

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования

Университет ИТМО

Факультет программной инженерии и компьютерной техники

Отчет по лабораторной работе №1 по дисциплине
“Экономика программной инженерии”
Вариант <https://gmz.su>

Выполнили: студенты группы
Р34131

Бусыгин Дмитрий Алексеевич и
Лазеев Сергей Максимович

Преподаватель:
Гаврилов Антон Валерьевич

Санкт-Петербург
2024

Задание:	3
Выполнение:	4
Оценка трудоемкости наивным методом.....	4
Оценка трудоемкости методом PERT.....	6
Метод критического пути.....	8
Метод функциональных точек.....	8
Определение типа оценки.....	8
Определение области оценки и границ продукта.....	8
Расчет функциональных точек, связанных с данными.....	8
Расчет функциональных точек транзакций.....	9
Определение суммы невыровненных ФТ (UFP).....	10
Определение фактора выравнивания (VAF).....	11
Расчет нормированного количества ФТ.....	11
COCOMO II.....	12
UseCase Points.....	12
UseCase диаграмма пользователя.....	12
Оценка веса прецедентов.....	12
Оценка веса акторов.....	13
Оценка веса технических факторов.....	13
Оценка веса факторов окружения.....	14
Вычисление UCP.....	14
Подсчет фактора продуктивности (PF).....	14
UseCase'ы прошлого проекта.....	14
Оценка веса прецедентов.....	15
Оценка веса акторов.....	15
Оценка веса технических факторов.....	15
Оценка веса факторов окружения.....	16
Вычисление UCP и PF.....	17
Применение фактора продуктивности (PF).....	17
Анализ результатов:	17

Задание:

Для выданного веб-проекта:

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 COCOMO 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 завершенный проект с известными временными трудозатратами, оценив его размер методом UCP.
6. Сравнить полученные результаты и сделать выводы.

Вариант: <https://gmz.su> - Гиагинский молзавод

Выполнение:

Оценка трудоемкости наивным методом

№	Название	Описание	Optimistic (h-h)	Pessimistic (h-h)	Optimal (h-h)
Проектирование					
1.1	Формализация требований заказчика	Необходимо разработать лендинг со статической информацией о “Гиагинском” молзаводе. Важно ограничить спектр работ и составить список функциональных и нефункциональных требований, согласованных с заказчиком	8	40	24
1.2	Составление политики конфиденциальности	Лендинг будет включать форму для обратной связи с персональными данными, поэтому необходимо сформировать правовую политику работы с этими данными	24	48	32
1.3	Выбор и аренда хостинга	Нужно оценить необходимую пропускную способность и выбрать оптимальный вариант хостинга по соотношению цена/качество	8	24	16
1.4	Выбор технологий	Проект представляет собой single-page application, но для качественной разработки и дальнейшей поддержки лучше выбрать фреймворк/библиотеку, нежели писать на plain js Остановимся на React	4	8	4
1.5	Дизайн	Необходимо составить полный дизайн будущего сайта в Figma	120	240	200
Frontend					
2.1	Навигационная панель	Необходимо схлапывать ее в “Бургер” при сжатии экрана, а также настроить анимацию	40	60	70
2.2	Главная	Адаптивная верстка главной страницы, плавные анимации при прокручивании	50	80	60
2.3	Карусель элементов	Навигация по карусели на главной,	8	32	20

