



8 (911) 950-99-44 | esscale@mail.ru | esscale.ru

Прайс-лист на разработку программного обеспечения

Подход, основанный на прозрачной оценке сложности

Здравствуйте!

Мы верим, что ключ к успешному проекту – это честность и предсказуемость. Вместо того чтобы пытаться "угадать" точное количество часов, мы используем современный подход из Agile-разработки – **оценку по относительной сложности**.

Это позволяет нам дать точный прогноз бюджета и сроков, основываясь не на интуиции, а на декомпозиции проекта и командной оценке.

Наша философия ценообразования: Esscale Points

Мы оцениваем каждую задачу в проекте в **Esscale Points (EP)** – условных единицах, отражающих её **объем, сложность и риски**. Для этого мы используем шкалу, основанную на числах Фибоначчи (1, 2, 3, 5, 8, 13...).

Почему это лучше для вас?

- **Прозрачность:** Вы видите, какие задачи являются самыми трудоемкими и почему.

- **Точность:** Оценивать задачи относительно друг друга («эта задача в два раза сложнее той») намного точнее, чем гадать в абсолютных часах.
- **Гибкость:** Если в проект нужно добавить новую фичу, мы легко можем оценить её в Esscale Points и показать, как это повлияет на общий бюджет и сроки.

$$Total\ cost = \sum_{i=1}^k EP_i * Price\ of\ ep$$

$$EP_i = \prod_{k=1}^{n_i} Coeff_{ik} * f_k(i) \sum_{j=1}^{m_i} ep_{ji} * g_j(i) \ ,$$

где

$$g_j(i) = \begin{cases} 1, & \text{если услуга выполняется} \\ 0, & \text{если услуга не осуществляется} \end{cases}$$

$$f_k(i) = \begin{cases} 1, & \text{если услуга выполняется} \\ \frac{1}{Coeff_{ik}}, & \text{если услуга не осуществляется} \end{cases}$$

Где EP_i – это сумма значений за один i-раздел;

$Price\ of\ ep$ – это фиксированная цена за ep , которая зависит от состава команды, работающей над проектом;

$Coeff_{ik}$ – это коэффициентное значение услуги (при необходимости только в Backend) в i-разделе, которая в той или иной степени усложняет работу и которую трудно оценить по числовой шкале Фибоначчи;

ep_{ji} – это значение оценки услуги по предложенному прайсу.

Ориентировочная оценка проектов в Esscale Points

Мы разделили проекты на три категории по объему и сложности:

Проект "Бессерверное приложение"

Мобильное приложение не требующее написания собственного сервера.

- **Объем:** до 200 Esscale Points
- **Данный проект может включать:**
 - Проектирование и UI/UX дизайн
 - Реализация анимаций и переходов
 - Сложная логика и математические модели
 - Применение сложных алгоритмов
 - Работа с медиаконтентом
 - Тестирование CI/CD

Проект "Стандарт"

Полноценное приложение с серверной частью.

- **Объем:** от 250 до 340 Esscale Points
- **Данный проект может включать (все из Бессерверного приложения +):**
 - Регистрация/авторизация, личный кабинет
 - Push-уведомления
 - Серверная логика и аналитика
 - Внедрение сторонних API
 - Поддержание банковских транзакций
 - Простые чаты
 - Поддержание ~10000 RPS
 - Обработка статики

Проект "Pro / Комплексный"

Сложный продукт с нестандартной логикой, высокой нагрузкой или множеством интеграций.

- **Объем:** от 340 Esscale Points
- **Данный проект может включать (все из Стандарт +):**
 - Сложная backend архитектура
 - Масштабирование
 - Поддержание ~1000000 RPS
 - Сложная аналитика
 - Разработка собственных математических моделей
 - Интеграция или разработка ML и DL
 - Работа с медиаконтентом в реальном времени

Детализированный каталог услуг и оценка в Esscale Points

1. Backend Разработка (C++, Boost)

№	Услуга	Краткое описание	Оценка (EP)	Обоснование оценки
Архитектура и Бизнес-логика				
1.1	Проектирование масштабируемой и отказоустойчивой архитектуры	Создание фундамента сервиса, способного выдерживать высокие нагрузки, работать без сбоев и легко расширяться в будущем.	8-13	Сложность: Требуется глубокого опыта и стратегического мышления для предвидения узких мест. Риски: Ошибки на этом этапе наиболее дороги в исправлении.
1.2	Реализация основной бизнес-логики сервера	Написание ключевого кода, который решает задачи вашего	8-13	Объем: Обычно это самая большая часть кодовой базы.

