

Федеральное государственное автономное образовательное
учреждение высшего образования

Университет ИТМО

Дисциплина: Экономика программной инженерии

Лабораторная работа 1

Вариант <http://nordfl.ru/>

Выполнили:

Марков Петр Денисович
Кривоносов Егор Дмитриевич

Группа: Р34111

Преподаватель:

Гаврилов Антон Валерьевич

2022 г.

Санкт-Петербург

Оглавление

Задание	4
Вариант	4
Выполнение	4
Наивный метод	4
PERT метод	8
Метод критического пути	10
Метод функциональных точек	12
Определение типа оценки	12
Определение области оценки и границ продукта	12
Подсчет функциональных точек, связанных с данными	12
Подсчет функциональных точек, связанных с транзакциями	13
Определение суммарного количества не выровненных функциональных точек (UFP)	15
Определение значения фактора выравнивания (FAV)	15
Расчет количества выровненных функциональных точек (AFP)	16
COCOMO II	17
Оценка размера программного продукта в KSLOC	17
Оценка уровней факторов масштаба	17
Оценка уровней множителей трудоемкости	17
Оценка трудоемкости проекта	18
Use Case Points	19
Use Case диаграмма для пользователя:	19
Use Case диаграмма для суперадмина:	20
Оценка веса прецедентов	20
Оценка веса акторов	21
Оценка веса технических факторов	21
Оценка веса факторов окружения	22
Подсчет UCP	22
Подсчёт фактора продуктивности (PF) на основе прошлого проекта	22
Список UseCase-ов:	23
Оценка веса прецедентов	23
Оценка веса акторов	24
Оценка веса технических факторов	24
Оценка веса факторов окружения	25
Подсчет UCP	25
Подсчет трудоемкости проекта:	25
Анализ результатов	26

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 COSOMO 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

<http://nordfl.ru/>

Выполнение

Наивный метод

#	Название	Описание	Optimistic (h-h)	Pessimistic (h-h)	Optimal (h-h)
1	Подготовка		234	508	360
1.1	Прототип сайта	Идет разработка сайт под заказ Северной футбольной лиги г. Санкт-Петербурга. Официальный бренд вместе с дизайном, лого и т.д. уже имеется. Требуется прописать функциональные пользовательские сценарии, прототип анимация и	50	100	70

		интерфейса и т.д.			
1.2	Выбор технологий (фундамент)	Важно определиться с используемыми технологиями: Frontend - AngularJS подойдет нам лучше всего, потому что информационный сайт с красивыми анимациями можно очень быстро на нем написать. Backend - простой PHP сервер для аутентификации пользователей и запись на турниры. Создание собственных турниров не предусмотрено.	8	32	16
1.3	Разрешение на обработку персональных данных	Т.к. мы работаем с футбольными командами, а также многими спонсорами, такое соглашение нам необходимо для правомерной работы сайта.	168	336	250
1.4	Аренда хостинга	В интернете довольно много вариантов для столь простого сайта, поэтому нужно найти лучший по отношению цена/качество.	8	40	24
2	Frontend		147	230	199
2.1	Главная	Адаптивная верстка, плавные анимации, удобное отображение таблиц для четкого понимания турниров и будущих игр, подобрать фото в общей стилистике	16	28	20
2.2	Новости	Адаптивная и корректная верстка новостной ленты, подгрузка новых постов и удобство использования	8	15	12
2.3	Турниры/Профиль команды	Является самой трудоемкой задачей, т.к. они постоянно меняются и нужно заполнять данные для новых команд и турниров, понятное и ненагражденное отображение статистики, верстка таблиц, фото в общей стилистике, а также дополнительные внутренние вкладки с соотв таблицами и статистикой, по сути - профиль для каждого турнира, т.к. профиля пользователя нет. Внутренние таблицы, их верстка	30	50	40

