



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

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Лабораторная работа №7 по курсу «Экономика программной инженерии»

Студент Прянишников А. Н., Ском Э. П.

Группа ИУ7-85Б

Оценка (баллы) _____

Преподаватель Барышникова М. Ю.

Москва — 2023 г.

Задание лабораторной работы

На основе своего варианта задания рассчитать количество функциональных точек для разрабатываемого программного приложения. С этой целью разработать программный инструмент. Произвести оценку трудозатрат и длительности разработки по методике COSOMO II с использованием моделей композиции приложения и ранней разработки архитектуры. Определить среднюю численность команды разработчиков. На основе экспертной оценки стоимости человеко-месяца произвести предварительную оценку бюджета проекта. Дать заключение о применимости метода функциональных точек и модели COSOMO II, а также их сравнение с базовой моделью COSOMO для решения поставленной задачи с учетом своего варианта.

Условие лабораторной работы по варианту

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.

Авторизация

На данной странице осуществляется ввод логина и пароля пользователя для входа в систему. Страница содержит два поля ввода и одну командную кнопку, а также флажок-переключатель, который активируется при необходимости запоминания параметров авторизации.

Биржевые сводки

Биржевые сводки отражают текущую ситуацию на бирже. Страница содержит таблицу, кнопку «Добавить» и диалоговое окно с одним полем для ввода и двумя командными кнопками.

Таблица содержит три колонки: Ценная бумага (имя бумаги), Цена (цена за одну ценную бумагу), Изменение (изменение цены бумаги со времени последнего закрытия биржи). Кнопка «Добавить» вызывает диалоговое окно для добавления новой бумаги (окно состоит из поля ввода и кнопок ОК, Cancel).

Заявки

Заявки содержат таблицу, отображающую текущие (еще не выполненные) заявки на покупку или продажу ценных бумаг. Таблица содержит четыре поля: Тип (покупка/продажа), Имя бумаги, Цена по которой готовы покупать/продаваться бумаги, Количество бумаг для покупки/продажи. При нажатии на любую строку таблицы появляется контекстное меню с возможностью удалить или изменить заявку.

Новая заявка

Страница позволяет оформить заявку на покупку или продажу ценной бумаги. Страница состоит из 4 полей: Бумага (имя бумаги), Цена (цена по которой необходимо купить/продать бумагу), Покупка (булева переменная в значение true обозначает покупку, false – продажа) и кнопки «Оформить» - для подтверждения оформления заявки.

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.

1. Обмен данными – 5 (Несколько внешних интерфейсов, несколько типов коммуникационных протоколов).
2. Распределенная обработка – 5 (Динамическое выполнение процессов в любом подходящем компоненте системы).
3. Производительность – 3 (Время реакции или пропускная способность являются критическими в обычное рабочее время. Время обработки ограничено взаимодействующими системами.).
4. Эксплуатационные ограничения по аппаратным ресурсам – 2 (Должны учитываться некоторые ограничения, связанные с безопасностью или временем реакции).
5. Транзакционная нагрузка – 3 (Ожидаются ежедневные пиковые периоды).
6. Интенсивность взаимодействия с пользователем (оперативный ввод данных) – 4 (От 24% до 30% транзакций требуют интерактивного ввода данных).
7. Эргономические характеристики, влияющие на эффективность работы конечных пользователей – 1 (От одной до трех функциональных возможностей).
8. Оперативное обновление – 4 (Онлайновое обновление основных внутренних логических файлов + необходимость специальной защиты от потери данных).
9. Сложность обработки – 4.
10. Повторное использование – 0 (Отсутствует).

11. Легкость инсталляции – 1 (Для установки требуется специальная процедура).
12. Легкость эксплуатации/администрирования – 2.
13. Портативность – – 2 (Приложение рассчитано на много установок для строго стандартной платформы (технические средства + программное обеспечение)).
14. Гибкость – 2.

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.

Надежность и уровень сложности (RCPX) разрабатываемой системы оцениваются как очень высокие, повторного использования компонентов не предусматривается (RUSE). Возможности персонала (PERS) – средние, его опыт работы в разработке систем подобного типа (PREX) низкий. Сложность платформы (PDIF) высокая. Разработка предусматривает очень интенсивное использование инструментальных средств поддержки (FCIL). Заказчик настаивает на жестком графике (SCED).

