

# План разработки SmartEstimate

## Обзор

Данный документ представляет собой подробный план разработки SmartEstimate — WPF-приложения для создания смет и коммерческих предложений. План разделен на семь этапов, каждый из которых включает задачи, сроки, результаты и роли команды.

## Этап 1: Прототипирование UI

**Продолжительность:** 3 недели

**Цель:** Разработка и прототипирование пользовательского интерфейса с использованием WPF и MahApps.Metro/Material Design in XAML.

**Команда:** UI/UX-дизайнер, WPF-разработчик

**Инструменты:** Figma, Visual Studio, MahApps.Metro, Material Design in XAML

## Задачи

### 1. Сбор требований (2 дня)

- Определение ключевых экранов: Главная панель, Список клиентов, Создание проекта, Редактор смет, Предпросмотр PDF, Аналитика.
- Определение пользовательских сценариев (например, создание новой сметы, экспорт в PDF).

### 2. Создание каркасов (вайрфреймов) (4 дня)

- Разработка низкоуровневых каркасов в Figma для всех ключевых экранов.
- Утверждение каркасов с заинтересованными сторонами.

### 3. Высокоуровневые прототипы (7 дней)

- Создание высокоуровневых макетов с темами Material Design in XAML.
- Обеспечение адаптивности интерфейса для разных разрешений экрана.

### 4. Реализация UI в WPF (6 дней)

- Настройка проекта WPF в .NET 8 с использованием паттерна MVVM и CommunityToolkit.Mvvm.
- Реализация базовых компонентов интерфейса (панель навигации, кнопки, формы) с использованием MahApps.Metro.
- Тестирование адаптивности и согласованности тем.

## Результаты

- Каркасы и макеты в Figma.
- Начальный проект WPF с каркасом интерфейса (навигация, базовые элементы).
- Руководство по стилю UI (цвета, типографика, иконки).

## Этап 2: База данных и авторизация

**Продолжительность:** 4 недели

**Цель:** Настройка базы данных MSSQL и реализация авторизации пользователей.

**Команда:** Бэкенд-разработчик, администратор базы данных, WPF-разработчик

**Инструменты:** MSSQL Server, Entity Framework Core, ASP.NET Identity

### Задачи

#### 1. Проектирование базы данных (5 дней)

- Разработка схемы для Клиентов, Проектов, Смет, Заказов и Пользователей.
- Определение связей (например, один-ко-многим между Клиентами и Проектами).
- Создание ER-диаграммы.

#### 2. Настройка базы данных (3 дня)

- Настройка экземпляра MSSQL Server (локально или Azure SQL для тестирования).
- Реализация базы данных с использованием Entity Framework Core (подход Code-First).

#### 3. Авторизация (7 дней)

- Реализация ASP.NET Identity для регистрации/входа пользователей.
- Добавление ролевой авторизации (например, Админ, Редактор, Читатель).
- Защита учетных данных с помощью хеширования bcrypt.

#### 4. Интеграция с WPF (5 дней)

- Создание сервисного слоя для подключения WPF-приложения к базе данных через REST API или прямое использование EF Core.
- Реализация экрана входа и управления сессиями пользователей в WPF.

### Результаты

- Схема базы данных MSSQL и ER-диаграмма.
- Функциональная система авторизации в WPF.
- Сервисный слой для взаимодействия с базой данных.

## Этап 3: Модуль расчета смет

**Продолжительность:** 5 недель

**Цель:** Разработка основного функционала для создания и расчета смет.

**Команда:** WPF-разработчик, бэкенд-разработчик

**Инструменты:** .NET 8, CommunityToolkit.Mvvm, MSSQL

## Задачи

### 1. Определение моделей (4 дня)

- Определение моделей для Проектов, Смет и Услуг (например, труд, материалы).
- Реализация валидации данных (например, неотрицательные стоимости).

### 2. Логика расчетов (7 дней)

- Разработка логики для добавления услуг, расчета итогов, применения налогов/скидок.
- Реализация моделей представления MVVM для создания смет.

### 3. Разработка UI (7 дней)

- Создание интерфейса редактора смет с динамическими полями (добавление/удаление услуг).
- Добавление предварительного просмотра расчетов в реальном времени.

### 4. Интеграция с базой данных (5 дней)

- Сохранение смет в базу данных.
- Реализация извлечения и редактирования существующих смет.

## Результаты

- Полностью функциональный модуль создания смет.
- Интерфейс для создания и редактирования смет.
- Интеграция с базой данных для хранения смет.

## Этап 4: Генератор PDF и отправка по email

**Продолжительность:** 4 недели

**Цель:** Реализация генерации PDF-документов для коммерческих предложений и функционала отправки по email.

**Команда:** WPF-разработчик, бэкенд-разработчик

**Инструменты:** QuestPDF, SMTPClient, MSSQL

## Задачи

### 1. Генерация PDF (7 дней)

- Настройка QuestPDF для создания брендированных PDF-документов.
- Создание шаблонов для коммерческих предложений (логотип, заголовков, подвал).
- Реализация динамической привязки данных смет.

### 2. Предпросмотр PDF (4 дня)

- Добавление окна предпросмотра PDF в WPF.
- Разрешение сохранения PDF локально.

### 3. Интеграция email (6 дней)

- Настройка SMTP-клиента для отправки писем (например, Gmail, Outlook).
- Реализация интерфейса для составления письма, включающего прикрепление PDF.
- Добавление логирования отправленных писем в базу данных.