		анимирование переходов			
2.4	О нас	Адаптивная верстка страницы “О нас”, плавные анимации при прокручивании	40	100	80
2.5	Как мы это делаем	Адаптивная верстка страницы “Как мы это делаем”, плавные анимации при прокручивании	30	60	40
2.6	Продукция	Создание компонента “продукт”, создание динамической конфигурации отображаемой продукции	40	120	80
2.7	Карточка продукта	Создание карточки продукта, создание динамической конфигурации информации о продукте	30	100	50
2.8	Где купить	Адаптивная верстка, динамическая конфигурация списка доступных магазинов	30	60	40
2.9	Внедрение виджета карты + обработка местоположения	Отображение карты (Yandex Maps) с возможностью взаимодействия, обработка запросов на поиск по магазинам, автоматическое определение местоположения клиента	20	48	30
2.10	Партнерам	Адаптивная верстка страницы, плавные анимации при прокручивании	50	80	60
2.11	Новости	Создание динамической конфигурации списка новостей	30	80	40
2.12	Новость	Создание компонента для отрисовки новости	20	60	30
2.13	Футер	Создание нижней части сайта с контактными данными и ссылками	20	40	24
2.14	Карта сайта	Создание карты сайта с доступными пользователям страницами	6	16	10
2.15	Задать вопрос	Создание формы для вопросов пользователей и реализация функционала отправки вопроса	20	40	24
2.16	Политика	Создание страницы с описанием	16	30	20

	конфиденциальности	политики конфиденциальности			
Тестирование					
3.1	Функциональное тестирование	Тестирование пользовательских сценариев на сайте	16	24	20
Релиз					
4.1	Настройка окружения	Размещение приложения на сервере, запуск	10	20	16
4.2	Настройка домена	Покупка и настройка доменного имени	10	20	16
4.3	Настройка SSL	Настройка https соединения с хостом	16	32	20
4.4	Проверка	Проверка сетевого доступа к хосту	5	15	10
Σ					1036

Оценка трудоемкости методом PERT

№	Название	Optimistic (h-h)	Pessimistic (h-h)	Optimal (h-h)	$E_i = \frac{(P_i + O_i + 4M_i)}{6}$	$CKO_i = \frac{(P_i - O_i)}{6}$
Проектирование						
1.1	Формализация требований заказчика	8	40	24	24,00	5,33
1.2	Составление политики конфиденциальности	24	48	32	33,33	4,00
1.3	Выбор и аренда хостинга	8	24	16	16,00	2,67
1.4	Выбор технологий	4	8	4	4,67	0,67
1.5	Дизайн	120	240	200	193,33	20,00
Frontend						
2.1	Навигационная панель	40	60	70	63,33	3,33
2.2	Главная	50	80	60	61,67	5,00
2.3	Карусель элементов	8	32	20	20,00	4,00
2.4	О нас	40	100	80	76,67	10,00

2.5	Как мы это делаем	30	60	40	41,67	5,00
2.6	Продукция	40	120	80	80,00	13,33
2.7	Карточка продукта	30	100	50	55,00	11,67
2.8	Где купить	30	60	40	41,67	5,00
2.9	Внедрение виджета карты + обработка местоположения	20	48	30	31,33	4,67
2.10	Партнерам	50	80	60	61,67	5,00
2.11	Новости	30	80	40	45,00	8,33
2.12	Новость	20	60	30	33,33	6,67
2.13	Футер	20	40	24	26,00	3,33
2.14	Карта сайта	6	16	10	10,33	1,67
2.15	Задать вопрос	20	40	24	26,00	3,33
2.16	Политика конфиденциальности	16	30	20	21,00	2,33
Тестирование						
3.1	Функциональное тестирование	16	24	20	20,00	1,33
Релиз						
4.1	Настройка окружения	10	20	16	15,67	1,67
4.2	Настройка домена	10	20	16	15,67	1,67
4.3	Настройка SSL	16	32	20	21,33	2,67
4.4	Проверка	5	15	10	10,00	1,67
$E = \Sigma_i = 1048.67$ $CKO = \sqrt{\Sigma CKO_i^2} = 34.3$ $E_{95\%} = E + 2 * CKO = 1117.26$						

