



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический  
университет имени Н.Э. Баумана»  
(МГТУ им. Н.Э. Баумана)

---

ФАКУЛЬТЕТ «Информатика и системы управления»  
КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

---

## Лабораторная работа № 7

Дисциплина	Экономика программной инженерии.
Тема	Оценка параметров программного проекта с использованием метода функциональных точек и модели СОСОМО II.
Студенты	Сиденко А.Г., Степанов А.О.
Группа	ИУ7-83Б
Оценка (баллы)	
Преподаватель	Барышникова М.Ю., Силантьева А.В.

Москва, 2021 г.

## 1. Постановка задачи

Компания получила заказ на разработку клиентского мобильного приложения брокерской системы. Программа позволяет просматривать актуальную биржевую информацию, производить сделки и отслеживать их выполнение. Приложение имеет 4 страницы: авторизация, биржевые сводки, заявки, новая заявка.

## 2. Расчет по методу функциональных точек

Произведем подсчет количества функциональных точек.

Для начала определимся с хранением, нам можно хранить данные как в обычном файле, так и в таблице СУБД. Всего будет 5 таблиц. 4 из них внутренние: 1 таблица – логин, пароль; 2 таблица – тип заявки, имя бумаги, цена, количество; 3 таблица – название бумаг (которые мы отслеживаем на бирже), 4 таблица – токен (если запоминаем параметры). И 1 внешняя с информацией о бирже – название бумаги, цена, изменения.

- Внешние вводы (EI)

Имеется 5 внешних вводов:

### **Добавить бумагу**

Он ссылается на один внутренний логический файл (3 таблица) и имеет 2 элемента данных (кнопка и название бумаги).

**FTR** = 1

**DET** = 2

### **Удалить заявку**

Он ссылается на один внутренний логический файл (2 таблица) и имеет 5 элементов данных (тип, имя, цена, количество, кнопка).

**FTR** = 1

**DET** = 5

### **Изменить заявку**

Он ссылается на один внутренний логический файл (2 таблица) и имеет 5 элементов данных (тип, имя, цена, количество, кнопка).

**FTR** = 1

**DET** = 5

### **Создать заявку**

Он ссылается на один внутренний логический файл (2 таблица) и имеет 5 элементов данных (тип, имя, цена, количество, кнопка).

**FTR** = 1

**DET** = 5

### **Сохранение токена**

Он ссылается на один внутренний логический файл (4 таблица) и имеет 4 элемента данных (логин, пароль, кнопка и флажок).

**FTR** = 1

**DET** = 4

Уровень сложности для всех внешних вводов низкий.

- Внешние выводы (EO)

В ПО имеется также 2 внешних вывода (EO).

**Вывод списка заявок**

Он ссылается на один внутренний логический файл (2 таблица) и имеет 4 элемента данных (тип, имя, цена, количество).

**FTR** = 1

**DET** = 4

**Вывод биржевых сводок**

Он ссылается на один внутренний логический файл (3 таблица) и внешний интерфейсный файл и имеет 4 элемента данных (имя, цена, изменения).

**FTR** = 2

**DET** = 4

Уровень сложности для всех внешних выводов низкий.

- Внешние запросы (EQ)

Имеется один внешний запрос на авторизацию. Он ссылается на один внутренний логический файл (1 таблица) и имеет 4 элемента данных (логин, пароль, кнопка и флажок).

**FTR** = 1

**DET** = 4

Уровень сложности для внешнего запроса низкий.

- Внутренние логические файлы (ILF)

Как мы определились, есть 4 внутренних логических файла, которые имеют до 4 различных типов элементов записей и до 4 элементов данных.

**RET** = 4

**DET** = 4

Соответственно уровень сложности для внутренних логических файлов низкий.

- Внешние интерфейсные файлы (EIF)

Также есть 1 внешний интерфейсный файл, 2 типа элементов записей и 3 элемента данных.

**RET** = 2

$$DET = 3$$

Таким образом, уровень сложности – низкий.

Рис. 1: Ненормированное количество функциональных точек.

Характеристика	Количество	Уровень сложности	Итог
Внешние вводы (EI)	5	Низкий (3)	15
Внешние выводы (EO)	2	Низкий (4)	8
Внешние запросы (EQ)	1	Низкий (3)	3
Внутренние логические файлы (ILF)	4	Низкий (7)	28
Внешние интерфейсные файлы (EIF)	1	Низкий (5)	5
			59

Зададим характеристики продукта:

Разработанное ПО состоит из трех компонентов. Первый компонент составляет по объему примерно 15% программного кода и будет написан на SQL, второй (около 60% кода) - на C#, а третий в объеме 25% кода - на Java.

- Обмен данными - 5.
- Распределенная обработка-5
- Производительность -3
- Эксплуатационные ограничения по аппаратным ресурсам-2
- Транзакционная нагрузка-3
- Интенсивность взаимодействия с пользователем (оперативный ввод данных) – 4
- Эргономические характеристики, влияющие на эффективность работы конечных пользователей - 1

- Оперативное обновление–4
- Сложность обработки–4
- Повторное использование – 0
- Легкость инсталляции – 1
- Легкость эксплуатации/администрирования – 2
- Портитруемость – 2
- Гибкость- 2

И тогда,

Рис. 2: Нормированное количество функциональных точек.