		бизнеса (обработка заказов, управление пользователями, расчеты и т.д.) .		Сложность : Требует полного погружения в бизнес-процессы заказчика и их точной реализации в коде .
1.3.1	Реализация многопоточности сервера (коэффициент)	Внедрение механизмов для одновременной обработки тысяч запросов без блокировок, используя современные подходы C++ .	1,3	Сложность : Высокая когнитивная нагрузка . Риски : Потенциальные трудноуловимые ошибки (гонки данных, deadlock), которые критически влияют на стабильность .
1.3.2	Реализация асинхронности сервера (коэффициент)		1,3	
1.4	Рефакторинг существующего кода (коэффициент)	Модернизация кода для повышения его надежности, скорости работы, соответствия стандартам и упрощения дальнейшей поддержки .	1,5-3	Объем : Часто затрагивает большие участки кодовой базы . Риски : Необходимо глубоко понимать существующий код и не сломать рабочую логику .

1.5	Выбор и обоснование структур данных и алгоритмов	Подбор и реализация наиболее эффективных решений для высокой производительности и экономии ресурсов (память, CPU).	3	Сложность: Требует сильных теоретических знаний и умения применять их на практике для решения конкретных задач.
1.6	Внедрение API сторонних сервисов	Интеграция с внешними системами (платежные шлюзы, службы доставки, социальные сети) для расширения функционала.	5-8	Сложность: Зависит от качества документации внешнего API. Риски: Нестабильная работа внешнего сервиса может влиять на ваш продукт.
1.7	Обработка банковских транзакций	Реализация логики взаимодействия с платежными системами, обеспечение безопасности и атомарности операций.	8	Риски: Максимальная ответственность. Любая ошибка может привести к финансовым потерям. Требует строгого следования протоколам безопасности.

2.3	Обработка и парсинг JSON формата	Написание кода для чтения, валидации и преобразования данных в формате JSON, используемого в большинстве API.	3	Объем: Рутинная, но необходимая задача. Сложность возрастает при работе со сложными вложенными структурами и валидацией.
2.4.1	Обработка бинарных фреймов	Реализация работы с бинарными протоколами для высокопроизводительного обмена данными (игры, стриминг).	5	Сложность: Требует точной работы с байтами, битами и структурами данных. Отладка сложнее, чем с текстовыми форматами.
2.4.2	Обработка текстовых фреймов	Реализация работы с текстовыми протоколами.	3	Сложность: Верный парсинг формата строк. Обработка строковых данных.
2.5	Обработка статических и медиа файлов	Реализация отдачи файлов (изображения, документы, видео) с сервера, включая	5-8	Объем: Включает в себя настройку MIME-типов, кэширования, потенциально –

		оптимизацию и контроль доступа.		работу с большими файлами.
Работа с базами данных (БД)				
3.1	Подключение к БД и логика переключения	Настройка соединения с базой данных и реализация надежной логики автоматического восстановления связи при сбоях.	5	Риски: Нестабильное соединение с БД может парализовать весь сервис. Требуется тщательного тестирования.
3.2	Написание структуры данных БД (схемы, таблицы)	Проектирование логической структуры базы данных, определение таблиц, полей и связей между ними.	5	Сложность: Ошибки в проектировании схемы приводят к проблемам с производительностью и сложностью поддержки в будущем.
3.3	Написание процедур и триггеров БД	Создание хранимых процедур и триггеров для реализации сложной логики на стороне самой базы данных.	5	Сложность: Перенос логики в БД усложняет отладку и тестирование. Требуются глубокие знания конкретной СУБД.

