Министерство науки и высшего образования Российской Федерации
Университет ИТМО
Мегафакультет компьютерных технологий и управления
Системное и прикладное программное обеспечение
09.03.04 Программная инженерия

Лабораторная работа №1

по “Экономике программной инженерии”

Вариант https://www.zenit.ru/

Выполнил:
Пурэвсурэн Билгуун
группа P34131

Преподаватель:
Машина Е. А.

Санкт-Петербург
2023 г.

Содержание

Текст задания *3*

Выполнение *3*

Наивный метод **3**

PERT метод **6**

Метод функциональных точек **9**

COCOMO II **14**

Use Case Points **17**

Анализ результатов **23**

Вывод *24*

Текст задания

Для https://www.zenit.ru/:

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.

Выполнение

Наивный метод

№	Название	Описание	Optimistic (h-h)	Pessimistic (h-h)	Optimal (h-h)
1	Подготовка		220	880	550
1.1	Анализ проекта	Идет анализ проекта и составляется требования к продукту	60	240	150
1.2	Планирование и проектирование	Выбор технологических платформ и составления архитектуры проекта	70	280	155
1.3	Планирование защиты информации	Так как проект является банком, информационная безопасность является приоритетом.	40	160	100

1.5	Создание инфраструктуры	Так как проект серьезный и масштабный, нам нужен контроль над инфраструктурой: сервера, хостинг, и.т.д	50	200	125
2	Frontend		470	1880	1175
2.1	Проектирование дизайна сайта/приложении	Приготовление дизайна сайта / приложения на платформе Figma	60	240	150
2.2	Создать клиентскую часть веб-сайта (React)	Веб сайт должен быть адаптивной с плавными анимациями, с интуитивным UI/UX	120	480	300
2.3	Создать приложение на смартфонах (React Native)	В наше время приложение банка на смартфонах важнее чем веб-сайт, и он должен быть прост в использовании, отказоустойчивы и при этом употреблять не много ресурсов	120	480	300
2.4	Система аутентификации и авторизации на клиентской части	Система аутентификации и авторизации должны быть комплексной, с высшим уровнем защиты информации	20	80	50
2.5	Создать приложение для внутренней использования	Нужны функционалы, которые не доступны клиентам, но нужны для работниками банка. Например, создать клиента.	150	600	375
3	Backend		502	2008	1255
3.1	Система аутентификации и авторизации на серверной части	Аналогично фронтенду. Приоритет защита информации	40	160	100
3.2	Настройки и подключение БД	Выбор и создание БД, создание схем, таблиц,	62	248	155

		функции и тригеров. Должен быть отказоустойчивым.			
3.3	Создать инфраструктурное решение	Мониторинг, бизнес-аналитика и.т.д является неотъемлемой частью банка.	80	320	200
3.4	Разработка основной банковской системы	Главный функционал банка для работы с деньгами	200	800	500
3.5	Соединение СБП	СБП – в нынешный день, это требование	40	160	100
3.6	Соединение с фронтом и обработка запросов (REST, GraphQL)	Что же бэкенд без фроненда и наоборот.	80	320	200
4	Testing		320	1280	800
4.1	Модульное тестирование	Проверка индивидуальные сервисы бэкенда на надежность и корректную работу.	120	480	300
4.2	Интеграционное тестирование	Тестирование бэкенд как единое целое, и корректное общение с фронтом.	80	320	200
4.3	Функциональное тестирование	Тестирование пользовательские сценарии и стресс тестирование	120	480	300
5	Release		97	388	242
5.1	Сертификаты и документы	Для работы банка нужна много документов	80	320	200
5.2	Деплоймент	Деплоим готовый бек и фронт на внутренней инфраструктуре	12	48	18
5.3	Настройка окружение	Подготовка к деплойменту	15	60	37
Итого			1619	6476	4047.5