E_i – оценка средней трудоемкости

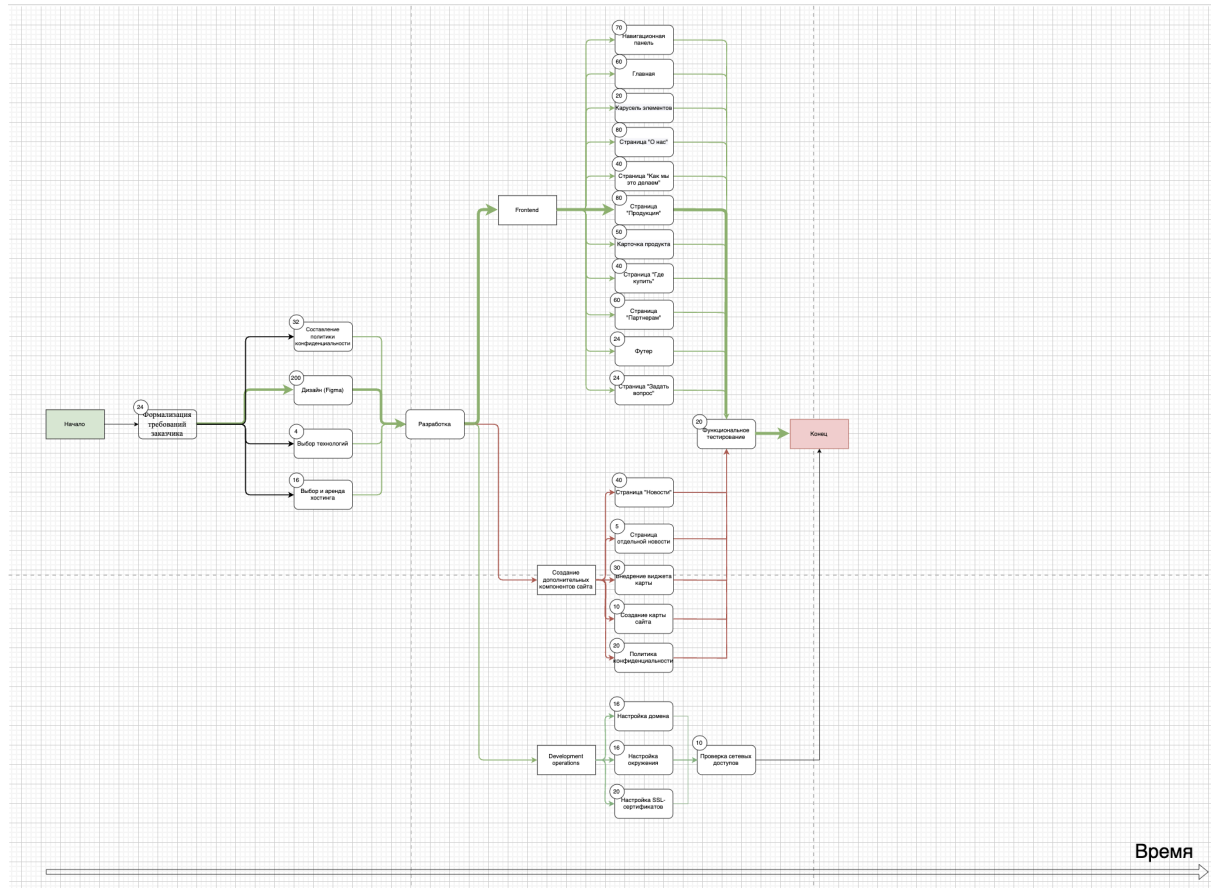
E – общая оценка статически независимых работ

CKO_i – среднеквадратичное отклонение

CKO – среднеквадратичное отклонение для оценки суммарной трудоемкости

$E_{95\%}$ – суммарная трудоемкость проекта (с вероятностью 95%)

Метод критического пути



Критический путь: 931 ч./ч.

Длинный путь: 1036 ч./ч.

Оценим время разработки по критическому пути:

Предполагаемая команда:

- 1) Менеджер
- 2) Frontend разработчик x4
- 3) Тестировщик
- 4) Дизайнер
- 5) DevOps-специалист

Предположим, что время в течение рабочего дня распределяется следующим образом: 5 часов разработки + 2 часа обсуждения + 1 час тех.перерыв

Таким образом получается 4 параллельные линии:

1:

На формализацию требований уйдет 24 ч./ч. или 5 дней

2:

На дизайн уйдет 200 ч./ч. или 40 дней

На выбор составление политики конфиденциальности 32 ч./ч. или 7 дней

На выбор технологий 4 ч./ч. или 1 день

На выбор и аренду хостинга 16 ч./ч. или 4 дня.