### 4. Тестирование (3 дня)

- Тестирование генерации PDF для смет разного объема.
- Проверка доставки писем через разные почтовые сервисы.

## Результаты

- Функционал генерации и предпросмотра PDF.
- Отправка писем с прикрепленным PDF.
- Логирование отправленных писем в базе данных.

## Этап 5: Экспорт в Excel

**Продолжительность:** 3 недели

**Цель:** Реализация экспорта смет в формат Excel.

**Команда:** WPF-разработчик

**Инструменты:** ClosedXML, MSSQL

## Задачи

### 1. Логика экспорта (5 дней)

- Использование ClosedXML для генерации Excel-файлов из данных смет.
- Создание настраиваемых шаблонов экспорта (включение/исключение столбцов).

### 2. Интеграция с UI (4 дня)

- Добавление кнопки "Экспорт в Excel" в редакторе смет.

- Реализация диалога сохранения файла для Excel.

### 3. Тестирование (3 дня)

- Тестирование экспорта для больших наборов данных.
- Проверка форматирования и точности данных.

## Результаты

- Функционал экспорта в Excel.
- Интерфейс для запуска экспорта.

## Этап 6: Логирование и аналитика

**Продолжительность:** 4 недели

**Цель:** Реализация логирования с помощью Serilog и аналитических панелей.

**Команда:** Бэкенд-разработчик, WPF-разработчик, аналитик данных

**Инструменты:** Serilog, Elasticsearch, Kibana, MSSQL

## Задачи

### 1. Настройка логирования (5 дней)

- Настройка Serilog для логирования по всему приложению.
- Настройка Elasticsearch и Kibana для хранения и визуализации логов.
- Логирование ключевых событий (создание смет, генерация PDF).

### 2. Бэкенд аналитики (7 дней)

- Разработка запросов для анализа прибыльности, удержания клиентов и тенденций проектов.
- Создание API-эндпоинтов для данных аналитики.

### 3. UI аналитики (6 дней)

- Создание панелей WPF с графиками (с использованием LiveCharts2 или OxyPlot).
- Отображение KPI: общий доход, средняя стоимость проекта.

### 4. Тестирование (2 дня)

- Проверка точности логов и отзывчивости панелей.

## Результаты

- Система логирования с Serilog и Kibana.
- Аналитические панели в WPF.

# Этап 7: Интеграция OpenAI и финальная сборка

**Продолжительность:** 4 недели

**Цель:** Интеграция OpenAI API для интеллектуальных функций и финализация приложения.

**Команда:** WPF-разработчик, бэкенд-разработчик, инженер по тестированию

**Инструменты:** OpenAI API, .NET 8, Visual Studio

## Задачи

- Интеграция OpenAI** (7 дней)
  - Подключение к OpenAI API для автоподсказок (например, названия услуг, оценка затрат).
  - Реализация NLP для разбора требований клиентов из текста.
- Доработка функций** (5 дней)
  - Улучшение UI/UX на основе отзывов пользователей.
  - Оптимизация производительности (например, ленивая загрузка для больших наборов данных).
- Тестирование** (5 дней)
  - Проведение сквозного тестирования (модульное, интеграционное, UI).
  - Тестирование функций ИИ на точность и задержку.
- Упаковка** (3 дня)
  - Создание установщика для Windows с использованием MSIX.
  - Подготовка к подаче в Microsoft Store.

## Результаты

- Функции на базе ИИ (автоподсказки, NLP).
- Финальная сборка приложения и установщик.
- Пакет для отправки в Microsoft Store.

## Сводная таблица сроков

Этап	Продолжительность	Дата начала	Дата окончания
Прототипирование UI	3 недели	5 мая 2025	23 мая 2025
База данных и авторизация	4 недели	26 мая 2025	20 июня 2025
Расчет смет	5 недель	23 июня 2025	25 июля 2025
PDF и Email	4 недели	28 июля 2025	22 августа 2025
Экспорт в Excel	3 недели	25 августа 2025	12 сентября 2025
Логирование и аналитика	4 недели	15 сентября 2025	10 октября 2025
OpenAI и финальная сборка	4 недели	13 октября 2025	7 ноября 2025

**Общая продолжительность:** 27 недель (примерно 6,5 месяцев)

Роли команды

- **UI/UX-дизайнер:** Каркасы, макеты, руководство по стилю.
- **WPF-разработчик:** Реализация UI, логика MVVM, интеграция фронтенда.
- **Бэкенд-разработчик:** База данных, API, логирование, интеграция OpenAI.
- **Администратор базы данных:** Настройка MSSQL, проектирование схемы.
- **Аналитик данных:** Запросы аналитики, дизайн панелей.
- **Инженер по тестированию:** Тестирование и исправление ошибок.

Риски и меры смягчения

- **Риск:** Задержки в интеграции OpenAI API из-за ограничений или изменений API.  
**Меры:** Раннее тестирование API на этапе 7 и кэширование ответов локально.
- **Риск:** Проблемы с производительностью UI при больших наборах данных.  
**Меры:** Реализация виртуализации и ленивой загрузки в WPF.
- **Риск:** Проблемы совместимости с версиями MSSQL.  
**Меры:** Использование Azure SQL для единообразной тестовой среды.

Заключение

Данный план обеспечивает структурированный подход к разработке SmartEstimate, гарантируя систематическую реализацию всех функций. Регулярные проверки с заинтересованными сторонами и итеративное тестирование обеспечат высокое качество и соответствие бизнес-целям.