

УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 Программная инженерия

Дисциплина «Экономика программной инженерии»

Лабораторная работа № 1

Выполнили: Копылова Полина Р34202

Савинцев Артем Р34202

Проверила: Машина Екатерина Алексеевна

г. Санкт-Петербург

2024

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
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.

Выданный веб проект - <https://www.postgresqltutorial.com/>

Выполнение:

Наивный метод

#	Название	Описание	Optimistic (h-h)	Pessimistic (h-h)	Optimal (h-h)
1	Подготовка		98	186	137
1.1	Прототип сайта	Разработка прототипа ресурса по PostgreSQL, которая включает в себя основные разделы - статьи, поиск, уроки	60	120	90

1.2	Выбор технологий	Frontend: Javascript дает нам достаточно функционала для реализации простого и удобного для пользователя дизайна. Backend: PostgreSQL используется как основная база данных для хранения контента	8	16	12
1.3	Настройка сервера для размещения сайта	Подготовка и базовая настройка серверного массива для размещения на нем сайта	30	50	35
2	Frontend		190	370	280
2.1	Главная страница	Создание адаптивной главной страницы с описанием уроков	20	40	30
2.2	Поиск по статьям	Реализация поиска по статьям и урокам, фильтры по темам	30	50	40
2.3	Статьи	Верстка статей с синтаксисом SQL	100	200	150
2.4	Продвинутые уроки	Реализация отдельных страниц с тематическими статьями для различных концептов: индексов, триггеров,	20	40	30

		представлений, администрирование базы данных			
2.5	Информация о сайте	Набор страниц предоставляющих доступ к информации об организации, форма для обратной связи, ссылки на другие ресурсы компании, и другое	20	40	30
3	Backend		80	170	130
3.1	Создание базы данных	Создание и настройка базы данных для хранения уроков, статей и т.д.	40	80	60
3.2	Работа с API	Разработка REST API для взаимодействия фронтенда и базы данных	40	90	70
4	Тестирование		80	160	120
4.1	Модульное тестирование	Тестирование серверной части	30	60	45
4.2	Интеграционное тестирование	Проверка взаимодействия фронтенда и бэкенда, правильности работы API	20	40	30
4.3	Функциональное тестирование	Тестирование пользовательских сценариев - поиск	30	60	45

		статей			
5	Релиз		36	72	53
5.1	Тестирование	Проверка корректного отображения всех страниц	20	40	30
5.2	SSL Certificate	Настройка SSL сертификата для обеспечения безопасного соединения	5	10	7
5.3	Выбор окружения	Выгрузка скомпилированной html страницы на сервер	4	8	6
5.4	Настройка	Финальная настройка хоста под наши нужды, проверка взаимодействия клиента и сервера	7	14	10
Σ			484	958	720

PERT метод

#	Название	Optimistic (h-h)	Pessimistic (h-h)	Optimal (h-h)	$E_i = \frac{(P_i + O_i + 4M_i)}{6}$	$CKO_i = \frac{P_i - O_i}{6}$
1.1	Прототип сайта	60	120	90	90,00	10,00
1.2	Выбор технологий	8	16	12	12,00	1,33
1.3	Настройка сервера для размещения сайта	30	50	35	36,67	3,33
2.1	Главная страница	20	40	30	30,00	3,33
2.2	Поиск по статьям	30	50	40	40,00	3,33
2.3	Статьи	100	200	150	150,00	16,67
2.4	Продвинутые уроки	20	40	30	30,00	3,33