COCOMO II

Модель COCOMO I основана на модели разработки Waterfall, но из-за освоения объектно-ориентированного подхода в процессе разработки программного обеспечения COCOMO I не дает точных результатов. COCOMO II позволяет учесть это при разработке проекта. Первоочередной целью модели COCOMO

II является создание возможностей поддержки для постоянного внесения поправок в модель и предоставление количественной аналитической структуры, методов и инструментов. Он также способен исследовать влияние усовершенствований программных технологий на жизненный цикл разработки программного обеспечения. Модели оценки, включенные в СОСОМО II, включают модель композиции приложения, модель раннего проектирования, модель повторного использования и модель постархитектуры.

Модель композиции приложения

Эта модель предназначена для использования с повторно используемыми компонентами и генерирует оценки разработки прототипа и работает на основе точек объекта. Модель больше подходит для разработки прототипа системы. Чтобы оценить общие усилия, выполняются следующие шаги.

Ранняя модель дизайна

Модель, используемая на этапе проектирования системы после получения требований. Его оценки создаются на основе функциональных точек, которые затем переводятся в несколько строк исходного кода.

Модель повторного использования

Это модель, которая вычисляет трудозатраты, необходимые для объединения повторно используемых компонентов или программного кода, который спонтанно создается инструментами проектирования или преобразования программ. Есть два типа повторно используемых кодов: черный ящик и код белого ящика. Код черного ящика используется, когда в нем нет понимания кода и модификации. И наоборот, белое поле используется при интеграции нового кода.

Реализация приложения

Для подсчета параметров проекта по методике COCOMO II было реализовано приложение. В нем можно рассчитать необходимые параметры в четырех вкладках. В каждом из экранов можно установить необходимые параметры по варианту.

На рисунке представлен вид приложения:

The screenshot shows the COCOMO2 application window with four tabs: 'Расчет показателя степени', 'Модель композиции', 'Метод функциональных точек' (selected), and 'Модель ранней разработки архитектуры'. The 'Метод функциональных точек' tab contains the following sections:

Характеристики продукта

Характеристика	Значение
Обмен данными	5
Распределенная обработка	5
Производительность	3
Эксплуатационные ограничения по аппаратным ресурсам	0
Транзакционная нагрузка	3
Оперативный ввод данных	2
Эргономические характеристики	0
Оперативное обновление	4
Сложность обработки	4
Повторное использование	3
Легкость инсталляции	0
Легкость эксплуатации	3
Портируемость	5
Гибкость	0

Низкие Средние Высокие Итого

EI	5	0	0	0
EO	3	0	0	0
EQ	5	0	0	0
ILF	2	0	0	0
EIF	1	0	0	0
Итого:				0

Процент использования ЯП

Язык	Процент	Язык	Процент
ASM	0,00	C#	0,00
C	0,00	Ada 95	0,00
Cobol	0,00	SQL	30,00
Fortran	0,00	Visual C++	0,00
Pascal	0,00	Delphi	0,00
C++	0,00	Perl	0,00
Java	60,00	Prolog	0,00
JS	10,00	Visual Basic	0,00

FP: 0

SLOC: 0

Рассчитать

По варианту

Рисунок 1: Основной экран приложения

Метод функциональных точек

Пройдемся по основным параметрам, которые нужно оценить:

1. *FTR* – количество связанных с каждым функциональным типом файлов типа ссылок.
2. *DET* – количество связанных с каждым функциональным типом элементарных данных.
3. *RET* – количество типов элементов записей.
4. *EI* – элементарный процесс, перемещающий данные из внешней среды в приложение.
5. *EO* – элементарный процесс, перемещающий данные, вычисленные в приложении, во внешнюю среду.
6. *EQ* – элементарный процесс, состоящий из комбинации «запрос/ответ», не связанный с вычислением производных данных или обновлением внутренних логических файлов (базы данных).
7. *ILF* – выделяемые пользователем логически связанные группы данных или блоки управляющей информации, которые поддерживаются внутри продукта и обслуживаются через внешние вводы.
8. *EIF* – выделяемые пользователем логически связанные группы данных или блоки управляющей информации, на которые ссылается продукт, но которые поддерживаются вне продукта.

В нашем приложении используются 3 внутренних файла:

- таблица с логинами и паролями;
- таблица с подробной информации о заявке: ее тип, имя бумаги, цена и количества;
- таблица с информацией о бумагах.

Также существует одна внешняя таблица с информацией о бирже с названием бумаги, ценой и изменением.

