*«Экономика программной инженерии».*

*Практическое занятие №5*

**Метод Function Points**

В настоящее время наиболее часто применяются метрики ПП, основанные на измерении функциональной полезности продукта, это так называемые функционально-ориентированные метрики.

Следует отметить, что применяемые при оценке трудоемкости модели содержат ряд параметров, численные значения которых определяют величину трудозатрат и являются нормативами (стандартами), принятыми на предприятии − разработчике ПП. Эти численные значения зависят от:

− квалификации разработчиков;

− используемых инструментальных средств;

− накопленного на предприятии опыта производства ПО.

Поэтому при применении соответствующей методики оценке трудоемкости изготовления ПО необходимо обоснованно задать численные значения соответствующим параметрам (коэффициентам) модели.

Область применения метода функциональных указателей – коммерческие информационные системы.

Достоинства функционально-ориентированных метрик:

− не зависят от языка программирования;

− легко вычисляются на любой стадии проекта.

Недостаток функционально-ориентированных метрик: результаты основаны на субъективных данных, используются не прямые, а косвенные измерения.

В методике функциональных указателей для определения объема работ используется 5 информационных характеристик:

внешний ввод – элементарный процесс, перемещающий данные из внешней среды в приложение. Данные могут поступать с экрана ввода или из другого приложения. Данные могут использоваться для обновления внутренних логических файлов. Данные могут содержать как управляющую, так и деловую информацию. Управляющие данные не модифицируют внутренние логические файлы;

внешний вывод – элементарный процесс, перемещающий данные, вычисленные в приложении, во внешнюю среду. Кроме того, в этом процессе могут обновляться внутренние логические файлы. Выводы означают отчеты, экраны, распечатки, сообщения об ошибках или выходные файлы, посылаемые другим приложениям. Отчеты и файлы создаются на основе внутренних логических файлов и внешних интерфейсных файлов. Дополнительно этот процесс может использовать вводимые данные: критерии поиска либо параметры, не поддерживаемые внутренними логическими файлами. Вводимые данные носят временный характер;

− внешний запрос – элементарный процесс, работающий как с вводимыми, так и с выводимыми данными. Его результат – данные, возвращаемые из внутренних логических файлов и внешних интерфейсных файлов. Входная часть процесса не модифицирует внутренние логические файлы, а выходная часть не несет данных, вычисляемых приложением (в этом и состоит отличие запроса от вывода);

− внутренний логический файл – распознаваемая пользователем группа логически связанных данных, которая размещена внутри приложения и обслуживается через внешние вводы;

− внешний интерфейсный файл – распознаваемая пользователем группа логически связанных данных, которая размещена внутри другого приложения и поддерживается им. Внешний файл данного приложения является внутренним логическим файлом в другом приложении.

Вводы, выводы и запросы относят к категории транзакция. Транзакция – это элементарный процесс, различаемый пользователем и перемещающий данные между внешней средой и программным приложением.

Оценка числа функциональных точек (ФТ) для программного продукта выводится на основе данных, которые определяются в результате анализа информационной области программного изделия и изучения особенностей его будущего функционирования.

Порядок расчета трудоемкости разработки ПО:

− определение количества и сложности функциональных информационных характеристик;

− определение количества связанных с каждым информационной характеристикой элементарных данных (DET), элементарных записей (RET) и файлов типа ссылок (FTR);

− определение сложности (в зависимости от количества DET, RET и FTR);

− подсчет количества функциональных точек приложения;

− подсчет количества функциональных точек с учетом общих характеристик системы оценка трудоемкости разработки (с использованием различных статистических данных).

Для транзакций ранжирование основано на количестве ссылок на файлы и количестве типов элементов данных. Для файлов ранжирование основано на количестве типов элементов-записей и типов элементов данных, входящих в файл.

Тип элемента-записи – подгруппа элементов данных, распознаваемая пользователем в пределах файла.

Тип элемента данных – уникальное не рекурсивное (неповторяемое) поле, распознаваемое пользователем. В качестве примера рассмотрим табл. 1.