PERT метод

№	Название	Optimistic (h-h)	Pessimistic (h-h)	Optimal (h-h)	$E_i = \frac{(P_i + O_i + 4M_i)}{6}$	$СКО_i = \frac{P_i - O_i}{6}$
1.1	Анализ проекта	60	240	150	150	30
1.2	Планирование и проектирование	70	280	155	161.6666667	35
1.3	Планирование защиты информации	40	160	100	100	20
1.5	Создание инфраструктуры	50	200	125	125	25
2.1	Проектирование дизайна сайта/приложении	60	240	150	150	30
2.2	Создать клиентскую часть веб-сайта (React)	120	480	300	300	60
2.3	Создать приложение на смартфонах (React Native)	120	480	300	300	60
2.4	Система аутентификации и авторизации на клиентской части	20	80	50	50	10
2.5	Создать приложение для внутренней использования	150	600	375	375	75
3.1	Система аутентификации и	40	160	100	100	20

	авторизации на серверной части					
3.2	Настройки и подключение БД	62	248	155	155	31
3.3	Создать инфраструктурное решение	80	320	200	200	40
3.4	Разработка основной банковской системы	200	800	500	500	100
3.5	Соединение СБП	40	160	100	100	20
3.6	Соединение с фронтом и обработка запросов (REST, GraphQL)	80	320	200	200	40
4.1	Модульное тестирование	120	480	300	300	60
4.2	Интеграционное тестирование	80	320	200	200	40
4.3	Функциональное тестирование	120	480	300	300	60
5.1	Сертификаты и документы	80	320	200	200	40
5.2	Деплоймент	12	48	18	22	6
5.3	Настройка окружение	15	60	37	37.16666667	7.5