3:

На Frontend $548 \text{ ч./ч.} / 4 = 137 \text{ ч.}$ или $548 \text{ ч./ч.} / 5 / 4 = 28 \text{ дней}$ (с учетом 4-ых разработчиков в команде)

На DevOps 52 ч./ч. или 11 дней

4:

На функциональное тестирование 20 ч./ч. или 4 дня

На проверку сетевых доступов 10 ч./ч. или 2 дня

Время разработки: $24 + 200 + 137 + 20 = \mathbf{381 \text{ ч.}}$

Общее время: $5 + 40 + 28 + 4 = \mathbf{77 \text{ дней}}$

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

Оценивается сложность существующего веб-сайта

Определение области оценки и границ продукта

Принимаем во внимание полный конечный автомат переходов между страницами сайта, воспринимая каждый из них за action. Границы системы можно оценить ниже на UseCase-диаграмме

Расчет функциональных точек, связанных с данными

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

- Название сыра - 1 DET
- Сыр Адыгейский (название, описание, кбжу и т.д) - много DET

RET (record element type) - логическая группа данных

- Продукт
- Новость

№	Название	RET	DET	Сложность	UFP
1	Раздел Продукция	Продукт (1)	Название, описание, фото, кКал, жиры, белки, углеводы, фасовка, срок годности, температура, объем упаковки, вес упаковки, состав, теги (14)	low	7
2	Раздел Где купить	Магазин (1)	Логотип, ссылка, пермалинк на	low	7

			карте (3)		
3	Форма ОС	Личная информация (1)	Имя, телефон, почта, вопрос, согласие на ОД (5)	low	7
4	Новости	Новость (1)	Заголовок, фото, дата, текст (4)	low	7

Таблица оценки сложности ФТ, связанных с данными

	1-19 DET	20-50 DET	51+
1 RET	low (7)	low (7)	average (10)
2-5 RET	low (7)	average (10)	high (15)
6+ RET	average (10)	high (15)	high (15)

Расчет функциональных точек транзакций

- EI (external inputs) — внешние входные транзакции, элементарная операция по обработке данных или управляющей информации, поступающих в систему извне.
- 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.
- EQ (external inquiries) — внешние запросы, элементарная операция, которая в ответ на внешний запрос извлекает данные или управляющую информацию.
- FTR (file type referenced) — позволяет подсчитать количество различных файлов (информационных объектов) модифицируемых, или считываемых в транзакции.
- DET (data element type) — неповторяемое уникальное поле данных. Примеры. EI: поле ввода, кнопка. EO: поле данных отчета, сообщение об ошибке. EQ: поле ввода для поиска, поле вывода результата поиска.

№	Название	Тип	FTR	DET	Сложность	UFP
1	Просмотр новостей	EQ	1	1	low	3
2	Просмотр страницы новости	EQ	1	4	low	3
3	Заполнение формы ОС	EI	1	5	low	3

4	Просмотр продукции	EQ	1	1	low	3
5	Просмотр страницы продукта	EO	1	14	low	4
6	Просмотр информационных статических страниц	EQ	1	1	low	3
7	Просмотр раздела Где купить	EQ	1	1	low	3

Таблицы оценки сложности возможностей транзакций

EI & EQ	1-5 DET	6-19 DET	20+
0-1 FTR	low (3)	low (3)	average (4)
2 FTR	low (3)	average (4)	high (6)
3+ FTR	average (4)	high (6)	high (6)

EO	1-5 DET	6-19 DET	20+
0-1 FTR	low (4)	low (4)	average (5)
2 FTR	low (4)	average (5)	high (7)
3+ FTR	average (5)	high (7)	high (7)

Референс: <https://www.codeproject.com/Articles/18024/Calculating-Function-Points>

Определение суммы невыровненных ФТ (UFP)

$$UFP = 28 + 22 = 50$$

Определение фактора выравнивания (VAF)

VAF - коэффициент добавленного значения, основан на TDI (суммарной степени влияния 14 общих характеристик системы).

