

Федеральное государственное автономное образовательное учреждение
высшего образования
Национальный Исследовательский Университет ИТМО

Лабораторная работа №1
Дисциплина: "Экономика программной инженерии"
Вариант: <https://garmanhomes.com>

Выполнил: Пушкин Антон Сергеевич
Факультет: Программной инженерии и компьютерной техники
Преподаватель: Машина Екатерина Алексеевна

Город Санкт-Петербург
2024 год

Описание лабораторной работы	2
Функциональные требования	2
Оценка трудоемкости разработки проекта наивным методом	5
Оценка трудоемкости разработки проекта методом PERT	7
Сетевая диаграмма	9
Метод функциональных точек	10
Определение типа оценки	10
Определение области оценки и границ продукта	10
Подсчет функциональных точек, связанных с данными	10
Подсчет функциональных точек, связанных с транзакциями	12
Определение суммарного количества не выровненных функциональных точек (UFP)	14
Определение значения фактора выравнивания (VAF)	14
Расчет количества выровненных функциональных точек (AFP)	15
COCOMO II	15
Оценка уровней факторов масштаба	15
Оценка уровней множителей трудоемкости	16
Оценка трудоемкости проекта	16
Use Case Points	17
Оценка веса прецедентов	17
Оценка веса акторов	17
Оценка веса технических факторов	17
Оценка веса факторов окружения	18
Подсчет UCP	19
Подсчёт фактора продуктивности (PF) на основе прошлого проекта	19
Список UseCase-ов	19
Оценка веса прецедентов	19
Оценка веса акторов	20
Оценка веса технических факторов	20
Оценка веса факторов окружения	21
Подсчет UCP	21
Подсчет трудоемкости проекта:	21
Анализ результатов	22

Описание лабораторной работы

1. Сформировать набор функциональных требований для разработки проекта.
2. Оценить трудоемкость разработки проекта наивным методом.
3. Оценить трудоемкость разработки проекта методом PERT (Project Evaluation and Review Technique). Нарисовать сетевую диаграмму взаимосвязи работ и методом критического пути рассчитать минимальную продолжительность разработки. Предложить оптимальное количество разработчиков и оценить срок выполнения проекта.

4. Оценить размер проекта методом функциональных точек, затем, исходя из предположения, что собранной статистики по завершенным проектам нет, рассчитать трудоемкость методом COSOMO II ([Обновленная таблица количества строк на точку для разных языков программирования](#))
5. Оценить размер проекта методом оценки вариантов использования (Use Case Points). Для расчета фактора продуктивности PF использовать любой свой завершённый проект с известными временными трудозатратами, оценив его размер методом UCP.
6. Сравнить полученные результаты и сделать выводы.

Функциональные требования

1. Система должна предоставлять возможность ознакомления с компанией и ее продукцией на нескольких уровнях вовлеченности, а именно:
 - 1.1. Ознакомительный уровень, в котором содержится:
 - 1.1.1. Страница с общей информацией о компании
 - 1.1.2. Блог
 - 1.1.3. Страница с описанием предоставляемых услуг
 - 1.1.4. Ссылки на социальные сети (Youtube, Instagram, Facebook)
 - 1.2. Уровень детального изучения, в котором содержится:
 - 1.2.1. Страница с отзывами
 - 1.2.2. Страница с примерами дизайнерских решений
 - 1.2.3. Страница с планом, по которому работает компания
 - 1.2.4. Модуль с часто задаваемыми вопросами
 - 1.3. Уровень полного погружения, в котором содержится:
 - 1.3.1. Политика конфиденциальности
 - 1.3.2. Страница с информацией о поставщике компании
 - 1.3.3. Страница с гарантийными условиями
2. Система должна предоставлять средства связи с компанией, такие как:
 - 2.1. Контактный номер телефона для моментального обращения
 - 2.2. Форма для оставления заявки с вопросом
3. Система должна предоставлять возможность взаимодействия для пользователей, которые:
 - 3.1. Ищут жилье и хотят получить информацию о доступных вариантах
 - 3.2. Ищут работу в компании и хотят получить доступ к вакансиям
4. Система должна предоставлять пользователям несколько уровней ознакомления с информацией на сайте:
 - 4.1. Краткий обзор в верхней части страницы – основные и ключевые сведения, позволяющие быстро ознакомиться с содержанием
 - 4.2. Подробная информация в средней части страницы – более детальные описания и дополнительные данные о продукции или услугах.

