



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»

ОТЧЕТ

к лабораторной работе №7

По курсу: «Экономика программной инженерии»

Тема: «Предварительная оценка параметров программного проекта»

Студенты ИУ7-82Б

Козлова И. В.

Зайцева А. А.

Сысоева В. Р.

Преподаватель

Барышникова М. Ю.

Силантьева А. В.

2023 г.

Цель работы

Продолжение знакомства с существующими методиками предварительной оценки параметров программного проекта и практическая оценка затрат по модели COSOMO II.

COSOMO II

3 модели оценки стоимости в COSOMO II:

1. **Модель композиции приложения** – модель, которая подходит для проектов, созданных с помощью современных инструментальных средств. Единицей измерения служит объектная точка (учитывается количество экранов, отчетов и компонентов).
 - a. рассматривается макетирование пользовательский интерфейсов
 - b. оценивается производительность
 - c. определяется степень зрелости технологии
2. **Модель ранней разработки архитектуры** – модель применяется для получения приблизительных оценок проектных затрат периода выполнения проекта перед тем как будет определена архитектура в целом. В качестве единиц измерения используются функциональные точки либо KSLOC.
3. **Постархитектурная модель** – наиболее детализированная модель COSOMO II, которая используется после разработки архитектуры проекта. В состав этой модели включены новые драйверы затрат, новые правила подсчета строк кода, а также новые уравнения

| Модель композиции приложения | Модель ранней разработки архитектуры | Постархитектурная модель |
|------------------------------|--------------------------------------|---|
| Грубые входные данные | Ясно понимаемые особенности проекта | Детальное описание проекта |
| Оценки низкой точности | Оценки умеренной точности | Высокоточные оценки |
| Приблизительные требования | Ясно понимаемые требования | Стабилизировавшиеся основные требования |
| Концепция архитектуры | Ясно понимаемая архитектура | Стабильная базовая архитектура |

| Плюсы СОСОМО II | Минусы СОСОМО II |
|--|--|
| 1. возможен учет многих факторов 2. универсальный метод 3. фактические данные подбираются в соответствии с реальными проектами 4. метод позволяет добавлять уникальные факторы для корректировки характеристик 5. результаты прогнозирования сопровождаются обязательной документацией 6. модель проста в освоении и применении | 7. все результаты зависят от размера программного продукта 8. игнорируются требования к характеристикам качества программного продукта 9. игнорируется изменяемость требований к программному продукту 10. игнорируются многие особенности, связанные с аппаратным обеспечением проекта |

Метод функциональных точек

В этом методе рассматривается не размер, а функциональность или полезность продукта, в качестве количественной характеристики применяется понятие количества функциональных точек.

- *Количество функциональных точек* – количественная характеристика, обозначается как FP (function points).

- *Функциональная точка* — единица измерения функциональности ПО. Функциональность программы связана с обработкой информации по запросу пользователя и не зависит от применяемых технических решений.
- *Пользователи* — отправители и целевые получатели данных (люди, смежные интегрированные информационные системы).
- Оценка трудозатрат может быть выполнена на ранних стадиях работы над проектом, поскольку в основе лежит изучение требований.

Функциональность системы определяется:

1. функциональными типами
2. элементарными процессами

FTR – количество связанных с каждым функциональным типом файлов типа ссылок.

DET – количество связанных с каждым функциональным типом элементарных данных.

RET – количество типов элементов записей.

Типы элементарных процессов, используемых в методе функциональных точек:

- внешний ввод (EI) — элементарный процесс, перемещающий данные из внешней среды в приложение;
- внешний вывод (EO) — элементарный процесс, перемещающий данные, вычисленные в приложении, во внешнюю среду;
- внешний запрос (EQ) — элементарный процесс, состоящий из комбинации «запрос/ответ», не связанный с вычислением производных данных;
- внутренний логический файл (ILF) — выделяемые пользователем логически связанные группы данных, которые поддерживаются внутри продукта и обслуживаются через внешние вводы;
- внешний интерфейсный файл (EIF) — выделяемые пользователем логически связанные группы данных или блоки управляющей информации, на которые ссылается продукт, но которые поддерживаются вне продукта.

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.

Характеристики команды, продукта и проекта

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.

