

РАСШИРЕНИЕ ФУНКЦИЙ ТРЕНАЖЁРА УПРАВЛЕНИЯ УЧЕБНЫМИ ПРОЕКТАМИ EDUFLOW НА ОСНОВЕ ОПТИМИЗАЦИИ ПОТОКА РАБОТ

Попов Н. Н., Хан Ахмади Р.

**Научный руководитель – канд. техн. наук, доцент Горлушкина Н. Н.
Университет ИТМО**

Введение

Традиционные подходы к визуализации задач в формате Канбан-досок при подготовке IT-специалистов зачастую не учитывают сложность современных циклов разработки ПО. Это приводит к скоплению устаревших задач, утрате связи с техническим контекстом (исходным кодом или пайплайнами CI/CD) и возникновению «иллюзии прогресса», когда крупные задачи остаются неподвижными неделями. Целью исследования является модернизация тренажёра управления учебными проектами «Eduflow» путем глубокой интеграции методологий разработки нацеленной на функции потоковой разработки (Feature-Driven & Flow-Optimized) с современной архитектурой программных решений [1].

Основная часть

Классические реализации Канбан-досок, изначально заимствованные из бережливого производства, сталкиваются с системными ограничениями в программной инженерии. Традиционная структура, состоящая всего из 4 этапов («Нужно сделать», «В работе», «На проверке» и «Готово») [2] часто маскируют скрытые узкие места, такие как ожидание код-ревью или интеграция сторонних API.

Предлагаемое решение трансформирует Канбан-доску для тренажера управления учебными проектами EDUFLOW [3] из пассивного инструмента визуализации в активный диагностический механизм управления жизненным циклом разработки. В основе методологии лежит внедрение «реактивных» колонок: «Бэклог», «Готово к разработке», «В работе», «На проверке», «Тестирование» и «Готово». «Бэклог» нужен для сырых идей. Этап «Нужно сделать» должен стать «Готово к разработке», строго требуя наличия четких требований, а этап «Тестирование» обязывает проводить верификацию в среде для тестирования. В дальнейшем сведения об активности членов команды передаются в модуль оценивания и влияют на оценку работы каждого члена команды.

Для поддержания высокой пропускной способности системы вводятся жесткие лимиты незавершенного производства (WIP-лимиты). Например, для микрокоманды из трех разработчиков устанавливается предел в пять активных задач на этапе «В работе». Задачи, заблокированные внешними зависимостями, помечаются визуальными маркерами, что инициирует немедленное командное взаимодействие для устранения блокировки. Кроме того, для поддержания гигиены бэклога внедрен автоматизированный механизм «анти-энтропии»: задачи, находящиеся в статусе «Готово к разработке» более двух недель без движения, автоматически архивируются.

Архитектурно-управленческая интеграция: ключевой инновацией системы является синхронизация управленческой доски с архитектурой кодовой базы на основе методологии Bulletproof-React. Задачи на доске и логика приложения группируются по изолированным функциональным фичам, а не по техническим слоям. Декомпозиция крупных эпиков на независимые задачи длительностью от 1 до 3 дней позволяет поддерживать ритмичность поставок.

С технической точки зрения инструментарий реализован на современном стеке: FastAPI на бэкенде и React с WebSockets на фронтенде, что обеспечивает мгновенную синхронизацию состояний доски у всех участников команды. Для минимизации когнитивной нагрузки при переключении контекста применяется подход «оптимистичного UI» через React Query, обеспечивающий бесшовный пользовательский опыт. Надежность и масштабируемость системы подтверждаются использованием многоэтапных Docker-сборок в средах, приближенных к продуктовым.

Выводы

Предложенный подход органично объединяет паттерны современной программной инженерии с методологиями гибкого управления. Ожидается, что строгая привязка архитектуры кода к Канбан-доске снизит время непосредственной работы команды над задачей (от момента начала разработки до выкладки результата), устранив проблему накопления технического долга на уровне управления и сформирует у студентов устойчивые навыки безопасной разработки в реалиях высокотехнологичного бизнеса.

Литература

1. Яровая Е. В. Принципы построения архитектуры программного обеспечения // E-Scio. 2022. №8 (71). [Электронный ресурс] URL: <https://cyberleninka.ru/article/n/printsiyu-postroeniya-arhitektury-programmnogo-obespecheniya> (дата обращения: 28.02.2026).
2. Котляр Е. В, Пушкарева Е. М. Система управления проектами канбан // Бизнес-образование в экономике знаний. 2020. №1 (15). [Электронный ресурс] URL: <https://cyberleninka.ru/article/n/sistema-upravleniya-proektami-kanban> (дата обращения: 28.02.2026).
3. Афанасьев А.В., Горлушкина Н.Н., Шутов Д.Э. Разработка информационного обеспечения системы управления проектной деятельностью студентов. Экономика. Право. Инновации. 2023. № 2. С. 92–101. [Электронный ресурс] URL: <https://books.ifmo.ru/file/pdf/3256.pdf> (дата обращения: 20.02.2026).