

Санкт-Петербургский Национальный Исследовательский
Университет ИТМО

Дисциплина: Экономика программной инженерии

Лабораторная работа №1

Вариант:

<https://www.abbeyroad.com>

Выполнили:

Виноградов Глеб Дмитриевич
Трофимченко Владислав Михайлович

Группа: Р34111

Преподаватель:
Машина Екатерина Алексеевна

Санкт-Петербург, 2024 г.

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 СОСМО 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 производительности PF использовать любой свой завершенный проект с известными временными трудозатратами, оценив его размер методом UCP.
6. Сравнить полученные результаты и сделать выводы.

Функциональные требования

1. Система должна предоставлять возможность просмотра новостей
 - 1.1. На “главной странице” должны отображаться в 3 последние новости в сокращенном виде
 - 1.1.1. В сокращенном виде новость должна представлять из себя: обложку, дату публикации, краткое описание в виде одного параграфа текста, ссылку на полную статью, набор тегов
 - 1.2. Должен быть реализован поиск публикации по заголовку новости на “странице новостей”
 - 1.3. Должна быть реализована категоризация новостей на основе тегов на “странице новостей”
 - 1.3.1. список категорий (тегов) динамический, новая категория создается при добавлении нового тега к одной из новостей
 - 1.3.2. Поиск по категориям должна поддерживать выдачу новостей из одной категории, объединение нескольких категорий при фильтрации не поддерживается
 - 1.4. На “странице новостей” новости в сокращенном виде должны отображаться по 9 штук на странице
 - 1.5. На “странице новостей” новости должны быть отсортированы по дате публикации в порядке убывания

- 1.6. На “странице новостей” должна быть реализована возможность загрузить следующую страницу новостей
2. Система должна предоставлять возможность отображать актуальные события студии
 - 2.1. Информация о событиях должна размещаться на “Главной странице” (по 5 штук)
 - 2.1.1. Каждое событие представляет собой: изображение, которое растягивается на весь экран; заголовок
 - 2.2. События должны автоматически циклически сменять друг друга на “Главной странице”
 - 2.3. При нажатии на событие должен происходить редирект на сторонний ресурс или раздел системы.
3. Система должна предоставлять возможность ознакомление с информацией об звукозаписывающих студиях
 - 3.1. Информация должна располагаться на отдельной странице “Студии” в обобщенном виде
 - 3.1.1. Для каждой студии должна размещаться: фотография, название, ссылка для перехода на страницу студии
 - 3.1.2. На странице “Студии” должны располагаться ссылки на внешний ресурс youtube.com на видеоматериалы, записанные в перечисленных студиях
 - 3.1.3. На странице “Студии” должен располагаться виджет в виде карты, с отмеченным адресом местонахождения студий
 - 3.1.4. На странице “Студии” должна располагаться форма обратной связи для записи/вопросов
 - 3.1.4.1. Форма должна содержать поля: имя - обязательно, email - обязательно, тип обращения (записаться в студию, задать вопрос)
 - 3.2. Подробная информация о каждой отдельной студии должна располагаться на странице “Студия”
 - 3.2.1. На странице “Студия” должна быть размещена краткая информация об истории студии
 - 3.2.2. На странице “Студия” должна быть ссылка на форму обратной связи (см. 3.1.4)
 - 3.2.3. На странице “Студия” должен быть размещен список недавних проектов, записанных в данной студии (4 шт.)
 - 3.2.4. На странице “Студия” должен быть размещен список оборудования с возможностью скачать .pdf версию
 - 3.2.4.1. План помещения
 - 3.2.4.2. Микрофоны
 - 3.2.4.3. Инструменты
 - 3.2.4.4. Другая звукозаписывающая техника
 - 3.2.5. На странице “Студия” должен быть размещен список сотрудников студии со ссылками на их персональные страницы
 - 3.2.5.1. На персональной странице сотрудника должна быть размещена следующая информация: Имя, биография, список работ