Характеристики продукта в условных единицах:

1. Обмен данными - 5
2. Распределенная обработка -5
3. Производительность -3
4. Эксплуатационные ограничения по аппаратным ресурсам – 2
5. Транзакционная нагрузка – 3
6. Интенсивность взаимодействия с пользователем (оперативный ввод данных) – 4
7. Эргономические характеристики, влияющие на эффективность работы конечных пользователей – 1
8. Оперативное обновление – 4
9. Сложность обработки – 4
- 10.Повторное использование – 0
- 11.Легкость инсталляции – 1
- 12.Легкость эксплуатации/администрирования – 2
- 13.Портируемость – 2
- 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.
- Повторное использования компонентов не предусматривается (RUSE).
- Возможности персонала (PERS) – средние.
- 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).

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.

Внутренний логический файл (ILF)

- Таблица в базе данных («Логин», «Пароль»)

RET = 1

- и логин, и пароль представляются в формате строки.

DET = 2 (логин, пароль).

- Локальный файл

RET = 1

- и логин, и пароль представляются в формате строки.

DET = 2 (логин, пароль).

Внешний ввод (EI)

- ввод логина и пароля для сверки с логином и паролем из БД и запоминания логина и пароля локально в системе

FTR = 2 (2 файла)

DET = 4 (поля логин, пароль, «Запомнить», «Войти»)

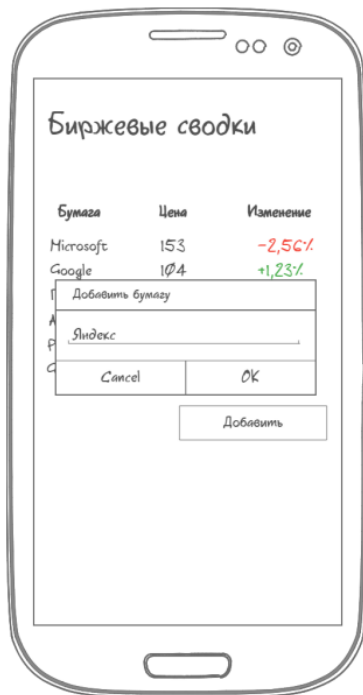
Внешние запросы (EQ)

- уведомление об ошибке (неправильный логин или пароль)

FTR = 1 (на логический файл)

DET = 4 (поля логин, пароль, флажок-переключатель «Запомнить», кнопка «Войти»)

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).

Внутренний логический файл (ILF)

RET = 2

- «Ценная бумага» представляется в формате строки
- «Цена» и «Изменение» в вещественном формате

DET = 2 (Ценная бумага, Цена, Изменение).

Внешний ввод (EI)

- Добавить новую бумагу

FTR = 1 (запрос к таблице)

DET = 3 (кнопка «Добавить», поле названия, кнопка «Ок»)

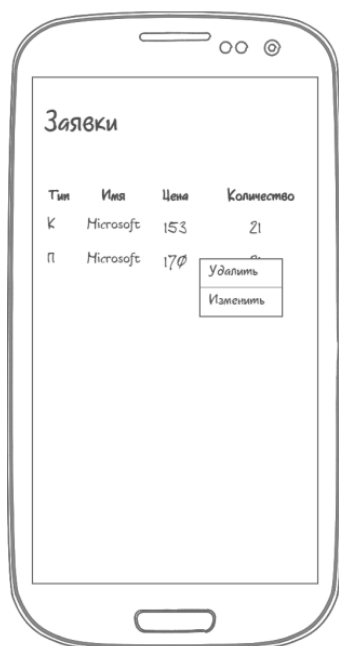
Внешние запросы (EQ)

- Вывод информации о ценных бумагах

FTR = 2 (запрос к таблице и внешний интерфейсный файл)

DET = 3 (поля «Бумага», «Цена», «Изменение»)

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 покупаться/продаваться бумаги, Количество бумаг для покупки/продажи. При нажатии на любую строку таблицы появляется контекстное меню с возможностью удалить или изменить заявку

Внутренний логический файл (ILF)

RET = 4

- «Тип» представляется логическим типом,
- «Имя» - строковый формат,
- «Цена» - вещественный,
- «Количество» - положительное целое число

