



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования

«Московский государственный технический
университет имени Н.Э. Баумана»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Программное обеспечение ЭВМ и информационные технологии

Лабораторная работа № 7

Дисциплина: «Экономика программной инженерии»

Студент Овчинникова А. П.

Группа ИУ7-85Б

Вариант 15

Москва, 2020 г.

Теоретическая часть

Функциональная точка — это единица измерения функциональности программного обеспечения. Функциональность программы связана с обработкой информации по запросу пользователя и не зависит от применяемых технических решений. Пользователи — это отправители и целевые получатели данных, ими могут быть как реальные люди, так и смежные интегрированные информационные системы.

Метод функциональных точек позволяет:

- оценивать категории пользовательских бизнес-функций;
- разрешить проблему, связанную с трудностью получения ЛОС – оценок на ранних стадиях жизненного цикла;
- определять количество и сложность входных и выходных данных, их структуру, а также внешние интерфейсы, связанные с программной системой.

Трудоемкость вычисляется на основе функциональности разрабатываемой системы, которая, в свою очередь, определяется путем выявления функциональных типов — логических групп взаимосвязанных данных, используемых и поддерживаемых приложением, а также элементарных процессов, связанных с вводом и выводом информации.

Типы элементарных процессов, используемых в методе функциональных точек:

- Внешний ввод (ЕІ, транзакция, получающая данные от пользователя) — элементарный процесс, перемещающий данные из внешней среды в приложение. Данные могут поступать с экрана ввода или из другого приложения. Данные могут содержать как управляющую, так и деловую информацию. Обработываемые данные могут соответствовать одному или нескольким внутренним логическим файлам.
- Внешний вывод (ЕО, транзакция передающая данные пользователю) — элементарный процесс, перемещающий данные, вычисленные в приложении, во внешнюю среду и связанный с созданием и/или обработ-

кой выходной информации приложения — выходного отчета, документа, экранной формы.

- Внешний запрос (EQ, интерактивный диалог с пользователем, требующий от него каких-либо действий) — элементарный процесс, состоящий из комбинации «запрос/ответ», не связанный с вычислением производных данных или обновлением внутренних логических файлов (базы данных).
- Внутренний логический файл (ILF, информация, которая используется во внутренних взаимодействиях системы) — выделяемые пользователем логически связанные группы данных или блоки управляющей информации, которые поддерживаются внутри продукта и обслуживаются через внешние вводы.
- Внешний интерфейсный файл (EIF, файлы, участвующие во внешних взаимодействиях с другими системами) — выделяемые пользователем логически связанные группы данных или блоки управляющей информации, на которые ссылается продукт, но которые поддерживаются вне продукта.

Расчет скорректированного количества функциональных точек:

$$FP = \text{Общее количество} \cdot (0.65 + 0.01 \cdot \sum Fi)$$

где Fi — 14 коэффициентов регулировки сложности.

Для пересчета FP-оценок в LOC-оценки используется количество операторов на один FP для конкретного ЯП (для ассемблера 320).

В модели COSOMO2 используются три модели оценки стоимости:

- Модель композиции приложения — это модель, которая подходит для проектов, созданных с помощью современных инструментальных средств. Единицей измерения служит объектная точка.
- Модель ранней разработки архитектуры. Эта модель применяется для получения приблизительных оценок проектных затрат периода выполнения проекта перед тем как будет определена архитектура в целом. В этом случае используется небольшой набор новых драйверов затрат и новых уравнений оценки. В качестве единиц измерения используются функциональные точки либо KSLOC.

- Постархитектурная модель – наиболее детализированная модель СОСО-МОП, 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 используется на ранней стадии конструирования ПО, когда:

- рассматривается макетирование пользовательских интерфейсов;
- оценивается производительность;
- определяется степень зрелости технологии.

Модель ориентирована на применение объектных точек. Объектная точка — средство косвенного измерения ПО. Подсчет количества объектных точек производится с учетом количества экранов (как элементов пользовательского интерфейса), отчетов и компонентов, требуемых для построения приложения.

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

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

PROD – оценка скорости разработки, зависит от опытности команды разработчиков.

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

В модели ранней разработки архитектуры

$$\text{Трудозатраты} = 2.45 * eArch * (\text{Размер})^p,$$

Размер – количество тысяч строк кода (kLOC).

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

$$\text{Время} = 3,0 * (\text{Трудозатраты})^{(0.33+0.2 \cdot (p-1.01))}$$

На показатель степени в модели СОСОМО2 влияет пять факторов:

$$p = \frac{(PREC \cdot FLEX \cdot RESL \cdot TEAM \cdot PMAT)}{100} + 1.01$$

Подсчет количества функциональных точек

В ПО имеется два внутренних логических файла (ILF).

Один для хранения информации о пользователях. Число типов элементов записей (RET) для этого файла равно двум (id - число, все остальное - строки). Число типов элементов данных (DET) внутреннего логического файла будет равно шести. Таким образом, уровень сложности внутреннего логического файла – низкий.

Второй ILF имеет три элемента данных. Число типов элементов записей равно двум. Уровень сложности низкий.

ПО имеет один внешний интерфейсный файл. Число типов элементов записей (RET) для этого файла равно трем. Число типов элементов данных (DET) внутреннего логического файла будет равно шести. Таким образом, уровень сложности внутреннего логического файла – низкий.