- 3.2.6. На странице “Студия” должна быть размещена ссылка на плейлист на внешнем ресурсе [spotify.com](https://open.spotify.com)
 - 3.2.7. На странице “Студия” должен располагаться виджет в виде карты, с отмеченным адресом местонахождения студии
 - 3.2.8. На странице “Студии” должна располагаться форма обратной связи для записи/вопросов
 - 3.2.8.1. Форма должна содержать поля: имя - обязательно, email - обязательно, тип обращения (записаться в студию, задать вопрос)
4. Система должна предоставлять возможность перехода на сайт института на внешнем ресурсе <https://abbeyroadinstitute.co.uk/>
5. В системе должны быть размещены статьи о студии
 - 5.1. Статья об истории студии
 - 5.2. Статья о Джайлзе Мартине
 - 5.3. Статья о команде студии
 - 5.4. Статья о Джордане Ракей
6. Система должна предоставлять возможность сведения музыки
 - 6.1. Онлайн сведение
 - 6.1.1. Должна быть возможность выбрать валюту, в которой отображаются цены и производится оплата
 - 6.1.1.1. GBP
 - 6.1.1.2. EUR
 - 6.1.1.3. USD
 - 6.1.2. Система должна предоставлять возможность выбрать пакет услуг (должна быть возможность выбрать несколько пакетов за раз)
 - 6.1.2.1. Стандартный пакет
 - 6.1.2.2. Улучшенный пакет
 - 6.1.2.3. Делюкс пакет
 - 6.1.3. Система должна предоставлять возможность выбрать инженера, который будет заниматься сведением
 - 6.1.4. Система должна предоставлять возможность ввести данные покупателя
 - 6.1.4.1. Адрес для отправки чека об оплате
 - 6.1.4.2. Имя
 - 6.1.4.3. Электронная почта
 - 6.1.5. Система должна предоставлять возможность ускорить процесс сведения трека за дополнительную плату
 - 6.1.6. Система должна предоставлять возможность ввести промокод для применения скидки
 - 6.1.7. Система должна отображать итоговую сумму заказа с учетом скидки и дополнительных опций
 - 6.1.8. Система должна предоставлять возможность загрузить желаемые трека в соответствии с количеством купленных пакетов
 - 6.1.9. Система должна предоставлять ссылку на внешнюю систему приема платежей
 - 6.1.10. Система должна запрашивать у клиента конечное подтверждение заказа перед его формированием
 - 6.2. Оффлайн сведение

- 6.2.1. Система должна предоставлять информацию о возможностях оффлайн сведения в студии
 - 6.2.1.1. Список услуг
 - 6.2.1.2. Список дополнительных услуг
 - 6.2.1.3. Список комнат для сведения
 - 6.2.1.4. Список инженеров по сведению
 - 6.2.1.5. Список партнеров по сведению
- 6.2.2. Система должна предоставлять возможность записаться на оффлайн сведение
- 7. Система должна предоставлять ссылки на социальные сети студии
 - 7.1. X
 - 7.2. Facebook*
 - 7.3. Instagram*
 - 7.4. Youtube
- 8. Система должна предоставлять возможность подписаться на почтовую рассылку новостей студии
- 9. Система должна предоставлять информацию об истории места в котором находится студия
 - 9.1. Система должна предоставлять информацию о достопримечательностях
 - 9.1.1. Пешеходный переход
 - 9.1.2. Стена граффити
 - 9.1.3. Здание студии
 - 9.1.4. Видеокамера, снимающая пешеходный переход
 - 9.1.5. Магазин
 - 9.2. Система должна предоставлять ссылки на товары в магазине мерчендайза
 - 9.3. Система должна предоставлять видео со знаменитого перехода
 - 9.3.1. В режиме реального времени
 - 9.3.2. В определенное время с 5:00 до 23:59 текущего дня
 - 9.4. Система должна предоставлять возможность поделиться ссылкой на трансляцию в соцсетях
 - 9.5. Система должна в случайное время делать снимки трансляции и отображать 3 из них на сайте в течении дня
- 10. Система должна реализовывать функции магазина атрибутики
 - 10.1. Должен быть реализован личный кабинет с функционалом:
 - 10.1.1. История заказов
 - 10.1.2. Статус заказа
 - 10.1.3. Избранное
 - 10.2. Должна быть реализована корзина с возможностью добавления и удаления товаров, применения промокодов
 - 10.3. Должен быть реализован поиск по товарам
 - 10.4. Должна быть реализована категоризация товаров