2.5	Информация о сайте	20	40	30	30,00	3,33
3.1	Создание базы данных	40	80	60	60,00	6,67
3.2	Работа с API	40	90	70	68,33	8,33
4.1	Модульное тестирование	30	60	45	45,00	5,00
4.2	Интеграционное тестирование	20	40	30	30,00	3,33
4.3	Функциональное тестирование	30	60	45	45,00	5,00
5.1	Тестирование	20	40	30	30,00	3,33
5.2	SSL Certificate	5	10	7	7,17	0,83
5.3	Выбор окружения	4	8	6	6,00	0,67
5.4	Настройка	7	14	10	10,17	1,17

$$E = \sum E_i = 720.33$$

$$CKO = \sqrt{\sum CKO_i^2} = 20.72$$

$$E_{95\%} = E + 2 * CKO = 761.77$$

E_i – оценка средней трудоемкости

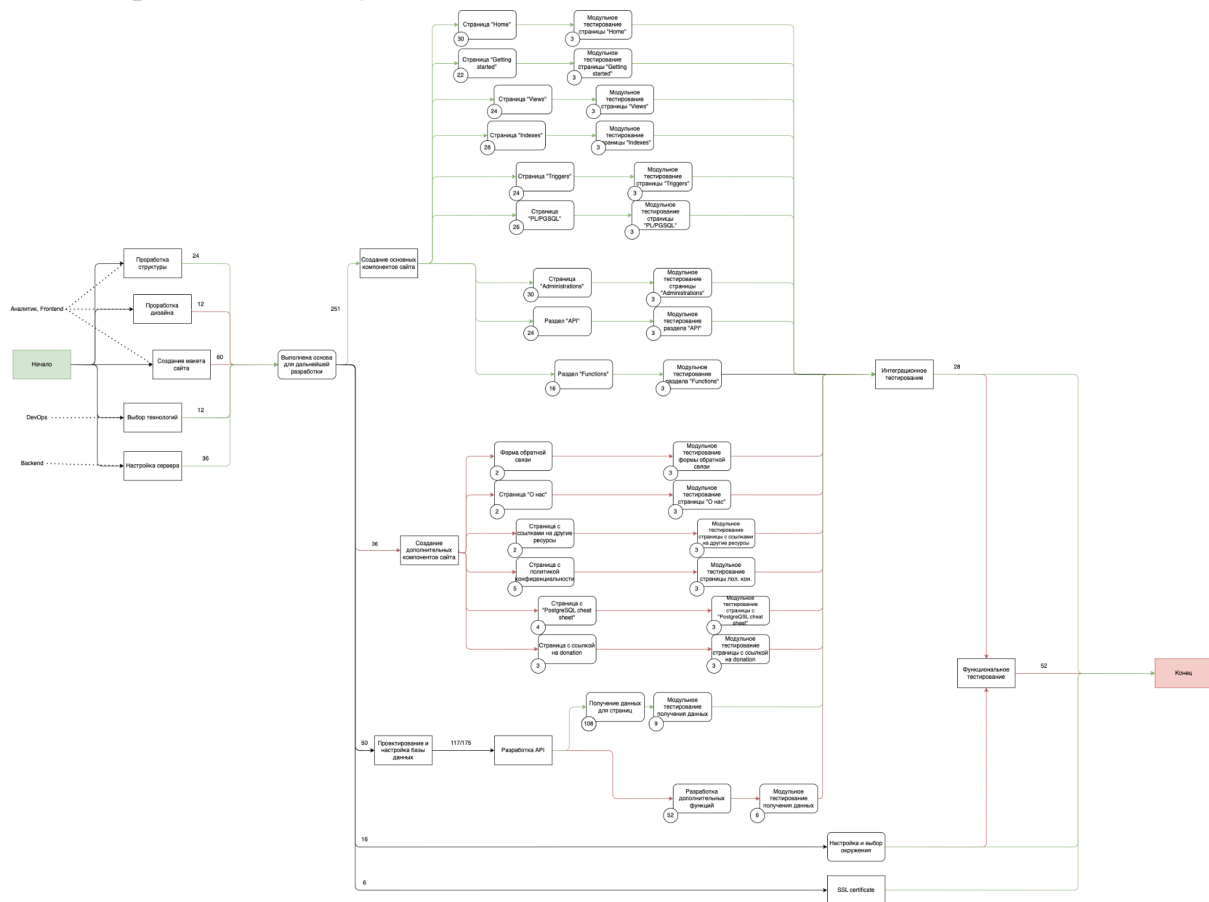
E – общая оценка статически независимых работ

CKO_i – среднеквадратичное отклонение

CKO – среднеквадратичное отклонение для оценки суммарной трудоемкости

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

Метод критического пути



Критический путь: 540 ч./ч.