- 4.3. Полная информация в нижней части страницы – доступ ко всей доступной информации
- 5. Система должна предоставлять пользователям возможность создания и управления личным аккаунтом за счет:
 - 5.1. регистрации в системе, создавая новый аккаунт, с указанием необходимых данных
 - 5.2. авторизации для входа в существующий аккаунт
- 6. Система должна предоставлять возможность для быстрого перехода на главную страницу
- 7. Система должна предоставлять возможность просмотра всего содержимого сайта с мобильных устройств и поддерживать динамическое изменение содержимого
- 8. Система должна предоставлять возможность поиска дома, используя следующий функционал
 - 8.1. Поиск дома осуществляется при помощи:
 - 8.1.1. Системы быстрого поиска дома
 - 8.1.2. Поиска сообщества(район,улица)
 - 8.2. Поиск сообщества осуществляется при помощи:
 - 8.2.1. Фильтра, указывая регион и город
 - 8.2.2. Карты
 - 8.3. Страница сообщества должна:
 - 8.3.1. Иметь возможность сохранения в избранное
 - 8.3.2. Иметь возможность поделиться страницей
 - 8.3.3. Содержать информацию:
 - 8.3.3.1. Список домов, где каждый дома содержит информацию:
 - 8.3.3.1.1. Цена
 - 8.3.3.1.2. Площадь
 - 8.3.3.1.3. количество ванных комнат
 - 8.3.3.1.4. Количество машин вмещающихся в гараж
 - 8.3.3.2. Фотографии района
 - 8.3.3.3. Карта района, содержащая информацию о проданных, строящихся и доступных домах
 - 8.3.3.4. Карта, отображающая наличие школ, парков, магазинов, ресторанов, больниц в округе
 - 8.3.3.5. Интервал параметров домов, таких как:
 - 8.3.3.5.1. Площадь
 - 8.3.3.5.2. Количество спальных мест
 - 8.3.3.5.3. Количество ванных комнат
 - 8.3.3.5.4. Количество машин, вмещающихся в гараж
 - 8.4. Быстрый поиск предоставляет список домов на основе фильтров, таких как:
 - 8.4.1. Цена

- 8.4.2. Город
- 8.4.3. Тип дома (Townhouse, Single family)
- 8.4.4. Площадь
- 8.5. Страница дома должна:
 - 8.5.1. Иметь возможность сохранения в избранное
 - 8.5.2. Иметь возможность поделиться страницей
 - 8.5.3. Содержать информацию:
 - 8.5.3.1. Цена
 - 8.5.3.2. Адрес
 - 8.5.3.3. Площадь
 - 8.5.3.4. Количество ванных комнат
 - 8.5.3.5. Количество машин вмещающихся в гараж
 - 8.5.3.6. Фотографии дома
 - 8.5.3.7. План дома
 - 8.5.3.8. Тип здания
 - 8.5.3.9. Виртуальная модель дома
 - 8.5.3.10. Карта, отображающая наличие школ, парков, магазинов, ресторанов, больниц в округе
 - 8.5.3.11. Форма для оставления заявки с вопросом
- 9. Система должна предоставлять возможность построения модели дома, имея:
 - 9.1. Список моделей домов
 - 9.2. Страница модели, которая содержит:
 - 9.2.1. Информацию о модели, а именно:
 - 9.2.1.1. Название модели
 - 9.2.1.2. Площадь
 - 9.2.1.3. Количество спальных мест
 - 9.2.1.4. Количество ванных комнат
 - 9.2.2. Виртуальное трехмерное представление
 - 9.2.3. Фотографии моделей домов
 - 9.2.4. Типы моделей домов
 - 9.2.5. Существующий дом в продаже, соответствующий модели
 - 9.2.6. Конструктор модели, позволяющий выбрать внешнюю отделку дома

