



УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники

Экономика программной инженерии

Лабораторная работа №1

Вариант: <https://puzinas.ru/>

Преподаватель: Гаврилов Антон Валерьевич

Выполнили: студенты Кульбако Артемий Юрьевич, Тарасов Александр
Станиславович; Р34112

Наивный метод

#	Название	Описание	P(ч-ч)	O(ч-ч)	M(ч-ч)
Подготовка					
1	Прототип дизайна	Сайт чётко соответствует дизайн-языку компании, множество качественных анимаций, адаптивная вёрстка. Всё выполнено на очень высоком уровне. Страниц достаточно много, хоть и не все особо сложные. К тому же, сайт построен на чистом HTML, CSS, JS, а не на каком-нибудь реактивном фреймворке.	80	50	70
2	Ключ для Telegram	Обе платформы позволяют легко и просто получить ключи для создания ботов.	1	0.5	1
3	Ключ для WhatsApp		1	0.5	1
4	Получение разрешения на обработку персональных данных и подготовка необходимых документов	Сайт хранит персональные данные своих пользователей, а значит является оператором персональных данных. Скорее всего, у данного бизнеса уже есть подобное разрешение, но знать наверняка мы не можем. Интернет подсказывает, что дело это не быстрое и не дешёвое.	300	210	270
Фронтенд					
5	Главная страница	Калькулятор, карусель статей из блога, форма обратной связи + статика.	32	22	28
6	Блог	Динамически подгружаемые статьи с помощью WordPress.	24	12	19
7	Портфолио	Динамически подгружаемые карточки с помощью WordPress.	24	12	19
8	Компания	Форма обратной связи и виджет Яндекс Карт.	24	12	19
9	Услуги	Полностью статичная страница, за исключением калькулятора расчёта стоимости работ с запросом на бекенд.	24	12	19
10	Прочие страницы	Исключительно статический контент, но страниц не мало.	48	30	40

11	Интеграционное тестирование	Проверить корректность взаимодействия фронтенда с бекендом	10	4	8
12	Модульное тестирование	Проверить адаптивность всех страниц и переходы (ручное тестирование)	4	2	3
Бекенд					
13	Telegram бот	Функциональность аналогична формам обратной связи с сайта, делается быстро.	14	6	8
14	WhatsApp бот	Функциональность аналогична формам обратной связи с сайта, делается быстро.	14	6	8
15	WordPress	Необходимо отдавать на фронт контент вида "текст и картинки", что не очень сложно, а сам WordPress достаточно простой в освоении.	16	9	13
16	Проектирование API	Формы обратной связи на фронте дублируют друг друга. Отличается лишь одна - калькулятор расчёта стоимости. Скорее всего, на бекенде не больше двух эндпоинтов.	2	1	2
17	Обработка запросов	Записать данные из форм в БД.	2	1	2
18	Рассылка email	Рассылка автоматизированных сообщений клиенту об успешном принятии заявки. Скорее всего, также сообщение отправляется менеджеру или дизайнеру с уведомлением о новой заявке.	2	1	2
19	Модульное тестирование	Тестирование обоих модулей бекенда.	4	2	3
Релиз					
20	Получение SSL-сертификата	Заказать SSL-сертификат.	8	4	6
21	Создание docker-образов	Написание Dockerfile, запуск и проверка.	6	2	3
22	Выбор сервера	Выбрать где разворачивать приложения, взвесить плюсы и минусы, развернуть HelloWorld для	14	2	8

		проверки.			
23	Деплой	Развернуть docker-образы на выбранном сервере.	12	3	6
Итого =			666	404	558

PERT

#	Название	P(ч-ч)	O(ч-ч)	M(ч-ч)	E	СКО
Подготовка						
1	Прототип дизайна	80	50	70	68,3	5,0
2	Ключ для Telegram	1	0.5	1	0,9	0,1
3	Ключ для WhatsApp	1	0.5	1	0,9	0,1
4	Получение разрешения на обработку персональных данных и подготовка необходимых документов	300	210	270	265,0	15,0
Фронтенд						
5	Главная страница	32	22	28	27,7	1,7
6	Блог	24	12	19	18,7	2,0
7	Портфолио	24	12	19	18,7	2,0
8	Компания	24	12	19	18,7	2,0
9	Услуги	24	12	19	18,7	2,0
10	Прочие страницы	48	30	40	39,7	3,0
11	Интеграционное тестирование	10	4	8	7,7	1,0
12	Модульное тестирование	4	2	3	3,0	0,3
Бекенд						
13	Telegram бот	14	6	8	8,7	1,3
14	WhatsApp бот	14	6	8	8,7	1,3
15	WordPress	16	9	13	12,8	1,2
16	Проектирование API	2	1	2	1,8	0,2