Вычислим все EI:

1. **Добавить заявку:** одна ссылка на файл ($FTR = 1$), 5 элементов – (4 поля ввода + кнопка). Уровень сложности – **Низкий(3)**

2. **Изменить заявку:** одна ссылка на файл ($FTR = 1$), 5 элементов – (4 поля ввода + кнопка). Уровень сложности – **Низкий(3)**
3. **Удалить заявку:** одна ссылка на файл ($FTR = 1$), 1 элемент – (кнопка). Уровень сложности – **Низкий(3)**
4. **Добавить бумагу:** одна ссылка на файл ($FTR = 1$), 2 элемента – (поле + кнопка). Уровень сложности – **Низкий(3)**

Подсчет всех ЕО:

1. **Вывод списка заявок:** одна ссылка на файл ($FTR = 1$), 4 элемента. Уровень сложности – **Низкий(4)**
2. **Вывод биржевых сводок:** внутренний логический файл и внешний интерфейсный файл ($FTR = 2$), 3 элемента. Уровень сложности – **Низкий(4)**.

Один EQ – **запрос на авторизацию:** одна ссылка на файл ($FTR = 1$), 4 элемента. Уровень сложности – **Низкий(3)**.

Вычисление ILF: три таблицы, у каждой уровень сложности – **Низкий(3)**.

Вычисление ELF: одна таблица. Уровень сложности – **Низкий(3)**.

Вычисление результата

Введем все параметры из условия. Результаты вычислений для проекта:

1. Количество функциональных точек – 49.
2. Количество функциональных точек с учетом корректирующих параметров – 50,47.
3. Количество строк кода – 3219 строк кода.

Результат вычислений представлен на рисунке:

Характеристики продукта		Низкие	Средние	Высокие	Итого
Обмен данными	5	4	0	0	12
Распределенная обработка	5	2	0	0	8
Производительность	3	1	0	0	3
Эксплуатационные ограничения по аппаратным ресурсам	2	3	0	0	21
Транзакционная нагрузка	3	1	0	0	5
Оперативный ввод данных	4	Итого: 49			
Эргономические характеристики	1	Процент использования ЯП			
Оперативное обновление	4	ASM	0,00	C#	60,00
Сложность обработки	4	C	0,00	Ada 95	0,00
Повторное использование	0	Cobol	0,00	SQL	15,00
Легкость инсталляции	1	Fortran	0,00	Visual C++	0,00
Легкость эксплуатации	2	Pascal	0,00	Delphi	0,00
Портируемость	2	C++	0,00	Perl	0,00
Гибкость	2	Java	25,00	Prolog	0,00
		JS	0,00	Visual Basic	0,00

FP:	50.47	<input type="button" value="Рассчитать"/>
SLOC:	3219	<input type="button" value="По варианту"/>

Рисунок 2: Метод функциональных точек

Расчет показателя степени

Подберем, исходя из условий, показатели степени, которые нужно использовать для расчета по остальным методам:

1. **Новизна проекта (PREC)** – почти полное отсутствие прецедентов, в значительной мере непредсказуемый проект.
2. **Гибкость процесса разработки (FLEX)** – по большей части согласованный процесс (график жесткий, точной регламентации нет).
3. **Разрешение рисков в архитектуре системы (RESL)** – некоторое (40%).
4. **Сплоченность команды (TEAM)** – некоторая согласованность (команда

новая, но были проведены определенные мероприятия по сплочению).

5. **Уровень развития процесса разработки (РМАТ)** – уровень 1+ (только начинают внедрять).

Итоговый результат: $P = 1.24$. Результат вычислений представлен на рисунке:

Факторы показателя степени модели

PREC	Почти полное отсутствие прецедентов, в значительной мере непредсказуе ▾
FLEX	Большей частью согласованный процесс ▾
RESL	Некоторое (40 %) ▾
TEAM	Некоторая согласованность ▾
PMAT	Уровень 1 CMM ▾

Показатель степени P: 1.2473

Рисунок 3: Расчет показателя степени

Модель ранней разработки архитектуры

Значение параметров для модели ранней разработки архитектуры были выбраны следующие значения:

1. **PERS** – номинальный уровень.
2. **RCPX** – очень высокий уровень.
3. **RUSE** – низкий уровень.
4. **PDIF** – высокий уровень.
5. **PREX** – низкий уровень.

6. **FCIL** – высокий уровень.

7. **SCED** – высокий уровень.

Средняя зарплата выбрана 100000 рублей.

