

Практична робота №3

Аналіз функціональних точок

Мета: Навчитися оцінювати характеристики програмних продуктів на основі аналізу функціональних точок.

Теоретичні відомості

Основні положення методу функціональних точок

Метод функціональних точок був вперше запропонований в 1983 р. Аланом Альбрехтом і на даний момент є основним для оцінки функціонального розміру програмних продуктів, як вже готових, так і таких, що знаходяться на стадії проектування або супроводження.

Функціональні точки є мірою функціональності програмних продуктів з позиції користувача. Функціональність визначається і вимірюється шляхом виконання наступного:

- аналізу логічних груп даних, які використовуються і підтримуються програмним продуктом і характеризують, по суті, функціональність даних;
- аналізу функціональності введеної і виведеної користувачем інформації, тобто функціональності здійснюваних транзакцій.

Загальна функціональність є сумою двох складових (рис. 3.1):

$$\text{Функціональність} = \text{Функціональність Даних} + \text{Функціональність Транзакцій}$$

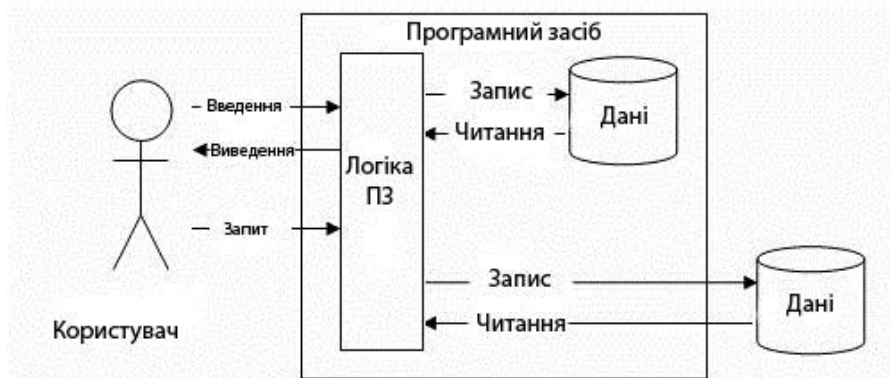


Рис. 3.1

Послідовність при застосуванні аналізу методом функціональних точок (рис. 3.2):

1. Визначення типу оцінки.
2. Визначення області оцінки і кордонів продукту.
3. Підрахунок функціональних точок, пов'язаних з даними.
4. Підрахунок функціональних точок, пов'язаних з транзакціями.
5. Визначення сумарного кількості не вирівняних функціональних точок (UFP).
6. Визначення значення фактора вирівнювання (FAV).
7. Розрахунок кількості вирівняних функціональних точок (AFP).

