



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»

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №7

Студент Ковель Александр Денисович, Солопов Юрий Витальевич

Группа ИУ7-86Б

Предмет Экономика программной инженерии

Студент

подпись, дата

Ковель А. Д.

фамилия, и.о.

Студент

подпись, дата

Солопов Ю.В.

фамилия, и.о.

Преподаватель

подпись, дата

Барышникова М.Ю.

фамилия, и.о.

2024 г.

1 Результаты работы 7 лабораторной работы

1.1 Аналитический раздел

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

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:

функциональными типами

элементарными процессами

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.

1.1.1 Задание

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

Характеристики продукта:

- 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) Портитруемость – 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, 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).

1.1.2 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.



Рисунок 1 – Авторизация

Внутренний логический файл (ILF)

1 соответствующая таблица в базе данных, поля «Логин», «Пароль» и 1 локальный файл

- Таблица в базе данных: Количество типов элементарных записей (RET) = 1 и логин, и пароль представляются в формате строки.
- Количество типов элементарных данных (DET) = 2 (логин, пароль).
- Локальный файл
- Количество типов элементарных записей (RET) = 1 и логин, и пароль представляются в формате строки.
- Количество типов элементарных данных (DET) = 2 (логин, пароль).

Внешний ввод (EI)

- запоминание данных пользователя
- FTR = 2 (2 файла)
- DET = 4 (поля логин, пароль, флажок-переключатель «Запомнить», кнопка «Войти»)

Внешние запросы (EQ)

- запрос на авторизацию
- FTR = 1 (на логический файл)
- DET = 4 (поля логин, пароль, флажок-переключатель «Запомнить», кнопка «Войти»)

1.1.3 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 (Ok, Cancel).

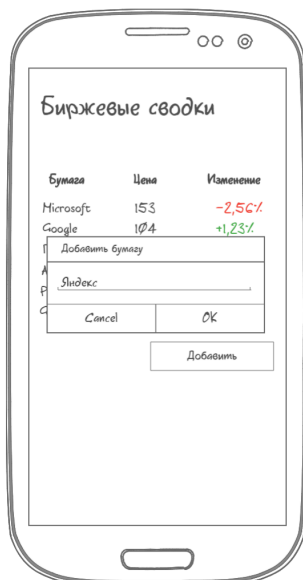


Рисунок 2 – Биржевые сводки

Внутренний логический файл (ILF)

1 соответствующая таблица в базе данных, поля «Ценная бумага», «Цена» (за одну ценную бумагу), «Изменение» (изменение цены бумаги со времени последнего закрытия биржи).

Количество типов элементарных записей (RET) = 2

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

Количество типов элементарных данных (DET) = 3 («Ценная бумага»,

«Цена», «Изменение»).

Внешний ввод (EI)

Добавить новую бумагу

FTR = 1 (запрос к таблице)

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

Внешний вывод (EO)

Вывод информации о ценных бумагах

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

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

1.1.4 Заявки

Заявки содержат таблицу, отображающую текущие (еще не выполненные) заявки на покупку или продажу ценных бумаг. При нажатии на любую строку таблицы появляется контекстное меню с возможностью удалить или изменить заявку.



Рисунок 3 – Заявки

Внутренний логический файл (ILF)

1 соответствующая таблица в базе данных, поля «Тип» (покупка или продажа), «Имя», «Цена», «Количество».

Количество типов элементарных записей (RET) = 4

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

Количество типов элементарных данных (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»)

Внешний вывод (EO)

Вывод информации о заявках

FTR = 1 (запрос к таблице)

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

1.1.5 Новая заявка

Страница позволяет оформить заявку на покупку или продажу ценной бумаги.



Рисунок 4 – Заявки

Внутренний логический файл (ILF)

1 соответствующая таблица в базе данных, поля «Бумага» (имя бумаги), «Цена», «Покупка» (булевая переменная в значении true – покупка, false - продажа). Используется та же таблица, что и для страницы «Заявки», но используется только 3 поля: «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 – низкий;
- 2) уровень сложности внешних выводов – низкий;

- 3) уровень сложности внешних вводов – низкий;
- 4) уровень внешних запросов – низкий;
- 5) уровень сложности внешних интерфейсных файлов – низкий.

СКОМО2

Метод функциональных точек Показатель степени Модель ранней разработки архитектуры Модель композиции

| Характеристики продукта | | Количество | Сложность | Итого |
|--|---|------------|------------|-----------|
| Обмен данными | 5 | 5 | Низкий (3) | 15 |
| Распределенная обработка | 5 | 2 | Низкий (4) | 8 |
| Производительность | 3 | 1 | Низкий (3) | 3 |
| Эксплуатационные ограничения по аппаратным ресурсам | 2 | 4 | Низкий (7) | 28 |
| Транзакционная нагрузка | 3 | 1 | Низкий (5) | 5 |
| Оперативный ввод данных | 4 | | | |
| Эргономические характеристики, влияющие на эффективность работы конечных пользователей | 1 | | | |
| Оперативное обновление | 4 | | | |
| Сложность обработки | 4 | | | |
| Повторное использование | 0 | | | |
| Легкость установки | 1 | | | |
| Легкость эксплуатации | 2 | | | |
| Портируемость | 2 | | | |
| Гибкость | 2 | | | |
| Итого: | | | | 59 |

