Федеральное государственное автономное образовательное учреждение высшего образования

Университет ИТМО

Дисциплина: Экономика программной инженерии

Лабораторная работа 1

Вариант: https://www.splitwise.com/

Выполнил:
Колесников Никита Алексеевич

Группа: P34131

Преподаватель:
Машина Екатерина Алексеевна

2023 г.

Санкт-Петербург

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://www.splitwise.com/

Выполнение

Наивный метод

#	Название	Описание	Optimistic (h-h)	Pessimistic (h-h)	Optimal (h-h)
1	Подготовка		264	568	410
1.1	Прототип сайта	Разработка сайта под заказ для организации всех расходов и долговых расписок в одном месте, чтобы каждый мог видеть, кому он должен. Нужно создать официальный бренд вместе с дизайном. Требуется прописать	80	160	120

		функциональные пользовательские сценарии, прототип анимации и интерфейса.			
1.2	Выбор технологий	Важно определиться с используемыми технологиями. Frontend – ReactJS отлично подойдёт для создания отзывчивого современного UI. Backend – Java Spring приложение для организации биллинговых систем, PostgreSQL бд для хранения информации о пользователях	8	32	16
1.3	Разрешение на обработку персональных данных	Т. к. мы работаем с чувствительной информацией, соглашение необходимо для хранения и обработки персональных данных, по которым, например, будут списываться счета	168	336	250
1.4	Аренда хостинга	Существует множество сервисов предоставляющих возможности хостинга сайта, даже для крупных проектов вроде Splitwise.	8	40	24
2	Frontend		271	567	419
2.1	Главная	Адаптивная верстка, плавные анимации, современный дизайн, наглядная презентация возможностей splitwise	32	48	40
2.2	Регистрация	Возможность создать нового пользователя через	10	20	15

		интуитивно понятный дизайн с капчей. Указание email и пароля, а также имени.			
2.3	Авторизация	Возможность войти в существующего пользователя с помощью email и пароля для получения доступа к dashboard-ам	8	16	12
2.4	Dashboard	Адаптивный интерфейс, который позволяет добавить расходы и конкретных друзей в таблицу, в которой наглядно будет отображаться информация сколько и кому пользователь должен.	16	32	24
2.5	Недавняя активность	Интерфейс, отображающий недавнюю активность пользователя, его транзакции, которые он совершал недавно.	10	20	15
2.6	Приглашение друзей	Возможность пригласить друзей по почте, через Facebook или Twitter (или X). Добавленные друзья отображаются в списке друзей, также их можно объединять в группы для более удобного поиска.	5	15	10
2.7	Баланс	Отображение баланса пользователя с учётом всех долгов и расходов	10	30	20
2.8	Приближающиеся траты	Список приближающихся дедлайнов по оплате счетов, в виде списка кому	10	30	20

		сколько и когда должен вернуть.			
2.9	Тренды	Возможность увидеть тенденцию оплат счетов за месяц, сколько пользователь заплатил, сколько разделил, сколько оплат совершил, сколько оплат получил и общее изменение баланса. Также возможно развернуть список и увидеть графики изменения тенденций баланса.	40	100	70
2.10	Калькулятор справедливости	Отдельная страница, на которой пользователь может выбрать один из нескольких калькуляторов. Представляет из себя набор карточек с названиями и ссылкой на калькулятор.	8	12	10
2.11	Калькулятор разделения арендной платы	Позволяет в несколько шагов рассчитать справедливое распределение арендной платы в зависимости от разных факторов.	24	48	36
2.12	Калькулятор получения страховки арендатора	В зависимости от штата цена страховки может отличаться, в определенных штатах это может быть очень выгодно, в других очень дорого.	18	36	24
2.13	Калькулятор прогрессивного налога	Страница с регулируемым графиком прогрессивного налога, которая демонстрирует фактический доход после вычета налогов при разной	80	160	120