$$E = \sum E_i = 4026$$

$$СКО = \sqrt{\sum СКО_i^2} = 206$$

$$E_{95\%} = E + 2 * СКО = 4438$$

Метод критического пути

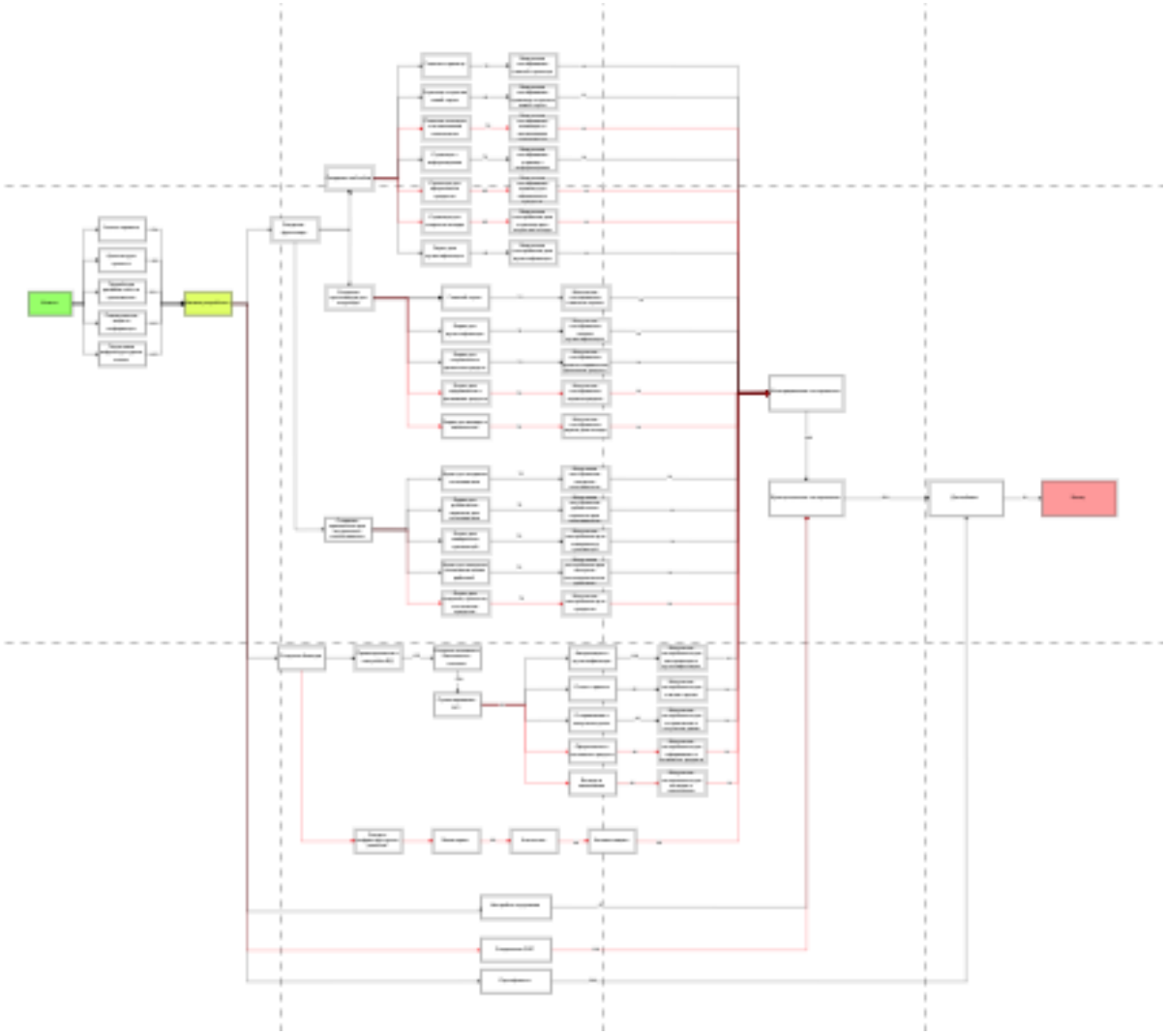


Рисунок 1 Сетевая диаграмма взаимосвязи работ

Критический путь: 3606 ч.ч

Полный путь: 4446 ч.ч

Выполнение проекта: при ориентире на минимальное время разработки получаем, что для выполнения нам необходимо 3606 ч.ч

Команда:

- Аналитик / Системный архитектор 1х
- Фронтэнд-разработчик: 5х
- Бэкэнд-разработчик: 5х
- Тестировщики: 3х
- Дизайнер: 2х
- Информационное безопасность: 5х
- Сетевой инженер: 1х

Рабочий день – 8 часов

Значит наша команда выполнит проект за:

- Анализ и архитектура: 305 часов (38 раб. дней)
- Инфраструктура желез: 125 часов (16 раб. дней)
- Дизайн сайта и приложение: 150 часов (10 раб. дней)
- Подготовка инф. безопасности: 500 часов (13 раб. дней)
- Frontend: 1313 часов (33 раб. дней)
- Бэкенд: 1350 часов (34 раб. дней)
- Интеграционное и функциональное тестирование: 500 часов (32 раб. дней)
- Сертификаты банка: 200 часов (25 раб. дней)
- Релиз: 18 часов (3 раб. дня)

План разработки:

До разработки нужно сделать анализ и архитектуру, до фронтенда нужен дизайн сайта и приложении. А до бэкенда и инфраструктуры нужно подготовить информационную безопасность.

Разработка фронтенда, бэкенда происходят параллельно. А интеграционное и функциональное тестирование делается после завершения разработки. И после все этого мы получаем сертификаты для банковской деятельности и готовимся к релизу.

Время разработки: 305 + 500 + MAX(150,125) + MAX(1313, 1350) + 500 + 200 + 18 = 3005

Общее время: 38 + 13 + MAX(16, 10) + MAX(33, 34) + 32 + 25 + 3 = 161 рабочих дней.

Метод функциональных точек

При анализе методом функциональных точек надо выполнить следующую последовательность шагов:

1. Определение типа оценки
2. Определение области оцеки и границ продукта
3. Подсчет функциональных точек, связанных с данными
4. Подсчет функциональных точек, связанных с транзакциями
5. Определение суммарного кол-ва не выровненных функциональных точек (UFP).
6. Определение значения фактора выравнивания (FAV)
7. Расчет кол-ва выровненных функциональных точек (AFP)

Определение типа оценки

Продукт. Оценивается объем уже существующего и установленного продукта.

Определение области оценки и границ продукта

Все функции. Расчитываем все необходимые (реально используемые), а не дополнительные или только основные функции. Границы продукта показаны в Use-Case диаграмме.

Подсчет функциональных точек, связанных с данными

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

- Фамилия и имя клиента – 2 DET
- Номер паспорта – 1 DET
- Адрес проживания (город, улица, дом, квартира, индекс, номер телефона) – 5 DET
- Контактная информация (номер телефона, электронная почта) – 2 DET

RET (record element type) – логическая группа данных.