		капчи или временной блокировки IP.		блокировки подозрительной активности.
5.6	Защита от Thundering Herd	Применение техник на уровне ОС или приложения для предотвращения "эффекта толпы" при пробуждении множества процессов.	5	Сложность: Низкоуровневая и специфическая задача, требующая глубокого понимания работы ОС и сетевого стека.

2. DevOps и Инфраструктура

№	Услуга	Краткое описание	Оценка (ЕР)	Обоснование оценки
Сборка и Контейнеризация				
1.1	Линковка зависимостей в CMake	Написание сборочных скриптов для корректного подключения всех необходимых библиотек и компиляции проекта.	1-3	Сложность: CMake имеет сложный синтаксис. Неправильная линковка приводит к ошибкам компиляции или выполнения.
1.2	Настройка Docker	Создание Dockerfile и docker-compose.yml для	3-5	Объем: Обеспечивает идентичность окружения на

3. Frontend Разработка (Web: React / Vue / Angular, JavaScript/TypeScript)

№	Услуга	Краткое описание	Оценка (ЕР)	Обоснование оценки
Проектирование и Архитектура				
1.1	Анализ UI/UX требований и макетов (Figma, Adobe XD)	Изучение дизайна и технического задания для планирования разработки, выявления сложных моментов и вопросов.	3	Сложность: Требуется внимательности и хорошей коммуникации с дизайнером/менеджером. Основа для всей последующей работы.
1.2	Проектирование компонентной архитектуры	Разделение интерфейса на логические, переиспользуемые компоненты (кнопки, формы, карточки) для чистоты кода и скорости разработки.	8	Сложность: Крайне важный этап. Ошибки в архитектуре приводят к сложностям в поддержке и масштабировании проекта.
1.3	Выбор и настройка архитектуры	Выбор и внедрение глобального хранилища данных	8	Сложность: Ключевое архитектурное решение.

		загрузки приложения.		и анализа производительности. Оказывает огромное влияние на UX.
4.2	Написание Unit и Integration тестов	Покрытие кода тестами для гарантии корректной работы компонентов и бизнес-логики, а также для предотвращения регрессий.	8	Объем: Написание тестов может занимать значительное время, но критически важно для долгосрочной поддержки больших проектов.

4. Frontend Разработка (Mobile: Flutter, Dart)

№	Услуга	Краткое описание	Оценка (ЕР)	Обоснование оценки
Проектирование и Архитектура				
1.1	Проектирование структуры виджетов и состояния приложения	Разработка архитектуры экранов и компонентов (виджетов),	8	Сложность: Фундамент приложения. Правильная иерархия

		определение потоков данных и состояний в приложении.		виджетов и управление состоянием напрямую влияют на производительность и простоту поддержки.
1.2	Выбор и обоснование архитектуры состояния (Provider, BLoC, Riverpod)	Выбор глобальной стратегии управления состоянием, которая будет использоваться во всем приложении.	8	Сложность: Ключевое решение, которое сложно изменить в будущем. Требуется опыта для выбора оптимального подхода под конкретный проект.
1.3	Разработка схемы навигации между экранами	Проектирование и реализация переходов между экранами, передача параметров, обработка deep linking.	5	Сложность: Может стать нетривиальной задачей в больших приложениях с вложенной навигацией и сложными сценариями.
Верстка и UI/UX Реализация				
2.1	Верстка экранов и	Создание интерфейса	5	Объем: Основная работа по

				риск platform-specific багов.
Тестирование и Развертывание				
4.1	Ручное и автоматизированное тестирование UI на разных устройствах и эмуляторах	Проверка работы приложения на реальных устройствах и эмуляторах для выявления багов, специфичных для платформ или моделей.	8	Объем: Очень трудоемкий процесс, необходимый для обеспечения качества. Автоматизация (Widget Tests, Integration Tests) снижает ручной труд в будущем.
4.2	Сборка и публикация приложения в App Store и Google Play	Подготовка сборок приложения, оформление страниц в сторках, прохождение ревью и публикация.	8	Сложность: Процесс связан с бюрократическими требованиями платформ, управлением сертификатами и профилями, что часто вызывает непредвиденные сложности.