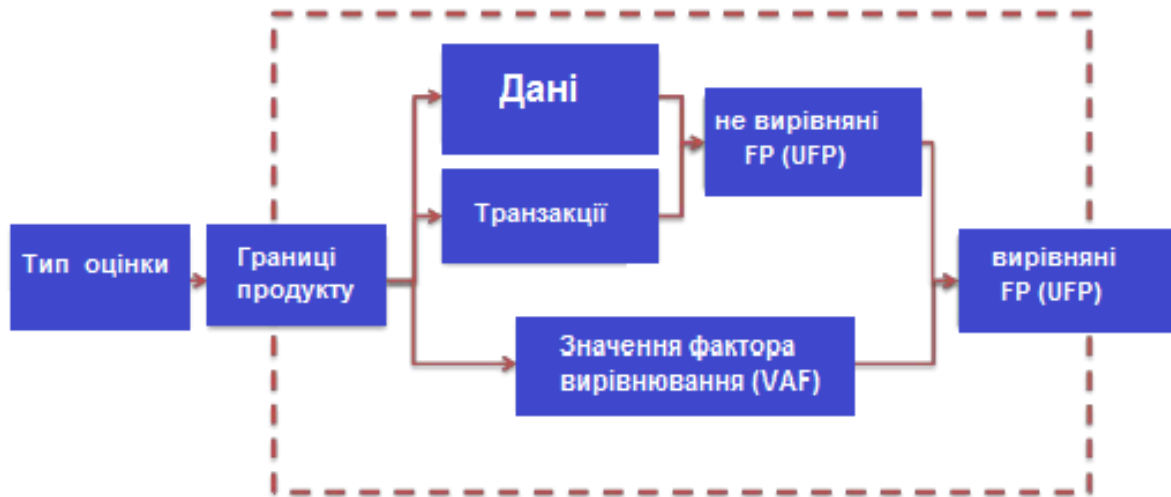


Рисунок 3.2 Схема аналізу функціональних точок

1) Визначення типу оцінки

Підрахунок функціональних точок може бути пов'язаний або з проектами, або з програмними продуктами. Існує три типи підрахунку функціональних точок:

- ✓ Проект створення програмного продукту
- ✓ Проект вдосконалення програмного продукту
- ✓ Програмний продукт

2) Визначення границі продукту

Може включати:

- ✓ Всі функції що розробляються *(для проекту розробки)*
- ✓ Всі функції що можуть додаватися, змінюватися і видалятися *(для проектів підтримки)*
- ✓ Тільки функції, що реально використовуються, або всі функції *(при оцінці продукту і / або продуктів)*

Границі продукту визначають:

- ✓ Що є «зовнішнім» по відношенню до оцінюваного продукту
- ✓ Де розташовується «границі системи», через яку проходять транзакції передаються або приймаються продуктом, з точки зору користувача
- ✓ Які дані підтримуються додатком, а які - зовнішні.

3) Зазвичай функціональність даних представляється файлами, таблицями баз даних, об'єктами та іншими одиницями зберігання інформації. При аналізі за методом функціональних точок розглядаються два види груп даних:

– внутрішній логічний файл (ILF - Internal Logical File) - логічно пов'язана група даних, що визначається користувачем і знаходиться всередині кордонів ПЗ (рис. 3.3 а);

– зовнішній інтерфейсний файл (EIF - External Interface File) - логічно пов'язана група даних, або блоки керуючої інформації, на які посилається продукт, але які підтримуються поза продукту (рис. 3.3 б).

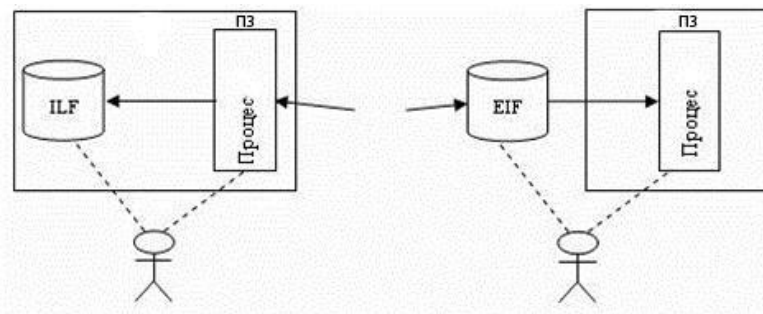


Рис.3.3.

4) Транзакції - це елементарні процеси, тобто найменші одиниці активності, мають сенс для користувача, вони відбуваються всередині ПЗ і породжують вхідну і вихідну інформацію. В аналізі, заснованому на методі функціональних точок, виділяють **три види транзакцій**:

- зовнішнє введення (EI - External Input) - процес введення даних і керуючої інформації в ПЗ (рис.3.4а). Керуюча інформація необхідна для правильної обробки даних. Дані, що надходять на вхід ПЗ, використовуються для підтримки внутрішнього логічного файлу. Зазвичай, процеси виду EI використовуються для додавання, зміни або видалення інформації;

- зовнішнє виведення (EO - External Output) - процес, що генерує дані або керуючу інформацію, які надходять на вихід ПЗ (рис.3.4б). Зазвичай процес виду EO є формування різних екранів, звітів, повідомлень;

- зовнішній запит (EQ - External Inquiry) - діалогове введення, яке призводить до негайної відповіді ПЗ в формі діалогового виведення (рис.3.4в). При цьому діалогове введення в самому ПЗ не зберігається, а діалоговий висновок не вимагає виконання обчислень. У цьому полягає головна відмінність EQ від EI і EO.