DET = 4 («Тип», «Имя», «Цена», «Количество»)

Внешний ввод (EI)

- Изменить заявку

FTR = 1 (запрос к таблице)

DET = 5 (строка из таблицы (поля «Тип», «Имя», «Цена», «Количество»), кнопка «Изменить»)

- Удалить заявку

FTR = 1 (запрос к таблице)

DET = 5 (строка из таблицы (поля «Тип», «Имя», «Цена», «Количество»), кнопка «Удалить»)

Внешние запросы (EQ)

- Вывод информации о заявках

FTR = 1 (запрос к таблице)

DET = 4 (поля «Тип», «Имя», «Цена», «Количество»)

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.

Внутренний логический файл (ILF)

RET = 3

- «Тип» представляется логическим типом,
- «Имя» - строковый формат,
- «Цена» - вещественный,

DET = 3 («Тип», «Имя», «Цена»,)

Внешний ввод (EI)

- Продать/купить ценную бумагу

FTR = 1 (запрос к таблице)

DET = 5 (поля «Бумага», «Цена», «Количество», флаг «Покупка», кнопка «Оформить»)

Внешний интерфейсный файл (EIF)

RET = 2 (строка, вещественное число)

DET = 3 (3 таблицы)

Итого:

1. уровень сложности логического файла (ILF) – низкий;

| Типы элементов-записей (RET) | Элементы данных (DET) | | |
|------------------------------|-----------------------|--------------|--------------|
| | 1-19 | 20-50 | >50 |
| 1 | Низкий (7) | Низкий (7) | Средний (10) |
| 2-5 | Низкий (7) | Средний (10) | Высокий (15) |
| >5 | Средний (10) | Высокий (15) | Высокий (15) |

2. уровень сложности внешних вводов (EI) – низкий;

| Ссылки на файлы (FTR) | Элементы данных (DET) | | |
|-----------------------|-----------------------|-------------|-------------|
| | 1-4 | 5-15 | >15 |
| 0-1 | Низкий (3) | Низкий (3) | Средний (4) |
| 2 | Низкий (3) | Средний (4) | Высокий (6) |
| >2 | Средний (4) | Высокий (6) | Высокий (6) |

3. уровень сложности внешних выводов (EO) – низкий;

| Ссылки на файлы (FTR) | Элементы данных (DET) | | |
|-----------------------|-----------------------|-------------|-------------|
| | 1-4 | 5-19 | >19 |
| 0-1 | Низкий (4) | Низкий (4) | Средний (5) |
| 2-3 | Низкий (4) | Средний (5) | Высокий (7) |
| >3 | Средний (5) | Высокий (7) | Высокий (7) |

4. уровень внешних запросов (EQ) – низкий;

| Ссылки на файлы (FTR) | Элементы данных | | |
|-----------------------|-----------------|-------------|-------------|
| | 1-4 | 5-19 | >19 |
| 0-1 | Низкий (3) | Низкий (3) | Средний (4) |
| 2-3 | Низкий (3) | Средний (4) | Высокий (6) |
| >3 | Средний (4) | Высокий (6) | Высокий (6) |

5. уровень сложности внешних интерфейсных файлов (EIF) – низкий.

| Типы элементов-записей (RET) | Элементы данных (DET) | | |
|------------------------------|-----------------------|--------------|--------------|
| | 1-19 | 20-50 | >50 |
| 1 | Низкий (5) | Низкий (5) | Средний (7) |
| 2-5 | Низкий (5) | Средний (7) | Высокий (10) |
| >5 | Средний (7) | Высокий (10) | Высокий (10) |

Интерфейс программы

The screenshot shows the COCOMO2 software interface. It includes a menu bar with 'Метод функциональных точек' and 'Модели композиции и ранней разработки архитектуры'. The main window is divided into several sections:

- Системные параметры приложения:** A list of parameters with dropdown menus, including 'Передачи данных' (5), 'Распределенная обработка данных' (5), 'Производительность' (3), 'Эксплуатационные ограничения' (2), 'Частота транзакций' (3), 'Оперативный ввод данных' (4), 'Эффективность работы конечных пользователей' (1), 'Оперативное обновление' (4), 'Сложность обработки' (4), 'Повторное использование' (0), 'Легкость инсталляции' (1), 'Легкость эксплуатации' (2), 'Количество возможных установок на различных платформах' (2), and 'Простота изменений(гибко)' (2).
- Процент использования языка:** A list of programming languages with input fields for their usage percentage: ASM (0), C (0), SQL (15), Delphi Pascal (0), Pascal (0), Visual C++ (0), Java (25), Ada 9 (0), C# (60), Cobol (0), C++ (0), Fortran (0), Prolog (0), and Perl (0).
- Исходные данные для расчета общего количества функциональных точек:** A table with columns for 'Количество/сложн', 'Низкая', 'Средняя', 'Высокая', and 'Результат'. It lists various types of functional points and their counts.
- Результат:** A section showing the calculated results: 'Количество строк исходного кода: 3877', 'Количество функциональных точек: 59', and 'Нормированное количество функциональных точек: 60.77'.

По расчетам:

Нормированное количество функциональных точек: 60.77

Количество функциональных точек: 59

Количество строк исходного кода: 3877

Расчет по модели COCOMO II

Показатели степени модели:

- *Новизна проекта (PREC)* – полное отсутствие прецедентов, полностью непредсказуемый проект (т.к. была сформирована новая команда

разработчиков, только отдельные члены имели некоторый опыт создания систем подобного типа)

- *Гибкость процесса разработки (FLEX)* – большей частью согласованный процесс (график жесткий, точной регламентации нет)
- *Разрешение рисков в архитектуре системы (RESL)* – некоторое (40%)
- *Сплоченность команды (TEAM)* – некоторая согласованность (команда новая, но были проведены определенные мероприятия по сплочению)
- *Уровень развития процесса разработки (PMAT)* – начальный уровень (только начинают внедрять)

Значения всех пяти показателей суммируются, сумма делится на 100, результат прибавляется к числу 1.01

Используем формулу: $p = \sum W_i / 100 * 1.01$, где W_i - показатель проекта.

| Показатель степени модели | |
|---|---|
| Новизна проекта (PREC) | Полное отсутствие прецедентов, полностью непредсказуемый проект |
| Анализ архитектуры системы и рисков (RESL) | Некоторое (40 %) |
| Гибкость процесса разработки (FLEX) | Большой частью согласованный п |
| Уровень развития процесса разработки (PMAT) | Уровень 1+ CMM |
| Сплоченность команды (TEAM) | Некоторая согласованность |

P = 1.244Рассчитать

Показатель степени P равен 1.244

Расчёт модели композиции проекта. По страницам:

- Авторизация – 3 простых поля и 1 средней сложности
- Биржевые сводки – 3 простых поля и 1 средней сложности

- Заявки - 1 простое поля и 2 средней сложности
- Новая заявка - 4 простых поля и 1 средней сложности

Средняя зарплата была принята равной 60 000 руб.

Итого:

- Полей простой сложности: **11**
- Полей средней сложности: **5**
- Полей высокой сложности: **0**
- Модулей на ЯП 3го поколения: **2**
- Повторное использование: **не предусматривается**
- Опытность команды: **низкая**

Формулы:

Модель композиции приложения

$NOP = (\text{Объектные точки}) \times [(100 - \%RUSE) / 100]$ – новые объектные точки

$\text{ТРУДОЗАТРАТЫ} = NOP / \text{PROD}$ [чел.-мес.]

PROD - оценка скорости разработки

Модель композиции приложения используется на этапе создания прототипов и анализа осуществимости

| Опытность/ возможности разработчика | Зрелость/ возможности среды разработки | PROD |
|---|--|------|
| Очень низкая | Очень низкая | 4 |
| Низкая | Низкая | 7 |
| Номинальная | Номинальная | 13 |
| Высокая | Высокая | 25 |
| Очень высокая | Очень высокая | 50 |

$\text{Время} = 3 * (\text{Трудозатраты})^{(0.33 + 0.2 * (p-1.01))}$

Модель композиции приложения

| Экранные формы | | Отчеты | |
|---|-------------------------------------|------------------|------------------------------------|
| 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"/> |
| %RUSE | <input type="text" value="0"/> | | |
| Модулей на языках 3 поколения | <input type="text" value="2"/> | | |
| Опытность команды/разработчика | <input type="text" value="Низкая"/> | Средняя зарплата | <input type="text" value="60000"/> |
| <input type="button" value="Рассчитать"/> | | | |
| Трудозатраты, чел/мес | 5.86 | | |
| Время, мес | 5.84 | | |
| Бюджет | 351600.0 | | |