$TDI = \text{Сумма } (DI \text{ 14 общих характеристик системы})$, где DI обозначает степень влияния.

№	Параметр	DI (вес)
1	Передача данных	2
2	Распределенная обработка данных	0
3	Производительность	1
4	Интенсивно используемая конфигурация	0
5	Транзакционная нагрузка	0
6	Диалоговый ввод данных	3
7	Эффективность для пользователя	3
8	Оперативное обновление	1
9	Сложность обработки данных	0
10	Повторное использование	1
11	Легкость установки	0
12	Простота использования основных операций	0
13	Распространенность	1
14	Расширяемость	0
$TDI = \sum DI = 12$ $VAF = (TDI * 0.01) + 0.65 = 0.77$		

Расчет нормированного количества ФТ

$$AFP = UFP * VAF = 50 * 0.77 = 38.5$$

В дальнейшем нормированное количество функциональных точек может быть использовано для получения оценки количества строк исходного кода (**SLOC – Source Lines of Code**) в ПС при помощи **метода обратного запуска (Backfire Method)**. Мы используем модель COCOMO II для дальнейшей оценки.

COCOMO II

Оценка размера программного продукта в KSLOC

Стек технологий:

JavaScript

Название фактора	Уровень фактора	Значение уровня
PREC	LOW	4.96
FLEX	HIGH	2.03
RESL	VERY HIGH	1.41
TEAM	VERY HIGH	1.1
PMAT	VERY LOW	7.8

Название фактора	Уровень фактора	Значение уровня
PERS	NOMINAL	1.0
RCPX	EXTRA LOW	0.49
RUSE	LOW	0.95
PDIF	EXTRA LOW	n/a
PREX	LOW	1.22
FCIL	LOW	1.10
CSED	NOMINAL	1.14