2.4	Команды	Просто удобное отображение в таблице существующих команд, высокое качество логотипов и понятное отображение названий команд. Отступы, адаптивная верстка, шрифты	5	10	8
2.5	Игроки	Аналогично Командам, но только с игроками и футбольными клубами. Отступы, адаптивная верстка, шрифты	10	15	12
2.6	Видео	Аналогично Новостям, только с видео вместо постов, соответственно корректная работа открывания поверх страницы видеоплеера и затемнение остальных элементов, корректная верстка плеера	15	30	23
2.7	Персонал	Аналогично Игрокам, только действующее руководство, сортировка по должности и текущему статусу	5	10	7
2.8	Партнеры	Небольшая вкладка с основными спонсорами, компактные карточки с нужной информацией, их верстка	5	10	7
2.9	Заявиться в соревнования	Форма в хэдере для того, чтобы зарегистрированный пользователь мог поучаствовать в турнирах, просмотр турниров, отображение основной информации	10	15	12
2.10	Форма для регистрации на турнир / О лиге	Аналогична Турниру, только с основной информацией о нем, основная информация об участниках, их сортировка по подтверждениям, верстка таблицы и подбор шрифтов	15	25	19
2.11	Обратная связь	Небольшая форма для обратной связи от футбольной лиги, выезжает вверх, затемнение фона, текст с email и отправка сообщение, возвращение к исходной странице по завершению	3	7	5
2.12	Поиск	Поиск по командам, турнирам и спортсменам. Ищет поля с предыдущих вкладок, поиск по разным именам/названиям. Верстка результатов во внутренних вкладках, переход на <i>Профили команд</i>	15	25	19

2.13	Войти	Простая логинка и соотв страница регистрации, просто ввод email и пароля, затемнение других окон, войти с помощью ВК и внутреннего профиля	10	20	15
3	Backend		160	314	258
3.1	Настройка и подключение БД	Выбор и создание БД, создание таблиц и настройка триггеров, нужна для хранения пользователей, турниров, команд и тд	64	110	90
3.2	Авторизация пользователей	Простейший PHP сервер для отправки запросов на логинку и регистрацию к БД	24	60	48
3.3	Обработка запросов	Отправка всех таблиц из БД, поиск нужных таблиц по БД	72	144	120
4	Testing		108	216	164
4.1	Модульное тестирование	Проверка бэка на надежность, на наличие SQL инъекций и корректную работу запросов к БД. Никакой более сложной логики не предусмотрено, как и других модулей, соотв требуется проверить основной функционал	48	96	72
4.2	Интеграционное тестирование	Тестирование общения фронта и бэка, корректная обработка форм и тд	12	24	20
4.3	Функциональное тестирование	Тестирование полных пользовательских сценариев	48	96	72
5	Релиз		70	144	108
5.1	Тестирование (alpha + beta)	Проверка основного функционала и корректного отображения сайта и его таблиц в различных верстках	48	96	72
5.2	SSL certificate	Self-Signed или отсутствие не подойдет, настройка https соединения	5	9	8
5.3	Выбор окружения	На выбранный хост скинуть скомпилированный html страничку с ангуляром, добавить php сервер	12	24	18
5.4	Настройка	Настройка хоста под наши нужды, проверка общения клиента и сервера	5	15	10
Σ			719	1442	1089

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	50	100	70	71.67	8.33
1.2	Выбор технологий (фундамент)	8	32	16	17.33	4.00
1.3	Разрешение на обработку персональных данных	168	336	250	250.67	28.00
1.4	Аренда хостинга	8	40	24	24.00	5.33
2.1	Главная	16	28	20	20.67	2.00
2.2	Новости	8	15	12	11.83	1.17
2.3	Турниры/ <i>Профиль команды</i>	30	50	40	40.00	3.33
2.4	Команды	5	10	8	7.83	0.83
2.5	Игроки	10	15	12	12.17	0.83
2.6	Видео	15	30	23	22.83	2.5
2.7	Персонал	5	10	7	7.17	0.83
2.8	Партнеры	5	10	7	7.17	0.83
2.9	Заявиться в соревнования	10	15	12	12.17	0.83
2.10	Форма для регистрации на турнир / О лиге	15	25	19	19.33	1.67
2.11	Обратная связь	3	7	5	5.00	0.67
2.12	Поиск	15	25	19	19.33	1.67
2.13	Войти	10	20	15	15.00	1.67
3.1	Настройка и подключение БД	64	110	90	89.00	7.67

