План разработки LMS с обязательным TDD

Версия: 2.0
Дата: 7 июля 2025
Статус: ОБЯЗАТЕЛЬНЫЙ К ИСПОЛНЕНИЮ

🚨 КРИТИЧЕСКИЕ ИЗМЕНЕНИЯ

❌ Больше НИКОГДА:

1. **Массовое создание тестов** - максимум 10 тестов в день
2. **Тесты без кода** - сначала тест, потом код, ВСЕГДА
3. **Фокус на метриках** - покрытие это побочный эффект, не цель
4. **Отложенный запуск** - тест должен быть запущен в течение 5 минут

✅ ВСЕГДА:

1. **Red → Green → Refactor** - святая троица TDD
2. **1 функция = 1 тест** - атомарность разработки
3. **100% зеленых тестов** - или работа не завершена
4. **Ежедневный запуск всех тестов** - обязательно

📋 Новая структура спринтов

Каждый день спринта:

```
09:00-09:30: Планирование дня
  - Выбор 3-5 функций для реализации
  - Написание тест-кейсов на бумаге

09:30-12:00: TDD циклы
  - Цикл 1: Тест → Запуск (красный) → Код → Запуск (зеленый) → Рефакторинг
  - Цикл 2: Тест → Запуск (красный) → Код → Запуск (зеленый) → Рефакторинг
  - Максимум 30 минут на цикл

12:00-13:00: Обед + запуск всех тестов проекта

13:00-16:00: TDD циклы продолжение
  - Цикл 3-5: Аналогично утренним

16:00-17:00: Интеграция и финальное тестирование
  - Запуск всех тестов
  - Исправление любых красных тестов
  - Коммит ТОЛЬКО если все зеленое

17:00-17:30: Дневной отчет
  - Сколько циклов TDD завершено
  - Все ли тесты зеленые
  - Какие проблемы встретились
```

🎯 Sprint 39: TDD Excellence (8-12 июля 2025)

Цель: Внедрить идеальный TDD процесс

День 1 (8 июля) - TDD Infrastructure

```
Задачи:
  1. Pre-commit hook:
     - Тест: hook блокирует коммит с красными тестами
     - Код: Реализация git hook
     - Проверка: Попытка закоммитить с падающим тестом

  2. CI/CD pipeline:
     - Тест: GitHub Action запускается на каждый push
     - Код: .github/workflows/tests.yml
     - Проверка: Push с зелеными тестами проходит

  3. Test runner script:
     - Тест: Скрипт запускает все тесты и показывает статистику
     - Код: scripts/run-all-tests.sh
     - Проверка: Красивый вывод с метриками

Метрики дня:
  - TDD циклов: 3
  - Новых тестов: 3
  - Все тесты зеленые: ДА
```

День 2 (9 июля) - Notification Service Refactoring

```
Задачи:
  1. Push notification handler:
     - Тест: Handler обрабатывает успешную отправку
     - Код: NotificationHandler.swift
     - Тест: Handler обрабатывает ошибки
     - Код: Обработка ошибок
     - Тест: Handler логирует события
     - Код: Логирование

  2. Notification queue:
     - Тест: Очередь сохраняет уведомления offline
     - Код: NotificationQueue.swift
     - Тест: Очередь отправляет при восстановлении сети
     - Код: Реализация retry логики

Метрики дня:
  - TDD циклов: 5
```

```
  - Новых тестов: 5
  - Все тесты зеленые: ДА
```

День 3 (10 июля) - Analytics Enhancement

```
Задачи:
  1. Event tracking:
     - Тест: Tracker записывает пользовательские события
     - Код: EventTracker.swift
     - Тест: Tracker batching для производительности
     - Код: Batch отправка

  2. Analytics export:
     - Тест: Export в CSV формат
     - Код: CSVExporter.swift
     - Тест: Export в JSON формат
     - Код: JSONExporter.swift

  3. Real-time dashboard:
     - Тест: Dashboard обновляется каждые 30 секунд
     - Код: RealtimeDashboard.swift

Метрики дня:
  - TDD циклов: 5
  - Новых тестов: 5
  - Все тесты зеленые: ДА
```

День 4 (11 июля) - Integration Testing

```
Задачи:
  1. E2E test: Полный flow регистрации
     - Тест: Пользователь может зарегистрироваться
     - Код: Интеграция всех компонентов

  2. E2E test: Прохождение курса
     - Тест: От записи до сертификата
     - Код: Исправление найденных проблем

  3. Performance test:
     - Тест: Загрузка главной < 2 секунд
     - Код: Оптимизация если нужно

Метрики дня:
  - TDD циклов: 3
  - Новых тестов: 3
  - Все тесты зеленые: ДА
```