*Принадлежит компании Meta, признанной экстремистской в РФ

Наивный метод

№	Название	Описание	Часы (min)	Часы (max)
Подготовка				
0.1	Макет	Разработать макет сайта в фигме: цветовые решения, формат экранов для разных платформ, айдентика	50	100
0.2	Архитектура	Определиться с используемыми технологиями. Определиться с делением системы на компоненты и с взаимосвязями между ними	25	50
0.3	Видеокамеры	Закупить, повесить и настроить видеокамеры для трансляции видео	10	20
0.4	Разрешение на съемку	Получить в мэрии разрешение на съемку	5	15
0.5	Серверное оборудование	Закупить и настроить серверное оборудование для хостинга системы	50	100
Функциональные требования				
1.1	Отображение новостей на главной странице	10	15	
1.2	Поиск по новостям	3	5	
1.3	Категоризация новостей	2	3	
1.4	Отображение новостей на “Странице новостей”	1	2	
1.5	Сортировка новостей по дате	1	2	
1.6	Деление новостей на страницы	5	10	
2.1	Размещение события на “Главной”	10	15	
2.2	Циклическая смена событий	5	7	
2.3	Редирект на страницу события	1	2	
3.1	Информация о студиях на “Странице студии”	15	20	
3.2	Подробная информация о конкретной студии на “Странице студии”	25	30	
4	Переход на сайт института	1	2	
5	Статьи о студии	5	7	

6.1	Онлайн сведение	75	100
6.2.1	Информация об оффлайн сведении	10	20
6.2.2	Запись на оффлайн сведение	5	7
7	Ссылки на социальные сети студии	2	3
8	Почтовая рассылка	7	10
9.1	Информация о достопримечательностях	15	20
9.2	Ссылки на товары мерчендайза	3	5
9.3	Трансляция видео с пешеходного перехода	50	75
9.4	Возможность поделиться ссылкой на трансляцию	2	3
9.5	Скриншоты трансляции	25	30
10.1	Личный кабинет магазин	50	75
10.2	Корзина магазина	50	75
10.3	Поиск по товарам	20	25
10.4	Категоризация товаров	10	15
Тестирование + Поставка			
11.1	Unit-тестирование	50	75
11.2	Интеграционное тестирование	80	100
11.3	Системное тестирование	50	75
11.4	Деплой приложения на продуктовое окружение	10	15
Сумма		738	1133

PERT

№	Название	Optimistic (h-h)	Pessimistic (h-h)	Optimal (h-h)	E_i	$СКО_i$
0.1	Макет	50	100	70	71,67	8,33
0.2	Архитектура	25	50	40	39,17	4,17
0.3	Видеокамеры	10	20	15	15	1,67

0.4	Разрешение на съемку	5	15	10	10	1,67
0.5	Серверное оборудование	50	100	70	71,67	8,33
1.1	Отображение новостей на главной странице	10	15	12	12,17	0,83
1.2	Поиск по новостям	3	5	4	4	0,33
1.3	Категоризация новостей	2	3	2	2,17	0,17
1.4	Отображение новостей на "Странице новостей"	1	2	1	1,17	0,17
1.5	Сортировка новостей по дате	1	2	1	1,17	0,17
1.6	Деление новостей на страницы	5	10	7	7,17	0,83
2.1	Размещение события на "Главной"	10	15	12	12,17	0,83
2.2	Циклическая смена событий	5	7	5	5,33	0,33
2.3	Редирект на страницу события	1	2	1	1,17	0,17
3.1	Информация о студиях на "Странице студии"	15	20	17	17,17	0,83
3.2	Подробная информация о конкретной студии на "Странице студии"	25	30	27	27,17	0,83
4	Переход на сайт института	1	2	1	1,17	0,17
5	Статьи о студии	5	7	6	6	0,33
6.1	Онлайн сведение	75	100	90	89,17	4,17
6.2.1	Информация об оффлайн сведении	10	20	15	15	1,67
6.2.2	Запись на оффлайн сведение	5	7	6	6	0,33
7	Ссылки на социальные сети студии	2	3	2	2,17	0,17
8	Почтовая рассылка	7	10	8	8,17	0,5
9.1	Информация о достопримечательностях	15	20	17	17,17	0,83

