

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский университет ИТМО»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ

ЛАБОРАТОРНАЯ РАБОТА № 1

по дисциплине

Экономика программной инженерии

Выполнили:

Студенты группы

Р34111

Журавлев Вячеслав

Бегина Анастасия

Преподаватель:

Машина Е.А.

Санкт-Петербург, 2024

Оглавление

Оглавление	2
Вариант	3
Функциональные требования	3
Оценка наивным методом	6
Оценка с помощью метода PERT	8
Метод функциональных точек	9
Метод COCOMO II	11
Оценка методом Use Case Points	14
Сравнение использованных результатов	16

Вариант

Сайт для анализа – <https://stepik.org>

Функциональные требования

1. Система должна предоставлять возможность регистрации и авторизации
 - 1.1. Система должна предоставлять возможность регистрации и входа через ВКонтакте
 - 1.2. Система должна предоставлять возможность регистрации и входа через Google Аккаунт
 - 1.3. Система должна предоставлять возможность регистрации и входа через GitHub
 - 1.4. Система должна позволять регистрироваться с именем и фамилией (опциональное поле), email и паролем
 - 1.5. Система должна валидировать email при регистрации
 - 1.6. Система должна предоставлять возможность входа по email и паролю
2. Система должна предоставлять возможность просматривать и искать курсы
 - 2.1. Система должна содержать каталог курсов
 - 2.2. Система должна содержать курсы по разделам в зависимости от тематики курса
 - 2.3. Система должна предоставлять возможность сортировать курсы в зависимости от тематики
 - 2.4. Система должна содержать подборки курсов по темам: “Новые курсы”, “Хиты”, “Пакеты курсов”, “Актуальные скидки”, “Выбирают компании”
 - 2.5. Система должна содержать блок “Рекомендованные курсы”, которые индивидуальны для каждого пользователя
 - 2.6. Система должна предоставлять возможность поиска по курсам
 - 2.7. Система должна предоставлять возможность выставить фильтры по языку, наличию сертификата и стоимости курса при поиске
3. Система должна предоставлять возможность просматривать детали курса
 - 3.1. Описание курса должно содержать автора программы, краткое описание, количество уроков и времени, которое займет курс
 - 3.2. Описание курса должно содержать начальные требования, программу курса, стоимость обучения
 - 3.3. Из деталей курс можно добавить в избранное

4. Система должна предоставлять возможность проходить курсы авторизованному пользователю
 - 4.1. Прохождение курса должно происходить уроком, которые разделены по шагам
 - 4.2. К каждому шагу необходимо предоставлять возможность оставить комментарий
 - 4.3. Система должна предоставлять возможность оценить каждый шаг пальцем вверх или вниз
 - 4.4. Система должна содержать проверочные вопросы
 - 4.4.1. Проверочные вопросы могут содержать только один правильный ответ
 - 4.4.2. Проверочные вопросы могут содержать несколько правильных ответов
 - 4.4.3. Проверочные вопросы могут содержать текстовое поле для ввода
 - 4.4.4. Проверочные вопросы могут содержать поле для ввода кода с форматированием
 - 4.4.5. Система должна предоставлять возможность запустить код из проверочного вопроса и показать вывод программы
 - 4.4.6. Проверочные вопросы могут содержать поле для загрузки файла с любым расширением
 - 4.5. Система должна предоставлять возможность просматривать содержание курса на каждом этапе
 - 4.6. Система должна предоставлять возможность автору курса выбрать возможность перехода между уроками
 - 4.7. Система должна предоставлять возможность открыть курс на полный экран при прохождении
 - 4.8. Система должна предоставлять возможность распечатать урок
 - 4.9. Система должна предоставлять возможность подписаться и отписаться на комментарии к уроку
5. Система должна предоставлять возможность купить курс авторизованному пользователю
 - 5.1. Система должна предоставлять возможность купить курс целиком
 - 5.2. Система должна предоставлять возможность оплатить курс "Долями"
 - 5.3. Система должна предоставлять возможность попробовать бесплатно курс, что включает первый урок
 - 5.4. Система должна предоставлять возможность купить курс в подарок
 - 5.5. Система должна предоставлять возможность купить курс от компании