3.2	Авторизация пользователей	24	60	48	46.00	6.00
3.3	Обработка запросов	72	144	120	116.00	12.00
4.1	Модульное тестирование	48	96	72	72.00	8.00
4.2	Интеграционное тестирование	12	24	20	19.33	2.00
4.3	Функциональное тестирование	48	96	72	72.00	8.00
5.1	Тестирование (alpha + beta)	48	96	72	72.00	8.00
5.2	SSL certificate	5	9	8	7.67	0.67
5.3	Выбор окружения	12	24	18	18.00	2.00
5.4	Настройка	5	15	10	10.00	1.67

$$E = \sum E_i = 1086.17$$

$$CKO = \sqrt{\sum CKO_i^2} = 37.08$$

$$E_{95\%} = E + 2 * CKO = 1160.33$$

E_i - оценка средней трудоемкости

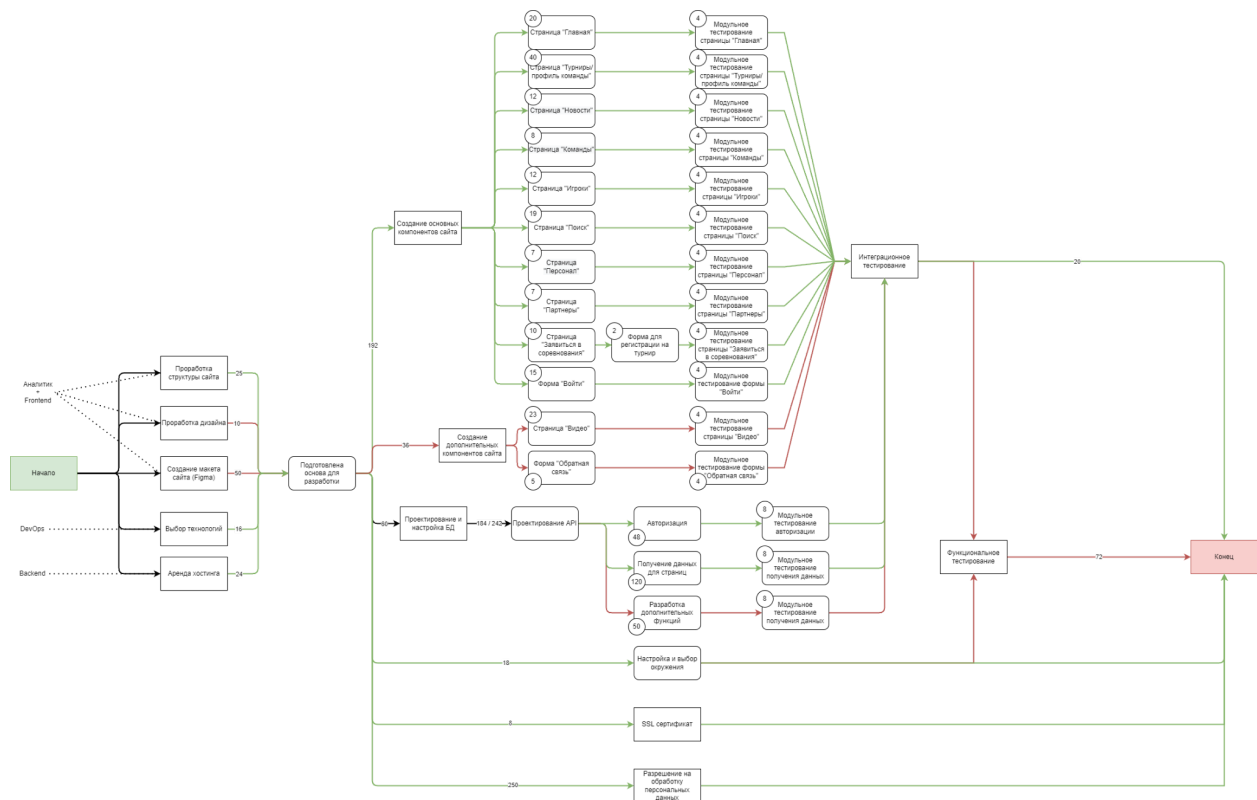
E - общая оценка статически независимых работ

CKO_i - среднеквадратичное отклонение

CKO - среднеквадратичное отклонение для оценки суммарной трудоемкости

$E_{95\%}$ - суммарная трудоемкость проекта (с вероятностью 95%)

Метод критического пути



[Original](#)

Критический путь: 797 ч./ч.

Длинный путь: 1029 ч./ч.

Выполнение проекта: при ориентире на минимальное время разработки (критический путь) получаем, что для выполнения нам необходимо 797 ч./ч.