- Адрес
- Контакты
- Информация авторизации

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

Таблица 1 Матрица сложности данных

№	Название	RET	DET	Сложность	UFP
1	Личная информация	Личная информация (1)	Фамилия, имя, дата рождения, адрес (5), контактная информация (2) = 10	Low	7
2	Форма создание счета	Личная информация, Данные счета (2)	Номер счета, баланс счета, валюта счета и тип счета. +	Average	10

			(Личная информация) = 13		
3	Форма создание карты	Личная информация, данные счета, данные карты (3)	Номер банковской карты, тип карты, адрес доставки + (данные счета, личная информация) = 16	Average	10

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 inquires) – внешние запросы, элементарная операция, которая в ответ на внешний запрос извлекает данные или управляющую информацию.
- **FTR** (file type referenced) – позволяет подсчитать кол-во 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 FTR	Low	Average	High
3+ FTR	Average	High	High

Таблица 2 Матрица сложности внешних входных транзакций (EI)

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

Таблица 3 Матрица сложности внешних выходных транзакций и внешних запросов (EO & EQ)

Отличия транзакции:

Функция	Тип транзакция		
	EI	EO	EQ
Изменяет поведение системы	Основная	Дополнительная	-
Поддержка одного или более внутренних логических файлов	Основная	Дополнительная	-
Представление информации пользователю	Дополнительная	Основная	Основная

Таблица 4 Отличия транзакции

№	Название	Тип	FTR	DET	Сложность	UFP
1	Внесение данных клиента при открытии нового счета	EI	1	Фамилия, имя, дата рождения, адрес (6), контактная информация (2) = 10	Low	3
2	Генерация выписки по счету в печатном или электронном формате.	EO	1	Номер счета (1), данные о транзакции (4), данные о клиенте (6), баланс счета (1), реквизиты счета (6) = 17	Average	5
3	Запрос остатка средств на счете	EQ	1	Информация о клиенте (6), данные о счетов клиента	Average	4

				(1), баланс счета (1) = 7		
4	Перевод между счетами	EI	2	Данные о счета отправляющей (5), данные о счета получателя (5), баланс отправителя (1), баланс получателя (1) = 12	Low	3
5	Закрытие свой банковский счет	EI	1	Данные о клиента (6), данные о счета (5) = 11	Low	3
6	Уведомление об операциях по счету через SMS	EO	1	Данные о транзакции (4), баланс счета (1) = 5	Low	4
7	Уведомление о предстоящем платеже по кредиту	EO	1	Детали о кредите (5), Детали о платеже (5), Баланс счета (1) = 11	Average	5
8	Запрос кредитного лимита	EQ	1	Информация о клиенте (6), Информация о кредита (8) = 14	Average	4

Определение суммарного количества не выровненных функциональных точек

UFP = 27 + 31 = 58

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	Вес (DI)
1	Обмен данными	5
2	Распределенная обработка данных	5
3	Производительность	5
4	Ограничения по аппаратным ресурсам	0
5	Транзакционная нагрузка	5
6	Интенсивность взаимодействия с пользователей	1
7	Эргономика	5
8	Интенсивность изменения данных	5
9	Сложность обработка	5
10	Повторное использование	5
11	Удобство инсталляции	0
12	Удобство администрирования	2
13	Портируемость	5
14	Гибкость	1
TDI = ΣDI = 49 $VAF = (TDI * 0.01) + 0.65 = 1.14$		

DI (degree of influence)

TDI (total degree of influence)

Расчет количества выровненных функциональных точек (AFP)

$AFP = UPF * VAF = 58 * 1.14 = 66.12$

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 JS (Javascript) – веб
- Flutter (Dart) – мобильное приложение
- Spring Boot (Java) - бэкенд

Разделим функциональность между слоями: 1/3 веб, 1/3 мобильное приложение и 1/3 бэкенд.

Подсчитаем размер по KSLOC:

Таблица коэффициентов

$$KSLOC = UFP * SIZE = \left(58 * \frac{1}{3} * \ 0.063\right) + \left(58 * \frac{1}{3} * 0.080\right) + \left(58 * \frac{1}{3} * 0.134\right) = 5.35$$

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	Значения уровня
PREC	Low	4.96
FLEX	Nominal	3.04
RESL	High	2.83
TEAM	Nominal	3.29
PMAT	Very Low	7.80

Таблица 5 Уровни факторов масштаба

Оценка уровней множителей трудоемкости

Для предварительной оценки прокта необходимо оценить уровень семи множителей трудоемкости M:

- PERS – квалификация персонала
- RCPS – сложность и надежность продукта
- RUSE – разработка для повторного использования
- PDIF – сложность платформы разработки
- PREX – опыт персонала
- FCIL - оборудование
- SCED – требуемое выполнение графика работ

Название	Уровень	Значение
PERS	Very High	0.63
RCPX	Very High	1.91
RUSE	Very Low	-
PDIF	Nominal	1
PREX	High	0.87

FCIL	Very High	0.87
SCED	Nominal	1

Таблица 6 Уровни множителей трудоемкостей

Оценка трудоескомти проекта

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

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

$E = 0.91 + 0.01 * (4.96 + 3.04 + 2.83 + 3.29 + 7.8) = 1.1292$

$PM = 2.94 * 5.35^{1.1292} * (0.63 * 1.91 * 1 * 0.87 * 0.87 * 1) = 17.79$ ч./мес

17.79 ч./мес = 2846.4 ч./ч.

Use Case Points

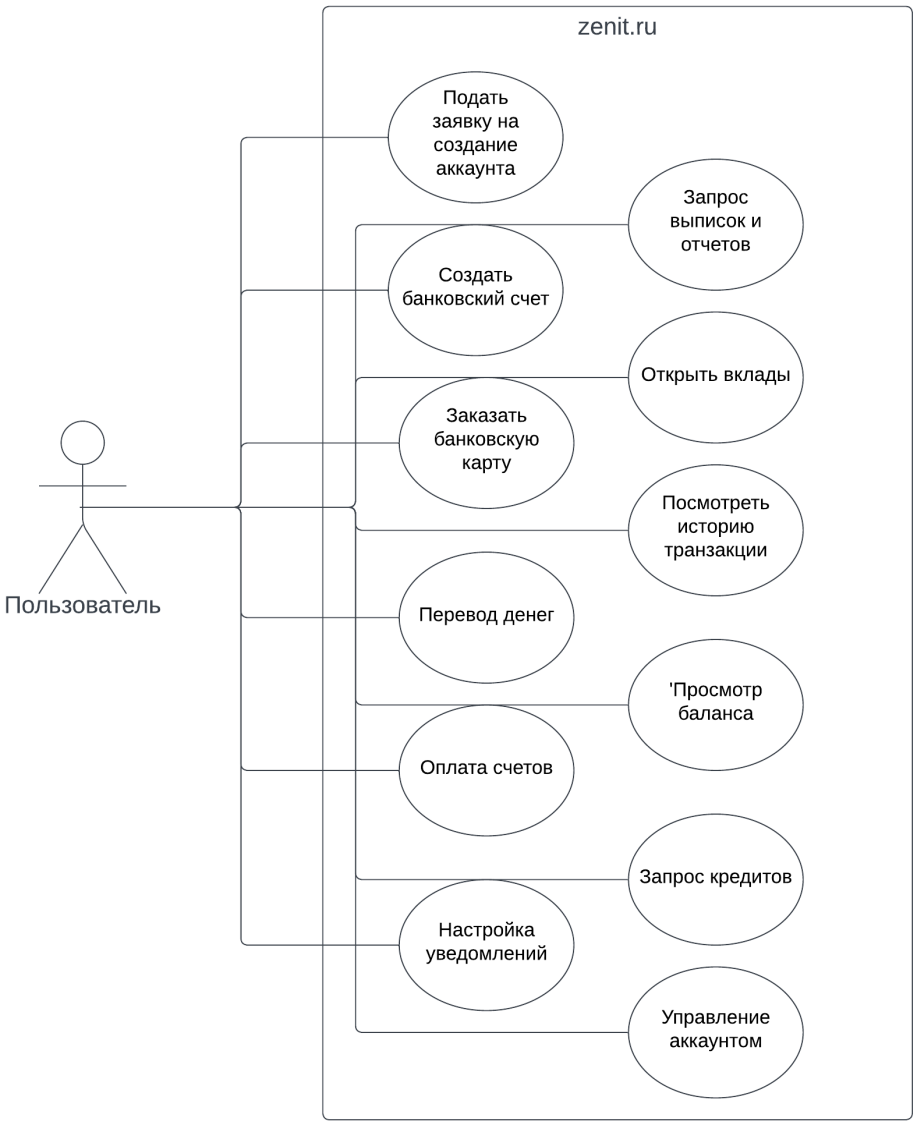


Рисунок 2 Прецеденты пользование пользователя

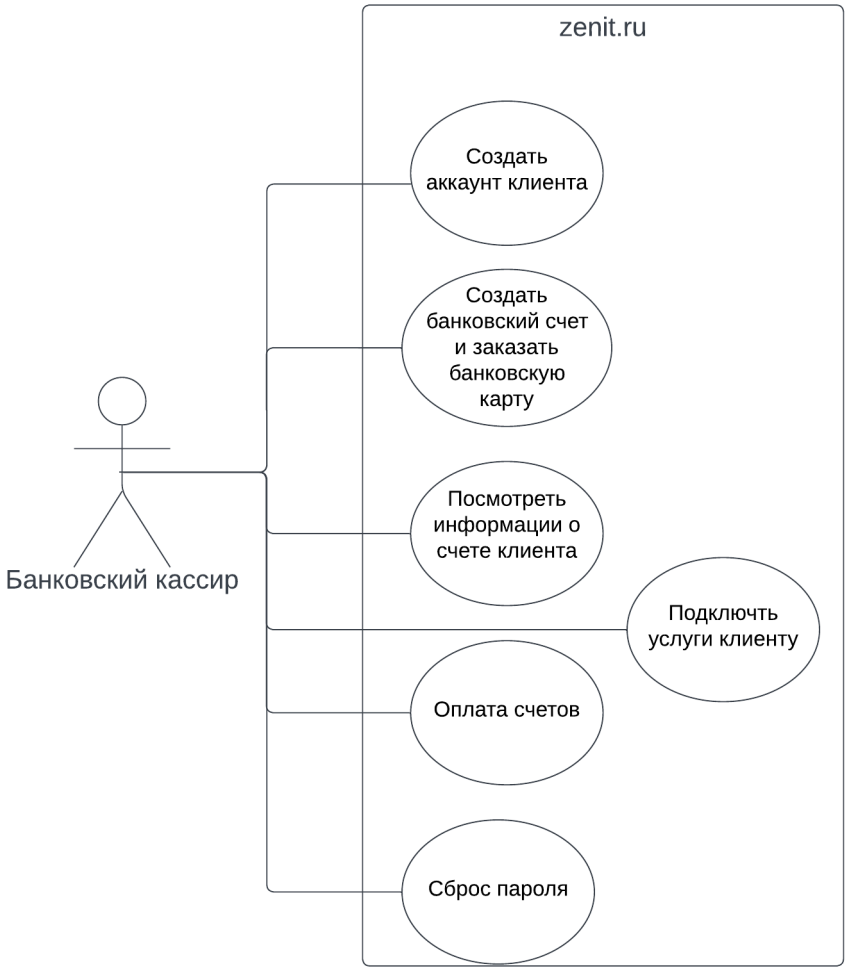
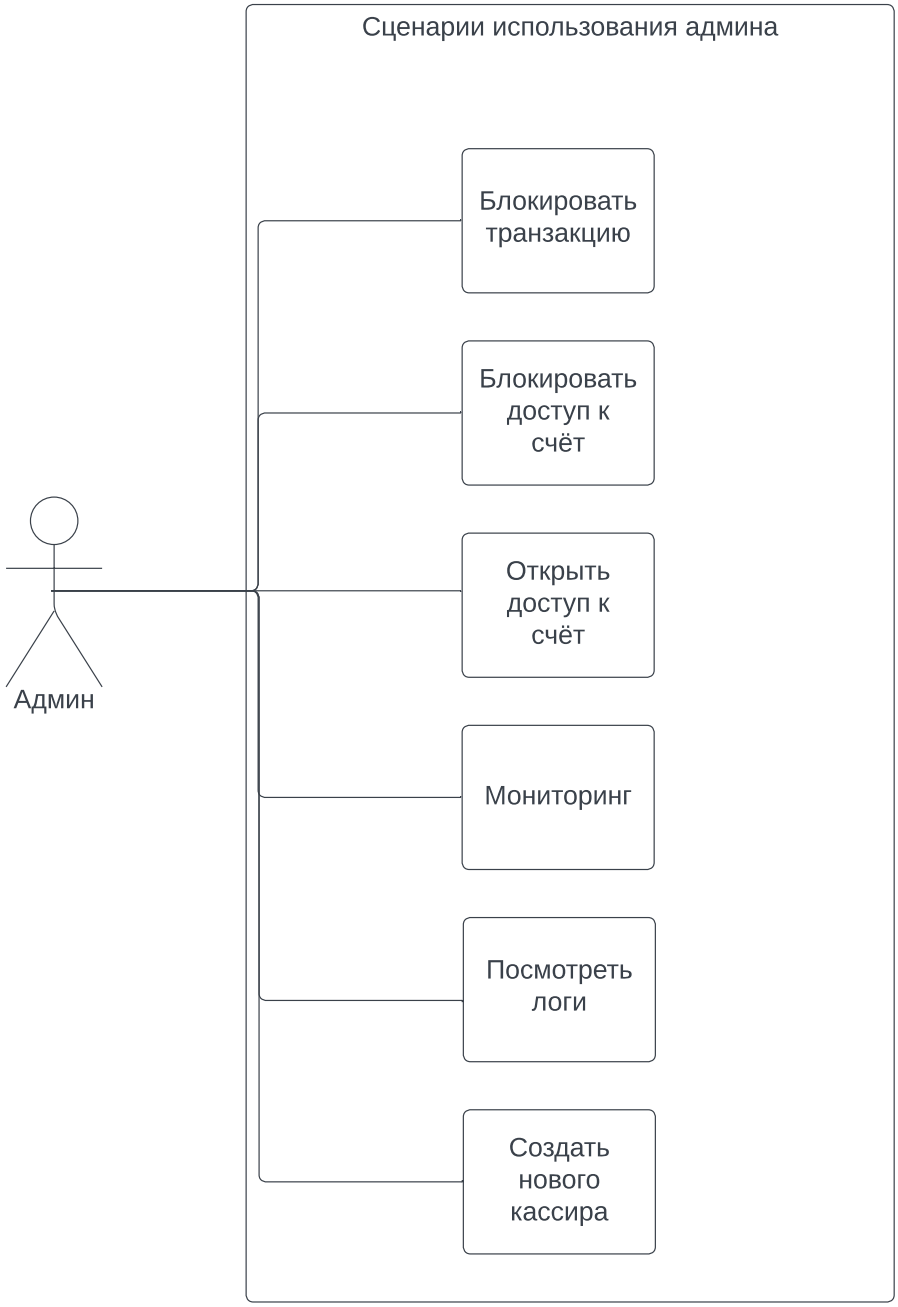