Длинный путь: 758 ч./ч.

Выполнение проекта: при ориентире на минимальное время разработки (критический путь) получаем, что для выполнения нам необходимо 540 ч./ч.

Команда:

Аналитик - 1х

Frontend-разработчик - 2х

Backend-разработчик - 2х

Тестировщика - 1х

За рабочий день принимаем 8 часов - 6 ч. 30 мин. работы, 45 мин. обед, 45 мин. технический перерыв.

Таким образом, сроки выполнения проекта:

Frontend (с модульным тестированием): 251 час (39 рабочих дней)

Backend (с модульным тестированием): 237 часов (37 рабочих дней)

Интеграционное тестирование: 25 часов (4 рабочих дня)

Релиз: 6 часов (1 раб. день)

Рассчитаем общее время разработки и завершения проекта: так как backend и frontend можно делать одновременно, время разработки уменьшится.

Время разработки: $251 + 25 = 276$ часов

Общее время: $39 + 4 + 1 = 44$ рабочих дня

Метод функциональных точек

Определение типа оценки: Продукт оценивается по объему уже существующего и установленного продукта.

Определение области оценки и границ продукта: Оцениваются все функции системы, включая все реально используемые (не только основные, но и дополнительные). Границы системы определены на диаграмме вариантов использования (Use Case).

Подсчет функциональных точек, связанных с данными: DET (тип элемента данных) — уникальное поле данных, которое не повторяется.

- Имя человека (имя), email (электронный адрес) — 1 DET.
- Адрес человека (индекс, страна, город, улица, дом, корпус, квартира) — 7 DET.

RET (тип элемента записи) — логически сгруппированные данные:

- Адрес

Количество RET	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	Форма обратной связи	Контакты пользователя, сообщение (2)	Имя, Email, сообщение (3)	Low	7

Подсчет функциональных точек, связанных с транзакциями

- 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) — неповторимое 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 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 FTR	Low	Average	High
3 FTR	Average	High	High

№	Название	Тип	FTR	DET	Сложность	UFP
1	Форма обратной связи	EI	1	4	Low	3
2	Поиск	EQ	3	1	Low	3
3	Просмотр статических страниц	EO	0	1	Low	3
4	Изменение темы	EI	1	1	Low	3
5	Просмотр Home статей	EO	0	4	Low	4
6	Просмотр Function статей	EO	0	3	Low	4
7	Просмотр "Advanced" статей	EO	0	4	Low	4
8	Просмотр API статей	EO	0	4	Low	4
9	Просмотр Administration статей	EO	0	4	Low	4
10	Просмотр статей "Getting Started"	EO	0	4	Low	4
11	Просмотр статей "PL/PGSQL"	EO	0	4	Low	4

Подсчет суммарного количества UFP (не выровненных функциональных точек):

$$UFP = 7 + 40 = 47$$

Определение значения VAF (фактора выравнивания):

Помимо функциональных требований продукт также имеет общесистемные требования, которые влияют на сложность разработки и ограничивают выбор решений. Чтобы учесть добавленную сложность

используем фактор выравнивания - VAF. Значение фактора выравнивания строится из 14 системных характеристик продукта

Характеристика	Значение
Резервное хранение данных	2
Обмен данными	2
Распределенные вычисления	0
Важность производительности	1
Загруженность оборудования	1
Производительность	1
Транзакционная нагрузка	0
Интенсивность изменения данных	0
Эргономика	3
Интенсивность взаимодействия с пользователем	2
Повторное использование	2
Преобразование данных и установка	0
Портируемость	1
Необходимость настройки и простота	0

$$TDI = \sum DI = 15$$

$$VAF = (TDI * 0.01) + 0.65 = 0.8$$

Подсчет количества AFP (выровненных функциональных точек):

$$AFP = UFP * VAF = 37.6$$

Метод анализа функциональных точек не предоставляет информации о трудозатратах на разработку оцениваемого продукта. Этот вопрос можно легко решить, если у компании-разработчика есть собственная статистика по трудозатратам на реализацию функциональных точек. В случае отсутствия такой статистики, для оценки трудоемкости и сроков проекта можно использовать метод COSOMO II.