Кожній з виявлених характеристик функціональності ПЗ (EI, EO, EQ, ILF, або EIF) ставиться у відповідність низький, середній або високий рівень складності, а потім присвоюється певна числова оцінка.

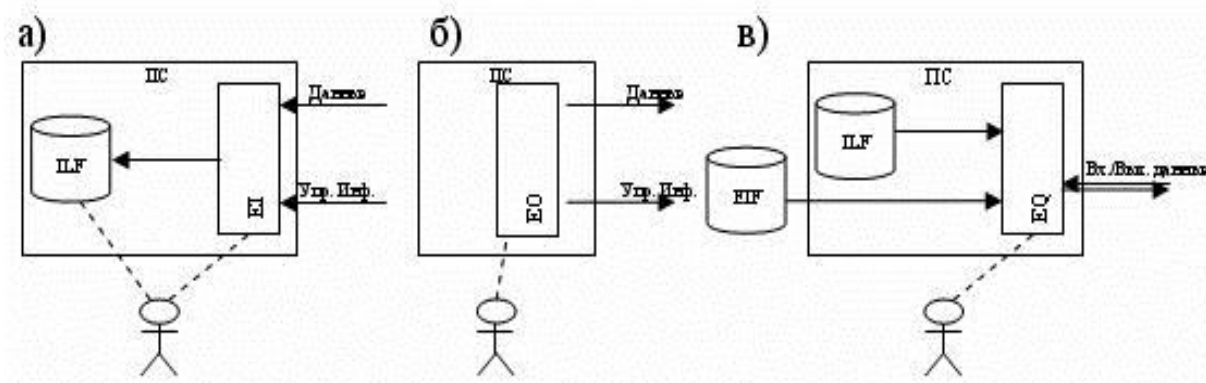


Рис. 3.4

Для зовнішніх і внутрішніх файлів (ILF і EIF) складність визначається і ранжується за допомогою кількості типів елементів записів (RET - Record Element

Types) і кількості типів елементів даних (DET - Data Element Types), що входять до відповідних логічних груп даних. При цьому під кількістю RET розуміється кількість різних форматів записів, що використовуються в даному файлі, а під кількістю DET - кількість різних полів в записах.

Рівні складності для внутрішніх і зовнішніх файлів (ILF і EIF) в залежності від кількості RET і DET наведені в табл. 3.1.

Таблиця 3.1 Рівні складності для внутрішніх і зовнішніх файлів (ILF і EIF)

| Record Element Types (RET) | Data Elements | | |
|-------------------------------|---------------|--------------|--------------|
| | 1 to 19 | 20 - 50 | 51 or More |
| 1 RET | Low (7) | Low(7) | Average (10) |
| 2 to 5 RET | Low (7) | Average (10) | High (15) |
| 6 or More RET | Average (10) | High (15) | High (15) |

Для транзакційних функцій (EI, EO і EQ) складність визначається і ранжується за допомогою кількості типів використовуваних файлів (FTR - File Types Referenced), тобто кількості ILF і EIF, що беруть участь в транзакційному процесі, а також кількості типів елементів даних DET (відмінних один від одного полів записів), що додаються, модифікуються, стертих або створюваних в вихідних даних.

Таблиця 3.2 Рівні складності для зовнішніх входів (EI)

| | | | |
|----------------|----------|----------|-----------------|
| | 1-4 DET | 5-15 DET | 16 і більше DET |
| 0-1 FTR | Низкий | Низкий | Середній |
| 2 FTR | Низкий | Середній | Високий |
| 3 і більше FTR | Середній | Високий | Високий |

Рівні складності для зовнішніх виходів (EO) в залежності від числа FTR і DET наведені в табл. 3.3.