Рисунок 3 Прецендeнты пользование банковского кассира



Оценка веса прецедентов

Сложность	Вес (UCW)	Количество	Затраты
Low	5	13	65
Medium	10	8	80
High	15	3	45
Нескорректированный вес прецента (UCW)			190

Таблица 7 Оценка веса прецедентов

Оценка веса акторов

Сложность	Вес (UAW)	Количество	Затраты
Low	1	0	0
Medium	2	0	0
High	3	3	9
Масса актера без корректировки (UAW)			9

Таблица 8 Оценка веса акторов

Оценка веса технических факторов

Фактор	Вес (W)	Номинальная стоимость (F)	Затраты
Распределенность	2	2	4
Производительность	3	3	9
Эффективность для пользователя	3	3	9
Сложная внутренняя обработка	2	3	6
Повторное использование кода	1	1	1
Простота установки	1	0	0
Простота использования	3	4	12
Переносимость	1	0	0
Простота изменений	2	3	6
Многопоточность	2	2	4
Дополнительные возможности безопасности	3	5	15
Доступ к другим системам	1	0	0
Необходимы тренажеры	1	0	0
Общий технический фактор (Tfactor)			66
TCF = 0.6 + (TF/100)			1.26

Таблица 9 Оценка веса технических факторов

Оценка веса факторов окружения

Фактор	Вес (W)	Номинальная стоимость (F)	Затраты
Знаком с моделью проекта, которая используется	1.5	4	6
Опыт применения	0.5	3	1.5

Опыт в веб/мобильной разработке	1.0	5	5
Возможность ведущего аналитика	0.5	2	1
Мотивация	1.0	2	2
Стабильные требования	1.5	3	4.5
Частичная занятость	-1.0	2	-2
Сложность языка программирования	-1.0	3	-3
Общий фактор окружающей среды (EFactor)			15
ECF = 1.4 + (-0.03 * EF)			0.95