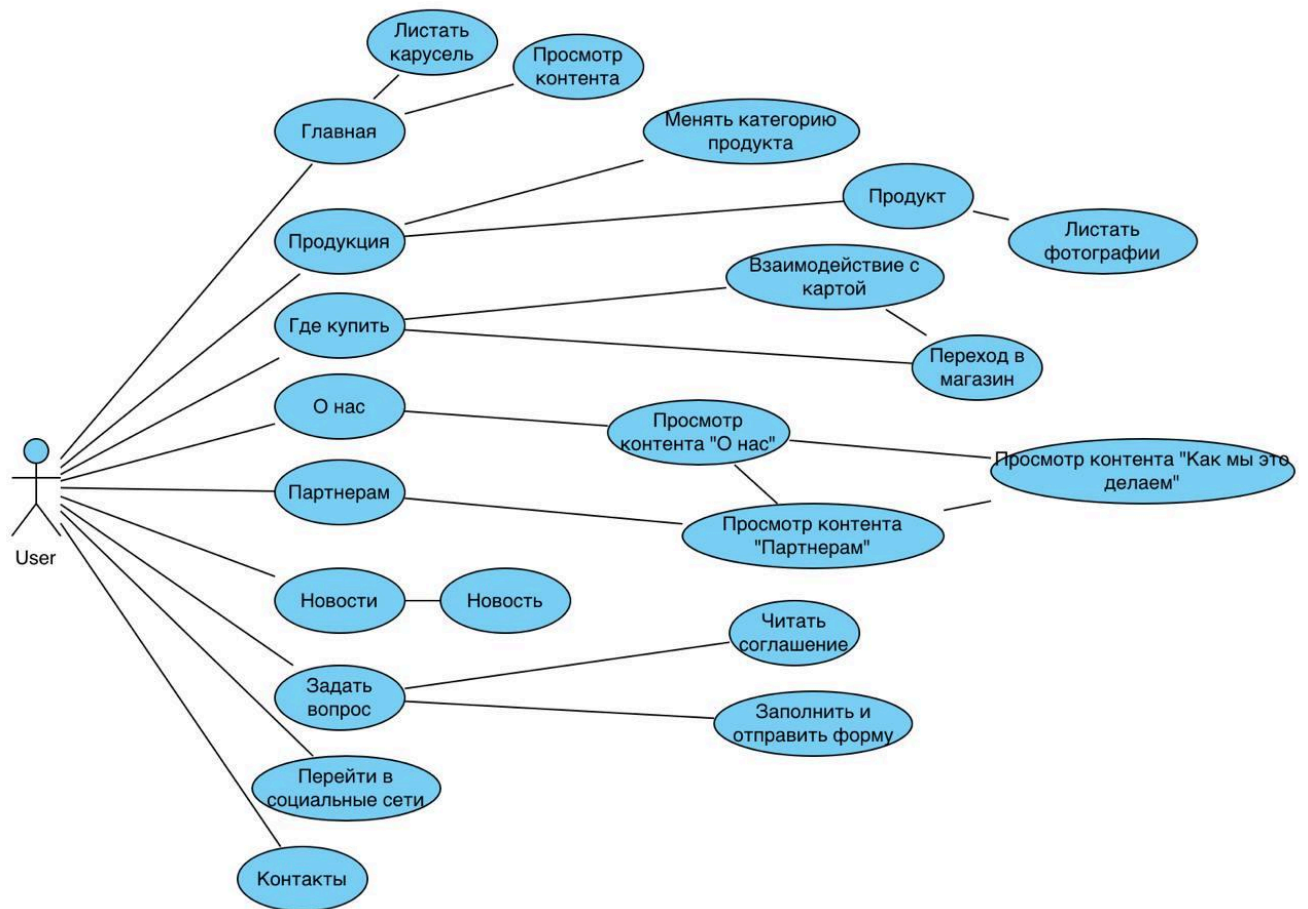
$$E = 0.91 + 0.01 * \sum SF_j = 0.91 + 0.01 * (4.96 + 2.03 + 1.41 + 1.1 + 7.8) = 1.083$$

$$PM = A * SIZE^E * \Pi (EM_j) = 2.94 * (50 * 0.053)^{1.083}$$

$$* (1 * 0.49 * 0.95 * 1.22 * 1.1 * 1.14) = 6 \text{ ч. /мес.} = 960 \text{ ч. /ч.}$$

UseCase Points

UseCase диаграмма пользователя



Оценка веса прецедентов

Сложность	Вес (UUCW)	Количество	Затраты
low	5	8	60
medium	10	6	60
high	15	8	120
UUCW=			240

Оценка веса акторов

Сложность	Вес (UAW)	Количество	Затраты
low	1	0	0
medium	2	1	2
high	3	0	0
UAW=			2

Оценка веса технических факторов

№	Фактор	Вес (W)	Значение (F)	Значение с учетом веса
T1	Распределенная система	2	0	0
T2	Высокая пропускная способность	1	3	3
T3	Работа конечных пользователей в режиме онлайн	1	2	2
T4	Сложная обработка данных	1	0	0
T5	Повторное использование кода	1	1	1
T6	Простота установки	0,5	1	0,5
T7	Простота использования	0,5	3	1,5
T8	Переносимость	2	0	0
T9	Простота внесения изменений	1	2	2
T10	Параллелизм	1	0	0
T11	Дополнительные требования безопасности	1	3	3
T12	Доступ к системе со стороны внешних пользователей	1	5	5
T13	Требования к обучению пользователей	1	0	0
TFactor				18