Таблица 1 – Отчет радиостанции

|  |  |  |  |
| --- | --- | --- | --- |
| **День** | **Уровень активно-сти дня недели,**  **хиты** | **В процентах от Суммы хитов** | **Сеансы**  **пользователя** |
| Понедельник  Вторник  Среда  Четверг  Пятница  Суббота  Воскресенье | 1887  1547  1975  1591  2209  1286  1004 | 16,41  13,45  17,17  13,83  19,21  11,18  8,73 | 201  177  195  191  200  121  111 |
| Сумма по рабочим дням | 9209 | 80,08 | 964 |
| Сумма по выходным дням | 2290 | 19,91 | 232 |

Таблица 2 – Правила учета элементов данных для транзакций

|  |  |
| --- | --- |
| **Информационная**  **характеристика** | **Элементы данных** |
| Внешние Вводы  Внешние Выводы  Внешние Запросы | Поля ввода данных, сообщения об ошибках, вычисляемые значения, кнопки  Поля данных в отчетах, вычисляемые значения, сообщения об ошиб-ках, заголовки столбцов, которые читаются из внутреннего файла  Вводимые элементы: поле, используемое для поиска, щелчок мыши.  Выводимые элементы – отображаемые на экране поля |

Таблица 3 – Правила учета элементов данных из графического интерфейса пользователя

|  |  |
| --- | --- |
| **Элемент данных** | **Правило учета** |
| Группа радиокнопок  Группа флажков  (переключателей)  Командные кнопки  Списки | Так как в группе пользователь выбирает только одну радио-кнопку, все радиокнопки группы считаются одним элементом данных  Так как в группе пользователь может выбрать несколько флажков, каждый флажок считают элементом данных  Командная кнопка может определять действие добавления, запроса. Кнопка ОК может вызывать транзакции (различных ти-пов). Кнопка Next может быть входным элементом запроса или вызывать другую транзакцию. Каждая кнопка считается от-дельным элементом данных  Список может быть внешним запросом, но результат запроса может быть элементом данных внешнего ввода |

Эти сообщения не образуют самостоятельного процесса, они являются частью другого процесса, то есть считаются элементом данных соответствующей транзакции.

С другой стороны, уведомление является независимым элементарным процессом. Например, при попытке получить из банкомата сумму денег, превышающую их количество на счете, генерируется сообщение «Не хватает средств для завершения транзакции». Оно является результатом чтения информации из файла счета и формирования заключения. Сообщение уведомления рассматривается как внешний вывод.

Данные для определения ранга и оценки сложности транзакций и файлов приведены в табл. 4 – 8 (числовая оценка указана в круглых скобках). Например, внешнему вводу, который ссылается на 2 файла и имеет 7 элементов данных, по табл. 4 назначается средний ранг и оценка сложности 4.

Таблица 4 – Ранг и оценка сложности внешних вводов

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

Таблица 5 – Ранг и оценка сложности внешних выводов

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

Таблица 6 – Ранг и оценка сложности внешних запросов

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

Таблица 7 – Ранг и оценка сложности внутренних логических файлов

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

Таблица 8 – Ранг и оценка сложности внешних интерфейсных файлов

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

Отметим, что если во внешнем запросе ссылка на файл используется как на этапе ввода, так и на этапе вывода, она учитывается только один раз. Такое же правило распространяется и на элемент данных (однократный учет).

После определения всех информационных характеристик программного продукта и их сложности приступают к расчету метрики – количества функциональных указателей FP (Function Points).

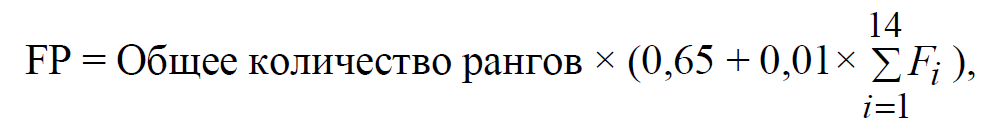
Исходные данные для расчета сводятся в табл. 9.

