

# Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский государствонный тохимисский

«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»

Студент Пронин А. С., Климов И. С.

Группа ИУ7-82Б

Вариант 1

### Подсчет количества объектных точек

Количество простых экранных форм примем равным восьми (просмотр и оплата штрафов, регистрация, авторизация для web-приложения и мобильного приложения, просмотр всех выплат, просмотр данных пользователей, их редактирование и создание новых пользователей).

Количество модулей, написанных на ЯП третьего поколения – пять.

- 1. Приложение для мобильного телефона
- 2. Веб-портал
- 3. Модуль регистрации и авторизации
- 4. Модуль обмена данными с системой ГИБДД
- 5. Модуль проведения платежных транзакций

Наименование объекта	Уровень сложности	Количество	Число точек
Форма	Низкий (1)	8	8
Модуль		5	50
Всего		13	58

Итоговое кол-во объектных точек - 58. Кол-во новых оъектных точек - 58, так как  $\mathrm{RUSE} = 0.$ 

## Подсчет количества функциональных точек

В ПО имеется два внутренних логических файла (ILF).

Один для хранения информаци о пользователях. Число типов элементов записей (RET) для этого файла равно двум (id - число, все остальное - строки). Число типов элементов данных (DET) внутреннего логического файла будет равно шести. Таким образом, уровень сложности внутреннего логического файла — низкий.

Второй для хранения платежных транзакций. ILF имеет четыре элемента данных. Число типов элементов записей равно двум. Уровень сложности низкий.

ПО имеет один внешний интерфейсный файл для хранения штрафов. Число типов элементов записей (RET) для этого файла равно трем. Число типов элементов данных (DET) внутреннего логического файла будет равно шести. Таким образом, уровень сложности внутреннего логического файла – низкий.

#### Внешние вводы ПО:

- Регистрация (мобильное приложение и веб портал). Ссылается на один внутренний логический файл и имеет пять элементов данных. Уровень сложности низкий.
- Авторизация (мобильное приложение и веб портал). Ссылается на один внутренний логический файл и имеет три элемента данных. Уровень сложности низкий.
- Оплата штрафа (мобильное приложение и веб-портал). Ссылается на один внешний интерфейсный файл, один внутренний логический файл и имеет пять элементов данных. Уровень сложности средний.
- Добавление пользователей в БД (веб-портал). Ссылется на один внутренний логический файл и имеет четыре элемента данных. Уровень сложности низкий.
- Получение списка штрафов (веб-портал). Ссылается на один внешний интерфейсный файл и имеет шесть элементов данных. Уровень сложности низкий.
- Получение сообщения об успешном или неуспешном оплате штрафа от ГИБДД (веб-портал). Имеет один элемент данных и не ссылается на внутренние файлы. Уровень сложности низкий.
- Ответ о результате оплаты от платежной системы (веб-портал). Имеет один элемент данных и ссылается на один внутренний файл. Уровень сложности низкий.

#### Внешние выводы ПО:

- Вывод сообщения о положительном или отрицательном результате авторизации (веб портал + мобильное приложение). Уровень сложности этого внешнего вывода низкий, так как он имеет один DET и не имеет FTR.
- Вывод сообщения о положительном или отрицательном результате оплаты штрафа (веб портал + мобильное приложение). Уровень сложности этого внешнего вывода низкий, так как он имеет один DET и один FTR.
- Вывод списка штрафов. Ссылается на один внешний интерфейсный файл и имеет один элемент данных. Уровень сложности низкий.
- Оповещение ГИБДД об оплате штрафа (веб-портал). Ссылается на один внутренний логический файл и имеет один элемент данных. Уровень сложности низкий.
- Запрос об оплате штафа (веб-портал). Имеет три элемента данных и ссылается на два внутренних файла. Уровень сложности низкий.