		зарплате			
3	Backend		400	666	533
3.1	Настройка и подключение БД	Создание таблиц и настройка триггеров для хранения пользователей, данных о долгах, датах списания долгов и т.д.	64	128	96
3.2	Авторизация пользователей	Возможность зарегистрироваться и авторизоваться в приложения для безопасного доступа к личным данным	48	96	72
3.3	Обработка запросов	Агрегирование актуальных данных о различных налогах, разделить расходы, записать долги, равное или неравное разделение, разделить по % или долям, рассчитать общий баланс, рекомендуемые выплаты, упростить долги, регулярные расходы	224	346	285
3.4	Облачные решения	Предоставить синхронизацию данных в облаке	64	96	80
4.	Мобильные приложения		400	600	500
4.1	Разработка приложения под Android	Реализовать аналогичный веб приложению интерфейс на android	200	300	250
4.2.	Разработка приложения под IOS	Реализовать аналогичный веб приложению интерфейс на IOS	200	300	250
5.	Тестирование		160	288	224
5.1	Модульное	Проверка бекэнда на	72	144	108

	тестирование	надежность, на наличие популярных уязвимостей и корректную работу сервисов.			
5.2	Интеграционное тестирование	Тестирование взаимодействия фронтенда и бекенда, тестирование обработки запросов	24	48	36
5.3	Функциональное тестирование	Тестирование полных пользовательских сценариев	64	96	80
6	Релиз		76	146	101
6.1	Тестирование (alpha+beta)	Проверка основного функционала и корректного отображения приложения в различных верстках	64	96	80
6.2	SSL сертификат	Self-Signed или отсутствие не подойдет, настройка https соединения	5	11	8
6.3	Выбор окружения	На выбранный хост добавить созданное react приложение, также запустить Spring приложение	12	24	18
6.4	Настройка	Настройка хоста под конечные нужды клиента, настройка общения клиента и сервера	5	15	10
Сумма			1571	2835	2187

PERT метод

#	Название	Optimistic (h-h)	Pessimistic (h-h)	Optimal (h-h)	$E_i = \frac{(P_i + O_i + 4M_i)}{6}$	$\text{СКО}_i = \frac{(P_i - O_i)}{6}$
1.1	Прототип сайта	80	160	120	120.00	13.33
1.2	Выбор технологий	8	32	16	17.33	4.00
1.3	Разрешение на обработку персональных данных	168	336	250	250.67	25.00
1.4	Аренда хостинга	8	40	24	24.00	5.33
2.1	Главная	32	48	40	40.00	2.66
2.2	Регистрация	10	20	15	15.00	1.66
2.3	Авторизация	8	16	12	12.00	1.33
2.4	Dashboard	16	32	24	24.00	2.66
2.5	Недавняя активность	10	20	15	15.00	1.66
2.6	Приглашение друзей	5	15	10	10.00	1.66
2.7	Баланс	10	30	20	20.00	3.33
2.8	Приближающиеся траты	10	30	20	20.00	3.33
2.9	Тренды	40	100	70	70.00	10.00
2.10	Калькулятор справедливости	8	12	10	10.00	0.66
2.11	Калькулятор разделения арендной платы	24	48	36	36.00	4.00
2.12	Калькулятор получения страховки арендатора	18	36	24	25.00	3.00
2.13	Калькулятор	80	160	120	120.00	13.33

	прогрессивного налога					
3.1	Настройка и подключение к бд	64	128	96	96.00	10.66
3.2	Авторизация пользователей	48	96	72	72.00	8.00
3.3	Обработка запросов	224	346	285	285.00	20.33
3.4	Облачные решения	64	96	80	80.00	5.33
4.1	Разработка под Android	200	300	250	250.00	16.66
4.2	Разработка под IOS	200	300	250	250.00	16.66
5.1	Модульное тестирование	72	144	108	108.00	12.00
5.2	Интеграционное тестирование	24	48	36	36.00	4.00
5.3	Функциональное тестирование	64	96	80	80.00	5.33
6.1	Тестирование (Alpha+beta)	64	96	80	80.00	5.33
6.2	SSL сертификат	5	11	8	8.00	1.00
6.3	Выбор окружения	12	24	18	18.00	2.00
6.4	Настройка	5	15	10	10.00	1.66