Расчёт модели ранней разработки архитектуры:

- Надежность и уровень сложности (RCPX) – **очень высокие**
- Повторное использования компонентов **не предусматривается** (RUSE).
- Возможности персонала (PERS) – **номинальные**
- Опыт работы в разработке систем подобного типа (PREX) – **низкий**.
- Сложность платформы (PDIF) – **высокая**.
- **Очень интенсивное** использование инструментальных средств поддержки (FCIL).
- Жесткий график (SCED)

Формулы:

Трудозатраты = 2,45*EArch*(Размер)^р,

где: Трудозатраты (работа) — число человеко-месяцев

EArch = PERS * RCPX * RUSE * PDIF * PREX * FCIL * SCED

Размер — KSLOC (предпочтительно для подсчета KSLOC предварительно подсчитать количество функциональных точек)

р — показатель степени

Время = 3,0 * (Трудозатраты)^{(0.33 + 0.2 * (р-1.01))}

Модель ранней разработки архитектуры

| | | | | | |
|------|---------------|------|-----------|------|---------------|
| PERS | Номинальный ▾ | RUSE | Низкий ▾ | FCIL | Очень высок ▾ |
| RCPX | Очень высок ▾ | PDIF | Высокий ▾ | PREX | Низкий ▾ |
| SCED | Очень высок ▾ | | | | |

Рассчитать

| | |
|------------------------------|--------------|
| Трудозатраты, чел/мес | 27.57 |
| Время, мес | 10.47 |
| Бюджет | 1654200.0 |

В ходе выполнения работы было предложено еще одно видение проекта

Расчёт **модели композиции** проекта. По страницам:

- Авторизация – 2 простых поля и 2 средней сложности
- Биржевые сводки – 3 простых поля и 1 средней сложности
- Заявки - 2 простое поля и 1 высокой сложности
- Новая заявка - 3 простых поля и 2 средней сложности

Средняя зарплата была принята равной 70 000 руб.

Итого:

- Полей простой сложности: **10**
- Полей средней сложности: **5**
- Полей высокой сложности: **1**
- Модулей на ЯП 3го поколения: **2**
- Повторное использование: **не предусматривается**
- Опытность команды: **номинальная**

Модель композиции приложения

| Экранные формы | | Отчеты | |
|---|--|------------------|------------------------------------|
| Простые | <input type="text" value="10"/> | Простые | <input type="text" value="0"/> |
| Средние | <input type="text" value="5"/> | Средние | <input type="text" value="0"/> |
| Сложные | <input type="text" value="1"/> | Сложные | <input type="text" value="0"/> |
| %RUSE | <input type="text" value="10"/> | | |
| Модулей на языках 3 поколения | <input type="text" value="2"/> | | |
| Опытность команды/разработчика | <input type="text" value="Номинальная"/> | Средняя зарплата | <input type="text" value="70000"/> |
| <input type="button" value="Рассчитать"/> | | | |
| Трудозатраты, чел/мес | 2.98 | | |
| Время, мес | 4.53 | | |
| Бюджет | 208600.0 | | |

При увеличении форм высокой сложности параметры проекты должны были повыситься, но за счет того, что опытность команды повысилась, показатели понизились.

Вывод

В ходе выполнения данной работы была освоена методология оценки параметров проекта COSOMO2 и разработан программный инструмент для её применения. Выполнен анализ выданного задания:

- рассчитаны функциональные точки и показатель степени модели (p)
- были определены факторы, влияющие на показатель степени
- рассчитаны трудозатраты и времени по моделям:
 - ранней разработки архитектуры приложения
 - композиции приложения

По модели композиции приложения прогноз более благоприятный, чем в модели ранней архитектуры приложения.

Методология COSOMO2 является более сложной по сравнению с COSOMO, но позволяет более тонко настраивать параметры плана, что даёт более точный и детальный прогноз.