COCCOMO II

Оценка размера продукта в KSLOC:

Используемый стек технологий:

- Frontend: Javascript
- Backend: PostgreSQL

Используем обновленную таблицу коэффициентов из задания:

<https://www.qsm.com/resources/function-point-languages-table>

$$KSLOC = UFP * SIZE = (47 * 0.06) = 2.82$$

Оценка уровней факторов масштаба:

1. PREC — прецедентность, наличие опыт аналогичных разработок (**High**)
2. FLEX — гибкость процесса разработки (**Low**).
3. RESL — архитектура и разрешение рисков (**Nominal**)
4. TEAM — сработанность команды (**High**)
5. PMAT — зрелость процессов (**Nominal**)

Факторы масштаба	
PREC	2.48
FLEX	3.04
RESL	4.24
TEAM	2.19
PMAT	4.68

1. PERS — квалификация персонала (**High**)
2. RCPX — сложность и надежность продукта (**High**)
3. RUSE — разработка для повторного использования (**Low**)
4. PDIF — сложность платформы разработки (**Low**)
5. PREX — опыт персонала (**High**)

6. FCIL — оборудование (**Nominal**)
7. SCED — сжатие расписания (**Nominal**)

Трудоемкость	
ь	
PERS	0.83
RCPX	1.33
RUSE	0.95
RDIF	0.87
PREX	0.87
FCIL	1
SCED	1

Оценка трудоемкости проекта:

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

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

$$A = 2.94$$

$$B = 0.91$$

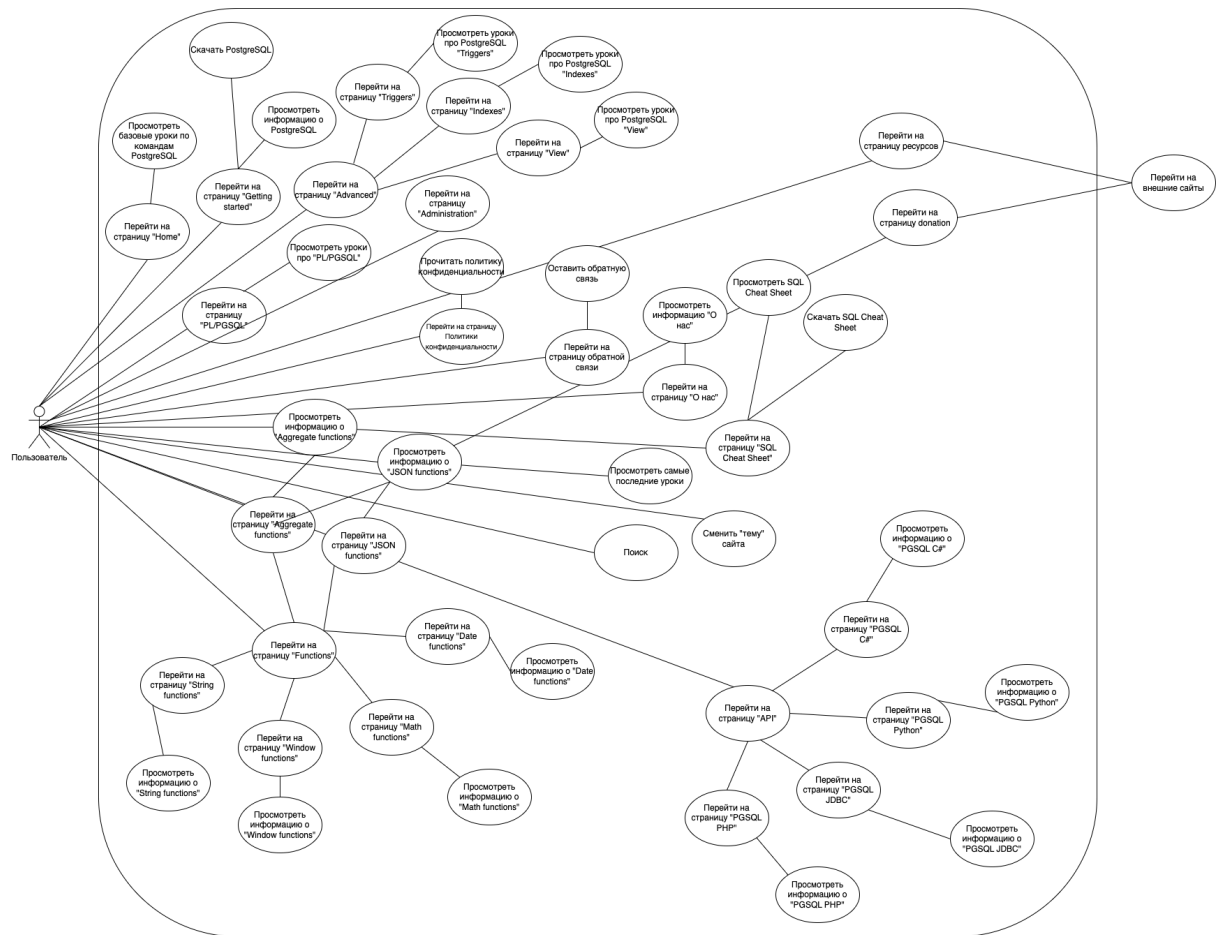
- SIZE — размер продукта в KSLOC
- EM_i — множители трудоемкости
- SF_j — факторы масштаба
- n=7 — для предварительной оценки
- n=17 — для детальной оценки