$$E = \sum E_i = 2202$$

$$\text{СКО} = \sqrt{\sum \text{СКО}_i^2} = 204{,}24$$

$$E_{95\%} = E + 2 * \text{СКО} = 2610{,}48$$

$E_i -$ оценка средней трудоёмкости

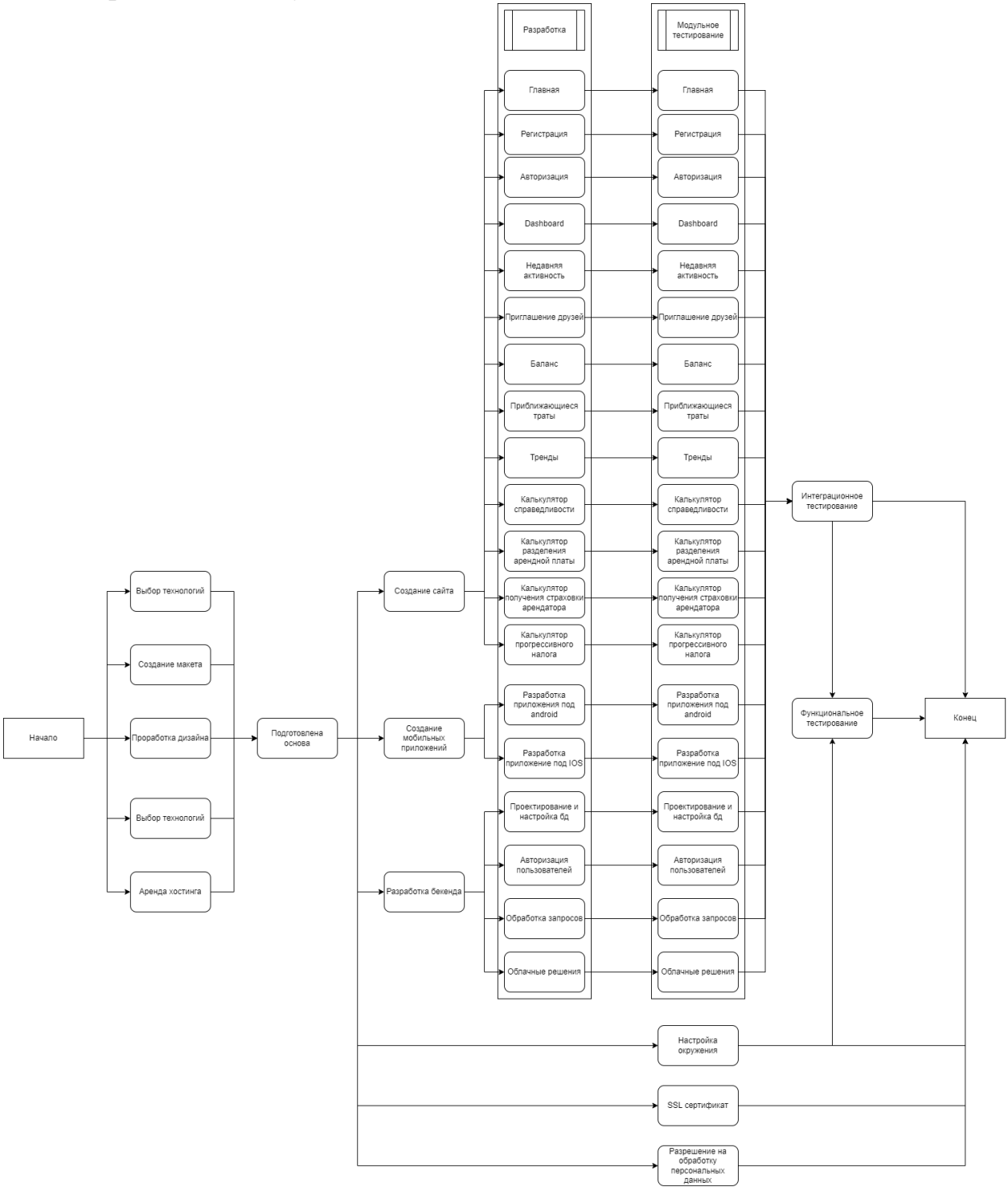
$E -$ общая оценка статически независимых работ

$\text{СКО}_i -$ среднеквадратичное уравнение

$\text{СКО} -$ среднеквадратичное уравнение оценки суммарной трудоемкости

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

Метод критического пути.