Таблиця 3.3 Рівні складності для зовнішніх виходів (EO)

| | 1-5 DET | 6-19 DET | 20 і більше DET |
|----------------|----------|----------|-----------------|
| 0-1 FTR | Низкий | Низкий | Середній |
| 2-3 FTR | Низкий | Середній | Високий |
| 4 і більше FTR | Середній | Високий | Високий |

Рівні складності для зовнішніх запитів (EQ) визначаються з використанням наступного алгоритму:

- 1) вхід зовнішнього запиту розглядається аналогічно зовнішньому входу (EI) див. табл. 2
- 2) вихід зовнішнього запиту розглядається аналогічно зовнішнього виходу (EO) див. табл. 3
- 3) в якості результату використовується найбільше значення з отриманих 1 і 2 рівнів складності.

Вагові коефіцієнти рівнів складності для різних характеристик функціональності ПЗ наведені в табл. 3.4.

Таблиця 3.4. Вагові коефіцієнти рівнів складності

| | Низький | Середній | Високий |
|-----|---------|----------|---------|
| ILF | 7 | 10 | 15 |
| EIF | 5 | 7 | 10 |
| EI | 3 | 4 | 6 |
| EO | 4 | 5 | 7 |
| EQ | 3 | 4 | 6 |

5) Загальна формула для обчислення ненормованої кількості функціональних точок (UFPC - Unadjusted Function Point Count):

$$UFPC = (3 * LEI) + (4 * AEI) + (6 * HEI) + (4 * LEO) + (5 * AEO) + (7 * HEO) + (7 * LILF) + (10 * AILF) + (15 * HILF) + (5 * LEIF) + (7 * AEIF) + (10 * HEIF) + (3 * LEQ) + (4 * AEQ) + (6 * HEQ)$$

де UFPC - ненормована кількість функціональних точок;

LEI - кількість зовнішніх входів низького рівня складності;

AEI - кількість зовнішніх входів середнього рівня складності,

HEI - кількість зовнішніх входів високого рівня складності;

LEO - кількість зовнішніх виходів низького рівня складності;

AEO - кількість зовнішніх виходів середнього рівня складності;

HEO - кількість зовнішніх виходів високого рівня складності;
LILF - кількість внутрішніх логічних файлів низького рівня складності;
AILF - кількість внутрішніх логічних файлів середнього рівня складності;
HILF - кількість внутрішніх логічних файлів високого рівня складності;
LEIF - кількість зовнішніх інтерфейсних файлів низького рівня складності;
AEIF - кількість зовнішніх інтерфейсних файлів середнього рівня складності;
HEIF - кількість зовнішніх інтерфейсних файлів високого рівня складності;
LEQ - кількість зовнішніх запитів низького рівня складності;
AEQ - кількість зовнішніх запитів середнього рівня складності;
HEQ - кількість зовнішніх запитів високого рівня складності.

6) Для отримання остаточного результату аналізу, тобто нормованого кількості функціональних точок (AFPC - Adjusted Function Point Count), необхідно також врахувати ряд загальних вимог до проекту, для чого отримана ненормована кількість функціональних точок множиться на спеціальним чином розрахований нормуючий фактор (VAF - Value Adjustment Factor).

У методі функціональних точок **нормуючий фактор** (VAF) визначається шляхом аналізу 14 основних характеристик програмних продуктів (GSC - General System Characteristics), метою яких і є облік загальних вимог до програмних продуктів. Всі передбачені методом функціональних точок основні характеристики системи (GSC).

Основні характеристики системи (GSC), кожна з наведених характеристик системи оцінюється експертним способом, числом від 0 (якщо вона не присутня або не має значення для даного ПЗ) до 5 (якщо вона має дуже сильний вплив на дане ПЗ):

1. *Обмін даними* (0 - продукт являє собою автономне додаток; 5 - продукт обмінюється даними по більш, ніж одному телекомунікаційному протоколу).

2. *Розподілена обробка даних* (0- продукт не переміщує дані; 5 - розподілена обробка даних виконується декількома компонентами системи).

3. *Продуктивність* (0- призначені для користувача вимоги по продуктивності не встановлені; 5 - час відгуку сильно обмежена критично для всіх бізнес-операцій, для задоволення вимог необхідні спеціальні проектні рішення і інструменти аналізу).