Таблица 9 – Исходные данные для расчета FP-метрик

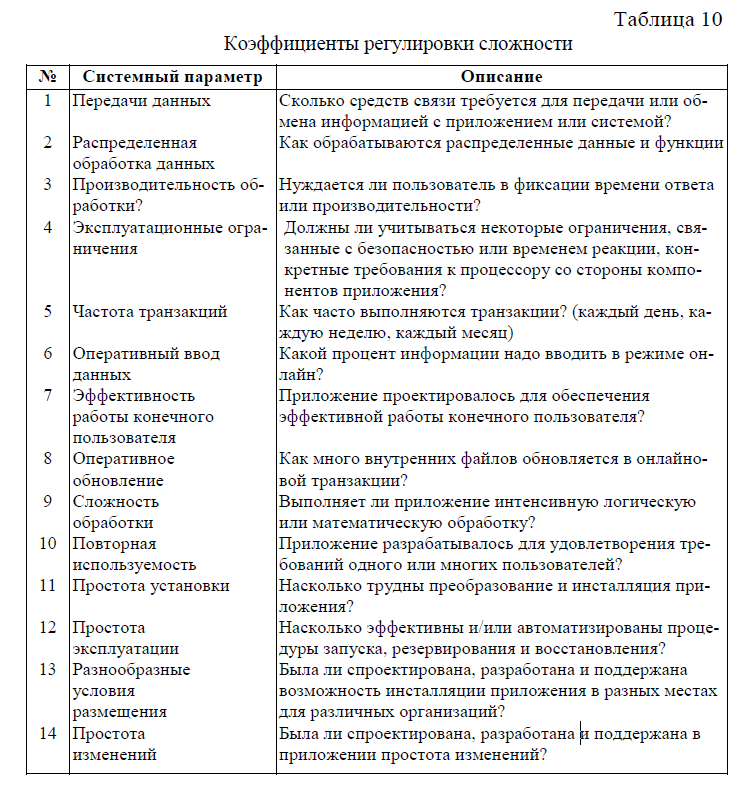
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Имя характеристики** | | | | **Ранг, сложность, количество** | | | |
| Низкий | | **Средний** | | **Высокий** | | **Итого** | |
| Внешние вводы  Внешние выводы  Внешние запросы  Внутренние логические файлы  Внешние интерфейсные файлы | x3 = \_\_\_  x4 = \_\_\_  x3 =\_\_\_  x7 = \_\_\_  x5 = \_\_\_ | | x4 = \_\_\_  x5 = \_\_\_  x4 = \_\_\_  x10=\_\_\_  x7 = \_\_\_ | | x6 =\_\_\_   х 7 =\_\_\_  x6 =\_\_\_  x15 =\_\_\_  x10 =\_\_\_ | | =  =   =  =  = |
| Общее количество рангов | | | | =  | | | |

В таблицу заносится количественное значение характеристики каждого вида (по всем уровням сложности). Места подстановки значений отмечены прямоугольниками (прямоугольник играет роль метки-заполнителя). Количественные значения характеристик умножаются на числовые оценки сложности. Полученные в каждой строке значения суммируются, давая полное значение для данной характеристики. Эти полные значения затем суммируются по вертикали, формируя общее количество.

Количество функциональных указателей вычисляется по формуле



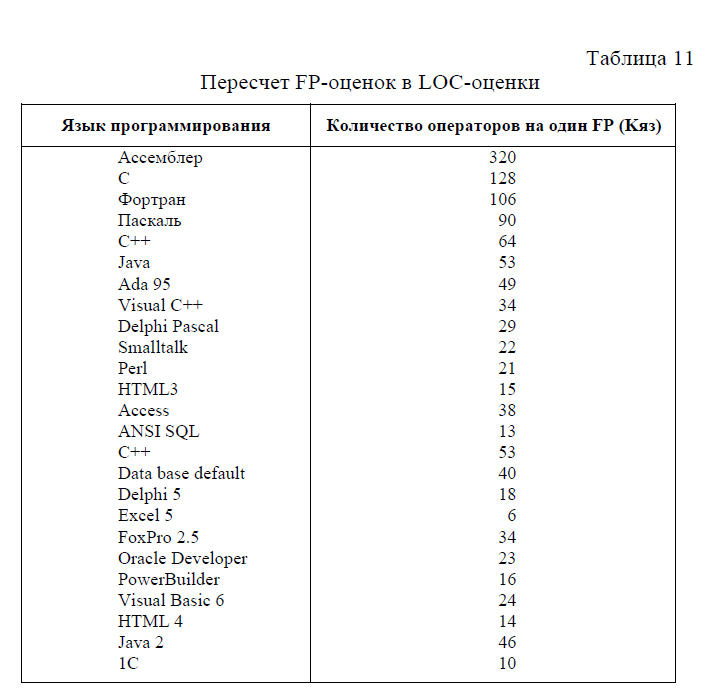
где Fi – коэффициенты регулировки сложности (табл. 10), принимающие целые значения: 0 – 5 в зависимости от сложности реализации соответствующей характеристики ПП.