Критический путь:

1642 ч./ч.

При ориентире на минимальное время разработки получается, что для выполнения необходимо 1642 ч.

Команда:

- 1 Аналитик
- 2 Frontend разработчика
- 2 Backend разработчика
- 1 Android разработчик
- 1 IOS разработчик
- 2 Тестировщика
- Рабочий день считаем 8 часов

Таким образом такая команда может выполнить проект за

- Frontend с модульным тестированием: 240 часов (30 рабочих дней)
- Backend с модульным тестированием: 296.5 часов (38 рабочих дней)
- Мобильные приложения: 250 часов (32 рабочих дня)
- Интеграционное тестирование: 18 часов (3 рабочих дня)
- Сбор данных: 250 часов (32 рабочих дня)

Рассчитать время разработки
Frontend и Backend создаются параллельно, мобильные приложения могут создаваться на базе Frontend-а, но они также могут разрабатываться параллельно. В реальности обычно мобильные приложения отстают от Frontend-а на 20–30. Часов после чего происходит общее интеграционное тестирование. Сбор данных также может идти параллельно

Время разработки: 296.5 + 30 + 18 + 250 = 594.5 часа

Общее время: 38 + 4 + 3 = 45 рабочих дней

Метод функциональных точек

При анализе методом функциональных точек надо выполнить следующую последовательность шагов:

- Определение типа оценки
- Определение области оценки и границ продукта
- Подсчет функциональных точек, связанных с данными
- Подсчет функциональных точек, связанных с транзакциями
- Определение суммарного количества не выровненных функциональных точек (UFP)
- Определение значения фактора выравнивания (FAV)
- Расчет количества выровненных функциональных точек (AFP)

Определение типа оценки

Продукт. Оценивается объем уже существующего и установленного продукта.

Определение области оценки и границ продукта

Все 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

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

№	Название	RET	DET	Сложность	UFP
1	Авторизация	Данные входа, личная информация (2)	Телефон (1)	Low	7
2	Регистрация	Данные входа, личная информация (2)	Имя, фамилия, пол, email, дата рождения, город, телефон (8)	Low	7
3	Приглашение друзей	Контакты пользователя, сообщения (2)	Имя, email, сообщение (3)	Low	7
4	Калькуляторы справедливости	Личная информация, информация о доходе, информация о месте жительства (3)	Имя, фамилия, отчество, дата рождения, информация о доходе, город (7)	Low	7

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.

- **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.

- **EQ** (external inquiries) — внешние запросы, элементарная операция, которая в ответ на внешний запрос извлекает данные или управляющую информаци
- **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

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

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

№	Название	Тип	FTR	DET	Сложность	UFP
1	Авторизация	EI	1	1	Low	3
2	Регистрация	EI	1	8	Low	3
3	Приглашение друзей	EO	2	3	Low	3
4	Калькулятор справедливости	EQ	1	7	Low	3
5	Калькулятор арендной платы	EQ	1	7	Low	3
6	Калькулятор	EQ	1	6	Low	3

	получения страховки					
7	Калькулятор прогрессивного налога	EQ	1	7	Low	4
8	Тренды	EI	0	1	Low	4
9	Приближающиеся траты	EQ	1	2	Low	3

Определение суммарного количества не выровненных функциональных точек (UFP)

UFP=51+29=80

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	DI
1	Обмен данными	2
2	Распределенная обработка данных	0
3	Производительность	0
4	Ограничения по аппаратным ресурсам	0
5	Транзакционная нагрузка	0
6	Интенсивность взаимодействия с пользователем	2
7	Эргономика	3
8	Интенсивность изменения данных	1
9	Сложность обработки	1
10	Повторное использование	1
11	Удобство инсталляции	0
12	Удобство администрирования	2
13	Портируемость	1
14	Гибкость	1
$$TDI = \sum DI = 14$$ $$VAF = (TDI * 0.01) + 0.65 = 0.79$$		

DI (degree of influence)
TDI (total degree of influence)

Расчет количества выровненных функциональных точек (AFP)