Команда:

- 1x Аналитик
- 2x Frontend-разработчик
- 2x Backend-разработчик
- 2x Тестировщика

Рабочий день считаем: 8 часов (6 разработки + 1 обед + 1 тех. перерыв)

Таким образом, наша команда сможет выполнить проект за:

- Frontend (с модульным тестированием): 217 часов (37 раб. день)
- Backend (с модульным тестированием): 310 часов (52 раб. день)
- Интеграционное тестирование: 20 часов (4 раб. день)
- Сбор данных: 250 часов (31 раб. день)

- Релиз: 6 часов (1 раб. день)

Рассчитаем время разработки и общее время для завершения проекта:

Frontend и Backend можно делать параллельно, после чего уже делать общее интеграционное тестирование, а также сбор данных для обработки может тоже происходить параллельно. В конце после всех настроек - релиз.

Время разработки: $310 + 20 + 250 = 580$ часов

Общее время: $52 + 4 + 1 = 57$ рабочих дней

Метод функциональных точек

При анализе методом функциональных точек надо выполнить следующую последовательность шагов:

1. Определение типа оценки
2. Определение области оценки и границ продукта
3. Подсчет функциональных точек, связанных с данными
4. Подсчет функциональных точек, связанных с транзакциями
5. Определение суммарного количества не выровненных функциональных точек (UFP)
6. Определение значения фактора выравнивания (FAV)
7. 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	Личный кабинет	Личная информация (1)	имя, фамилия, пол, дата рождения, город, телефон (7)	Low	7
2	Форма регистрации	Данные входа, личная информация (2)	Email, пароль, имя, фамилия, пол, дата рождения, город, телефон (9)	Low	7
3	Форма обратной связи	Контакты пользователя, сообщение (2)	Имя, email, сообщение (3)	Low	7
4	Форма для регистрации на турнир	Личная информация, информация клуба и игрока (3)	имя, фамилия, отчество, дата рождения, команда, тренер команды (имя, фамилия, отчество) игровой номер (10)	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 информацию.

- **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 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
4+ FTR	Average	High	High

Чем отличаются транзакции:

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

№	Название	Тип	FTR	DET	Сложность	UFP
1	Форма обратной связи	EI	1	4	Low	3
2	Форма для регистрации на турнир	EI	1	11	Low	3
3	Форма регистрации	EI	1	10	Low	3
4	Просмотр профиля	EQ	1	8	Low	3
5	Поиск	EQ	3	1	Low	3
6	Просмотр свежих новостей	EO	0	1	Low	4
7	Просмотр видео	EQ	0	1	Low	3
8	Просмотр статических страниц	EI	1	1	Low	3
9	Информация о команде	EO	3	1	Low	4

Определение суммарного количества не выровненных функциональных точек (UFP)

$$UFP = 49 + 29 = 78$$

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	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	0
10	Повторное использование	1
11	Удобство инсталляции	0
12	Удобство администрирования	2
13	Портируемость	1
14	Гибкость	0
$TDI = \sum DI = 12$ $VAF = (TDI * 0.01) + 0.65 = 0.77$		

DI (degree of influence)

TDI (total degree of influence)

Расчет количества выровненных функциональных точек (AFP)

$$AFP = UPF \times VAF = 78 * 0.77 = 60.06$$

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 COSOMO II.

COCOMO II

Оценка размера программного продукта в KSLOC

Стек технологий:

- Angular (TS / JS)
- Backend (PHP)

Разделим функциональность между слоями: $\frac{3}{4}$ - frontend и $\frac{1}{4}$ - backend. Подсчитаем размер по KSLOC:

[Таблица коэффициентов](#)

$$KSLOC = UFP * SIZE = (78 \times \frac{3}{4} \times 0.060) + (78 \times \frac{1}{4} \times 0.053) = 3.51 + 1.0335 = 4.5435$$

Оценка уровней факторов масштаба

- PREC - прецедентность, наличие опыта аналогичных разработок
- FLEX - гибкость процесса разработки
- RESL - архитектура и разрешение рисков
- TEAM - слаботанность 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