	Низкий	Средний	Высокий	Итого
Вневшние вводы	6*3	1*6	0*4	33
Внешние выводы	5*4	0*5	0*7	20
Внешние запросы	0*3	0*4	0*6	0
Внутренний логический файлы	2*7	0*10	0*15	14
Внешние логический файлы	1*5	0*7	0*10	5
Всего				63

Итоговое кол-во функциональных точек - 63. Скорректированное количество функциональных точек равно 64.26.

С учетом выбранных ЯП, итоговое кол-во строк кода равно 3944.

Показатель степени равен 1.01 + (6.2 + 5.07 + 1.41 + 1.1 + 4.68)/100), то есть 1.19.

Ввод данных для метода функциональных точек представлен на рисунке 1. Ввод данных для методики СОСОМО2 и результаты рассчетов представлены на рисунке 2.

Hanameth	Просто		Средне	Средне		Сложно		Mana
	Количество	Коэффі	ициентКоличеств		нтКолич	ествоКоэффи	циент	Итого
Внешние входы (EI)	6	÷3	1	<b>÷</b> 6	0	<del>•</del> 4	24	
Внешние выходы (ЕО)	5	<b>‡</b> 4	0	<del>•</del> 5	0	<del>÷</del> 7	20	
Внешние запросы (EQ)	0	÷3	0	÷ 4	0	<del>÷</del> 6	0	
Знутренние логические файлы (ILF)	2	<b>÷</b> 7	0	<del>•</del> 10	0	<b>÷</b> 15	14	
Внешние логические файлы (EIF)	1	<del>•</del> 5	0	÷ 7	0	€ 10	5	
		Общее н	количество				63	
Системные параметры приложения	1							
Передача данных	5		🗦 Оператив	Оперативное обновление				
Распределенная обработка данных	¢ 5		Сложност	Сложность обработки		4		
Производительность	3		🗦 Повторна:	Повторная используемость		3		
Эксплуатационные ограничения	0		🗦 Легкость і	Легкость инсталляции				<u>;</u>
Частота транзакций	3		🗦 Легкость з	Легкость эксплуатации		3		
Оперативный ввод данных	2	Количество возми на различных пл		во возможных установон чных платформах	5			:
Эффективность работы конечных пользователей	0		Простота	Простота изменений (гибкость)				
	SQL		× 30		: %			
	JavaScript		~ <u>10</u>		<b>?</b> %			
	Java		~ <b>70</b>		÷ %			
			Рассчитать					

Рис. 1: Ввод данных для метода функциональных точек.

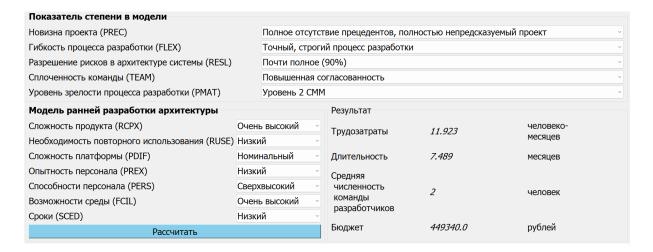


Рис. 2: Ввод данных для методики СОСОМО2 (часть1).

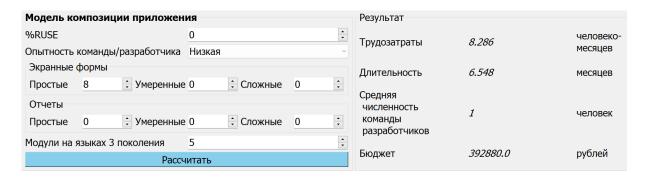


Рис. 3: Ввод данных для методики СОСОМО2 (часть2).

# Вывод

Модель COCOMO2 позволяет более полно учитывать факторы, влияющих на экономические характеристики производства сложных программных продуктов, а также учитывать уникальные факторы для корректировки экономических характеристик, связанные со специфическим проектом и организацией.

COCOMO2 и метод функциональных точек предоставляет возможность оценить объем проекта если собственный опыт аналогичных проектов отсутствует, а экспертное мнение недоступно.