6. Система должна позволять отслеживать учебный прогресс авторизованному пользователю
 - 6.1. Система должна предоставлять возможность отслеживать прогресс во время прохождения курса
 - 6.2. Система должна предоставлять возможность отслеживать прогресс в личном кабинете
 - 6.3. Система должна предоставлять возможность отслеживать прогресс в разделе “Мое обучение”
 - 6.3.1. Раздел “Мое обучение” должен содержать информацию о пройденных курсах
 - 6.3.2. Раздел “Мое обучение” должен содержать текущие курсы
 - 6.3.3. Раздел “Мое обучение” должен содержать избранные курсы
 - 6.3.4. Раздел “Мое обучение” должен содержать курсы, которые пользователь хочет пройти
 - 6.3.5. Раздел “Мое обучение” должен содержать архив курсов
 - 6.3.6. Раздел “Мое обучение” должен содержать классы пользователя
 - 6.3.7. Раздел “Мое обучение” должен содержать уведомления, которые получил пользователь
 - 6.3.8. Раздел “Мое обучение” должен содержать раздел ”Помощь”, где будет информация о разделе
7. Система должна позволять получать сертификаты после завершения курса авторизованному пользователю
 - 7.1. Система должна показывать сертификаты в личном кабинете пользователя
 - 7.2. Система должна предоставлять возможность получить сертификат с отличием при прохождении курса больше чем на 80%
8. Система должна содержать личный кабинет пользователя
 - 8.1. Личный кабинет пользователя должен содержать имя и фамилию
 - 8.2. Личный кабинет пользователя должен содержать ID пользователя
 - 8.3. Личный кабинет пользователя должен содержать аватар пользователя
 - 8.4. Личный кабинет пользователя должен содержать данные об активности за последний год
 - 8.4.1. Данные об активности должны содержать количество дней без перерыва
 - 8.4.2. Данные об активности должны содержать максимальное число дней без перерыва
 - 8.4.3. Данные об активности должны содержать количество решенных задач

- 8.5. Личный кабинет пользователя должен содержать количество репутации и знаний пользователя
 - 8.5.1. Репутация рассчитывается из оставленных комментариев пользователя
 - 8.5.2. Знания рассчитываются из пройденных пользователем уроков
- 8.6. Личный кабинет пользователя должен содержать информацию, когда пользователь присоединился
- 8.7. Личный кабинет пользователя должен содержать сертификаты пользователя
- 8.8. Личный кабинет пользователя должен предоставлять возможность посмотреть, как профиль видят другие пользователи
- 9. Система должна иметь сообщество: социальные сети, отзывы и рейтинги курсов
 - 9.1. Рейтинги курсов должны формироваться из оценок пользователей
 - 9.2. Отзывы по курсам должны оставлять пользователи после или во время прохождения курса
 - 9.3. Сервис должен содержать ссылки на социальные сети
- 10. Система должна иметь мобильное приложение
- 11. Система должна иметь возможность перейти на стороннюю страницу создания курсов на Stepik
- 12. Система должна предоставлять возможность сменить язык на русский, белорусский, датский, английский, испанский, португальский, украинский и китайский