Оценка трудоемкости разработки проекта наивным методом

#	Название	Описание	Оценка, мин./чел. час	Оценка, макс./чел. .час	Оптимизированная оценка
1	Подготовка		58	130	95

1.1	Прототип сайта	Разработка сайта по поиску домов. Видео, фото и дополнительные данные домов имеются. Требуется прописать функциональные сценарии, прототип анимации и интерфейса и тд	50	100	75
1.2	Аренда хостинга	Поиск лучшего хостинга	8	30	20
2	Backend		200	430	320
2.1	Настройка БД	Проектирование структуры БД, выполнение CRUD-операций для хранения информации о домах, пользователях и отправленных ответах	40	120	80
2.2	Разработка API	Разработка API для обработки данных домов	80	120	100
2.3	Обработка запросов с форм, логика отправки ответов	Обработка фильтрации на сервере и формирование ответов с отфильтрованными данными. Работа с данными, полученными с форм, сохранение статуса, пользователя и его данных.	40	70	50
2.4	Отображение домов на карте	Обработка и предоставление данных для отображения на карте (генерация списка домов, их координат и связанная информация)	40	120	90
3	Frontend		180	510	330
3.1	Главная страница с отображением домов	Дизайн и верстка: создание пользовательского интерфейса, отображение элементов, карты, изображений и видео.	20	60	40
3.2	Второстепенные страницы	Дизайн и верстка для страниц сообществ, домов,	100	300	200

		быстрого поиска, макетов			
3.3	Отображение домов на карте	Интеграция карты на странице (с помощью JavaScript-библиотек для работы с Google Maps). Реализация отображения маркеров на карте, взаимодействие с пользователем.	30	60	40
3.4	Фильтры поиска и сортировки	Реализация пользовательского интерфейса для фильтров (поля, кнопки, выпадающие списки)	10	30	15
3.5	Формы для отправки ответов:	Разработка форм для ввода данных (HTML, CSS, JavaScript) и валидация на стороне клиента.	10	30	20
3.6	Регистрация/авторизация	Простая логика и соотв страница регистрации (ввод email и пароля, затемнение других окон, войти с помощью ВК и внутреннего профиля)	10	30	15
4	Тестирование		96	192	144
4.1	Модульное тестирование	Проверка бэка на надежность, на наличие SQL инъекций и корректную работу запросов к БД. Никакой более сложной логики не предусмотрено, как и других модулей, соотв требуется проверить основной функционал	40	80	60
4.2	Интеграционное	Тестирование общения	16	32	24

	тестирование	фронта и бэка, корректная обработка форм и тд			
4.3	Функциональное тестирование	Тестирование полных пользовательских сценариев	40	80	60
5	Релиз		45	95	70
5.1	Тестирование (alpha + beta)	Проверка основного функционала и корректного отображения сайта и его таблиц в различных верстках	40	80	60
5.2	Настройка	Настройка хоста под наши нужды, проверка общения клиента и сервера	5	15	10
Сумма часов			579	1357	959

Оценка трудоемкости разработки проекта методом PERT

PERT

#	Название	Оценка, мин./чел. час	Оценка, макс./чел. час	Оптимальная оценка	$E_i = \frac{(P_i + O_i + 4M_i)}{6}$	$CKO_i = \frac{(P_i - O_i)}{6}$
1.1	Прототип сайта	50	100	75	75	4,17
1.2	Аренда хостинга	8	30	20	19,7	2
2.1	Настройка БД	40	120	80	80	6,7
2.2	Разработка API	80	120	100	100	3,3
2.3	Обработка запросов с форм, логика отправки ответов	40	70	50	51,7	1,7
2.4	Отображение домов на карте	40	120	90	86,7	8,3
3.1	Главная страница с отображением домов	20	60	40	40	3,3

3.2	Второстепенные страницы	100	300	200	200	16,7
3.3	Отображение домов на карте	30	60	40	41,7	1,7
3.4	Фильтры поиска и сортировки домов	10	30	15	16,7	0,8
3.5	Формы для отправки ответов:	10	30	20	20	1,7
3.6	Регистрация/авто ризация	10	30	15	16,7	0,8
4.1	Модульное тестирование	40	80	60	60	3,3
4.2	Интеграционное тестирование	16	32	24	24	1,3
4.3	Функциональное тестирование	40	80	60	60	3,3
5.1	Тестирование (alpha + beta)	40	80	60	60	3,3
5.2	Настройка	5	15	10	10	0,8