17	Обработка запросов	2	1	2	1,8	0,2
18	Рассылка email	2	1	2	1,8	0,2
19	Модульное тестирование	4	2	3	3,0	0,3
Релиз						
20	Получение SSL-сертификата	8	4	6	6,0	0,7
21	Создание docker-образов	6	2	3	3,3	0,7
22	Выбор сервера	14	2	8	8,0	2,0
23	Деплой	12	3	6	6,5	1,5
$E = \sum_i E_i = 601.67$ $\sum \sqrt{CKO_i^2} = 18.03$ $E_{95\%} = E + 2 * CKO = 637.73$						

Метод функциональных точек

Определение типа оценки

Тип оценки - продукт.

Определение области оценки и границ продукта

Область оценки - все функции продукта.

Внутренние логические файлы

Внешние интерфейсные файлы - Telegram API, WhatsApp API, почтовый сервер.

Граница системы - определены на UseCase диаграмме.

Подсчет функциональных точек, связанных с данными

Область оценки - все функции продукта.

#	Имя	RET	DET	Сложность	UPF
1	Простая форма обратной связи	Контакты пользователя	(2) Имя, email	Low	7
2	Задания	Основы бренда	(7) Название, логотип, визитная карточка, фирменный бланк,	Average	10

			папка, конверт, вывеска		
		Фирменный стиль	(7) Брендинг одежды, брендинг транспорта, оформление интерьера, сувенирная продукция, упаковка/этикетка, листовка/флаер, дизайн рекламы		
		web	(5) Корпоративный сайт, интернет-магазин, оформление соц. сетей, продвижение сайта, продвижение в соц. сетях		
3	Сложная обратная связь	Контакты пользователя	(7) Я, хочу, мне нужно, имя, email, телефон, сообщение	Low	7
4	Бриф	База	(5) С нуля, название, сфера деятельности, значение, когда создано	High	15
		Экономика	(6) Территория, сайт, конкуренты, преимущество, характеристики, ценовая категория		
		Сурсы	(5) Прошлый дизайн, носители, формат, образы, стиль		
		Дизайн-язык	(4) Ассоциации, образы, слоган, тон		
		Референсы	(2) Не рекомендуемые логотипы, не рекомендуемые стили		
		Задачи	(3) Носители, сайт,		

			продвижение		
		Менеджмент	(5) Готовность, совет управления, имя, email, телефон		

Подсчет функциональных точек, связанных с транзакциями

#	Имя	Тип	FTR	DET	Сложность	UPF
1	Отправка формы обратной связи	EI	1	2	Low	3
2	Отправка брифа	EI	7	30	High	6
3	Общение через мессенджер	EQ	1	1	Low	3
4	Калькулятор стоимости	EO	3	20	High	6
5	Просмотр блога	EI	1	4	Low	3
6	Просмотр отзывов	EI	1	4	Low	3
7	Просмотр статических страниц	EI	1	1	Low	3
8	Рассылка почты	EQ	2	4	Average	4

Определение суммарного количества не выровненных функциональных точек (UFP)

$$\Sigma UFP = 39 + 31 = 70$$

Определение значения фактора выравнивания (FAV)

#	Фактор	Вес
1	Обмен данными	1
2	Распределенная обработка данных	0
3	Производительность	0
4	Ограничения по аппаратным ресурсам	0
5	Транзакционная нагрузка	0
6	Интенсивность взаимодействия с пользователем	0
7	Эргономика	3

8	Интенсивность изменения данных	1
9	Сложность обработки	0
10	Повторное использование	2
11	Удобство инсталляции	0
12	Удобство администрирования	3
13	Портируемость	1
14	Гибкость	0
$TDI = \sum DI = 10$ $VAF = (TDI * 0.01) * 0.65 = 0.75$		

Расчет количества выровненных функциональных точек (AFP)

$$AFP = UFP * VAF = 70 * 0.75 \approx 53$$

Метод функциональных точек слабо подходит для расчёта полного цикла создания ПО (нельзя учесть проектирование, составление ТЗ, а также задачи по созданию статического контента), поэтому работа с документами, а также этапа прототипирования и релиза. будут взяты из PERT. Одну функциональную точку оценим в 8 ч. Таким образом:

$$\Sigma = (53 * 8) + 23 + 330 = 777 \text{ ч/ч}$$

COCOMO II

Оценка размера программного продукта в KSLOC

Стеком проекта будет JS + JQuery на фронте (это можно узнать достоверно), а бэкэнд на WordPress и, предположим, Spring MVC. Большая часть проекта - это фронт, поэтому разделим функциональные между слоями как $\frac{3}{4}$ фронт и $\frac{1}{4}$ бэк. Подсчитаем размер по KSLOC:

$$KSLOC = UFP * SIZE = (70 * \frac{3}{4} * 0.056) + (70 * \frac{1}{4} * 0.061) = 3.949$$