$AFP = UPF \times VAF$ = 80 * 0. 79 = 63.2

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:

- React (TS / JS)
- Backend (Java)

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

Таблица коэффициентов

$KSLOC = UPF * SIZE$ = (80 × 3/4 × 0. 053) + (80 × 1/4 × 0. 050) = 3.18 + 1= 4.18

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	High	2.19
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	Уровень	Значение
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

$$PM = A * SIZE^E * \prod_{i=1}^{n} EM_i \qquad A = 2.94$$

$$E = B + 0.01 * \sum_{j=1}^{5} SF_j \qquad B = 0.91$$

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

E = 0. 91 + 0. 01 × (2. 48 + 2. 03 + 5. 65 + 2. 19 + 7. 80) = 1. 1115

PM = 2. 94 × $4.18^{1.1115}$ × (1. 00 * 0. 60 * 0. 95 * 0. 87 * 0. 87 * 1. 00 * 1. 00) ≈ 6. 21867 ч. /мес

6.21867 ч. /мес = 994.98 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 (AUW)	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
Распределенность	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	1	1	1

возможности безопасности			
Доступ к другим системам	1	2	2
Необходимы тренажеры для пользователей	1	0	0
Общий технический фактор (TFactor)			48
TCF = 0.6 + (TF/100)			1.08

Подсчет UCP

UCP` = (UCW + UAW) * TCF * ECF = 303

Подсчёт фактора продуктивности (PF) на основе прошлого проекта

В качестве примера я выбрал проект по БЛПС, в которую входит выполнение всех лабораторных работ для двух человек.

Список UseCase-ов:

#	**Сценарий**
1	Регистрация
2	Авторизация
3	Поиск статьи
4	Поиск автора
5	Просмотр статьи
7	Создать статью
8	Отредактировать статью
9	Выставить рейтинг статье
12	Оставить комментарий
13	Оценить комментарий
14	Поделиться статьей

Оценка веса прецедентов

Сложность	**Вес (UUCW)**	**Количество**	**Затраты**
Low	5	12	60
Medium	10	4	40
High	15	0	0

Оценка веса акторов

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

Оценка веса технических факторов

Фактор	Вес (W)	Номинальная стоимость(F)	Затраты
Распределенность	2.0	0	0.0
Производительность	1.0	1	1.0
Эффективность для пользователя	1.0	2	2.0
Сложная внутренняя обработка	1.0	1	1.0
Повторное использование кода	2.0	1	2.0
Простота установки	0.5	1	0.5
Простота использования	0.5	1	0.5
Переносимость	2.0	1	2.0
Простота изменений	2.0	3	6.0
Многопоточность	1.0	1	1.0
Дополнительные возможности безопасности	1.0	1	1.0
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

UCP = 303 - для нашего сайта

E = PF * UCP = 285 ч/ч + юр работа + работа с партнерами + заполнение всех данных = 285 + 250 + 160 + 40 = 735 ч/ч

Анализ результатов

Метод	Затраты (h-h)
Наивный	2187
PERT	2610,48
Функциональных точек	1642
COCOMO II	995
UCP	735

Можно увидеть, что COCOMO II и наивный метод демонстрируют почти идентичные результаты. Это неудивительно, потому что в COCOMO II мы учитываем различные реальные параметры, которые мы анализируем на основе личного опыта, что похоже на наивный метод. Они схожи, потому что оба основаны на наших предположениях и реальном опыте.

В случае использования метода UCP время расчета оказалось ниже ожидаемого. Возможно, это произошло из-за неточного расчета для уже завершенного проекта. Вероятно, мы неправильно оценили количество UCP, содержащихся в нем, или ошибочно указали слишком короткое время его реализации. Также стоит учесть, что лабораторные работы значительно отличаются от реальной разработки, поэтому значительные различия в проектах могут также повлиять на результаты.

Вывод

В ходе выполнения лабораторной работы я познал на собственном опыте, что роль менеджера включает в себя задачу адекватного определения функций проекта и оценки времени, необходимого для его выполнения. Я почувствовал, что оценка является одной из сложных задач, с которой сталкиваются в программировании, так же, как и определение реалистичных сроков выполнения.