4. *Обмеження по апаратних ресурсів* (0- немає обмежень; 5 - продукт цілком повинен функціонувати на певному процесорі і не може бути розподілений).

5. *Транзакційна навантаження* (0- транзакцій не багато, без піків; 5 - число транзакцій велике і нерівномірно, потрібні спеціальні рішення і інструменти).

6. *Інтенсивність взаємодії з користувачем* (0- всі транзакції обробляються в пакетному режимі; 5 - більше 30% транзакцій - інтерактивні).

7. *Ергономіка*(Ефективність роботи кінцевих користувачів) (0 - немає спеціальних вимог; 5 - вимоги щодо ефективності дуже жорсткі).

8. *Інтенсивність зміни даних*(ILF) користувачами (0 - не потрібні; 5 - зміни інтенсивні, жорсткі вимоги по відновленню).

9. *Складність обробки* (0- обробка мінімальна; 5 - вимоги безпеки, логічна і математична складність, багатопоточність).

10. *Повторне використання*(0 - не вимагається; 5 - продукт розробляється як стандартний багаторазовий компонент).

11. *Зручність інсталяції* (0- немає вимог; 5 - установка і оновлення ПЗ проводиться автоматично).

12. *зручність адміністрування*(0 - не вимагається; 5 - система автоматично самовідновлюється).

13. *портіруемість*(0 - продукт має тільки 1 інсталяцію на єдиному процесорі; 5 - система є розподіленою і передбачає установку на різні «залізо» і ОС).

14. *Гнучкість* (0- не вимагається; 5 - гнучка система запитів і побудова довільних звітів, модель даних змінюється користувачем в інтерактивному режимі).

Значення всіх 14 характеристик підсумовуються для отримання підсумкового ступеня впливу (TDI - Total Degree of Influence):

$$TDI = \sum DI$$

Нормуючий фактор (VAF) розраховується за формулою:

$$VAF = (TDI * 0.01) + 0.65$$

7 Нормована кількість функціональних точок представляє собою добуток ненормований кількості функціональних точок на нормуючий фактор:

$$AFPC = UFPC \cdot VAF$$

Надалі нормовану кількість функціональних точок може бути використано для отримання оцінки кількості рядків вихідного коду (SLOC - Source Lines of Code).

Статистичними методами визначено значення «мовних множників – рівня мови» для основних мов програмування. Таким чином, якщо мова реалізації обрана, то можна оцінити кількість рядків вихідного коду, шляхом множення нормованої кількості функціональних точок на відповідний множник:

$$SLOC = AFPC \cdot LM$$

де LM -множник мови програмування (додаток 2).

Початкова оцінка кількості вирівняних функціональних точок враховує лише нову функціональність, що реалізується у продукті.

Проект розробки продукту оцінюється в DFP (development functional point) за формулою:

$$DFP = (UFP + CFP) * VAF,$$

де CFP (conversion functional point) — функціональні точки, підраховані для додаткової функціональності, яка потрібна для встановлення продукту, наприклад, міграції даних.

Проект доопрацювання оцінюється в EFP (enhancement functional point) за формулою:

$$EFP = [(ADD + CHGA + CFP) * VAFA] + (DEL * VAFB),$$

де

ADD — це невідкоригована кількість функціональних точок тих функцій, які були додані в рамках проекту вдосконалення;

CHGA — це невідкоригована кількість функціональних точок тих функцій, які були змінені в рамках проекту вдосконалення (відображає функції після модифікацій);

CFP — кількість функціональних точок, додана в результаті перетворення;

VAFA — величина фактора вирівнювання програми після завершення проекту;

DEL — невідкоригована кількість функціональних точок тих функцій, які були видалені проектом вдосконалення. Важливо враховувати абсолютне значення DEL, а не від'ємне значення.