$$E = \Sigma E_i = 962$$

$$CKO = \sqrt{\Sigma CKO_i^2} = 21,93$$

$$E_{95\%} = E + 2 * CKO = 1005,85, \text{ где}$$

E_i — оценка средней трудоемкости

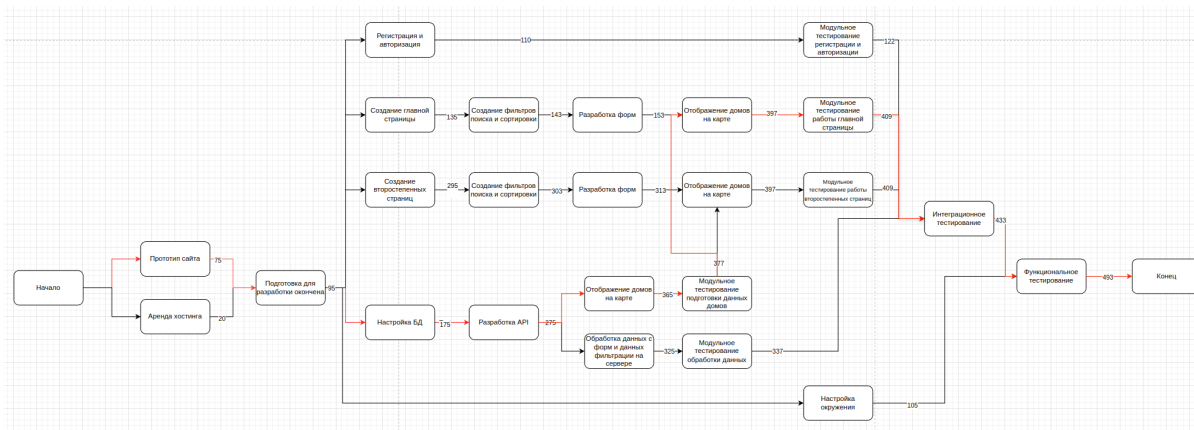
E — общая оценка статически независимых работ

CKO_i — среднеквадратичное отклонение

CKO — среднеквадратичное отклонение для оценки суммарной трудоемкости

$E_{95\%}$ — суммарная трудоемкость проекта

Сетевая диаграмма



<https://app.diagrams.net/?src=about#G1EZ2wkbmlguRV8wXZnF46Pk7QBDU5O0aA#%7B%22pageId%22%3A%22ARU6mwUwu6-PiVxgrLoo%22%7D>

Критический путь: 493 ч/ч

Чтобы параллельные задачи выполнялись без задержек и в срок необходим такой состав:

Команда:

- 2 Frontend- разработчика
- 2 Backend-разработчика
- 2 Тестировщика

Данное количество работников

Расчет времени разработки и общее время для завершения проекта:

Общее время разработки совпадает с критическим путем, так как большое количество задач идет в параллель с критическим путем, а при использовании оптимального количества работников.

Время разработки: 493 часа

Общее время(учитывая 8-ми часовой день): 62 рабочих дня

Метод функциональных точек

При анализе методом функциональных точек надо выполнить следующую последовательность шагов:

1. Определение типа оценки.
2. Определение области оценки и границ продукта.
3. Подсчет функциональных точек, связанных с данными.
4. Подсчет функциональных точек, связанных с транзакциями.
5. Определение суммарного количества не выровненных функциональных точек (UFP).
6. Определение значения фактора выравнивания (FAV).
7. Расчет количества выровненных функциональных точек (AFP).

Определение типа оценки

Продукт. Оценивается объем уже существующего и установленного продукта.

Определение области оценки и границ продукта

Все разрабатываемые функции. Рассчитываем все необходимые (реально используемые), а не дополнительные или только основные функции. Границы системы определены на UseCase диаграмме.

Подсчет функциональных точек, связанных с данными

DET (data element type) - неповторяемое уникальное поле данных.