Оценка наивным методом

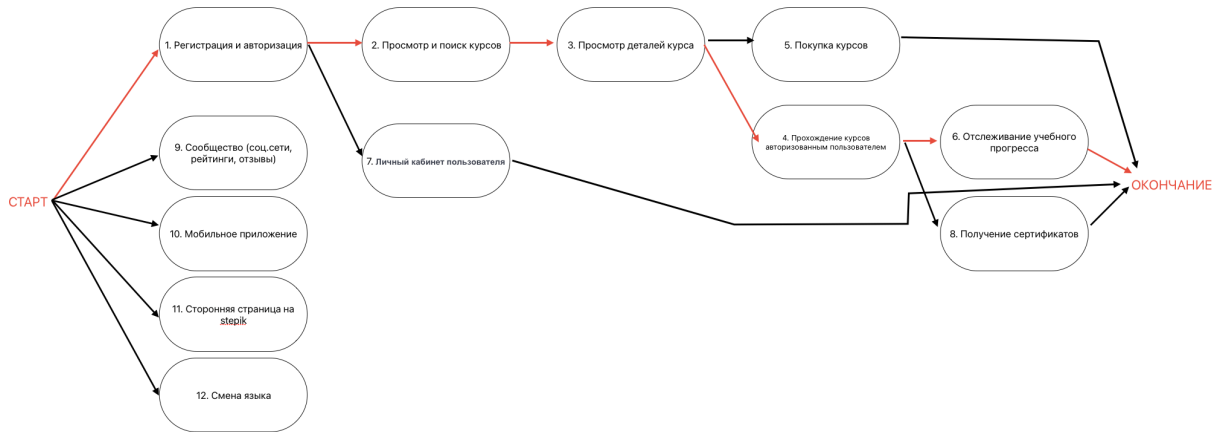
- 1. Регистрация и авторизация:
 - Минимум: 1 неделя * 40 = 40
 - Вероятно: 1.5 недели * 40 = 60
 - Максимум: 2 недели * 40 = 80
- 2. Просмотр и поиск курсов:
 - Минимум: 2 недели * 40 = 80
 - Вероятно: 2.5 недели * 40 = 100
 - Максимум: 3 недели * 40 = 120
- 3. Просмотр деталей курса:
 - Минимум: 1 неделя * 40 = 40
 - Вероятно: 1.5 недели * 40 = 60
 - Максимум: 2 недели * 40 = 80
- 4. Прохождение курсов авторизованным пользователем:

- Минимум: 3 недели * 40 = 120
 - Вероятно: 3.5 недели * 40 = 140
 - Максимум: 4 недели * 40 = 160
5. Покупка курсов:
- Минимум: 2 недели * 40 = 80
 - Вероятно: 2.5 недели * 40 = 100
 - Максимум: 3 недели * 40 = 120
6. Отслеживание учебного прогресса:
- Минимум: 2 недели * 40 = 80
 - Вероятно: 2.5 недели * 40 = 100
 - Максимум: 3 недели * 40 = 120
7. Получение сертификатов:
- Минимум: 2 недели * 40 = 80
 - Вероятно: 2 недели * 40 = 80
 - Максимум: 2 недели * 40 = 80
8. Личный кабинет пользователя:
- Минимум: 2 недели * 40 = 80
 - Вероятно: 2.5 недели * 40 = 100
 - Максимум: 3 недели * 40 = 120
9. Сообщество (социальные сети, отзывы и рейтинги):
- Минимум: 1 неделя * 40 = 40
 - Вероятно: 1.5 недели * 40 = 60
 - Максимум: 2 недели * 40 = 80
10. Мобильное приложение:
- Минимум: 4 недели * 40 = 160
 - Вероятно: 5 недель * 40 = 200
 - Максимум: 6 недель * 40 = 240
11. Сторонняя страница на Stepik:
- Минимум: 1 неделя * 40 = 40
 - Вероятно: 1.5 недели * 40 = 60
 - Максимум: 2 недели * 40 = 80
12. Смена языка:
- Минимум: 2 недели * 40 = 80
 - Вероятно: 2.5 недели * 40 = 100
 - Максимум: 3 недели * 40 = 120

Итого: 60+100+60+140+100+100+80+100+60+200+60+100=1160 человеко-часов

Оценка с помощью метода PERT

Сетевая диаграмма взаимосвязи работ и критический путь



Рассчитаем минимальное время разработки по критическому пути:

- Регистрация и авторизация (1.5 недели) →
- Просмотр и поиск курсов (2.5 недели) →
- Просмотр деталей курса (1.5 недели) →
- Прохождение курсов (3.5 недели) →
- Отслеживание прогресса (2.5 недели) →
- Получение сертификатов (2 недели)

Таким образом, суммарное время критического пути = $1.5 + 2.5 + 1.5 + 3.5 + 2.5 + 2 = 13.5$ недель.