Полученная FP-оценка пересчитывается в LOC-оценки V

V = Kяз × FP,

где коэффициент Kяз зависит от языка программирования, используемого для реализации ПО, берется из табл. 11.



Для пересчета объема программы в условных строках V в трудозатраты T используется промежуточная модель СОСОМО, в соответствии с которой номинальную трудоемкость (без учета коэффициентов затрат труда, стоимостных факторов и сложности) можно вычислить по формуле

*Т = N1 × KSLOCN2*,

где KSLOC (тыс. строк) = V / 1000

Значения N1 и N2 определяются по табл. 12 [3].

Таблица 12 – Коэффициенты N1 и N2

|  |  |  |
| --- | --- | --- |
| **Тип ПО** | **N1** | **N2** |
| Распространенное | 3,2 | 1,05 |
| Полунезависимое | 3,0 | 1,12 |
| Встроенное | 2,8 | 1,20 |

Распространенное ПО − ПО небольшого объема (не более 50 KSLOC), разрабатываемое относительно небольшой группой опытных специалистов в стабильных условиях.

Полунезависимое ПО − ПО среднего объема (не более 300 KSLOC), разрабатываемое неоднородной группой специалистов средней квалификации.

Встроенное ПО − ПО с жесткими ограничениями (система резервирования авиабилетов, система управления воздушным движением и т.п.).

Время разработки вычисляется по формуле

*tразр = 2,5 × ТN3*,

где значения N3 берутся из табл. 13 [3].

Таблица 13 – Коэффициент N3

|  |  |
| --- | --- |
| **Тип ПО** | **N3** |
| Распространенное | 0,38 |
| Полунезависимое | 0,35 |
| Встроенное | 0,32 |

Рекомендуемое правило распределения затрат проекта – 40−20−40:

- на анализ и проектирование приходится 40% затрат (из них на планирование и системный анализ – 5%);

- на кодирование – 20%;

- на тестирование и отладку – 40%.

***Список использованных источников***

1. Вендров, А. М. Проектирование программного обеспечения эконо-мических информационных систем: учебник для студ. экон. вузов, обу-чающихся по спец. "Прикл. информатика (по обл.)" и "Прикл. математика и информатика" / Вендров А. М. − 2-e изд., перераб. и доп. − М.: Финансы и статистика, 2006. − 543 с.: ил.

2. Вендров, А. М. Практикум по проектированию программного обес-печения экономических информационных систем: учебное пособие для ву-зов / Вендров А. М. − 2-e изд., перераб. и доп. − М.: Финансы и статистика, 2006. − 191 с.: ил.

3. Орлов, С. А. Технологии разработки программного обеспечения. Современный курс по программной инженерии: учебник для вузов / Орлов С. А., Цилькер Б. Я. − 4-e изд. − Санкт-Петербург [и др.]: Питер, 2012. − 608 с.: ил. (Стандарт третьего поколения).

**Задание**

При оформлении отчета по практическому занятию № 5 «Метод Function Points» следует:

* провести расчёт *объёма программного проекта и времени разработки* для реализации своего IT-проекта, в качестве исходных данных для расчёта надлежит использовать вводные предыдущих практических занятий;
* представить затраты на IT-проект в табличной форме и графически.