| Процент использования ЯП | | | |
|--------------------------|-------|------------|-------|
| ASM | 0,00 | C# | 60,00 |
| C | 0,00 | Ada 95 | 0,00 |
| Cobol | 0,00 | SQL | 15,00 |
| Fortran | 0,00 | Visual C++ | 0,00 |
| Pascal | 0,00 | Delphi | 0,00 |
| C++ | 0,00 | Perl | 0,00 |
| Java | 25,00 | Prolog | 0,00 |

| | |
|-------------|--------------|
| AFPC | 60.77 |
| FP | 59 |
| SLOC | 3877 |

Рассчитать

Рисунок 5 – Метод функциональных точек

Нормированное количество функциональных точек: 60.77

Количество функциональных точек: 59

Количество строк исходного кода: 3877

1.2 Методика 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).
 - рассматривается макетирование пользовательский интерфейсов
 - оценивается производительность
 - 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 COCOMO 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 |

Преимущества COCOMO2:

- 1) возможен учет многих факторов;
- 2) универсальный метод;
- 3) фактические данные подбираются в соответствии с реальными проекта-

ми;

- 4) метод позволяет добавлять уникальные факторы для корректировки характеристик;
- 5) результаты прогнозирования сопровождаются обязательной документацией;
- 6) модель проста в освоении и применении

Недостатки СОСОМО2:

- 1) все результаты зависят от размера программного продукта;
- 2) игнорируются требования к характеристикам качества программного продукта;
- 3) игнорируется изменяемость требований к программному продукту;
- 4) игнорируются многие особенности, связанные с аппаратным обеспечением проекта;

Показатели проекта:

- 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 (команд новая, но были проведены определенные мероприятия по сплочению)
- Уровень развития процесса разработки (PMAT) – начальный уровень (только начинают внедрять)

Результат расчёта показателя степени Р

Скриншот интерфейса программы COCOMO2. Вкладка «Показатель степени».

Факторы показателя степени модели

| Фактор | Значение |
|--------|---|
| PREC | Полное отсутствие прецедентов, полностью непредсказуемый проект |
| FLEX | Большой частью согласованный процесс |
| RESL | Некоторое (40 %) |
| TEAM | Некоторая согласованность |
| PMAT | Уровень 1+ CMM |

Рассчитать

Результат: **P = 1.2441**

Рисунок 6 – Показатель степени

(Средняя ЗП принята равной 60 000)

Расчёт модели композиции проекта. По страницам:

- Авторизация – 3 простых поля и 1 средней сложности, 3 поколение
- Биржевые сводки – 3 простых поля и 1 средней сложности, 3 поколение
- Заявки - 1 простое поля и 2 средней сложности
- Новая заявка - 4 простых поля и 1 средней сложности

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

The screenshot shows the COCOMO2 software interface. At the top, there are four tabs: "Метод функциональных точек", "Показатель степени", "Модель ранней разработки архитектуры", and "Модель композиции", with the last one being active. Below the tabs, there are several input fields and a "Рассчитать" button.

| Экранные формы | | Отчеты |
|--------------------------------|--------|--------|
| Простые | 11 | 0 |
| Средние | 5 | 0 |
| Сложные | 0 | 0 |
| Модулей на языках 3 поколения | 2 | |
| %RUSE | 0 | |
| Опытность команды/разработчика | Низкая | |

Below the input fields is a "Рассчитать" button. At the bottom, the calculated results are displayed:

Трудозатраты, чел/мес 5.86
 Время, мес 5.38
 Бюджет 351600.0

Рисунок 7 – Метод композиции

Расчёт модели ранней разработки архитектуры:

- Надежность и уровень сложности (RCPX) – 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.
- Очень интенсивное использование инструментальных средств поддержки (FCIL).
- Жесткий график (SCED)

The screenshot shows the COCOMO2 software interface with the following elements:

- Navigation Tabs:**
 - Метод функциональных точек
 - Показатель степени
 - Модель ранней разработки архитектуры** (Active)
 - Модель композиции
- Input Fields:**
 - Средняя ЗП: 60000
 - PERS: Номинальный
 - RCPX: Очень высокий
 - RUSE: Низкий
 - PDIF: Высокий
 - PREX: Низкий
 - FCIL: Очень высокий
 - SCED: Очень высокий
- Buttons:**
 - Рассчитать
- Notes:**

Примечание: перед расчётом внесите данные проекта во вкладки "Метод функциональных точек" и "Показатель степени"
- Results:**
 - Трудозатраты, чел/мес: 27.57
 - Время, мес: 10.47
 - Бюджет: 1654200.0

Рисунок 8 – Модель ранней разработки архитектуры

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
- 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.