9.2	Ссылки на товары мерчендайза	3	5	4	4	0,33
9.3	Трансляция видео с пешеходного перехода	50	75	60	60,83	4,17
9.4	Возможность поделиться ссылкой на трансляцию	2	3	2	2,17	0,17
9.5	Скриншоты трансляции	25	30	27	27,17	0,83
10.1	Личный кабинет магазин	50	75	60	60,83	4,17
10.2	Корзина магазина	50	75	60	60,83	4,17
10.3	Поиск по товарам	20	25	22	22,17	0,83
10.4	Категоризация товаров	10	15	12	12,17	0,83
11.1	Unit-тестирование	50	75	60	60,83	4,17
11.2	Интеграционное тестирование	80	100	90	90	3,33
11.3	Системное тестирование	50	75	60	60,83	4,17
11.4	Деплой приложения на продуктовое окружение	10	15	12	12,17	0,83

$$E_i = \frac{P_i + O_i + 4M_i}{6} - \text{оценка средней трудоемкости}$$

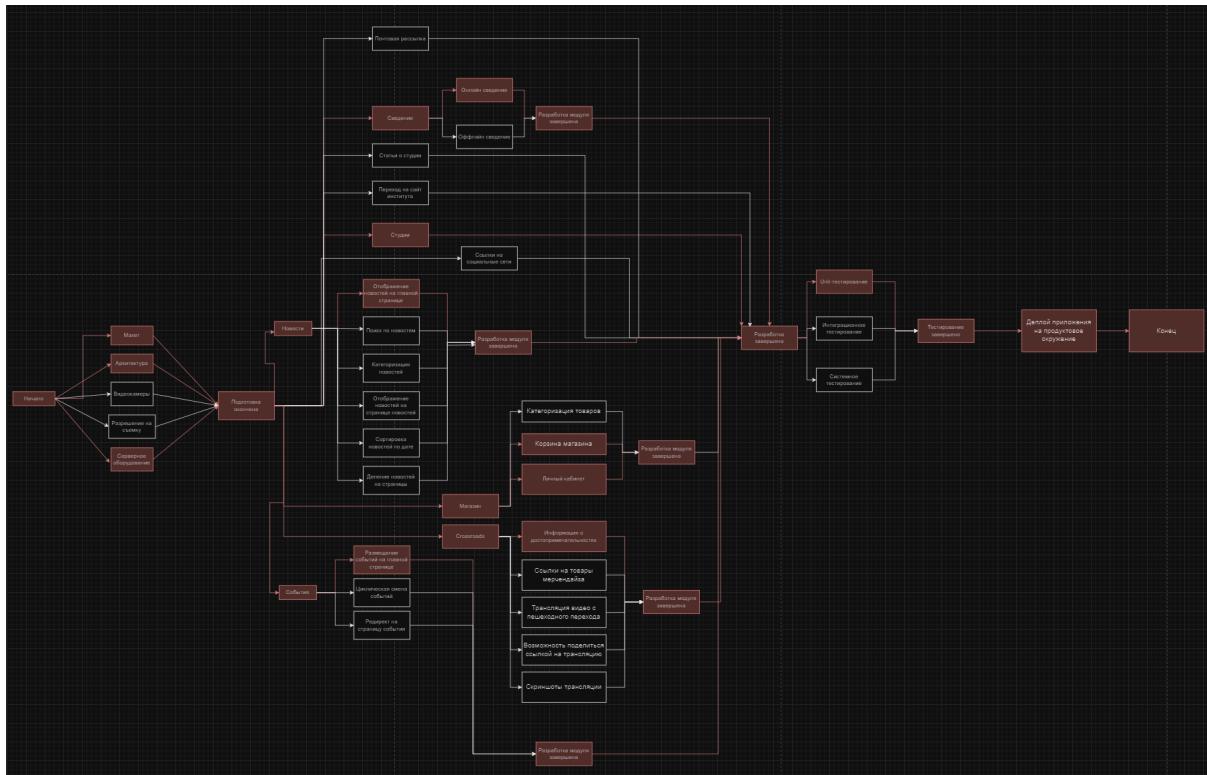
$$\text{СКО}_i = \frac{P_i - O_i}{6} - \text{среднеквадратичное отклонение}$$

$$E = \Sigma E_i = 917,17 - \text{общая оценка статистически независимых работ}$$

$$\text{СКО} = \sqrt{\Sigma \text{СКО}^2} = 16,96 - \text{среднеквадратичное отклонение для оценки общей трудоемкости}$$

$$E_{95\%} = E + 2 * \text{СКО} = 951,09 - \text{суммарная трудоемкость проекта (с вероятностью 95\%)}$$