Обмен данными	5	Характеристика	Количество	Уровень сложности	Итог
Распределенная обработка данных	5				
Производительность	3	Внешние вводы (EI)	5	Низкий (3)	15
Эксплуатационные ограничения по аппаратным ресурсам	2				
Транзакционная нагрузка	3	Внешние выводы (EO)	2	Низкий (4)	8
Интенсивность взаимодействия с пользователем (оперативный ввод данных)	4				
Эргономические характеристики, влияющие на эффективность работы конечных пользователей	1	Внешние запросы (EQ)	1	Низкий (3)	3
Оперативное обновление	4				
Сложность обработки	4	Внутренние логические файлы (ILF)	4	Низкий (7)	28
Повторное использование	0				
Легкость инсталляции	1	Внешние интерфейсные файлы (EIF)	1	Низкий (5)	5
Легкость эксплуатации/администрирования	2				
Портитруемость	2	<b>Нормированное количество функциональных точек: 60.77</b> <b>Количество функциональных точек: 59</b> <b>Количество строк исходного кода: 3877</b> <b>Коэффициент: 1.03</b>			59
Гибкость	2				
Рассчитать		Выбрать технологический стек			

### 3. Произвести оценку трудозатрат и длительности разработки по методике СОСОМО II с использованием модели композиции приложения.

Для реализации проекта была сформирована новая команда разработчиков, у отдельных членов которой имеется некоторый опыт создания систем подобного типа. В целях сплочения команды были проведены определенные мероприятия, что обеспечило на старте проекта приемлемую коммуникацию внутри коллектива. Заказчик не настаивает на жесткой регламентации процесса, однако график реализации проекта довольно жесткий. Несмотря на то, что предметная область является для разработчиков относительно

новой, анализу архитектурных рисков было уделено лишь некоторое внимание. Организация только начинает внедрять методы управления проектами и формальные методы оценки качества процесса разработки.

Определим показатели проекта:

- Новизна проекта – Почти полное отсутствие прецедентов, в значительной мере непредсказуемый проект (у некоторых членов команды имеется некоторый опыт создания подобных систем)
- Гибкость процесса разработки – Большей частью согласованный процесс (график жесткий, но точной регламентации нет)
- Разрешение рисков в архитектуре системы – Некоторое
- Сплоченность команды – Некоторая согласованность (благодаря мероприятиям, но команда все равно новая)
- Уровень зрелости процесса разработки – Уровень 1+ (только начинают внедрять)

Рис. 3: Факторы, влияющие на показатель степени в модели.

Факторы, влияющие на показатель степени в модели	
Новизна (PREC)	Почти полное отсутствие прецедентов, в значительной мере непредсказуемый п
Гибкость разработки (FLEX)	Большей частью согласованный процесс
Разрешение рисков в архитектуре системы (RESL)	Некоторое (40%)
Сплоченность команды (TEAM)	Некоторая согласованность
Уровень зрелости процесса разработки (PMAT)	Уровень 1+ CMM
<b>P = 1.2317</b>	<div>Выбрать технологический стек</div> <div>Средняя зарплата 100000</div>

Повторное использование – 0%.

Опытность команды – Низкая.

### Объектные точки.

Рассмотрим каждую из страниц:

- Страница авторизации – 3 простых поля и 1 средней сложности (обращение к бд)
- Страница биржевых сводок – 3 простых поля и 1 средней сложности (обращение к бд)
- Страница заявок – 1 простое поле и 2 средней сложности (обращение к бд)
- Страница новой заявки – 4 простых поля и 1 средней сложности (обращение к бд)

Также есть 2 модуля написанные на языках третьего поколения.

Рис. 4: Время и трудозатраты по методике COSOMO II с использованием модели композиции приложения.

Модель композиции изображения			
RUSE (%)	<input type="text" value="0"/>	Опытность команды/разработчика	<input type="text" value="Низкая"/>
<b>Экранные формы</b>		<b>Отчеты</b>	
Простые	<input type="text" value="11"/>	Простые	<input type="text" value="0"/>
Средние	<input type="text" value="5"/>	Средние	<input type="text" value="0"/>
Сложные	<input type="text" value="0"/>	Сложные	<input type="text" value="0"/>
Модули на языках 3 поколения	<input type="text" value="2"/>	<input type="button" value="Рассчитать"/>	
Трудозатраты(чел/мес): 6		Время(мес): 6	Объектн. точки: 41    Бюджет: 600000

4. Произвести оценку трудозатрат и длительности разработки по методике COSOMO II с использованием модели ранней разработки архитектуры.

PERS – номинальный

RCPX – очень высокий

RUSE – низкий

PDIF – высокий

PREX – низкий

FCIL – очень высокий

SCED – очень высокий

KSLOC примерно 4, из метода функциональных точек

Рис. 5: Время и трудозатраты по методике COSOMO II с использованием модели ранней разработки архитектуры.

Модель ранней разработки архитектуры			
PERS	Номинальный ▾	RCPX	Очень высокий ▾
RUSE	Низкий ▾	PDIF	Высокий ▾
PREX	Низкий ▾	FCIL	Очень высокий ▾
SCED	Очень высокий ▾	Рассчитать	
Трудозатраты(чел/мес): 27		Время(мес): 10	Бюджет: 2700000

## 5. Вывод.

В отличие от COSOMO учитывает больше человечески факторов: сплоченность, вовлеченность заказчика. Так же учитывается множество системных параметров и функциональных точек, каждая из которых отвечает за свой набор действий.

Так что модель позволяет оценивать разнообразные проекты, от начальных до крупных, от начинающих команд до опытных разработчиков. И для каждой найдутся свои параметры для расчетов.