VAFB — величина фактора вирівнювання розрахункового до початку проекту

На практиці часто буває що $VAFA = VAFB = VAF$, таким чином, рівняння стає таким:

$$EFP = (ADD + CHGA + CFP + DEL) * VAF$$

Проект після вдосконалення програми

$$AFP = [(UFPB + ADD + CHGA) - (CHGB + DEL)] * VAFA$$

де:

UFPB = Нескоригована кількість функціональних балів перед покращенням.

AFP = підрахунок точки функції програми

DEL = кількість видалених функціональних точок (від'ємне значення).

Усі інші аббревіатури такі ж, як і раніше.

Звичайно, розрахунок покращення може додавати та/або видаляти функції з UFPB. Додана функціональність може бути за рахунок нових компонентів або додана функціональність може бути за рахунок збільшення розміру існуючих компонентів.

Завдання

1. Розробити модель для аналізу функціональних точок з переліком і докладним описом всіх внутрішніх логічних (ILF) і зовнішніх інтерфейсних (EIF) файлів, а також всіх транзакцій (EI, EO, EQ).
2. Виконати оцінки кількості RET і DET для внутрішніх логічних (ILF) і зовнішніх інтерфейсних (EIF) файлів, а також оцінки кількості FTR і DET для зовнішніх введень (EI), виведень (EO) і запитів (EQ).
3. Провести аналіз ступеня впливу основних характеристик системи.
4. Виконати розрахунки по моделі функціональних точок.
5. **Всі розрахунки** внести в робочий лист, що наведений в додатку 1.
6. Захистити практичну роботу.

****Розрахунки мають бути проведені з використанням власного програмного продукту.***

Структура звіту з практичної роботи

Звіт про виконання практичної роботи №3 має містити наступні складові елементи:

1. Титульний аркуш.
2. Опис програми для якої будуть виконуватися розрахунки, посилання на програму або скріншоти по яким надалі будуть виконуватися розрахунки.
3. Отримані результати у вигляді робочого листа що наведений в додатку 1.
4. Висновки по практичній роботі.

Контрольні питання

1. Які метрики зазвичай використовують для оцінки розміру?
2. Які метрики розміру називають внутрішніми, а які зовнішніми і чому?
3. Які переваги є у функціональних точок у порівнянні з рядками вихідного коду при оцінці розміру на ранніх стадіях проекту?
4. Що розуміється під функціональністю даних?
5. Що розуміється під функціональністю транзакцій?
6. Що таке внутрішній логічний файл (ILF)?
7. Що таке зовнішній інтерфейсний файл (EIF)?
8. У чому основна відмінність між внутрішнім логічним файлом (ILF) і зовнішнім інтерфейсним файлом (EIF)?
9. Що таке зовнішнє введення (EI)?
10. Що таке зовнішній виведення (EO)?
11. Що таке зовнішній запит (EQ)?
12. У чому основна відмінність між зовнішнім введенням (EI) і зовнішнім запитом (EQ)?
13. У чому основна відмінність між зовнішнім виводом (EO) і зовнішнім запитом (EQ)?
14. Як оцінюються рівні складності внутрішніх логічних (ILF) і зовнішніх інтерфейсних файлів (EIF)?

15. Як оцінюються рівні складності зовнішніх введів (EI), виведень (EO) і запитів (EQ)?

16. Яким чином проводиться аналіз ступенів впливу основних характеристик системи і обчислюється нормируючий фактор VAF?

20. У чому принципова відмінність ненормованого кількості функціональних точок від нормованого? Як сильно вони можуть відрізнятися?

21. З яких основних кроків (етапів) складається процес аналізу функціональних точок?

22. Що служить вихідною інформацією для процесу аналізу функціональних точок?

