

Оценка трудоемкости программных проектов

В.А. Макаров

Новгородский государственный
университет

Аксиома

Фирма, занимающаяся промышленной разработкой программного обеспечения, должна владеть методикой и иметь инструментальные средства оценки трудоемкости программных проектов

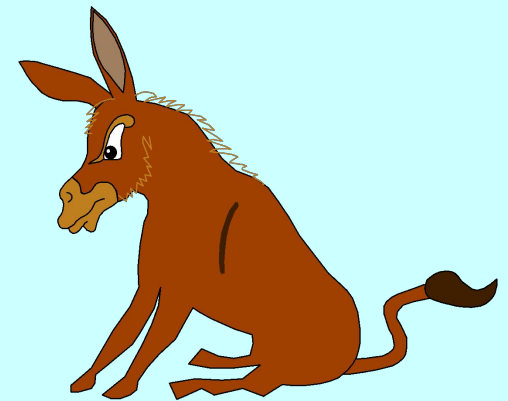
Методики оценки размеров ПО

Внешние метрики и факторы	Внутренние метрики
Методика Функциональных точек Function Points [IFPUG, Mark II]	Строки исходного кода (SLOC) Source Lines of Code (SLOC)
Оценка пользовательских требований к функциональности Functional User Requirements [COSMIC]	Операторы и Операнды Operators and Operands [Maurice Halstead]
Тип приложения или Производительность разработки Application Type or Productivity [SLIM]	Метрики для объектно - Ориентированного ПО Metrics for Object-Oriented Software Engineering (MOOSE) [Chidamer and Kemerer]
Основные характеристики системы General System Characteristics [Task points]	Характерная точка алгоритмов Feature Point Algorithms [Capers Jones]
Проектные ограничения Project Constraints [COCOMO Development Mode]	Метод функциональных точек Experience Pro [Pekka Forselius]

Этапы определения трудозатрат

7 этапов:

- **Описание проекта**
- **Оценка функционального размера в *fp***
- **Анализ повторного использования**
- **Анализ ситуации**
- **Поиск аналогов**
- **Общий расчет трудозатрат**
- **Оценка риска**



Описание проекта

- Общая информация
- Тип программного продукта
- Среда функционирования
- Инструментальная среда разработки
- Основные этапы разработки
- Предварительное описание бригады проекта

Оценка функционального размера в *fp*

Функции программы

- **Сущности**
- **Ввод**
- **Вывод**
- **Внешние
интерфейсы**
- **Запросы**
- **Алгоритмы**



СУЩНОСТИ (ENTITIES)

Логически целостный элемент данных
определенный в программе

- **Классы** в ООП
- **Функции и процедуры** в функционально – ориентированных языках
- **DB - таблицы** в базах данных

Общая классификация сущностей

СУЩНОСТЬ	Общее число параметров сущности				
	1-3	4-6	7-16	17-35	36 и более
	2	5	7	10	13

ВВОД (INPUT)

Любые действия направленные на создание, изменение или удаление входных данных программы.

Ввод с экрана, пакетный ввод, сообщения от других систем.

Общая классификация функций ввода

ВВОД Общее число сущностей, для которых вводимые параметры являются входными	Общее число вводимых параметров				
	1-2	3-4	5-11	12-19	20 и более
1	1	1	3	3	4
2	1	3	4	4	6
3-4	3	4	4	4	6
5-7	3	4	4	6	10
8 и более	4	6	6	10	10

ВЫВОД (OUTPUT)

Вывод результатов работы программы.

Вывод результатов на экран, получение отчетов (reports), посылка электронной почты.

Общая классификация функций вывода

ВЫВОД Общее число сущностей, которые задействованы в процессе подготовки выводимых параметров	Общее число выводимых параметров				
	1-4	5-10	11-18	19-26	27 и более
1	2	2	4	4	5
2-3	2	4	5	5	7
4-5	4	5	5	5	7
6-9	4	5	5	7	11
10 и более	5	7	7	11	11

ИНТЕРФЕЙСЫ (INTERFACES)

Элементы данных, которые выходят за границы текущего приложения и связывают разрабатываемую систему с внешними модулями (средство взаимодействия систем)

- **Совокупность данных (batch)**, передаваемых как единое целое в другие приложения.
- **Сообщения (messages)** - код, сообщаемый другому приложению или полученный от внешнего приложения

Общая классификация интерфейсов

ИНТЕРФЕЙСЫ	Общее число параметров интерфейса				
	1-4	5-9	10-30	31-60	
	2	5	7	10	Более 60 13

ЗАПРОСЫ (INQUIRIES)

Операции, направленные на получение информации о состоянии данных программы без изменения самих данных.

Меню, иконки, формы для ввода информации, пользовательские **запросы**.
Передача данных между компонентами программы.

Общая классификация запросов

Запросы Общее число сущностей, которые задействованы в процессе обработки запроса	Общее число параметров запроса				
	1	2-3	4-10	11-19	20 и более
1	1	1	3	3	4
2-3	1	3	4	4	6
4-5	3	4	4	4	6
6-7	3	4	4	6	10
8 и более	4	6	6	10	10

АЛГОРИТМЫ (ALGORITHMS)

Математическая и/или логическая
последовательность операций,
направленная на изменение исходных
данных

Общая классификация функций

Алгоритм

АЛГОРИТМ число элементарных блоков обработки данных	Общее число обрабатываемых параметров				
	1-3	4-6	7-12	13-19	30 и более
1- 4	1	1	3	3	4
5-7	1	3	4	4	6
8-10	3	4	4	4	6
11-15	3	4	4	6	10
16 и более	4	6	6	10	10

Определение суммарной оценки в fp

$$FP_{\Sigma} = FP_I + FP_O + FP_Q + FP_A + FP_E + FP_F$$

$$FP_X = \sum_i fp_i$$

- FP_I - трудоемкость функций Ввод,
- FP_O - трудоемкость функций Вывод
- FP_Q - трудоемкость функций Запрос
- FP_A - трудоемкость функций Алгоритм
- FP_E - трудоемкость функций Сущность
- FP_F - трудоемкость функций Интерфейс.