TCF=0.6+(TFactor/100)	0.78
-----------------------	------

Референс: <https://studfile.net/preview/9769909/page:11/>

Оценка веса факторов окружения

Фактор	Вес	Влияние	Затраты
Знакомство с моделью проекта	1.5	3	4,5
Частичная занятость	-1	0	0
Аналитические способности команды	0.5	2	1
Опыт применения подобных систем	0.5	4	2
Опыт в веб-разработке	1	4	4
Мотивация	1	2	2
Сложные технологии	-1	1	-1
Неизменность требований	2	2	4
EFactor			16.5
ECF=1.4+(-0.03*EFactor)			0.905

Вычисление UCP

$$UCP = (UUCW + UAW) * TCF * ECF = 171$$

Подсчет фактора продуктивности (PF)

Повторим процедуру оценки и вычислим UCP на основе прошлого проекта: курсовой работы по ИСБД

UseCase'ы прошлого проекта

№	Сценарий
1	Просмотр страницы с фильмами
2	Регистрация/авторизация
3	Просмотр страницы подписок
4	Просмотр страницы фильма
5	Оставить отзыв на фильм

6	Покупка подписки
7	Просмотр коллекций
8	Просмотр подборок по жанрам
9	Просмотр рекомендаций

Оценка веса прецедентов

Сложность	Вес (UUCW)	Количество	Затраты
low	5	5	25
medium	10	3	30
high	15	1	15
UUCW=			70

Оценка веса акторов

Сложность	Вес (UAW)	Количество	Затраты
low	1	0	0
medium	2	1	2
high	3	0	0
UAW=			2

Оценка веса технических факторов

№	Фактор	Вес (W)	Значение (F)	Значение с учетом веса
T1	Распределенная система	2	0	0
T2	Высокая пропускная способность	1	4	4
T3	Работа конечных пользователей в режиме онлайн	1	4	4
T4	Сложная обработка данных	1	3	3
T5	Повторное использование кода	1	2	2

T6	Простота установки	0,5	2	1
T7	Простота использования	0,5	4	2
T8	Переносимость	2	0	0
T9	Простота внесения изменений	1	2	2
T10	Параллелизм	1	3	3
T11	Дополнительные требования безопасности	1	4	4
T12	Доступ к системе со стороны внешних пользователей	1	5	5
T13	Требования к обучению пользователей	1	0	0
TFactor				30
TCF=0.6+(TFactor/100)				0.9

Оценка веса факторов окружения

Фактор	Вес	Влияние	Затраты
Знакомство с моделью проекта	1.5	4	6
Частичная занятость	-1	2	-2
Аналитические способности команды	0.5	1	0.5
Опыт применения подобных систем	0.5	4	2
Опыт в веб-разработке	1	4	4
Мотивация	1	3	3
Сложные технологии	-1	1	-1
Неизменность требований	2	0	0
EFactor			12.5
ECF=1.4+(-0.03*EFactor)			1.025

Вычисление UCP и PF

$$UCP = (UUCW + UAW) * TCF * ECF = 66$$

На проект было потрачено 50 чч, значит

$$PF = E/UCP = 50/66 = 0.75$$

Применение фактора продуктивности (PF)

Итого имеем:

$$E = UCP * PF + CONST = 171 * 0.75 + 24 + 32 + 16 + 4 + 200 = 405 \text{ чч}$$

где CONST - затраты персонала, отличного от разработчиков

Анализ результатов

Метод оценки	Затраты (чч)
Наивный	1036
PERT	1117.26
COCOMO II	960
UCP	405

Видим, что оценки первыми тремя методами весьма близки друг к другу, но по методу UCP значение получилось весьма заниженным. Это связано в первую очередь с тем, что он не учитывает ничего, кроме разработки. Еще одна причина - различие областей деятельности в оцениваемом проекте и проекте, взятом как пример, их, пожалуй, нельзя оценивать по одному и тому же набору метрик

Вывод:

В результате выполнения лабораторной работы мы сравнили методы оценки затрат человеко-часов на разработку заданного по варианту продукта. Разобрались какие величины учитывают эти методы и какие недостатки у них есть.

Можно сказать, что все методы привели нас к приблизительно одинаковым результатам. Кроме того мы практиковались разбивать поставленный проект на задачи и зоны ответственности.