$$E = 0.91 + 0.01 * (2.48 + 3.04 + 4.24 + 2.19 + 4.68) = 1.0763$$

$$PM = 2.94 * 2.82^{1.0763} * (0.83 * 1.33 * 0.95 * 0.87 * 0.87 * 1 * 1) \approx 7.1227 \text{ ч. /мес} \approx 1139 \text{ ч. /ч}$$

Use Case Points:

Use Case диаграмма для пользователя:



Оценка веса прецедентов:

Сложность	Вес (UUCW)	Количество	Затраты
Low	5	42	210
Mid	10	10	100
High	15	3	45
Нескорректированный вес варианта использования(UUCW)			355

Оценка веса акторов:

Сложность	Вес (AWi)	Количество	Затраты
Low	1	3	3
Mid	2	1	2
High	3	1	3
Масса актера без корректировки (UAW)			9

Оценка веса технических факторов:

Фактор	Описание	Вес	Сложность
T1	Распределенная система	1	0
	Время отклика / целевые показатели		3
T2	производительности	1	
	Эффективность для конечного		5
T3	пользователя	3	
T4	Сложность внутренней обработки	1	0
T5	Повторное использование кода	1	2
T6	Простота установки	0,5	2
T7	Простота использования	2	5
T8	Переносимость на другие платформы	1	3
T9	Обслуживание системы	1	2
T10	Многопоточность	1	1
T11	Функции безопасности	1	2
T12	Доступ к другим системам	1	0
T13	Обучение конечных пользователей	1	0

TF 39
TCF 0.99

Оценка веса факторов окружения:

Фактор	Описание	Вес	Влияние
E1	Знакомство с использованным процессом разработки	1.5	4
E2	Опыт применения технологий	0.5	3
E3	Объектно-ориентированный опыт команды	1	3
E4	Возможности ведущего аналитика	0.5	1
E5	Мотивация команды	1	2
E6	Стабильность требований	2	3
E7	Персонал, занятый неполный рабочий день	-1	2
E8	Сложный язык программирования	-1	3
		EF	14
		ECF	0.98

Подсчет UCP

$$UCP' = (UCW + UAW) * TCF * ECF = 353$$

Подсчет фактора продуктивности на основе прошлого проекта

В качестве примера мы выбрали проект по предмету «Разработка мобильных приложений», который разрабатывала команда из 4-х студентов

