Минобрнауки России

Федеральное государственное автономное образовательное   
учреждение высшего образования   
  
«Национальный исследовательский университет   
«Московский институт электронной техники»

**Лабораторная работа №7 по дисциплине   
«Конструирование программного обеспечения»**

«Экономические аспекты разработки ПО»

Подготовили:

Студенты группы ПИН-36

Бойков И. И.

Бозюкова Л.С.

Карпухин Д.И.

Силантьев М.В.

Москва, 2024

1. Определение типа оценки.

На данном этапе проекта разработки программного обеспечения мы оцениваем объем функциональности, который будет предоставлен пользователям в первом релизе продукта. Эта оценка важна для определения масштабов работы и планирования ресурсов.

При проектировании конечного результата нашей системы мы первично будем опираться на эти данные.

1. Определение области оценки и границ продукта.

Область оценки охватывает все функции, связанные с управлением базой данных кассет, регистрацией пользователей, обработкой операций аренды и возврата кассет, а также генерацией отчетов. Четкое определение границ продукта позволяет сфокусироваться на ключевых аспектах разработки и избежать ненужных расходов.

1. Подсчет функциональных точек, связанных с данными.

Внутренние файлы (ILF):

* Таблица кассет: 1 ILF
* Таблица клиентов: 1 ILF
* Таблица арендных записей: 1 ILF

Внешние файлы (EIF):

* Интерфейс к внешней системе оплаты: 1 EIF

1. Подсчет функциональных точек, связанных с транзакциями.

**Внешние входы (EI)**:

* Ввод новой кассеты: 1 EI
* Регистрация нового клиента: 1 EI
* Запись аренды кассеты: 1 EI

**Внешние выходы (EO)**:

* Генерация отчетов по арендованным кассетам: 1 EO
* Уведомления о задержке возврата: 1 EO

**Взаимодействие с пользователем (EQ)**:

* Поиск кассеты по названию: 1 EQ
* Проверка статуса аренды: 1 EQ

5. Определение суммарного количества не выровненных функциональных точек (UFP).

Путем суммирования всех функциональных точек получаем:

**ILF**: 3 средних внутренних файла: 3 \* 4 = 12

**EIF**: 1 простой внешний файл: 5 \* 1 = 5

**EI**: 3 простых внешних входа: 3 \* 3 = 9

**EO**: 2 простых внешних выхода: 4 \* 2 = 8

**EQ**: 2 простых внешних запроса: 3 \* 2 = 6

Общее количество не выравненных функциональных точек (**UFP**) = 12 (ILF) + 5 (EIF) + 9 (EI) + 8 (EO) + 6 (EQ) = 40 UFP

1. Определение значения фактора выравнивания (FAV).

Оценка 14 аспектов системы, таких как сложность обработки данных, производительность, поддержка многопользовательского режима и другие.

Допустим, общий коэффициент выравнивания (FAV) составляет 1.2, что отражает требования к качеству и сложности проекта.

1. Расчет количества выровненных функциональных точек (AFP).

Рассчитаем факторы выравнивания:

* Размер функции (ФР) = 40
* Оценки характеристик:

1. Требуется ли резервное копирование данных? - 4 (часто)
2. Требуется обмен данными? - 3 (средне)
3. Используются распределенные вычисления? - 2 (редко)
4. Важна ли производительность? - 4 (часто)
5. Программа выполняется на сильно загруженном оборудовании? - 2 (редко)
6. Требуется ли оперативный ввод данных? - 5 (всегда)
7. Используется много форм для ввода данных? - 3 (средне)
8. Поля базы данных обновляются оперативно? - 4 (часто)
9. Ввод, вывод, запросы являются сложными? - 3 (средне)
10. Внутренние вычисления сложны? - 2 (редко)
11. Код предназначен для повторного использования? - 2 (редко)
12. Требуется преобразование данных и установка программы? - 3 (средне)
13. Требуется много установок в различных организациях? - 1 (иногда)
14. Требуется поддерживать возможность настройки и простоту использования? - 4 (часто)

* Суммируем оценки характеристик (S): 4+3+2+4+2+5+3+4+3+2+2+3+1+4=42
* Рассчитываем AFP (по формуле ФР \* (0,65 + 0,01 \* S):

*AFP = 40 \* (0,65 + 0,01 \* 42) = 42,8*

Определим ожидаемое количество строк кода, учитывая, что программный продукт разрабатывается на языке Java (в среднем 46 LOC/UFP):

*LOC = 56 \* UFP = 40 \* 56 = 1840*

**Оценим реальное количество строк кода в разработанном в Лабораторной работе № 6 минимально жизнеспособном продукте:**

***(52 + 46) + (44 + 124) + (39 + 44) + (47 + 41) + 110 = 547***

**Вывод**: **Результат оценки подтверждает проведённые расчёты, так как база данных, её контроллер и часть функций интерфейса ещё не реализованы.**

**Метод функциональных точек является гибким способом оценки размера программного продукта, так как позволяет оценить его размеры на начальных этапах проектирования программного продукта.**