Оценка уровней факторов масштаба

#	Фактор	Уровень	Значение
1	PREC	High	2.48
2	FLEX	High	2.03
3	RESL	Low	5.65

4	TEAM	Extra High	0.00
5	PMAT	Very Low	7.80

Оценка уровней множителей трудоемкости

#	Фактор	Уровень	Значение
1	PERS	Nominal	1.00
2	RCPX	Extra Low	0.49
3	RUSE	Extra Low	n/a
4	PDIF	Extra Low	n/a
5	PREX	High	0.87
6	FCIL	Nominal	1.00
7	SCED	High	1.00

Оценка трудоёмкости проекта

$$PM = A * SIZE^E * \prod_{i=1}^n EM_i, \quad A = 2.94, \quad E = B + 0.01 * \sum_{j=1}^5 SF_j, \quad B = 0.91$$

$$PM = 2.94 * 3.949^{(0.91+0.01*(2.48+2.03+5.65+0+7.8))} * 1 * 0.49 * 0.87 * 1 * 1 = 4.948 \text{ чел/мес}$$

$$4.948 * 20 * 8 \approx 792 \text{ ч/ч}$$

$$UCM = \sum UCW_i * N_i = 140$$

Оценка веса акторов

Сложность	Вес (AUW)	Количество	Затраты
Low	1	2	2
Medium	2	0	0
High	3	2	6
$UAW = \sum AW_i * N_i = 8$			

Оценка веса технических факторов

Фактор	Вес (W)	Сложность (F)	Затраты
Распределённость	2	0	0
Производительность	1	0	0
Эффективность для пользователя	1	4	4
Сложная внутренняя обработка	1	2	2
Повторное использование кода	1	0	0
Простота установки	0.5	0	0
Простота использования	0.5	5	2.5
Переносимость	2	1	2
Простота модификации	1	1	1
Многopotочность	1	0	0
Дополнительная безопасность	1	1	1
Взаимодействие с другими системами	1	2	2
Необходимость в тренажёрах	1	0	0
$TCF = C_1 + C_2 * \sum W_i * F_i = 0.6 + 0.01 * 14.5 = 0.745$			

Определение веса факторов окружения

Фактор	Вес (W)	Важность (F)	Затраты
Качественная модель	1.5	1	1.5
Количество работников на неполный рабочий день	-1	0	0
Опыт аналитика	0.5	1	0.5
Опыт работы с средствами разработки	0.5	4	2
Опыт ОО разработки	1	5	5
Мотивация разработчиков	1	2	2
Сложность технологического стека	-1	0	0
Неизменность требований	2	3	6
$ECF = C_1 + C_2 * \sum W_i * F_i = 1.4 + (-0.03 * 17) = 0.89$			

Подсчёт UCP

$$UCP = (UCW + UAW) * TCF * ECF = (140 + 8) * 0.745 * 0.89 \approx 98$$

Прошлый похожий проект нашими разработчиками был выполнен за 750 ч/ч.

Исходя из этого, рассчитаем фактор продуктивности:

$$PF = E/UCP = 750/98 \approx 8$$

Оценим трудоёмкость проекта:

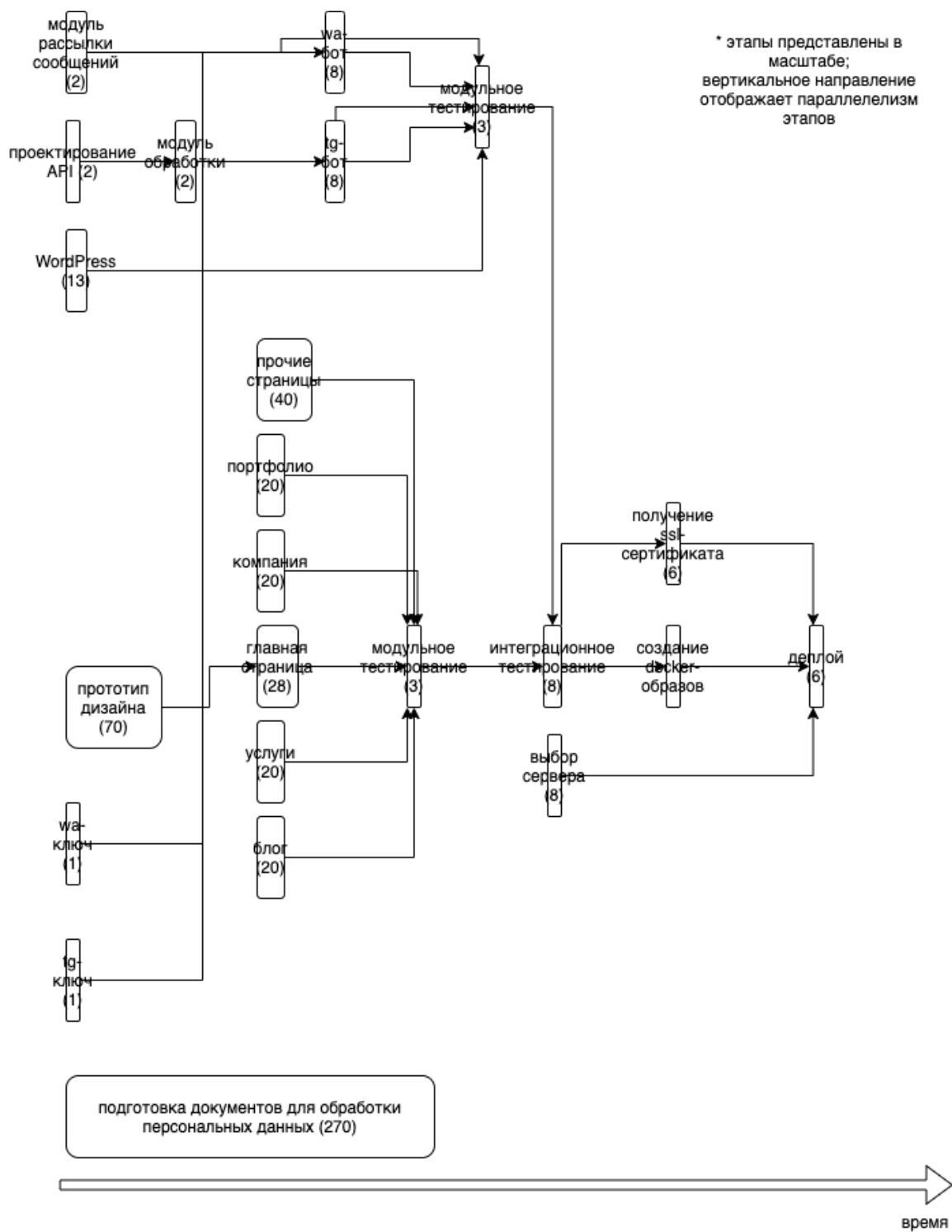
$$E = UCP * PF = 98 * 8 = 784 \text{ ч/ч}$$

Команда

Смотря на план разработки, для минимизации времени разработки проекта нам потребуется дизайнер, юрист (или нотариус, или адвокат, или вообще кто-то, кто разбирается в законах), фронтендер, бекендер, тестировщик для всех этапов и девопс. Естественно, такая команда излишняя для проекта, который вряд ли будет активно развиваться. Созданием ботов для мессенджеров может заняться бекендер, пока фронтендер будет делать макет сайта, таким образом, эти этапы должно завершиться одновременно, далее каждый из них займётся тестированием своей части и вместе деплоем. Человека, который займется бюрократической волокитой, можно единоразово привлечь со стороны. Таким образом: фронтендер, бекендер, юрист.

$$\Sigma = (\text{общие ч/ч} - \text{время на релиз}) / 3 + \text{время на релиз} = (601 - 23) / 3 + 23 = 21$$

План разработки



Вывод

Метод	Затраты (ч/ч)
-------	---------------

Наивный	558
PERT	638
Функциональных точек	777
УСР	784
COCOMO II	792

В процессе выполнения лабораторной работы нам на время пришлось побыть менеджерами проекта и на личном опыте убедиться, что подобная деятельность несколько не легче непосредственно программирования, ибо от адекватности планирования и сроков во многом зависит успех проекта. При планировании разработки мы максимально попытались распараллелить этапы разработки, чтобы сократить время общее время создания проекта.

Интересная тенденция, что с усложнением метода увеличивалось предполагаемое количество ч/ч, радуется, что разница не достигла множителя 2х.

Лабораторная показала, что расчёт плана проекта - очень сложный процесс, требующий опыта от менеджера, ведь большая часть оценок, даже при разбивке на usecase-ы или функциональные точки, субъективны.

Использованная литература

- http://citforum.ru/SE/project/arkhipenkov_lectures/12.shtml
- http://citforum.ru/SE/project/arkhipenkov_lectures/13.shtml
- https://ru.wikipedia.org/wiki/Capability_Maturity_Model
- <https://habr.com/ru/company/orioninc/blog/574956/>
- <https://coderlessons.com/tutorials/akademicheskii/izuchite-metody-otsenki/metody-otsenki-tochki-ispolzovaniia>