Метод критического пути



https://drive.google.com/file/d/1N0-_oc62nkIPRquc4_Ir4FK8NuFS0mWy/view?usp=sharing

Критический путь: 552 ч./ч. при ориентире на среднюю оценку трудоемкости и функции, необходимые для MVP

Состав команды:

- 1 аналитик
- 2 frontend-разработчика
- 2 backend-разработчика
- 1 QA-инженер
- 1 Тимлид

Рабочий день - 8 часов (из них полезной работы 7 часов)

Таким образом:

Подготовка (аналитик, тимлид, frontend-разработчики): 13 дней

Разработка: 42 дня

Тестирование: 9 дней

Поставка: 1 день

Общая длительность: 65 дней

Оценка методом функциональных точек

Этапы оценки:

1. Идентификация функциональных компонентов сайта.
2. Классификация каждого компонента по сложности.
3. Присвоение функциональных точек в зависимости от сложности.
4. Суммирование функциональных точек для получения общего размера проекта.

1. Определение функциональных компонентов

Внешние входы (External Inputs - EI):

- **Форма входа/регистрации пользователей** — 1 EI (средней сложности, так как требует ввода данных, проверки и сохранения).
- **Форма подписки на рассылку** — 1 EI (простая).
- **Форма обратной связи (contact us)** — 1 EI (простая).
- **Форма бронирования студийных услуг** — 1 EI (сложная, так как включает множество полей и логики).

Внешние выходы (External Outputs - EO):

- **Отображение информации о студиях и услугах** — 1 EO (средняя сложность).
- **Вывод подтверждения бронирования** — 1 EO (сложная).
- **Вывод страницы новостей и блогов** — 1 EO (простая).

Внешние запросы (External Queries - EQ):

- **Запрос информации о студиях и услугах (поиск по услугам)** — 1 EQ (средней сложности).
- **Запрос доступных слотов бронирования** — 1 EQ (сложная).
- **Запрос информации по ценам** — 1 EQ (средней сложности).

Внутренние логические файлы (Internal Logical Files - ILF):

- **База данных пользователей** — 1 ILF (средняя сложность).
- **База данных бронирований** — 1 ILF (сложная).
- **База данных услуг и цен** — 1 ILF (средняя сложность).
- **База данных новостей и контента** — 1 ILF (простая).

Внешние интерфейсные файлы (External Interface Files - EIF):

- **Интеграция с платежной системой** — 1 EIF (сложная).
- **Интеграция с внешними медиа-файлами (например, видео, музыка)** — 1 EIF (средней сложности).

2. Присвоение функциональных точек на основе сложности

Категория	Компонент	Сложность	Количество FP
EI	Форма входа/регистрации	Средняя	4
	Форма подписки на рассылку	Простая	3
	Форма обратной связи	Простая	3
	Форма бронирования услуг	Сложная	6
EO	Отображение информации о студиях	Средняя	5
	Подтверждение бронирования	Сложная	7
	Новости/блог	Простая	4
EQ	Поиск по услугам	Средняя	4
	Поиск доступных слотов	Сложная	6
	Запрос информации по ценам	Средняя	4
ILF	База данных пользователей	Средняя	7
	База данных бронирований	Сложная	10
	База данных услуг и цен	Средняя	7
	База данных контента	Простая	5
	Интеграция с платежной системой	Сложная	6
	Внешние медиафайлы	Средняя	5

3. Итоговая оценка

Теперь сложим все функциональные точки для оценки общего размера проекта:

- Внешние входы (EI): $4+3+3+6=16$ + 3 + 3 + 6 = $16+3+3+6=16$
- Внешние выходы (EO): $5+7+4=16$ + 7 + 4 = $16+7+4=16$
- Внешние запросы (EQ): $4+6+4=14$ + 6 + 4 = $14+6+4=14$
- Внутренние логические файлы (ILF): $7+10+7+5=29$ + 10 + 7 + 5 = $29+10+7+5=29$
- Внешние интерфейсные файлы (EIF): $6+5=11$ + 5 = $11+5=11$

Общее количество функциональных точек:

$$16+16+14+29+11 = 86 \quad 16 + 16 + 14 + 29 + 11 = 86 \quad 16+16+14+29+11=86$$

4. Оценка

Оценка проекта сайта [Abbey Road Studios](#) по методу функциональных точек составляет **86 функциональных точек**. Это средний проект по размеру, с достаточно разнообразной функциональностью, включая сложные формы бронирования, интеграцию с платежными системами и внешними медиа-файлами.