Список Use Case:

№	Описание
1	Регистрация
2	Авторизация
3	Поиск акций

4	Добавление акций в портфель
5	Создание портфеля
6	Посмотреть список акций
7	Посмотреть статистику
8	Посмотреть историю транзакций
9	Изменение данных пользователя в личном кабинете
10	Посмотреть список портфелей

Оценка веса прецедентов:

Сложность	Вес (UUCW)	Количество	Затраты
Low	5	5	25
Mid	10	5	50
High	15	0	0
Нескорректированный вес варианта использования(UUCW)			75

Оценка веса акторов:

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

Оценка веса технических факторов:

Фактор	Описание	Вес	Сложность
T1	Распределенная система	1	0
T2	Время отклика / целевые показатели производительности	1.5	2
T3	Эффективность для конечного пользователя	3	4
T4	Сложность внутренней обработки	1	2
T5	Повторное использование кода	1	1
T6	Простота установки	0.5	1
T7	Простота использования	2	3
T8	Переносимость на другие платформы	1	0
T9	Обслуживание системы	1	1
T10	Многопоточность	1	1
T11	Функции безопасности	1	1
T12	Доступ к другим системам	1	0
T13	Обучение конечных пользователей	1	1

TF 28.5
TCF 0.885

Оценка веса факторов окружения:

Фактор	Описание	Вес	Влияние
E1	Знакомство с использованным процессом разработки	1.5	4
E2	Опыт применения технологий	0.5	3
E3	Объектно-ориентированный опыт команды	1	3
E4	Возможности ведущего аналитика	0.5	2
E5	Мотивация команды	1	2
E6	Стабильность требований	2	3
E7	Персонал, занятый неполный рабочий день	-1	4
E8	Сложный язык программирования	-1	4

EF 11.5
ECF 1.055

Подсчет UCP:

$$UCP' = (UCW + UAW) * TCF * ECF = 78$$

Подсчет трудоемкости проекта:

Предыдущая работа была выполнена за 120 часов на 4-х человек.

$$PF = E/UCP' = 120 / 78 = 1.54$$

$$E = PF * UCP' = 543 \text{ ч./ч.}$$

Анализ результатов:

Метод	Затраты (h-h)
Наивный	600
PERF	762
COMOCO II	1139
UCP	543

Наивный метод - Это самый простой способ оценки, который основан на первоначальных предположениях или эмпирическом опыте. Он показывает минимальные трудозатраты, что потенциально указывает на недооценку сложности проекта. Повысить объективность метода можно используя Покер планирование.

PERF - Этот метод более сложен и учитывает вероятные факторы риска и особенности проекта. Оценка выше наивного метода, что указывает на более детализированный подход к оценке проекта, который предполагает дополнительные сложности или непредвиденные трудности.

COMOCO II - Этот метод выдает наибольшую оценку трудозатрат, что указывает на высокий уровень детализации и учет всех возможных рисков и сложностей. Это наиболее консервативный подход к оценке, поэтому он может быть предпочтительнее при разработке сложных или важных систем.

UCP - Метод на основе UCP дает оценку ниже, чем наивный метод. Это означает, что проект был оценен как менее сложный с точки зрения пользовательских сценариев. Однако UCP может недооценивать трудозатраты в случае более сложной внутренней логики системы. Также завязка на сторонние проекты может привести к неточности результатов в случае сравнения с проектом в другой сфере разработки

Вывод

В ходе лабораторной работы мы научились использовать различные методы оценки трудоемкости проектов. Каждый из методов показал разные результаты по трудозатратам, что позволило нам получить более полное представление о возможных затратах времени и ресурсов на реализацию проекта. Подобное оценивание - трудная задача с которой приходится сталкиваться в сфере программирования, как и задача выделения реалистичных сроков на выполнение проекта.

Таким образом, в зависимости от сложности и требований к проекту, а также опыта предыдущей разработки схожих проектов, можно выбрать тот метод оценки, который наиболее точно отражает характер проекта, и использовать полученные результаты для эффективного планирования и распределения ресурсов.