Результат вычислений по варианту представлен на рисунке:

Средняя ЗП	<input type="text" value="100000"/>
PERS	<input type="text" value="Номинальный"/>
RCPX	<input type="text" value="Очень высокий"/>
RUSE	<input type="text" value="Низкий"/>
PDIF	<input type="text" value="Высокий"/>
PREX	<input type="text" value="Низкий"/>
FCIL	<input type="text" value="Высокий"/>
SCED	<input type="text" value="Высокий"/>

Трудозатраты, чм 26.17
Время, мес 10.29
Бюджет 2617000.0

Рисунок 4: Модель ранней разработки архитектуры

Итоговые результаты:

- Требуемую трудоемкость – 26.17 человеко-мес.
- Время разработки – 10.29 месяцев.
- Бюджет – 2617000 рублей.

Модель композиции

Следует определить следующие параметры:

1. RUSE – 0%, так как по условию сказано, что повторного использования компонентов не предусматривается.
2. Средняя зарплата – 100000 рублей.
3. Опытность команды – низкая, так как по условию у персонала низкий опыт разработки систем подобного типа.
4. Модулей у проекта три – так как в ТЗ указано, что у реализации будет три компонента.

Экранных форм всего 4.

- **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).

Из отчетов можно выделить 2 средней сложности: вывод всех заявок и биржевых сводок.

Результат вычислений по варианту представлен на рисунке:

Средняя ЗП	<input type="text" value="100000"/>	
	Экранные формы	Отчеты
Простые	<input type="text" value="11"/>	<input type="text" value="0"/>
Средние	<input type="text" value="5"/>	<input type="text" value="2"/>
Сложные	<input type="text" value="0"/>	<input type="text" value="0"/>
Модулей на языках 3 поколения	<input type="text" value="3"/>	
%RUSE	<input type="text" value="0"/>	
Опытность команды	<input type="text" value="Низкая"/>	
<input type="button" value="По варианту"/>		<input type="button" value="Рассчитать"/>
<p>Трудозатраты, чм 8.71 Время, мес 6.79 Бюджет 871000.0</p>		

Рисунок 5: Модель композиции

Итоговые результаты:

- Требующуюся трудоемкость – 8.71 человеко-месяц.
- Время разработки – 6.79 месяцев.
- Бюджет – 871000 рублей.

COCOMO I

Для выполнения сравнения между моделями требуется проверить результат от модели COCOMO I.

Результат вычислений по варианту представлен на рисунке:

COCOMO

Прянишников Александр, ИУ7-85Б Лабораторная работа №6 по ЭПИ Вариант 4 Просмотр ТЗ

RELY	Очень высокий ▾	STOR	Номинальный ▾	AEXP	Высокий ▾	MODP	Высокий ▾
DATA	Номинальный ▾	VIRT	Номинальный ▾	PCAP	Высокий ▾	TOOL	Высокий ▾
CPLX	Высокий ▾	TURN	Высокий ▾	VEXP	Высокий ▾	SCED	Номинальный ▾
TIME	Номинальный ▾	ACAP	Номинальный ▾	LEXP	Высокий ▾		

SIZE (KLOC) Режим проекта: Промежуточный ▾ Рассчитать проект Задача

Распределение работ по видам деятельности WBS

	Бюджет (%)	Человеко-месяцы
Анализ требований	4	0.46
Проектирование продукта	12	1.38
Программирование	44	5.04
Планирование тестирования	6	0.69
Верификация и аттестация	14	1.6
Канцелярия проекта	7	0.8
УК и ОК	7	0.8
Создание руководств	6	0.69
ИТОГО	100	11.46

Результат:

Время разработки: 7.77 мес.
Трудоемкость: 11.46 Ч/м

	Работа	Время
Определение требований	0.85	2.06
Проектирование продукта	1.91	2.06
Детальное проектирование	2.65	1.03
Кодирование и тестирование	2.76	1.03
Интеграция и тестирование	3.29	1.6
Итого без планирования	10.61	5.71
Итого	11.46	7.77

Рисунок 6: COCOMO I

Итоговые результаты:

- Требуемую трудоемкость – 11.46 человеко-месяц.
- Время разработки – 7.77 месяцев.
- Бюджет – 1146000 рублей.

Выводы

В качестве моделей для оценки использовалась модель композиции и модель ранней разработки архитектуры. Выяснилось, что оценка второй модели оказалась по трудозатратам в три раза выше, чем первой – это связано с более точной оценкой в модели ранней разработки архитектуры. В частности, на ее результат влияет метод функциональных точек и более точное разбиение задач по языкам программирования.

По сравнению с COCOMO I, модель ранней разработки архитектуры увеличивает трудозатраты примерно в два раза, но модель композиции предполагает на 15% меньше усилий.