$13.5 * 40 = 540$ человеко-часов.

1. Минимальное время для выполнения проекта при разделении работы параллельно будет зависеть от критического пути, то есть от выполнения этапов друг за другом.
2. Если взять за цель сократить срок до 6.5 недель (около половины критического пути), можно посчитать количество разработчиков, разделив общий объем человеко-часов на это время

$$\text{необходимое число разработчиков} = \frac{\text{целевое время выполнения}}{\text{общий объем человеко-часов}} = \frac{1160}{6.5 * 40} = 4.5$$

Округляя, для оптимального выполнения проекта за 6.5 недель потребуется 5 универсальных разработчиков

Метод функциональных точек

Шаг 1: Определение типов функций

Сначала определим категории функций, которые существуют в проекте:

1. Внешние входы (EI) – ввод данных пользователем (например, регистрация и авторизация, покупка курсов).
2. Внешние выходы (EO) – отображение данных пользователю (например, просмотр курсов, просмотр деталей курса).
3. Внешние запросы (EQ) – ввод-вывод с системой, которая подразумевает взаимодействие с пользователем (например, социальные сети, отзывы и рейтинги).
4. Внутренние логические файлы (ILF) – локальные хранилища информации в системе (например, личный кабинет, отслеживание прогресса).
5. Внешние интерфейсные файлы (EIF) – данные, к которым система обращается извне (например, сторонняя страница на Stepik).

Шаг 2: Оценка функциональных точек для каждой категории

Для каждой функции в проекте оценим количество и сложность на основе следующих значений (высокая, средняя и низкая сложность):

- Внешние входы (EI):
 - Низкая сложность: 3 ФП
 - Средняя сложность: 4 ФП
 - Высокая сложность: 6 ФП
- Внешние выходы (EO):
 - Низкая сложность: 4 ФП
 - Средняя сложность: 5 ФП
 - Высокая сложность: 7 ФП
- Внешние запросы (EQ):
 - Низкая сложность: 3 ФП
 - Средняя сложность: 4 ФП
 - Высокая сложность: 6 ФП
- Внутренние логические файлы (ILF):
 - Низкая сложность: 7 ФП
 - Средняя сложность: 10 ФП
 - Высокая сложность: 15 ФП
- Внешние интерфейсные файлы (EIF):
 - Низкая сложность: 5 ФП
 - Средняя сложность: 7 ФП
 - Высокая сложность: 10 ФП

Шаг 3: Расчет функциональных точек для каждой функции

Функция	Тип	Сложность	ФП
Регистрация и авторизация	EI	Средняя	4
Просмотр и поиск курсов	EO	Средняя	5
Просмотр деталей курса	EO	Низкая	4
Прохождение курсов	ILF	Высокая	15
Покупка курсов	EI	Средняя	4
Отслеживание учебного прогресса	ILF	Средняя	10
Получение сертификатов	EO	Средняя	5
Личный кабинет пользователя	ILF	Средняя	10
Сообщество, отзывы и рейтинги	EQ	Низкая	3
Мобильное приложение	EIF	Средняя	7
Сторонняя страница на Stepik	EIF	Низкая	5
Смена языка	EQ	Средняя	4

Общие функциональные точки = $4+5+4+15+4+10+5+10+3+7+5+4 = 76$

Шаг 4: Применение коэффициента сложности

Далее применим **коэффициент сложности** для уточнения размера проекта. Обычно для корпоративных приложений коэффициент сложности варьируется от 1 до 1.4. Предположим, что этот проект имеет среднюю сложность, и используем коэффициент 1.2.

Итоговые функциональные точки = $76 \times 1.2 = 91.2$ ФП

Шаг 5: Оценка в человеко-часах

Теперь переведем функциональные точки в человеко-часы.