Расчет стоимости проекта scandere

Проект: Рейтинговое приложение

Задача: Расчет KPI, возможность финансового управления предприятия через приложение, предоставление основной отчетности по деятельности.

1. Оценка задач в Esscale Points (EP):

Написание сервера без обработки изображений:

1.1. Backend Разработка (C++, Boost):

- Проектирование масштабируемой и отказоустойчивой архитектуры: 8 EP
- Реализация основной бизнес-логики сервера: 13 EP
- Реализация многопоточности сервера (коэффициент): 1,3
- Реализация асинхронности сервера (коэффициент): 1,3
- Выбор и обоснование структур данных и алгоритмов: 3 EP
- Обработка и хранение сетевых подключений по WebSocket: 5 EP
- Обработка и парсинг JSON формата: 3 EP
- Обработка текстовых фреймов: 3 EP
- Подключение к БД и логика переподключения: 5 EP
- Написание структуры данных БД: 5 EP
- Написание процедур и триггеров БД: 5 EP
- Оптимизация запросов и структуры БД: 5 EP
- Реализация системы репликации БД: 5 EP
- Кэширование запросов к БД: 5 EP
- Кэширование общей информации для масштабирования: 5 EP
- Реализация подписок Redis: 8 EP
- Система авторизации и регистрации пользователей: 8 EP
- Хэширование и безопасное хранение паролей: 1 EP
- Настройка SSL шифрования: 3 EP
- Защита от SQL-инъекций: 3 EP
- Защита от брутфорс-атак: 5 EP
- Защита от Thundering Herd: 8 EP

1.2. DevOps и Инфраструктура

- Линковка зависимостей в CMake: 3 EP
- Настройка Docker: 5 EP
- Настройка проксирования Nginx: 3 EP
- Развертывание единственного экземпляра Redis: 2 EP

Добавление обработки изображения:

1.3. Backend Разработка (C++, Boost):

- Обработка и хранение сетевых подключений по HTTP (Beast): 3 EP
- Обработка бинарных фреймов: 5 EP
- Обработка статических и медиа файлов: 8 EP

1.4. DevOps и Инфраструктура

- Настройка раздачи статики Nginx: 2 EP

Итого по Backend-разработке: 221,18 Esscale Points

1.5. Frontend Разработка (Mobile: Flutter, Dart)

- Проектирование структуры виджетов и состояния приложения: 8 EP
- Выбор и обоснование архитектуры состояния: 8 EP
- Разработка схемы навигации между экранами: 5 EP
- Верстка экранов и компонентов по макетам с использованием Flutter Widgets: 5 EP
- Планирование и реализация адаптивного дизайна: 8 EP
- Реализация сложных анимаций и переходов: 5 EP
- Реализация сложной клиентской логики: 13 EP
- Вызов API бэкенда (http, dio), обработка ответов: 8 EP
- Ручное и автоматизированное тестирование UI на разных устройствах и эмуляторах: 8 EP
- Сборка и публикация приложения в App Store и Google Play: 8 EP

Итого по Frontend-разработке: 76 Esscale Points

Итого Esscale Points: 297,18 Esscale Points

2. Расчет стоимости:

- Предположим, стоимость 1 Esscale Point для вашего проекта составляет 4 150 ₽.
- **Итоговая стоимость проекта: $297,18 \text{ EP} \times 4\,150 \text{ ₽/EP} = 1\,233\,297 \text{ ₽}$**

3. Расчет сроков:

- Наша средняя производительность (velocity) – около 15 EP в неделю.
 - Примерный срок выполнения: 297,18 EP / 15 EP в неделю \approx 20 недель.
-

Следующие шаги

Готовы перевести идею в конкретный план? Свяжитесь с нами для бесплатной консультации. Мы:

1. Обсудим цели и задачи вашего проекта.
2. Проведем первичную декомпозицию и оценку объема в Esscale Points.
3. Подготовим для вас детализированное коммерческое предложение.

Давайте создадим продукт, которым будем гордиться!