

Оценка качества сложной программной системы

(продолжение)

Оценка качества

Метрики Лоренца и Кидда

Подразделяются на группы:

- метрики размера, основанные на подсчёте свойств и операций классов, средних значений объектно-ориентированной программы;
- метрики наследования, учитывающие способы повторного использования операций в иерархии классов;

Оценка качества

- внутренние метрики, отвечающие на вопросы связности и кодирования;
- внешние метрики, изучающие сцепление и повторное использование

Метрики Лоренца и Кидда включают десять метрик:

- 1 Размер класса CS (Class Size) – общий размер класса определяется на основании определения следующих показателей:
 - общее количество операций;
 - количество свойств.

Оценка качества

Указанные измерения в обоих случаях следует проводить с учетом приватных и наследуемых экземплярных операций, которые инкапсулируются внутри класса:

$$CS = C_{\Sigma} + S_{\Sigma} , \quad (2.22)$$

где C_{Σ} – количество инкапсулированных классом методов (операций);

S_{Σ} – количество инкапсулированных классом свойств.

Оценка качества

- 2 Количество операций, переопределяемых подклассом NOO (Number of Operations Overridden by a Subclass). Переопределением называют случай, когда подкласс замещает операцию, унаследованную от суперкласса, своей собственной версией. Большие значения NOO указывают на возникшие проблемы проектирования.
- 3 Количество операций, добавленных подклассом NOA (Number of Operations Added by a Subclass), определяется количеством добавленных относительно родительского класса собственных методов (операций):

Оценка качества

$$NOA = N_{\Sigma} , \quad (2.23)$$

где N_{Σ} – количество новых методов класса, добавленных относительно суперкласса.

С увеличением NOA подкласс приобретает меньшую общность со своим суперклассом, что требует больших трудозатрат по тестированию и внесению изменений. При увеличении высоты дерева иерархии классов (с ростом значения DIT) должно уменьшаться количество новых методов классов нижних уровней. Для рекомендуемых граничных значений размера класса ($CS = 20$) и высоты дерева иерархии классов ($DIT = 6$) значение NOA ограничено значением 4 ($NOA < 4$).

Оценка качества

- 4 Индекс специализации SI (Specialization Index) характеризует грубую оценку степени специализации каждого подкласса при добавлении, удалении или переопределении операций. Индекс специализации определяется по формуле

$$SI = \frac{NOO * u}{M_{\text{общ}}} , \quad (2.24)$$

где NOO – количество операций, переопределяемых подклассом;

u – номер уровня в иерархии, на котором находится подкласс;

$M_{\text{общ}}$ – общее количество методов класса.

Оценка качества

Чем выше значение метрики SI, тем выше вероятность того, что в иерархии классов есть отдельные экземпляры, нарушающие абстракцию суперкласса. Рекомендуемое значение показателя SI ограничено сверху величиной 0,15, т. е. $SI < 0,15$.

- 5 Средний размер операции AOS (Average Operation Size) определяется количеством сообщений, порождаемых операцией. В качестве оценки размера может использоваться количество строк программы, однако LOC-оценки приводят к известным проблемам. Иным вариантом может являться количество сообщений, посланных операцией. Рост значения данного показателя означает, что обязанности размещены в классе не очень удачно.

Оценка качества

Рекомендуемое значение метрики AOS не должно превышать 9. Увеличение среднего размера относительно этой границы рассматривают как показатель неудачного проектирования обязанностей класса.

- 6 Сложность операции ОС (Operation Complexity) может быть вычислена на основе стандартных метрик сложности (например, с помощью LOC- или FP-оценок, метрики цикломатической сложности, метрики Холстеда). Лоренц и Кидд предложили вычислять значение ОС суммированием оценок с весовыми коэффициентами, приведенными в таблице 2.1.

Оценка качества

Таблица 2.1 – Весовые коэффициенты

Действие	Вес
Определение переменной-параметра	0,3
Определение временной переменной	0,5
Присваивание значения	0,5
Вложенное выражение	0,5
Сообщение без параметра	1
Арифметическая операция	2
Сообщение с параметрами	3
Вызов стандартной функции интерфейса	5
Вызов пользовательской функции	7

Рекомендуемое значение метрики ограничено числом 65 ($OC < 65$).

Оценка качества

- 7 Среднее количество параметров на операцию ANP (Average Number of Parameters per operation) определяется отношением числа параметров к количеству операций (методов) класса. Чем больше параметров у операции, тем сложнее взаимодействие между объектами. Поэтому значение метрики ANP должно быть как можно меньшим. Рекомендуемое значение $ANP = 0,7$.
- 8 Количество описаний сценариев NSS (Number of Scenario Scripts) измеряется или количеством классов, реализующих требования к программному обеспечению, или количеством состояний для каждого класса, или количеством методов класса.

Оценка качества

При своем не совсем обычном способе измерения метрика NSS является достаточно эффективным индикатором размера создаваемой программы. Рекомендуется не менее одного сценария. Рост количества сценариев неминуемо ведет к увеличению размера программы.

- 9 Количество ключевых классов NКС (Number of Key Classes) характеризует объем работы по программированию. Рекомендуется ограничивать значения метрики снизу значением 0,2. Если значение метрики $NКС < 0,2$ от общего количества классов системы, то необходимо пересмотреть выделение классов.

