**Лабораторная работа №6**

**Оценка размера продукта методом функциональных точек**

выполнил студент III курса ТЭФ

группы ТВ-61

Калашников-Травин Владислав

вариант №11

Киев - 2019

**Задача:**

Для обліку оренди квартир створюється автоматизована система. У внутрішній базі зберігається інформація про угоди, клієнта - Номер клієнта, ПІБ ,Серія і номер паспорта, а також інвентарні номери квартир та дані держреєстру квартир. Система забезпечує: введення/редагування даних про клієнта, квартиру та угоду; підготовку звітів про чинні угоди та про клієнтів; пошук потрібної квартири. Методом функціональних точок оцінити розмір системи при TDI=42.

Метод функциональных точек позволяет определить объём работ на начальном этапе разработки проекта с точки зрения пользователя. Метод определяет объём продукта количеством функционала, необходимым заказчиком и поставляемым разработчиком. Метод не зависит от платформы/языка/технологии разработки и предоставляет одинаковый подход к оценке всех проектов компании.

Результатом работы метода является объём продукта, представленный в виде суммы количеств информационных единиц и элементарных операций (транзакций).

Оценка продукта происходит в несколько шагов.

**Шаг 1. Определение рабочих файлов**

Количество рабочих (логических и внешних) файлов напрямую влияет на оценку. Файлы представляют собой логические наборы данных, состоящие из более мелких частей. Файлы можно разделить на две категории:

* внутренний логический (отвечает за представление данных внутри системы); сокр. ILF - Internal Logical File
* внешний интерфейсный (отвечай за вывод данных пользователю); сокр. External Interface File

Обычно, одному ILF отвечает один EIF.

Файлы можно разбить на подтипы:

* RET (Record Element Type) - логическая группа данных (запись)
* DET (Data Element Type) - элементарные части (поля), из которых состоят RET

Подсчитаем количества таких файлов.

RET 1. Договор (итого 5 DET):

* номер договора
* дата заключения
* поставщик
* покупатель
* квартира

RET 2. Клиент (итого 6 DET):

* номер клиента
* RET 2.1. ФИО клиента (итого 3 DET)
* RET 2.2. Паспортные данные: серия и номер (итого 2 DET)

RET 3. Квартира (итого 10 DET)

* инвентарный номер квартиры
* RET 3.1. Адрес: страна, город, улица, номер дома (итого 4 DET)
* площадь
* стоимость
* RET 3.2. Данные госреестра (3 DET)

Итого мы получили 3 файла, 7 RET и 21 DET. Примем это количество как за ILF, так и за ELF.

Оценим уровень сложности данных, используя матрицу сложности.



Исходя из таблицы, уровень сложности данных системы высокий.

**Шаг 2. Определение транзакций**

Транзакция представляет собой элементарный процесс или действие, которое влияет на работу с пользователем, т.е. является существенной. Отличают три вида транзакций:

* внешняя входящая (данные приходят из-вне)
* внешняя исходящая (подготовка и выдача данных наружу)
* внешний запрос (извлечение данных из файлов)

Определим транзакции в соответствии с заданием:

Внешние входящие:

* ввод данных о квартире (1 файл, 3 RET и 10 DET) - низкая сл.
* ввод данных о пользователе (1 файл, 3 RET и 6 DET) - низкая сл.
* ввод данных о договоре (1 файл, 1 RET и 5 DET) - низкая сл.
* редактирование данных о квартире (повторяется) - низкая сл.
* редактирование данных о пользователе (повторяется) - низкая сл.
* редактирование данных о договоре (повторяется) - низкая сл.

Внешние исходящие:

* подготовка отчётов по договорам (2 файла, 5 DET) - низкая сл.
* подготовка отчётов по клиентам (1 файл, 6 DET) - низкая сл.
* поиск квартир по критериям (1 файл, 10 DET) - низкая сл.

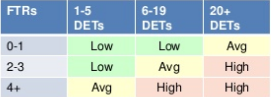
Внешние запросы:

* отображение отчётов по договорам (2 файла, 10 DET (условно)) - средняя сл.
* отображение отчётов по клиентам (1 файл, 11 DET (условно)) - средняя сл.
* отображение результатов поиска квартир (1 файл, 10 DET) - низкая сл.

Оценим уровень сложности транзакций , используя соответствующие матрицы.



Уровни сложности входящих транзакций



Уровни сложности исходящих транзакций/внешних запросов

Весы сложности для каждого уровня установлены таблицей:

|  |  |  |  |
| --- | --- | --- | --- |
| Тип | Низкий | Средний | Высокий |
| Внешняя входящая | 3 | 4 | 6 |
| Внешняя исходящая | 4 | 5 | 7 |
| Внешний запрос | 3 | 4 | 6 |

Сложим уровни, учитывая их весы:

TotalFP = 6 \* (1 \* 3) + 2 \* 4 + 1 \* 4 + 1 \* 4 + 2 \* 4 + 1 \* 4 + 1 \* 3 = 49 FP

Применим выравнивание функциональных точек, используя TDI (Total Degree of Influence) = 42. Этот уровень отображает вес общесистемных требований и представляет собой группу из двенадцати требований. Для выравнивания же используют VAF (Value Added Factor), который напрямую зависит от TDI и общее влияния требований на функциональные точки.

VAF = TDI \* 0.01 + 0.65 = 42 \* 0.01 + 0.65 = 1.07

Получим выровненное количество функциональных точек:

TotalAligned = TotalFP \* VAF = 49 \* 1.07 ~= 53 FP