Средняя оценка – это **10 человеко-часов на 1 функциональную точку**.

$91.2 \times 10 = 912$ человеко-часов

Результат:

Проект можно оценить как **91 функциональную точку**, что примерно соответствует **912 человеко-часам** работы. Это совпадает с предыдущими расчетами, что предполагает необходимость **5 разработчиков на срок около 6 недель** для выполнения проекта.

Метод COSOMO II

Шаг 1: Определение исходных данных

Для начала определим объем работы и основные параметры:

- **Размер проекта:** мы будем использовать размер в **тысячах строк кода (KLOC)** или **функциональных точках (ФП)**, которые можно преобразовать в KLOC.

Мы рассчитали **91 функциональную точку**. Для перевода в KLOC возьмем коэффициент: $1 \text{ ФП} \approx 50 \text{ строк кода}$. Так как проект предполагается на Java (47 строк) + JavaScript (53 строки)

$$KLOC = 91 \times 50 / 1000 = 4.55$$

Шаг 2: Определение режима разработки

COSOMO II определяет три режима:

1. **Органический:** Для небольших проектов с небольшим числом сложностей.
2. **Полусвободный:** Для проектов средней сложности.
3. **Встроенный:** Для крупных проектов с высокой сложностью и жёсткими ограничениями.

Поскольку наш проект предполагает среднюю сложность и корпоративную направленность, выбираем **полусвободный режим**.

Шаг 3: Расчет базовой трудоемкости

Для расчета базовой трудоемкости в COCOMO II используется формула:

$$\text{Трудоемкость} = A \times (KLOC)^B$$

где:

- A и B — коэффициенты, зависящие от выбранного режима,
- Для полусвободного режима обычно $A=3.0$ и $B=1.12$.

Подставляем значения:

$$\text{Трудоемкость} = 3.0 \times (4.55)^{1.12} = 16.37$$

Трудоемкость проекта по методу COCOMO II составит примерно **16.37 человеко-месяцев**.

Затем высчитаем трудоемкость с факторами масштаба:

Шаг 1: Вычисляем значение фактора масштаба

$$\sum_{j=1}^5 SF_j = PREC + FLEX + RESL + TEAM + PMAT$$

Используем значения:

- **PREC (Low)** = 4.96
- **FLEX (Nominal)** = 3.04
- **RESL (Nominal)** = 4.24
- **TEAM (Nominal)** = 3.29
- **PMAT (Nominal)** = 4.68

$$\sum_{j=1}^5 SF_j = 20.21$$

Шаг 2: Вычисляем множитель трудоемкости

$$EM = PERS \times RCPX \times RUSE \times PDIF \times PREX \times FCIL \times SCED$$

Используем значения:

- **PERS (Nominal)** = 1.00
- **RCPX (High)** = 1.33
- **RUSE (Low)** = 0.95
- **PDIF (Nominal)** = 1.00
- **PREX (Nominal)** = 1.00
- **FCIL (Nominal)** = 1.00
- **SCED (Nominal)** = 1.00

$$EM = 1.2635$$

Шаг 3: Расчет трудоемкости PM_{NS}

Предположим, что трудоемкость в человеко-месяцах без учета SCED равна 16.37, как указано ранее.

Шаг 4: Вычисляем время разработки TDEV по формуле

$$TDEV = C \times (PM_{NS})^{D+0,2 \times 0,01 \times \sum_{j=1}^5 SF_j} \times \frac{SCED}{100}$$

Где:

- C=3.67
- D=0.28
- PM_{NS} =16.37
- SCED=16.37

Подставим значения и произведем расчеты:

$$TDEV = 8.9882$$

Рассчитанное время разработки TDEV составляет примерно 9 месяцев. Это означает, что проект потребует около 9 человеко-месяцев для выполнения, учитывая выбранные параметры и коэффициенты.