Внешние вводы ПО:

- Регистрация (мобильное приложение и веб портал). Ссылается на один внутренний логический файл и имеет четыре элемента данных. Уровень сложности - низкий.
- Оплата штрафа (мобильное приложение и веб-портал). Ссылается на один внешний интерфейсный файл и имеет шесть элементов данных. Уровень сложности низкий.
- Добавление пользователей в БД (веб-портал). Ссылется на один внутренний логический файл и имеет четыре элемента данных. Уровень сложности низкий.
- Получение списка штрафов (веб-портал). Ссылается на один внешний интерфейсный файл и имеет семь элементов данных. Уровень сложности низкий.
- Получение сообщения об успешном или неуспешном оплате штрафа от ГИБДД (веб-портал). Имеет один элемент данных и не ссылается на внутренние файлы. Уровень сложности низкий.
- Ответ о результате оплаты от платежной системы (веб-портал). Имеет три элемента данных и ссылается на два внутренних файла. Уровень сложности низкий.

Внешние выводы ПО:

- Вывод сообщения о положительном или отрицательном результате оплаты штрафа. Уровень сложности этого внешнего вывода – низкий, так как он имеет один DET и один FTR (веб портал + мобильное приложение).
- Запрос на получение списка штрафов. Ссылается на один внешний интерфейсный файл и имеет один элемент данных. Уровень сложности низкий.
- Оповещение ГИБДД об оплате штрафа (веб-портал). Ссылается на два внутренних логических файла и имеет восемь элементов данных. Уровень сложности средний.
- Запрос об оплате штрафа (веб-портал). Имеет три элемента данных и ссылается на два внутренних файла. Уровень сложности низкий.

Ввод данных для метода функциональных точек представлен на рисунке 1. Ввод данных для методики СОСОМО2 и результаты расчетов представлены на рисунках 2 - 3.

Количество простых экранных форм примем равным восьми. Количество модулей, написанных на ЯП третьего программирования – пять (на Java и на JavaScript).

MainWindow

Параметр	Просто		Средне		Сложно		Итого
	Количество	Коэффициент	Количество	Коэффициент	Количество	Коэффициент	
Внешние входы (EI)	7	3	0	6	0	4	Nan
Внешние выходы (EO)	4	4	1	5	0	7	Nan
Внешние запросы (EQ)	0	3	0	4	0	6	Nan
Внутренние логические файлы (ILF)	2	7	0	10	0	15	Nan
Внешние логические файлы (EIF)	1	5	0	7	0	10	Nan
Общее количество							Nan

Системные параметры приложения

Передача данных	5	Оперативное обновление	4
Распределенная обработка данных	5	Сложность обработки	4
Производительность	3	Повторная используемость	3
Эксплуатационные ограничения	0	Легкость инсталляции	3
Частота транзакций	3	Легкость эксплуатации	3
Оперативный ввод данных	2	Количество возможных установок на различных платформах	5
Эффективность работы конечных пользователей	0	Простота изменений (гибкость)	0

Язык программирования

SQL	30	%
JavaScript	10	%
Java	60	%

Рассчитать

Рис. 1: Ввод данных для метода функциональных точек.

Dialog

Экспертная оценка стоимости человеко-месяца

50000 руб

Показатель степени в модели

Новизна проекта (PREC) Полное отсутствие прецедентов, полностью непре...

Гибкость процесса разработки (FLEX) Точный, строгий процесс разработки

Разрешение рисков в архитектуре системы (RESL) В целом (75%)

Сплоченность команды (TEAM) Повышенная согласованность

Уровень зрелости процесса разработки (PMAT) Уровень 2 CMM

Модель ранней разработки архитектуры

Сложность продукта (RCPX) Очень выск

Необходимость повторного использования (RUSE) Низкий

Сложность платформы (PDIF) Номиналь

Опытность персонала (PREX) Низкий

Способности персонала (PERS) Очень выск

Возможности среды (FCIL) Высокий

Сроки (SCED) Номиналь

Рассчитать

Результат

Трудозатраты	8.398	человеко-месяцев
Длительность	6.62	месяцев
Средняя численность команды разработчиков	1	человек
Бюджет	331000.0	рублей

Модель композиции приложения

Результат

Рис. 2: Ввод данных для методики COCOMO2 (часть1).

Dialog

Модель ранней разработки архитектуры

Сложность продукта (RCPX) Очень вык ▾

Необходимость повторного использования (RUSE) Низкий ▾

Сложность платформы (PDIF) Номиналь ▾

Опытность персонала (PREX) Низкий ▾

Способности персонала (PERS) Очень вык ▾

Возможности среды (FCIL) Высокий ▾

Сроки (SCED) Номиналь ▾

Рассчитать

Результат

Трудозатраты 8.398 человеко-месяцев

Длительность 6.62 месяцев

Средняя численность команды разработчиков 1 человек

Бюджет 331000.0 рублей

Модель композиции приложения

%RUSE 0 ▾

Опытность команды/разработчика Низкая ▾

Экранные формы

Простые 8 ▾ Умеренные 0 ▾ Сложные 0 ▾

Отчеты

Простые 0 ▾ Умеренные 0 ▾ Сложные 0 ▾

Модули на языках 3 поколения 5 ▾

Рассчитать

Результат

Трудозатраты 8.286 человеко-месяцев

Длительность 6.865 месяцев

Средняя численность команды разработчиков 1 человек

Бюджет 343250.0 рублей

Рис. 3: Ввод данных для методики COSOMO2 (часть 2).

Вывод

Модель COSOMO2 позволяет более полно учитывать факторы, влияющие на экономические характеристики производства сложных программных продуктов, а также учитывать уникальные факторы для корректировки экономических характеристик, связанные со специфическим проектом и организацией.

COSOMO2 и метод функциональных точек предоставляет возможность оценить объем проекта если собственный опыт аналогичных проектов отсутствует, а экспертное мнение недоступно.