Таблица 10 Оценка веса факторов окружения

Подсчет UCP

$UCP' = (UCW + UAW) * TCF * ECF = (165 + 9) * 1.26 * 0.95 = 208.278$

Подсчет фактора продуктивности (PF) на основе прошлого проекта

Выбранный проект – 3 лабы по БЛПС:

Список Use Case-ов:

№	Сценарий
1	Регистрация
2	Авторизация
3	Поиск статей
4	Просмотр статья
5	Создать статью
6	Удалить свою статью
7	Редактировать свою статью
8	Редактировать свой профиль
9	Оставить комментарий
10	Лайкнуть / дизлайкнуть статью
11	(Admin) Аппрувнуть создание статья
12	(Admin) Удалить статью
13	(Automatic) Уведомить админов на аппрув

Оценка веса прецедентов

Сложность	Вес (UCW)	Количество	Затраты
Low	5	10	50
Medium	10	3	30
High	15	0	0

Нескорректированный вес варианта использования (UCW)	80

Таблица 11 Оценка веса прецедентов предыдущего проекта

Оценка веса акторов

Сложность	Вес (UAW)	Количество	Затраты
Low	1	0	0
Medium	2	0	0
High	3	2	6
Масса актера без корректировки (UAW)			6

Таблица 12 Оценка веса акторов предыдущего проектов

Оценка веса технических факторов

Фактор	Вес (W)	Номинальная стоимость (F)	Затраты
Распределенность	1	1	1
Производительность	1	2	2
Эффективность для пользователя	2	2	4
Сложная внутренняя обработка	1	0	0
Повторное использование кода	1	1	1
Простота установки	1	0	0
Простота использования	2	2	4
Переносимость	1	0	0
Простота изменений	2	2	4
Многопоточность	1	0	0
Дополнительные возможности безопасности	1	1	1
Доступ к другим системам	1	0	0
Необходимы тренажеры	1	0	0
Общий технический фактор (Tfactor)			17
TCF = 0.6 + (TF/100)			0.77

Таблица 13 Оценка веса технических факторов предыдущего проекта

Оценка веса факторов окружения

Фактор	Вес (W)	Номинальная стоимость (F)	Затраты

Знаком с моделью проекта, которая используется	1.5	2	7.5
Опыт применения	0.5	3	1.5
Опыт в веб разработке	1.0	3	3
Возможность ведущего аналитика	0.5	1	0.5
Мотивация	1.0	3	3
Стабильные требования	1.5	1	1.5
Частичная занятость	-1.0	3	-3
Сложность языка программирования	-1.0	3	-3
Общий фактор окружающей среды (EFactor)			11
ECF = 1.4 + (-0.03 * EF)			1.07

Таблица 14 Оценка веса факторов окружения предыдущего проекта

Подсчет UCP предудыщего проекта

$UCP' = (UCW + UAW) * TCF * ECF = (80 + 6) * 0.77 * 1.07 = 70.9$

Подсчет трудоемкости проекта.

Предыдущая работа была выполнена за 210 часов на 2 человек.

PF = E / UCP = 6

UCP = 209 – для zenit.ru

E = PF * UCP = 1254 ч/ч + подготовка, аналитика, архитектура, дизайн, сертификаты = 1254 + 305 + 125 + 150 + 500 + 200 = 2534 ч./ч

Анализ результатов

Метод	Затраты (ч./ч)
Наивный	4047.5
PERT	4438
Функциональных точек	4446
COCOMO II	2846.4
UCP	2534

Как видно из наших результатов, различные подходы приводят к разным выводам. Возможно, низкий результат UCP объясняется неправильным расчетом весов прецедентов. В случае COCOMO II, мы ориентируемся на реальные параметры, вытекающие из анализа личного опыта, но даже в

этом случае результат оказался ниже ожидаемого. Возможно, мой наивный метод дает слишком пессимистичные прогнозы.

Вывод

В ходе лабораторной работы я познакомился с пятью методами оценка проекта. Почувствовал что оценка проекта является одним из сложнейших, не интересных аспектов в области технологии. Так же как и определение реалистичных сроков выполнения. Понял, что чтобы правильно оценить проект нужен огромный опыт в одинаковых проектов.