Оценка методом Use Case Points

В качестве завершеного проекта возьмем последнюю лабораторную работу по БЛПС

Шаг 1: Определение акторов и вариантов использования

Для лабораторной работы были следующие акторы и варианты использования:

- Акторы:
 - Простой: 1 (пользователь)
 - Средний: 1 (администратор)
 - Сложный: 1 (внешняя система)
- Варианты использования:
 - Простой: 2 (вход в систему, регистрация)
 - Средний: 2 (просмотр данных, создание заказов)
 - Сложный: 1 (отправка заказов с таймером)

Шаг 2: Оценка баллов

1. Акторы:

- Простой: $1 \times 1 = 1$
- Средний: $1 \times 2 = 2$
- Сложный: $1 \times 3 = 3$

Общее количество баллов акторов:

$$UAW = 1 + 2 + 3 = 6$$

2. Варианты использования:

- Простой: $2 \times 5 = 10$
- Средний: $2 \times 10 = 20$
- Сложный: $1 \times 15 = 15$

Общее количество баллов вариантов использования:

$$UUC = 10 + 20 + 15 = 45$$

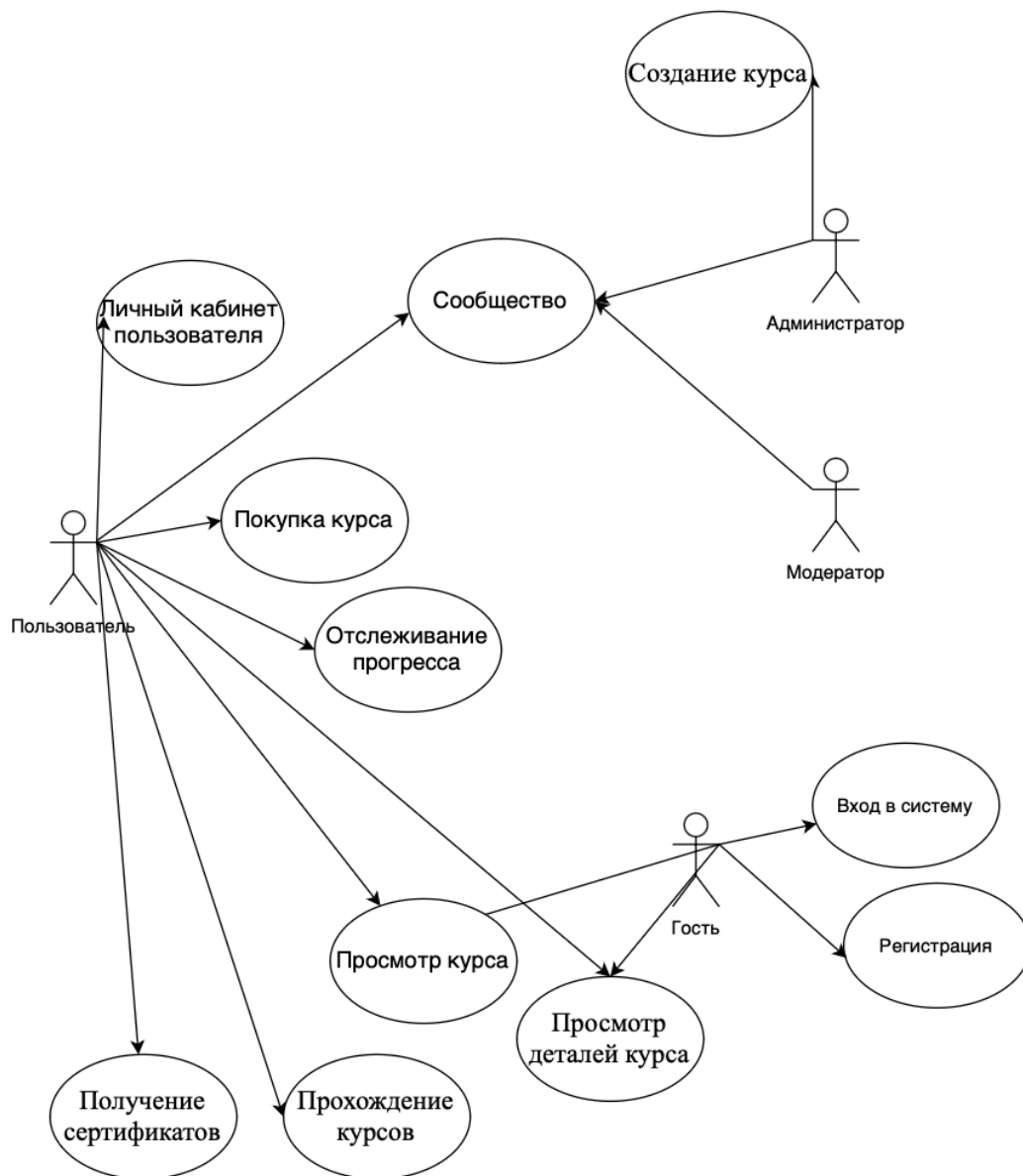
Шаг 3: Общий UCP

$$UCP = UAW + UUC = 6 + 45 = 51$$

Шаг 4: Расчет фактора продуктивности (PF)

$$\circ PF = \frac{\text{время на проект}}{\text{общее кол-во UCP}} = \frac{80}{51} \approx 1.56 \text{ часа на UCP}$$

Шаг 5: Определение UCP для stepik



Акторы

- Простой: 2 (обычный пользователь, гость)
- Средний: 2 (администратор, модератор)
- Сложный: 1 (API)

$$\circ UAW = 2 + 4 + 3 = 9$$

Варианты использования

- Простой: 5 (вход в систему, регистрация, просмотр курсов, личный кабинет пользователя, сообщество)
- Средний: 4 (покупка курса, отслеживание прогресса, получение сертификатов, детали курса)
- Сложный: 2 (прохождение курсов, интеграция с внешними системами(портал для создания курсов))

$$UCC = 25 + 40 + 30 = 95$$

$$UCP = UAW + UUC = 9 + 95 = 104$$

Шаг 6: Оценка трудозатрат для нового проекта