- Имя человека (имя) - 1 DET
- Адрес человека (индекс, страна, город, улица, дом, корпус, квартира) - 7 DET

RET (record element type) - логическая группа данных

- адрес
- паспорт
- Телефонный номер

Оценка количества не выровненных функциональных точек, зависит от сложности данных, которая определяется на основании матрицы сложности

Матрица сложности:

	1-10 DET	11-20 DET	20+ DET
1 RET	Low	Low	Average
2-5 RET	Low	Average	High
6+ RET	Average	High	High

Оценка данных в не выровненных функциональных точках (UFP)

Сложность данных	Количество UFP (ILF)	Количество UFP (EIF)
Low	7	5
Average	10	7
High	15	10

№	Название	RET	DET	Сложность	UFP
1	Личный кабинет	Личная информация (1)	имя, фамилия, пол, дата рождения, телефон (7)	Low	7
2	Форма регистрации	Данные входа, личная информация (2)	Email, пароль, имя, фамилия, пол, дата рождения, телефон (9)	Low	7
3	Форма обратной связи	Контакты пользователя, сообщение (2)	Имя, email, сообщение (3)	Low	7
4	Поиска дома	Местоположение дома, характеристики дома, характеристики района(3)	Регион, город, район, цена, площадь, количество мест в гараже, количество ванных, количество спальных мест, тип дома, заведения рядом(10)	Average	10

Подсчет функциональных точек, связанных с транзакциями

Транзакция — это элементарный неделимый замкнутый процесс, представляющий значение для пользователя и переводящий продукт из одного консистентного состояния в другое.

В методе различаются следующие типы транзакций (Таблица 9):

- EI (external inputs) — внешние входные транзакции, элементарная операция по обработке данных или управляющей информации, поступающих в систему из вне.
- EO (external outputs) — внешние выходные транзакции, элементарная операция по генерации данных или управляющей информации, которые выходят за пределы системы. Предполагает определенную логику обработки или вычислений информации из одного или более ILF.

- EQ (external inquiries) — внешние запросы, элементарная операция, которая в ответ на внешний запрос извлекает данные или управляющую информацию из ILF или EIF.

Основные отличия между типами транзакций.

Функция	Тип транзакции		
	EI	ЕО	EQ
Изменяет поведение системы	Основная	Дополнительная	Не применима
Поддержка одного или более ILF	Основная	Дополнительная	Не применима
Представление информации пользователю	Дополнительная	Основная	Основная

Оценка сложности транзакции основывается на следующих ее характеристиках:

- FTR (file type referenced) — позволяет подсчитать количество различных файлов (информационных объектов) типа ILF и/или EIF модифицируемых или считываемых в транзакции.
- DET (data element type) — неповторяемое уникальное поле данных. Примеры. EI: поле ввода, кнопка. ЕО: поле данных отчета, сообщение об ошибке. EQ: поле ввода для поиска, поле вывода результата поиска.

Матрица сложности внешних входных транзакций (EI)

EI	1-4 DET	5-15 DET	16+ DET
0-1 FTR	Low	Low	Average
2 FTR	Low	Average	High
3+ FTR	Average	High	High

Матрица сложности внешних выходных транзакций и внешних запросов (EO & EQ)

EO & EQ	1-5 DET	6-19 DET	20+ DET
0-1 FTR	Low	Low	Average
2-3 FTR	Low	Average	High

4+ FTR	Average	High	High
--------	---------	------	------

Сложность транзакций в не выровненных функциональных точках (UFP)

Сложность транзакций	Количество UFP (EI & EQ)	Количество UFP (EO)
Low	3	4
Average	4	5
High	6	7

№	Название	Тип	FTR	DET	Сложность	UFP
1	Форма обратной связи	EI	1	4	Low	3
2	Форма регистрации	EI	1	11	Low	3
3	Просмотр профиля	EQ	1	8	Low	3
4	Просмотр видео	EQ	0	1	Low	3
5	Поиск дома	EQ	3	7	Average	4
6	Информация о доме	EO	3	3	Low	4

Определение суммарного количества не выровненных функциональных точек (UFP)

UFP: $31 + 20 = 51$

Определение значения фактора выравнивания (VAF)

Помимо функциональных требований на продукт накладываются общесистемные требования, которые ограничивают разработчиков в выборе решения и увеличивают сложность разработки. Для учета этой сложности применяется фактор выравнивания (VAF). Значение фактора VAF зависит от 14 параметров, которые определяют системные характеристики продукта:

№	Параметр	Вес (DI)
---	----------	----------

1	Обмен данными	2
2	Распределенная обработка данных	0
3	Производительность	0
4	Ограничения по аппаратным ресурсам	0
5	Транзакционная нагрузка	1
6	Интенсивность взаимодействия с пользователем	2
7	Эргономика	2
8	Интенсивность изменения данных	1
9	Сложность обработки	0
10	Повторное использование	1
11	Удобство инсталляции	0
12	Удобство администрирования	2
13	Портируемость	1
14	Гибкость	0
$TDI = \sum DI = 12$ $VAF = (TDI * 0.01) + 0.65 = 0.77$		

TDI - total degree of influence

VAF - фактор выравнивания производится

Расчет количества выровненных функциональных точек (AFP)

$$AFP = UPF \times VAF = 51 * 0.77 = 39.27$$

Метод анализа функциональных точек ничего не говорит о трудоемкости разработки оцененного продукта. Вопрос решается просто, если компания разработчик имеет собственную статистику трудозатрат на реализацию функциональных точек. Если такой статистики нет, то для оценки трудоемкости и сроков проекта можно использовать метод COCOMO II.

COCOMO II

Оценка размера программного продукта в KSLOC

Стек технологий:

- Angular (JS)
- Backend (Java)

Разделим функциональность между слоями: 3/4 - frontend и 1/4 - backend. Подсчитаем размер по KSLOC:

$$KSLOC = UFP * SIZE = (51 \times 3/4 \times 0.047) + (51 \times 1/4 \times 0.053) = 1.798 + 0.676 = 2.474$$

Оценка уровней факторов масштаба

- PREC - прецедентность, наличие опыта аналогичных разработок
- FLEX - гибкость процесса разработки
- RESL - архитектура и разрешение рисков
- TEAM - сработанность команды
- PMAT - зрелость процессов

Название фактора	Уровень фактора	Значение уровня
PREC	High	2.48
FLEX	High	2.03
RESL	Low	5.65
TEAM	Norminal	3.29
PMAT	Very Low	7.80

Оценка уровней множителей трудоемкости

Для предварительной оценки проекта необходимо оценить уровень семи множителей трудоемкости M:

- PERS - квалификация персонала
- RCPX - сложность и надежность продукта
- RUSE - разработка для повторного использования
- PDIF - сложность платформы разработки
- PREX - опыт персонала
- FCIL - оборудование
- CSED - требуемое выполнение графика работ

Название фактора	Уровень фактора	Значение уровня
PERS	Nominal	1.00
RCPX	Very Low	0.60
RUSE	Low	0.95
PDIF	Low	0.87
PREX	High	0.87
FCIL	Nominal	1.00
CSED	Nominal	1.00

Оценка трудоемкости проекта

$$PM = A \times SIZE^E \times \prod_{i=1}^n EM_i$$

$$A = 2,94$$

$$E = B + 0,01 \times \sum_{j=1}^5 SF_j$$

$$B = 0,91$$

- SIZE — размер продукта в KSLOC
- EM_i — множители трудоемкости
- SF_j — факторы масштаба
- n=7 — для предварительной оценки
- n=17 — для детальной оценки

$$E = 0.91 + 0.01 \times (2.48 + 2.03 + 5.65 + 3.29 + 7.80) = 1.1225$$

$$PM = 2.94 \times 2.474^{1.1225} \times (1.00 * 0.60 * 0.95 * 0.87 * 0.87 * 1.00 * 1.00) = 3.506 \text{ ч./мес.}$$
$$3.506 \text{ ч./мес.} = 560,96 \text{ ч./ч.}$$

Use Case Points

Оценка веса прецедентов

Сложность	Вес (UUCW)	Количество	Затраты
Low	5	20	100
Medium	10	5	50
High	15	6	90
Нескорректированный вес варианта использования (UUCW)			240

Оценка веса акторов

Сложность	Вес (UAW)	Количество	Затраты
Low	1	2	2
Medium	2	0	0
High	3	2	6
Масса актера без корректировки (UAW)			8

Оценка веса технических факторов

Фактор	Вес (W)	Номинальная стоимость(F)	Затраты
--------	---------	--------------------------	---------

Распределённость	1	0	0
Производительность	2	3	6
Эффективность для пользователя	3	4	12
Сложная внутренняя обработка	1	0	0
Повторное использование кода	2	1	2
Простота установки	1	1	1
Простота использования	3	3	9
Переносимость	1	0	0
Простота изменений	3	5	15
Многопоточность	1	0	0
Дополнительные возможности безопасности	1	1	1
Доступ к другим системам	1	2	2
Необходимы тренажеры 1 0 0 2 1 для пользователей	1	0	0
Общий технический фактор (TFactor)			48
$TCF = 0.6 + (TF/100)$			1.08