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

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

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

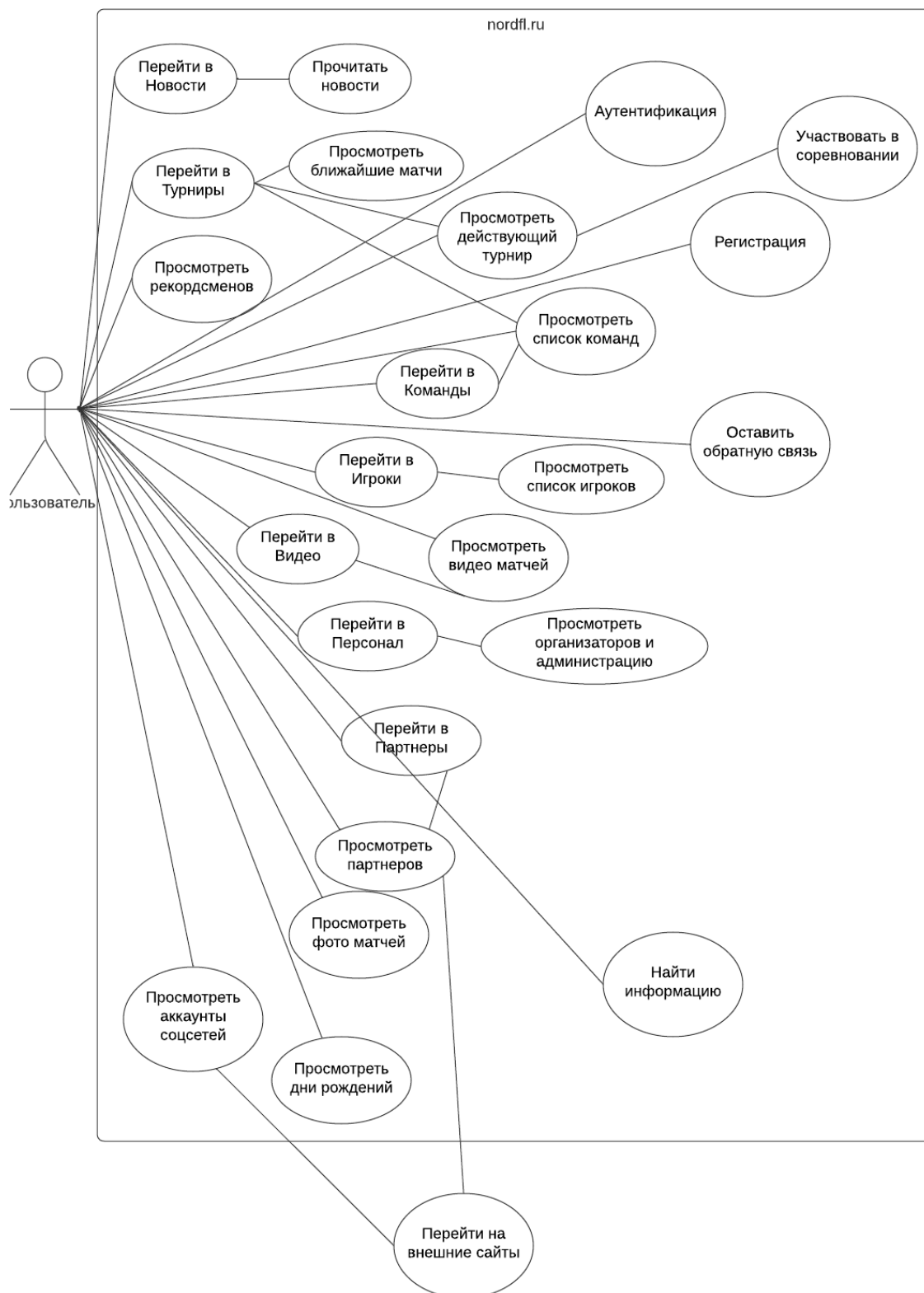
$$E = 0.91 + 0.01 \times (2.48 + 2.03 + 5.65 + 2.19 + 7.80) = 1.1115$$

$$PM = 2.94 \times 4.5435^{1.1115} \times (1.00 * 0.60 * 0.95 * 0.87 * 0.87 * 1.00 * 1.00) \approx 6.82261 \text{ ч. /мес.}$$