Расчет трудоемкости методом COCOMO II

Шаги расчета:

1. Преобразование функциональных точек в количество строк кода (SLOC).
2. Использование модели COCOMO II для расчёта трудоёмкости на основе SLOC.
3. Применение формулы COCOMO II для оценки трудоемкости в человеко-месяцах.

Шаг 1: Преобразование функциональных точек в SLOC

Для разных языков программирования существует своя таблица преобразования функциональных точек в строки кода. Выбор языка зависит от технологии, используемой для разработки сайта. Например, для веб-разработки часто используются такие языки и технологии, как:

- **HTML/CSS/JavaScript** для фронтенда;
- **PHP, Python, Ruby** или **Node.js** для серверной логики;
- **SQL** для работы с базами данных.

В обновленной таблице функциональных точек на строку кода используются следующие значения (приблизительные):

Язык программирования	Строк на 1 FP
HTML/CSS	15
JavaScript	50
PHP/Python	50
SQL	100

Допустим, что проект использует **PHP** для серверной части, **JavaScript** для клиентской логики, **HTML/CSS** для верстки, и **SQL** для взаимодействия с базой данных.

Шаг 2: Расчёт SLOC для проекта

Мы ранее оценили проект в **86 функциональных точек**. Теперь распределим их по языкам:

- **HTML/CSS**: Примерно 20% функциональности можно связать с фронтендом и версткой (статические страницы, формы).
- **JavaScript**: Примерно 30% на клиентскую логику (интерактивные элементы, формы бронирования).
- **PHP**: Примерно 40% функциональности — это серверная логика, обработка данных, запросов.

- **SQL:** Примерно 10% на запросы к базе данных.

Теперь рассчитаем количество строк кода для каждого языка:

- **HTML/CSS:** $86 \times 0.2 \times 15 = 258$ строк кода.
- **JavaScript:** $86 \times 0.3 \times 50 = 1290$ строк кода.
- **PHP:** $86 \times 0.4 \times 50 = 1720$ строк кода.
- **SQL:** $86 \times 0.1 \times 100 = 860$ строк кода.

Итоговое количество строк кода (SLOC):

$$258 + 1290 + 1720 + 860 = 4128 \text{ строк кода}$$

Шаг 3: Использование модели СОСОМО II

Теперь, когда мы знаем общее количество строк исходного кода, можем применить модель СОСОМО II. Формула для расчета трудоемкости в человеко-месяцах (PM) в модели СОСОМО II выглядит так:

$$PM = A \times (SLOC/1000)^B \times \prod EM_i$$

Где:

- A — базовый коэффициент (2.94 для органического режима);
- $SLOC$ — количество строк кода;
- B — экспонента, зависящая от характеристик проекта (обычно от 1.05 до 1.25);
- EM_i — факторы, зависящие от специфики проекта (мы примем их равными 1, так как данных нет).

Органический режим (характерен для небольших проектов с хорошо определенными требованиями):

$$A = 2.94$$

$$B = 1.05 \text{ (для проекта небольшой или средней сложности)}$$

Теперь подставим значения:

$$PM = 2.94 \times \left(\frac{4128}{1000}\right)^{1.05}$$

$$PM = 2.94 \times 4.128^{1.05} \approx 2.94 \times 4.24 \approx 12.47 \text{ человеко-месяцев}$$

Вывод

Согласно методу СОСОМО II, проект сайта [Abbey Road Studios](#) потребует около **12.5 человеко-месяцев** на разработку при использовании языка PHP для серверной части,

JavaScript для клиентской логики, HTML/CSS для верстки и SQL для работы с базой данных.