Оценка функционального размера

ДОСТОИНСТВА

- Детализация программы
- Понятна заказчику и разработчику
- Слабо зависит от инструмента разработки
- Оценка производительности труда

НЕДОСТАТКИ

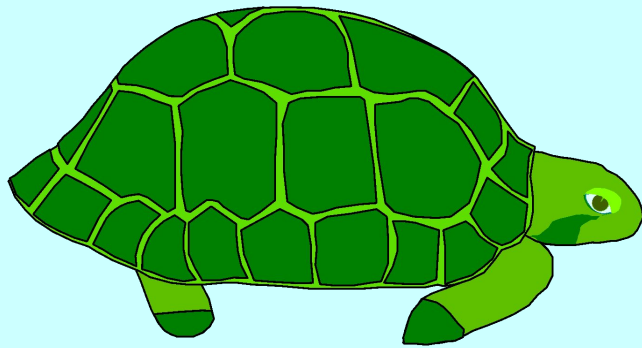
- Трудность в детализации функций
- Точная оценка требует большой статистики
- Плохо оценивает большие проекты

Оценка функционального размера

Основные принципы:

- Максимальная детализация
- Баланс категорий функций
- Оптимальный размер 400-600 *fp* (< 2000)
- Дублирование функций по типам
- Упрощение правил классификации

Анализ повторного использования



- Оценка возможности и уровня повторного использования компонент
- Коррекция ранее полученного значения трудоемкости, выраженной в *fp*.
- Внесение изменений в календарный план проекта

Анализ повторного использования

(Расчет функционального размера)

$$FP_x = \sum_{j=1}^{j=n} K_x(j) \cdot fp_x(j)$$

$K_x(j)$ - коэффициент использования j -ой функции

Анализ ситуации

- Оценивается *21* фактор проектной ситуации.
- Рассчитывается коэффициент производительности

Анализ ситуации

- факторы проекта (5 факторов);
- факторы процесса (5 факторов);
- факторы продукта (6 факторов);
- человеческие факторы (5 факторов).

Анализ ситуации

Факторы производительности	- -	-	+ -	+	++
<i>Факторы проекта</i>					
Взаимодействие в группе					
Уровень инструментальных средств					
Степень готовности персонала					
Число взаимосвязей					
Уровень напряженности календарного плана					

Анализ ситуации

<i>Факторы процесса</i>					
Использование стандартов					
Методы программирования					
Использование инструментальных средств					
Стабильность задания					
Стабильность процесса					

Анализ ситуации

<i>Факторы программного продукта</i>					
Логическая сложность разработки					
Объем программы (число сущностей)					
Число интерфейсов					
Требования качества					
Требования эффективности					
Требования к внедрению					

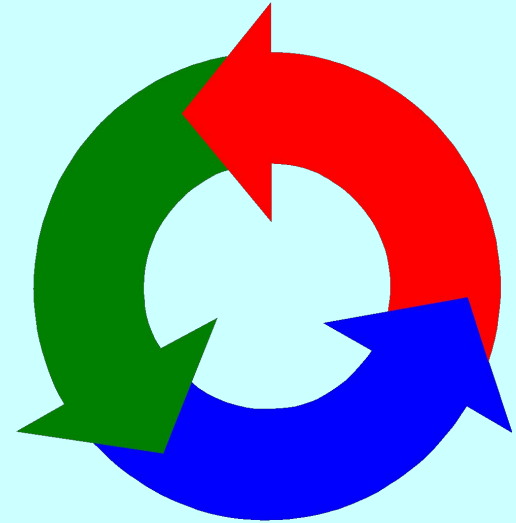
Анализ ситуации

<i>Человеческие факторы</i>					
Анализ навыков					
Знание прикладной области					
Знание инструментальных средств					
Опыт руководителя проекта					
Навыки согласованной работы					
Итого:	α	β	γ	φ	μ

$$T = 1.2^{\alpha} \cdot 1.1^{\beta} \cdot 1.0^{\gamma} \cdot 0.9^{\varphi} \cdot 0.8^{\mu}$$

Поиск аналогов

- **Главная цель** – обеспечение поступательного увеличения скорости разработки - производительности
- **Главный результат** – определение значения P (*час/фп*) для рассматриваемого проекта.



$P - ?$

Итоговая оценка трудоемкости

$$S = FP \cdot T \cdot P$$

- S - итоговое значение трудоемкости в часах
- FP –трудоемкость проекта в fp , полученная после этапа расчета повторного использования
- T - коэффициент производительности
- P - производительность (час/ fp)

Анализ рисков

Особенности модели

- два варианта моделей для оценки риска: **Top21** и **Расширенная модель риска** .
- Обе модели рассматривают все факторы риска, исходя из **5-уровней**. Риск каждого уровня имеет **5 - состояний**.

Анализ рисков

$$Risk = \sum_{i=1}^{i=n} V_i \cdot Im_i$$

Risk – интегрированный показатель факторов риска

V_i - состояние *i*-го фактора риска (диапазон 1...5)

Im_i - воздействие *i*-го фактора риска (диапазон 1...5)

n - число факторов риска, включенных в оценку.