День 5 (12 июля) - TestFlight Release

```
Утро:
  1. Запуск ВСЕХ тестов (должно быть 100% зеленых)
  2. Code review всех изменений
  3. Проверка метрик качества

День:
  1. Сборка release версии
  2. Загрузка в TestFlight
  3. Написание release notes с фокусом на качество

Вечер:
  1. Мониторинг crash reports
  2. Подготовка отчета о спринте

Метрики спринта:
  - TDD циклов всего: 16
  - Новых тестов: 16
  - Все тесты зеленые: ДА
  - Покрытие кода: НЕ ВАЖНО (но вероятно выросло)
```

📊 Метрики качества (НЕ количества!)

Ежедневные метрики:

1. **TDD Compliance Rate** = (Тестов написанных первыми / Всего тестов) × 100%
2. **Test Stability** = (Зеленых тестов / Всего тестов) × 100%
3. **Average TDD Cycle Time** = Общее время / Количество циклов
4. **Refactoring Rate** = (Циклов с рефакторингом / Всего циклов) × 100%

Sprint метрики:

1. **Sprint Test Stability** = Дней с 100% зелеными тестами / Всего дней
2. **TDD Process Adherence** = Соблюдение процесса в %
3. **Technical Debt** = Количество TODO в коде
4. **Code Quality** = Результаты линтеров и анализаторов

🚨 Контрольные точки

Ежедневно:

- [ ] Все новые функции имеют тесты, написанные ПЕРВЫМИ
- [ ] Все тесты запущены минимум 3 раза за день
- [ ] 100% тестов зеленые перед завершением дня
- [ ] Дневной отчет содержит TDD метрики

Еженедельно:

- [ ] Code review на соблюдение TDD
- [ ] Анализ метрик качества
- [ ] Ретроспектива процесса
- [ ] Обновление документации

По завершению спринта:

- [ ] Все acceptance criteria имеют тесты
- [ ] TestFlight build только с 100% зелеными тестами
- [ ] Отчет о соблюдении TDD процесса
- [ ] План улучшений на следующий спринт

🛡️ Защитные механизмы

1. Pre-commit hook

```
#!/bin/bash
# Запускаем все тесты
./scripts/run-all-tests.sh
if [ $? -ne 0 ]; then
    echo "❌ Коммит отклонен: есть падающие тесты!"
    exit 1
fi
```

2. CI/CD Gate

```
- name: Run Tests
  run: |
    xcodebuild test -scheme LMS -destination 'platform=iOS
Simulator,name=iPhone 16 Pro'
    if [ $? -ne 0 ]; then
      echo "::error::Tests failed! Push rejected."
      exit 1
    fi
```

3. Daily Test Report

```
## TDD Report - Day X

### Compliance ✅
- Tests written first: 5/5 (100%)
- All tests green: YES
- TDD cycles completed: 5

### Quality Metrics
- Average cycle time: 25 minutes
```

```
- Refactoring done: 3/5 cycles
- Code coverage: 18.5% (+1.3%)

### Issues
- None

Signed-off by: AI Agent
```

📈 Ожидаемые результаты

Через 1 спринт:

- Стабильность тестов 100%
- Скорость разработки +20% (за счет меньшего количества багов)
- Уверенность в коде 100%

Через 3 спринта:

- TDD станет естественной привычкой
- Покрытие кода органически вырастет до 30-40%
- Количество багов в production упадет на 80%

Через 6 спринтов:

- Полная автоматизация тестирования
- Возможность безопасного рефакторинга
- Документация через тесты

⚠ Критически важно

1. **НЕ ГНАТЬСЯ ЗА МЕТРИКАМИ ПОКРЫТИЯ**
2. **НЕ ПИСАТЬ ТЕСТЫ ЗАДНИМ ЧИСЛОМ**
3. **НЕ КОММИТИТЬ КРАСНЫЕ ТЕСТЫ**
4. **НЕ ПРОПУСКАТЬ РЕФАКТОРИНГ**

💡 Философия

> "TDD - это не о тестировании, это о дизайне. Тесты - это побочный эффект хорошего дизайна."

Каждый тест должен:

1. Быть понятным как документация
2. Проверять одну вещь
3. Быть независимым от других тестов
4. Выполняться быстро (<1 секунда)

🎯 Конечная цель

Создать культуру качества, где:

- Разработчики уверены в своем коде
- Рефакторинг не страшен
- Баги находятся сразу, а не в production
- Код самодокументируется через тесты
- Новые разработчики быстро понимают систему

Помните: Качество > Количество. Всегда.