ИТМО)

Научный руководитель – канд. техн. наук, доцент Коцюба И.Ю. (ИТМО)

Введение. Ускорение темпов цифровизации и рост сложности мобильных и облачных программных систем создаёт противоречие между объёмами проектной документации и человеческими возможностями аналитика [1]. Существующие экосистемы управления требованиями — как гибкие трекинговые среды, так и специализированные корпоративные платформы — не обеспечивают автоматизированного семантического контроля качества спецификаций, оставляя пространство для логических противоречий и несоответствий ещё на ранних этапах жизненного цикла [2]. Данное исследование направлено на проектирование подхода, объединяющего возможности больших языковых моделей (LLM) с процессами верификации требований в рамках интеллектуальной среды управления.

Основная часть. Теоретическая база исследования сформирована через сравнительный анализ методологий управления проектами: установлено, что каскадная модель Waterfall обеспечивает строгость фиксации требований, однако обладает низкой адаптивностью к изменениям, тогда как гибкие методологии Agile и Scrum предполагают итеративное уточнение бэклога, создавая повышенную нагрузку на контроль консистентности документации [3]. В обоих случаях критическим узлом жизненного цикла выступает этап управления требованиями, где накапливается наибольший потенциал ошибок, стоимость устранения которых экспоненциально возрастает на поздних стадиях разработки [4].

Ключевым решением исследования стало проектирование интеллектуальной среды управления требованиями на базе совмещения Jira и LLM, интегрирующего этап валидации со стороны LLM непосредственно в Канбан-поток разработки. Введение WIP-лимитов на данном этапе обеспечивает балансировку нагрузки и исключает накопление непроверенных артефактов. Также применена изменённая V-модель с параллельным аудитом требований, при котором LLM-сервис выполняет их проверку ещё до начала технического проектирования [5].

Центральным результатом работы стала формализация пошагового алгоритма автоматизированной проверки требований по критериям SMART, где каждый параметр получает измеримую машинную интерпретацию: S (Specific) — наличие конкретного действия и объекта; M (Measurable) — числовые показатели (мс, %, количество кликов); A/R (Achievable/Relevant) — отсутствие противоречий с ограничениями проекта; T (Time-bound) — указание условий или триггеров. Для обоснования применимости Kanban как управляющей структуры был проанализирован принцип ограничения незавершённой работы, демонстрирующий эффективность WIP-контроля в сложных системах [6].

Работоспособность алгоритма SMART-верификации подтверждена на конкретных примерах требований. Требование «Система должна быстро загружать данные в отчёты» получило оценку S: 0, M: 0, A: 1, R: 1, T: 0 и было преобразовано LLM-сервисом в верифицированную спецификацию: «Время формирования PDF-отчёта при объёме данных до 100 МБ не должно превышать 5 секунд». Аналогично, физически неосуществимое требование «обеспечить мгновенную синхронизацию при полном отсутствии интернет-соединения» (A: 0) было переформулировано в технически корректное: «реализовать локальное сохранение изменений с автоматической отправкой на сервер при восстановлении связи (Ping < 100 мс)» [7]. Подобные примеры демонстрируют, что алгоритм переводит абстрактные критерии качества в измеримые и технически реализуемые параметры, снижая зависимость результата от субъективной интерпретации аналитика.

Выводы. Проведённое исследование показало, что интеграция больших языковых моделей в процессы управления требованиями программных проектов позволяет перевести

этап верификации документации из реактивного в проактивный режим, обеспечивая обнаружение логических противоречий ещё до начала разработки. Спроектированная архитектура интеллектуальной среды с механизмами параллельного аудита со стороны LLM и алгоритмом SMART-верификации формирует воспроизводимую основу для снижения рисков неполноты и несоответствия требований в информационных системах. Выявленные практические барьеры (безопасность данных, риск галлюцинаций) определяют вектор дальнейших исследований в направлении детального проектирования информационной архитектуры, формализации экспертных промптов и разработки регламента внедрения гибридной среды.

Список использованных источников:

1. Короходкина Ю. И., Гагарина С. Н. Современные методы управления проектами // Экономика и бизнес: теория и практика. – 2022. – № 1 – 2. – С. 38 – 42.
2. Кожевникова Г. В., Грошева Н. Б. Методы проектного управления: использование гибких методологий (Agile, Scrum) в реализации проектов цифровой трансформации в банковской сфере // Международный научно-исследовательский журнал. – 2025. – № 4 (154). – С. 1–12.
3. Винтайкина О. А. Традиционный подход или Agile в современном проектном управлении? Сравнительный анализ. // Скиф. – 2019. – № 4 (32). – С. 375 – 382.
4. Невлюдов И. Ш., Андрусевич А. А., Евсеев В. В. Анализ жизненного цикла разработки программного обеспечения для корпоративных информационных систем // ВЕЖПТ. – 2010. – № 8 (48). – С. 25–27.
5. Моргунов Е. П. Система поддержки принятия решений при исследовании эффективности сложных систем: принципы разработки, требования и архитектура // Сибирский аэрокосмический журнал. – 2007. – № 3. – С. 59 – 63.
6. Huang, Chun-Che, Kusiak, Andrew Overview of Kanban systems // International Journal of Computer Integrated Manufacturing – 1996. – Vol. 9(3), P. 169 – 189.
7. Amna A. R., Poels G. Systematic Literature Mapping of User Story Research // IEEE Access. – 2022. – Vol. 10.