Оценка размера проекта методом оценки вариантов использования (Use Case Points)

Чтобы оценить проект сайта Abbey Road Studios с помощью метода оценки вариантов использования (Use Case Points), мы сначала проведем следующие этапы:

1. **Оценка проекта [ISDB-course](#)** с помощью метода Use Case Points и расчёт фактора продуктивности (Productivity Factor, PF).
2. **Оценка сайта **Abbey Road Studios** методом Use Case Points.**
3. **Применение рассчитанного PF для оценки трудозатрат проекта сайта **Abbey Road Studios**.**

Шаг 1: Оценка проекта **ISDB-course** методом Use Case Points

1.1. Определение основных метрик

Метод UCP основывается на следующих элементах:

- **Акторы** (Actors) — пользователи или другие системы, взаимодействующие с системой.
- **Варианты использования** (Use Cases) — сценарии взаимодействия акторов с системой.

Классификация акторов:

1. **Простые** — акторы, взаимодействующие через API (например, другой сервер). Вес — 1.
2. **Средние** — акторы, использующие текстовый интерфейс (например, командная строка). Вес — 2.
3. **Сложные** — акторы с графическим интерфейсом пользователя (GUI). Вес — 3.

Классификация вариантов использования:

1. **Простые** — включают до 3 транзакций. Вес — 5.
2. **Средние** — включают от 4 до 7 транзакций. Вес — 10.
3. **Сложные** — включают более 7 транзакций. Вес — 15.

1.2. Определение количества акторов и вариантов использования в проекте **ISDB-course**

По анализу проекта ISDB-course (<https://github.com/TivM/ISDB-course>), его функциональность связана с управлением базой данных (ISDB) в рамках учебного курса, и включает в себя взаимодействие с пользователем через интерфейс и работу с данными.

Акторы:

- **Администратор** (сложный) — управляет данными, имеет доступ к различным функциям интерфейса.
- **Пользователь** (средний) — взаимодействует с интерфейсом для просмотра и обновления данных.

Варианты использования:

- **Авторизация пользователя** (простой).
- **Просмотр информации из базы данных** (средний).
- **Обновление записей в базе данных** (средний).
- **Администрирование** (сложный) — создание, редактирование и удаление записей.

1.3. Подсчет количества акторов и вариантов использования для ISDB-course

- **Акторы:**
 - 1 сложный актор (администратор) — 3.
 - 1 средний актор (пользователь) — 2.
- **Итого по акторам:** $3 + 2 = 5$
- **Варианты использования:**
 - 1 простой вариант (авторизация) — 5.
 - 2 средних варианта (просмотр и обновление) — $2 \times 10 = 20$.
 - 1 сложный вариант (администрирование) — 15.
- **Итого по вариантам использования:** $5 + 20 + 15 = 40$

1.4. Подсчет необработанных точек вариантов использования (UUCP)

$$UUCP = (\text{Суммавесаакторов}) + (\text{Суммавесавариантовиспользования})$$

$$UUCP = 5 + 40 = 45$$

1.5. Подсчет технических и экологических факторов

Технические факторы (Technical Complexity Factors, TCF) оцениваются на основе таких характеристик, как сложность архитектуры, производительность, интеграция с другими системами и т.д. Обычно используется 13 технических факторов, каждый из которых оценивается по шкале от 0 до 5. Мы примем стандартные значения для учебного проекта, так как детали этих факторов не указаны:

$$TCF = 0.6 + (0.01 \times \sum W_i) = 0.6 + (0.01 \times 40) = 1.0$$

Внешние факторы (Environmental Factors, EF) также оцениваются с помощью набора факторов (например, опыт команды, использование новых технологий). Примем их по умолчанию для учебного проекта:

$$EF = 1.4 - (0.03 \times \sum W_i) = 1.4 - (0.03 \times 20) = 0.8$$

1.6. Подсчет UCP

$$UCP = UUCP \times TCF \times EF$$

$$UCP = 45 \times 1.0 \times 0.8 = 36$$

1.7. Известные трудозатраты

Предположим, что для проекта ISDB-course были затрачены **200 часов** (это трудоёмкость проекта по имеющимся данным).

1.8. Подсчет Productivity Factor (PF)

$$PF = \frac{\text{Трудоемкость (часы)}}{UCP}$$
$$PF = \frac{200}{36} \approx 5.56 \text{ человека} - \text{часов на } UCP$$

Шаг 2: Оценка сайта Abbey Road Studios методом Use Case Points

Теперь повторим аналогичную процедуру для сайта Abbey Road Studios.