Оценка веса факторов окружения

Фактор	Вес (W)	Номинальная стоимость(F)	Затраты
Знаком с моделью проекта, которая используется	1.5	4	6.0
Опыт применения	0.5	3	1.5
Опыт в веб разработке	1.0	4	4.0
Возможность	0.5	2	1.0

ведущего аналитика			
Мотивация	1.0	2	2.0
Стабильные требования	1.5	2	1.5
Частичная занятость	-1.0	3	-3.0
Сложность языка программирования	-1.0	4	-4.0
Общий фактор окружающей среды (EFactor)			9.0
ECF = $1.4 + (-0.03 * EF)$			1.13

Подсчет UCP

$$UCP' = (UCW + UAW) * TCF * ECF = 303$$

Подсчёт фактора продуктивности (PF) на основе прошлого проекта

В качестве проекта был выбран проект по БЛПС

Список UseCase-ов

№	Сценарий
1	Регистрация
2	Авторизация
3	Поиск машины
4	Поиск продавца
5	Просмотр автомобиля
6	Изменить личные данные в профиле пользователя (Личный кабинет)
7	Создать объявление
8	Отредактировать объявление
9	Выставить рейтинг продавцу
10	Добавление машины в любимое
11	Подписаться на продавца
12	Оставить комментарий
13	Оценить комментарий
14	Поделиться автомобилем
15	Пожаловаться на пользователя

16	Пожаловаться на объявление
----	----------------------------

Оценка веса прецедентов

Сложность	Вес (UUCW)	Количество	Затраты
Low	5	12	60
Medium	10	4	40
High	15	0	0
Нескорректированный вес варианта использования (UUCW)			100

Оценка веса акторов

Сложность	Вес (UAW)	Количество	Затраты
Low	1	1	1
Medium	2	0	0
High	1	1	3
Масса актера без корректировки (UAW)			4

Оценка веса технических факторов

Фактор	Вес (W)	Номинальная стоимость(F)	Затраты
Распределённость	2	0	0
Производительность	1	1	1
Эффективность для пользователя	1	2	2
Сложная внутренняя обработка	1	1	1
Повторное использование кода	2	1	2
Простота установки	0.5	1	0.5
Простота использования	0.5	1	0.5
Переносимость	2	1	2
Простота изменений	2	3	6
Многопоточность	1	1	1

Дополнительные возможности безопасности	1	1	1
Доступ к другим системам	1	2	1
Необходимы тренажеры 1 0 0 21 для пользователей	1	0	1
Общий технический фактор (TFactor)			19
$TCF = 0.6 + (TF/100)$			0.79

Оценка веса факторов окружения

Фактор	Вес (W)	Номинальная стоимость(F)	Затраты
Знаком с моделью проекта, которая используется	1.5	2	3
Опыт применения	0.5	2	1
Опыт в веб разработке	1.0	5	5
Возможность ведущего аналитика	0.5	0	0
Мотивация	1.0	1	1
Стабильные требования	1.5	2	3
Частичная занятость	-1.0	3	-3
Сложность языка программирования	-1.0	4	-4
Общий фактор окружающей среды (EFactor)			15
$ECF = 1.4 + (-0.03 * EF)$			0.95

Подсчет UCP

$UCP' = (UCW + UAW) * TCF * ECF = 75$

Подсчет трудоемкости проекта:

Предыдущая работа была выполнена за 35 часов на 2-ух человек $PF = E/UCP = 0.94$ 25 UCP = 303 - для сайта

$E = PF * UCP = 285 \text{ ч/ч} + \text{работа с партнерами} + \text{заполнение всех данных} = 285 + 160 + 40 = 485$
ч/ч

Анализ результатов

Метод	Затраты (h-h)
Наивный	959
PERT	1006
Метод критического пути	493
COCOMO II	561
UCP	485

Значения наивного и pert методов вышли схожими, так как не имеют строгих критериев, параметров и основываются лишь на внутренней оценке и понимании. Другие 3 метода имеют показатели, меньшие почти в 2 раза. Они в свою очередь используют более глубокий анализ