Теперь, зная PF и общее количество UCP нового проекта, можем рассчитать трудозатраты

$$\text{Трудозатраты} = UCP \times PF = 104 \times 1.56 = 163 \text{ часов}$$

Сравнение использованных результатов

Наивный метод: 1160 человеко-часов

PERT: 540 человеко-часов

COCOMO II: 8.9 человеко-месяцев

UCP: 163 человеко-часов

Анализ результатов:

1. Наивный метод:

Наивный метод дает наиболее высокую оценку трудозатрат, составившую 1160 человеко-часов. Это может быть связано с недостатком детализации и учета различных факторов, влияющих на трудоемкость.

2. PERT:

Метод PERT предлагает значительно более низкую оценку в 540 человеко-часов. Этот метод учитывает различные сценарии (оптимистичный, пессимистичный и наиболее вероятный), что позволяет более точно оценить трудозатраты.

3. COCOMO II:

Оценка методом COCOMO II составляет 8.9 человеко-месяцев, что в пересчете на человеко-часы составляет примерно 132 человеко-часа. Это указывает на то, что метод COCOMO II учитывает факторы масштабируемости и трудоемкости, предлагая более реалистичную оценку для опытных разработчиков.

4. UCP:

Метод оценки по вариантам использования (UCP) показывает оценку в 163 человеко-часа. Это значение меньше остальных, так как рассчитывалась на данных о нашем проекте, который был менее трудным.

Общие выводы:

- Наивный метод предоставляет высокие оценки, что может привести к неоправданным ожиданиям и завышенным бюджетам.
- Методы PERT, COCOMO II дают более сбалансированные и обоснованные оценки, позволяя лучше планировать ресурсы и время выполнения проекта.
- Для более точной оценки трудозатрат рекомендуется использовать комбинацию различных методов, что позволит учесть различные аспекты проекта и улучшить планирование.