Итоги ревью планов разработки LMS

Дата: 7 июля 2025

Провел: Al Development Agent

Контекст: Анализ нарушений TDD в Sprint 33-38

📊 Резюме проведенного анализа

Выявленные критические проблемы:

1. Массовое создание тестов без кода

- Sprint 33: 301 тест за 20 минут (15 тестов/минуту)
- Sprint 34: 258 тестов за спринт
- Тесты создавались "пачками" для достижения метрик

2. Нарушение TDD принципов

- o Test-Last вместо Test-First
- Отсутствие цикла Red-Green-Refactor
- Тесты не запускались после написания

3. Фокус на неправильных метриках

- Цели типа "достичь 15% покрытия"
- Количество тестов как показатель успеха
- Игнорирование качества ради скорости

Внесенные изменения

1. Новый план разработки (TDD-focused)

Файл: docs/project/TDD_F0CUSED_DEVELOPMENT_PLAN.md

Ключевые принципы:

- Максимум 10 тестов в день
- Обязательный TDD цикл для каждой функции
- Качество > Количество
- 100% зеленых тестов или работа не завершена

2. Обновление методологии (v2.0.0)

Файлы обновлены:

- cursorrules добавлены усиленные правила TDD
- reports/methodology/METHODOLOGY_UPDATE_v2.0.0_TDD_ENFORCEMENT.md

Основные изменения:

- Жесткие ограничения на процесс
- Защитные механизмы (pre-commit hooks, CI/CD gates)
- Новые метрики качества вместо количества

3. Инструменты контроля

Новые файлы:

- scripts/pre-commit автоматическая проверка TDD compliance
- Планируется: .github/workflows/tdd-compliance.yml

Функциональность:

- Блокировка коммитов с красными тестами
- Проверка наличия тестов для нового кода
- Ограничение количества тестов за коммит

✓ Сравнение подходов

Метрика	Старый подход	Новый подход	Улучшение
Фокус	Покрытие (%)	Качество дизайна	00
Тестов/день	100-200	5-10	Качество ↑
Красные тесты	Игнорировались	Блокируют работу	100% ↑
TDD compliance	~0%	100% (цель)	00
Время на баги	30-40%	5-10% (прогноз)	75% ↓

© План внедрения

Немедленные действия (День 176):

- 🗸 Обновлена методология (v2.0.0)
- 🗸 Создан TDD-focused план разработки
- 🗸 Разработан pre-commit hook
- ▼ Установить hook в git/hooks/
- 📱 Настроить CI/CD pipeline

Sprint 39 (8-12 июля):

- 1. День 1: Инфраструктура TDD
 - Установка защитных механизмов
 - Обучение новому процессу
 - Первые 3 TDD цикла
- 2. **День 2-4**: Практика
 - ∘ 5-7 TDD циклов в день
 - Измерение новых метрик

- Постоянная обратная связь
- 3. День 5: TestFlight с качеством
 - Только с 100% зелеными тестами
 - Release notes о качестве
 - Анализ первой недели

🦞 Ключевые выводы

Что мы поняли:

- 1. LLM может нарушать TDD без явных ограничений AI оптимизирует скорость, а не качество
- 2. Метрики формируют поведение фокус на покрытии привел к формальному подходу
- 3. Автоматизация критична ручной контроль не работает в долгосрочной перспективе

Что изменится:

- 1. Культура качества TDD станет образом мышления
- 2. Доверие к тестам 100% зеленых тестов всегда
- 3. Скорость через качество меньше багов = быстрее разработка

📊 Метрики успеха нового подхода

Неделя 1:

- ☐ TDD Compliance Rate > 80%
- ☐ Test Stability = 100%
- 🔲 0 коммитов с красными тестами

Месяц 1:

- П Снижение багов на 50%
- Увеличение скорости на 20%
- Покрытие 25-30% (органически)

Квартал 1:

- П TDD стандартная практика
- Покрытие 40-50% (побочный эффект)
- 🗌 Уверенность в коде 100%

🚀 Заключение

Проведенный ревью выявил системные проблемы в применении TDD при LLM-разработке. Созданные изменения направлены на:

- 1. Предотвращение повторения ошибок через автоматизацию
- 2. Изменение фокуса с количества на качество
- 3. Создание культуры где TDD это норма, а не исключение

Главный вывод: При работе с LLM критически важно устанавливать жесткие рамки и автоматические проверки, иначе стремление к эффективности приведет к накоплению технического долга.

[&]quot;Медленно и правильно быстрее, чем быстро и неправильно" - новый девиз разработки