Робочий лист аналізу за методом функціональних точок

| | | | | |
|--|---|--|----------|---------------------|
| Визначення типу оцінки | | | | |
| Визначення границі продукту | <i>*Повний опис винести додатково скріншоти програми або посилання на програму*</i> | | | |
| Підрахунок кількості функцій в кожній категорії | | | | |
| | Застосування вагових коефіцієнтів складності | | | |
| | Простий | Середній | Складний | Функціональні точки |
| Кількість введів | _*3 | _*4 | _*6 | |
| Кількість виводів | _*4 | _*5 | _*7 | |
| Кількість запитів | _*3 | _*4 | _*6 | |
| Кількість файлів | _*7 | _*10 | _*15 | |
| Кількість інтерфейсів | _*5 | _*7 | _*10 | |
| Обчислення ненормованої кількості функціональних точок UFPC | | | | |
| Основні характеристики системи (GSC) | | | | |
| Фактори середовища | | Рейтинг (0,1,2,3,4,5) *з поясненням вибору* | | |
| Обмін даними | | | | |
| Розподілена обробка даних | | | | |
| Вимоги до продуктивності | | | | |
| Обмеження по апаратним ресурсам | | | | |
| Транзакційне навантаження | | | | |
| Інтенсивність взаємодії з користувачем | | | | |
| Ергономіка(Ефективність роботи кінцевих користувачів) | | | | |
| Інтенсивність зміни даних(ILF) користувачами | | | | |
| Складність обробки | | | | |
| Повторне використання | | | | |
| Зручність інсталяції | | | | |
| Зручність адміністрування | | | | |
| Використання декількох вузлів (портів) | | | | |
| Гнучкість | | | | |
| Обчислення TDI | | | | |

| Загальні розрахунки по проекту | |
|--|--|
| Обчислення нормуючого фактора VAF | |
| Обчислення нормованої кількості функціональних точок AFPS | |
| Обчислення оцінки кількості рядків вихідного коду | |
| Обчислення DFP | |
| Обчислення EFP | |
| Обчислення AFP | |

| QSM SLOC/FP Data | | | | |
|-------------------------------|-----|--------|-----|------|
| Language | Avg | Median | Low | High |
| ABAP (SAP) * | 28 | 18 | 16 | 60 |
| ASP* | 51 | 54 | 15 | 69 |
| Assembler * | 119 | 98 | 25 | 320 |
| Brio + | 14 | 14 | 13 | 16 |
| C * | 97 | 99 | 39 | 333 |
| C++ * | 50 | 53 | 25 | 80 |
| C# * | 54 | 59 | 29 | 70 |
| COBOL * | 61 | 55 | 23 | 297 |
| Cognos Impromptu Scripts + | 47 | 42 | 30 | 100 |
| Cross System Products (CSP) + | 20 | 18 | 10 | 38 |
| Cool:Gen/IEF * | 32 | 24 | 10 | 82 |
| Datastage | 71 | 65 | 31 | 157 |
| Excel * | 209 | 191 | 131 | 315 |
| Focus * | 43 | 45 | 45 | 45 |
| FoxPro | 36 | 35 | 34 | 38 |
| HTML * | 34 | 40 | 14 | 48 |
| J2EE * | 46 | 49 | 15 | 67 |
| Java * | 53 | 53 | 14 | 134 |
| JavaScript * | 47 | 53 | 31 | 63 |
| JCL * | 62 | 48 | 25 | 221 |
| LINC II | 29 | 30 | 22 | 38 |
| Lotus Notes * | 23 | 21 | 19 | 40 |
| Natural * | 40 | 34 | 34 | 53 |
| .NET * | 57 | 60 | 53 | 60 |
| Oracle * | 37 | 40 | 17 | 60 |
| PACBASE * | 35 | 32 | 22 | 60 |
| Perl * | 24 | 15 | 15 | 60 |
| PL/I * | 64 | 80 | 16 | 80 |
| PL/SQL * | 37 | 35 | 13 | 60 |
| Powerbuilder * | 26 | 28 | 7 | 40 |
| REXX * | 77 | 80 | 50 | 80 |
| Sabretalk * | 70 | 66 | 45 | 109 |
| SAS * | 38 | 37 | 22 | 55 |
| Siebel * | 59 | 60 | 51 | 60 |

| | | | | |
|----------------|----|----|----|----|
| SLOGAN * | 75 | 75 | 74 | 75 |
| SQL * | 21 | 21 | 13 | 37 |
| VB.NET * | 52 | 60 | 26 | 60 |
| Visual Basic * | 42 | 44 | 20 | 60 |