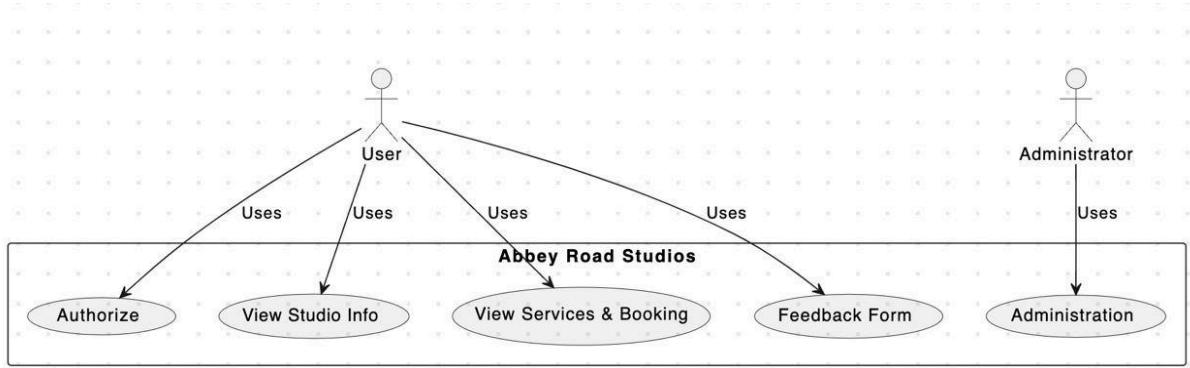
2.1. Определение количества акторов и вариантов использования

Акторы:

- **Пользователь** (сложный) — взаимодействует с графическим интерфейсом (GUI), имеет доступ к бронированию, новостям и услугам.
- **Администратор** (средний) — управляет данными через административный интерфейс.

Варианты использования:

- **Авторизация** (простой).
- **Просмотр информации о студии** (средний).
- **Просмотр услуг и бронирование** (сложный).
- **Обратная связь (форма)** (простой).
- **Администрирование** (сложный).



2.2. Подсчет количества акторов и вариантов использования

- **Акторы:**
 - 1 сложный актор — 3.
 - 1 средний актор — 2.
- **Итого по акторам:** $3 + 2 = 5$
- **Варианты использования:**
 - 2 простых варианта (авторизация, обратная связь) — $2 \times 5 = 10$.
 - 1 средний вариант (просмотр информации) — 10.
 - 2 сложных варианта (бронирование, администрирование) — $2 \times 15 = 30$.
- **Итого по вариантам использования:** $10 + 10 + 30 = 50$.

2.3. Подсчет необработанных точек вариантов использования (UUCP)

$$UUCP = (\text{Суммавесаакторов}) + (\text{Суммавесавариантовиспользования})$$

$$UUCP = 5 + 50 = 55$$

2.4. Подсчет TCF и EF

Примем такие же значения, как и для проекта ISDB-course:

$$TCF = 1.0, EF = 0.8$$

2.5. Подсчет UCP

$$UCP = UUCP \times TCF \times EF$$

$$UCP = 55 \times 1.0 \times 0.8 = 44$$

Шаг 3: Оценка трудозатрат проекта Abbey Road Studios

Теперь, используя рассчитанный PF, оценим трудозатраты для проекта Abbey Road Studios.

Трудозатраты = $UCP \times PF = 4 \times 5.56 = 244.64$ человека – часов ≈ 1.53 человека – месяца

Вывод

Для проекта сайта Abbey Road Studios, по методу Use Case Points и с учетом продуктивности на основе проекта ISDB-course, оценка трудоёмкости составляет **примерно 1.6 человека-месяца**.

Сравнить полученные результаты и сделать выводы

Оценка очень сильно варьируется в зависимости от метода оценки. Кажется, главная причина - разная логика декомпозирования проекта на подзадачи. Разные методы предполагают деление проекта на задачи разного уровня абстракции, а также используемой точки зрения: в Use Case Points мы ориентируемся на варианты использования системы клиентом, а в наивном методе смотрим на проект с точки зрения разработчика/тимлида