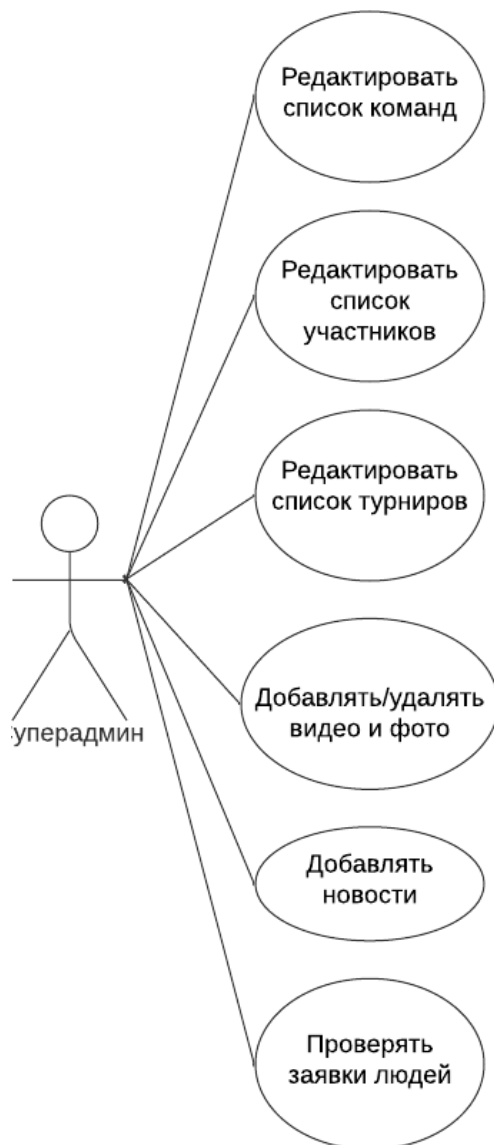
$$6.82261 \text{ ч. /мес.} = 1088 \text{ ч. /ч.}$$

Use Case Points

Use Case диаграмма для пользователя:



Use Case диаграмма для суперадмина:



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
Распределённость	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
Многopotочность	1	0	0
Дополнительные возможности безопасности	1	1	1
Доступ к другим системам	1	2	2
Необходимы тренажеры	1	0	0

для пользователей			
Общий технический фактор (TFactor)			48
TCF = 0.6 + (TF/100)			1.08

Оценка веса факторов окружения

Фактор	Вес (W)	Номинальная стоимость(F)	Затраты
Знаком с моделью проекта, которая используется	1.5	4	6.0
Опыт применения	0.5	3	1.5
Опыт в веб разработке	1.0	4	4.0
Возможность ведущего аналитика	0.5	2	1.0
Мотивация	1.0	2	2.0
Стабильные требования	1.5	2	1.5
Частичная занятость	-1.0	3	-3.0
Сложность языка программирования	-1.0	4	-4.0
Общий фактор окружающей среды (EFactor)			9.0
ECF = 1.4 + (-0.03 * EF)			1.13

Подсчет UCP

$$UCP' = (UCW + UAW) * TCF * ECF = 303$$

Подсчёт фактора продуктивности (PF) на основе прошлого проекта

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

Список UseCase-ов:

#	Сценарий
1	Регистрация
2	Авторизация
3	Поиск машины
4	Поиск продавца
5	Просмотр автомобиля
6	Изменить личные данные в профиле пользователя (Личный кабинет)
7	Создать объявление
8	Отредактировать объявление
9	Выставить рейтинг продавцу
10	Добавление машины в любимое
11	Подписаться на продавца
12	Оставить комментарий
13	Оценить комментарий
14	Поделиться автомобилем
15	Пожаловаться на пользователя
16	Пожаловаться на объявление

Оценка веса прецедентов

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

Нескорректированный вес варианта использования (UUCW)	100
--	-----

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
Распределённость	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
Многopotочность	1.0	1	1.0
Дополнительные возможности безопасности	1.0	1	1.0

Доступ к другим системам	1.0	1	1.0
Необходимы тренажеры для пользователей	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

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

Анализ результатов

Метод	Затраты (h-h)
Наивный	1089
PERT	1160
Функциональных точек	1029
COCOMO II	1088
UCP	735

Как мы видим, COCOMO II и наивный метод показывают практически идентичные результаты. Неудивительно, в COCOMO II мы ориентируемся на различные реальные параметры, которые мы анализируем из личного опыта, что схоже с наивным методом. Они схожие, т.к. ориентируются на наши предположения и реальный опыт. В случае с использованием метода UCP время расчета получилось меньше ожидаемого. Скорее всего такая ситуация могла получиться из-за не совсем верного расчета для уже готового проекта. Вероятно он был оценен в большем количестве UCP, чем на самом деле содержит, либо мы ошиблись с реальным временем его реализации, указав слишком малое время. Кроме того, стоит учитывать, что лабораторные работы сильно отличаются от реальной разработки, так что сильные различия в проектах тоже могут повлиять.

Вывод

В процессе выполнения лабораторной работы мы примерили на себе роль менеджера, пытающегося адекватно выделить функции требуемого проекта и оценить время-затратность его выполнения. Ощутили, что оценивание одна из непростых задач, с которыми приходится сталкиваться в сфере программирования, как и реалистичные сроки выполнения.