Оценка качества

10 Количество подсистем NSUB (Number of subsystem) определяется непосредственным подсчетом. Количество подсистем обеспечивает понимание таких вопросов, как размещение ресурсов, планирование, общие затраты на интеграцию. Рекомендуется выделять в программном комплексе не менее трёх подсистем.

Метрики Абреу

Набор метрик Абреу (MOOD, Metrics for Object Oriented Design) включает в себя показатели качества программных средств:

Оценка качества

- фактор закрытости метода (MHF);
- фактор закрытости свойства (AHF);
- фактор наследования метода (MIF);
- фактор наследования свойства (AIF)
- фактор полиморфизма (POF);
- фактор сцепления (COF).

Метрики MHF и AHF относятся к первому принципу объектно-ориентированного программирования – инкапсуляции. Метрики MIF и AIF относятся ко второму принципу объектно-ориентированного программирования – наследованию.

Оценка качества

Метрики POF относятся к третьему принципу объектно-ориентированного программирования – полиморфизму. Метрики COF относятся к отсылке сообщений.

Показатели качества метрик Абреу:

- 1 Фактор закрытости метода MNF (Method Hiding Factor) – показатель, характеризующий процентное количество классов, из которых метод невидим. Значение показателя MNF определяется по формуле

Оценка качества

ТС

$$\sum_{i=1} M_h (C_i)$$

$i=1$

$$MHF = \frac{\sum_{i=1}^{ТС} M_h (C_i)}{\sum_{i=1}^{ТС} M_d (C_i)}, \quad (2.25)$$

ТС

$$\sum_{i=1} M_d (C_i)$$

$i=1$

где $M_h (C_i)$ – реализация класса, характеризующая количество скрытых методов в классе;

$M_d (C_i)$ – общее количество методов, определённых в классе (не учитываются наследованные методы);

C_i – класс с номером i , $i = 1, \dots, ТС$;

Оценка качества

ТС – количество классов в создаваемой программной системе.

Общее количество методов в классе определяется по формуле

$$M_d(C_i) = M_v(C_i) + M_h(C_i), \quad (2.26)$$

где $M_v(C_i)$ – интерфейс класса, определяющий количество видимых методов в классе.

Видоизменённая формула расчёта показателя MNF имеет вид:

Оценка качества

$$\sum_{i=1}^{TC} \sum_{m=1}^{M_d(C_i)} (1 - V(M_{mi}))$$

$$MHF = \frac{\sum_{i=1}^{TC} M_d(C_i)}{\sum_{i=1}^{TC} \sum_{m=1}^{M_d(C_i)} (1 - V(M_{mi}))}, \quad (2.27)$$

$$\sum_{i=1}^{TC} M_d(C_i)$$

где $V(M_{mi})$ – процентное количество классов, которые способны видеть m -й метод i -го класса, $i = 1, \dots, TC$;

C_i – класс с номером i , $i = 1, \dots, TC$;

Оценка качества

$M_d(C_i)$ – общее количество методов, определённых в классе (не учитываются наследованные методы);

ТС – количество классов в создаваемой программной системе.

Процентное количество классов, которые способны видеть m -й метод i -го класса определяется по формуле

$$\sum_{i=1}^{ТС} \mathbf{is_visible}(M_{mi}, C_j)$$

$$V(M_{mi}) = \frac{\sum_{i=1}^{ТС} \mathbf{is_visible}(M_{mi}, C_j)}{ТС - 1}, \quad (2.28)$$

Оценка качества

где **is_visible**(M_{mi} , C_j) – уровень видимости m -го метода i -го класса из j -го класса, $i = 1, \dots, TC$.

Уровень видимости m -го метода i -го класса из j -го класса определяется по формуле

$$\mathbf{is_visible}(M_{mi}, C_j) = \quad (2.29)$$

1, если $j \neq 1$ и C_j может вызвать M_{mi}

0, в остальных случаях

где M_{mi} – m -е свойство i -го класса;

C_j – класс с номером j .

С увеличением значения метрики МНФ уменьшается плотность дефектов в системе, что означает сокращение затрат для устранения ошибок.

Оценка качества

2 Фактор закрытости свойства AHF (Attribute Hiding Factor) – показатель, представляющий процентное количество классов, из которых данное свойство невидимо. Значение показателя AHF определяется по формуле

$$AHF = \frac{\sum_{i=1}^{TC} A_h(C_i)}{\sum_{i=1}^{TC} A_d(C_i)}, \quad (2.30)$$

Оценка качества

где $A_h(C_i)$ – реализация класса, характеризующая количество скрытых свойств в классе;

$A_d(C_i)$ – общее количество свойств, определённых в классе (не учитываются наследованные свойства);

C_i – класс с номером i , $i = 1, \dots, TC$;

TC – количество классов в создаваемой программной системе.

Общее количество свойств, определённых в классе C_i определяется по формуле

$$A_d(C_i) = A_v(C_i) + A_h(C_i), \quad (2.31)$$

где $A_v(C_i)$ – интерфейс класса, определяющий количество видимых свойств в классе C_i , $i = 1, \dots, TC$.

Оценка качества

Видоизменённая формула расчёта показателя АНФ имеет вид:

$$\sum_{i=1}^{TC} A_d(C_i) \sum_{m=1} (1 - V(A_{mi}))$$

$$АНФ = \frac{\sum_{i=1}^{TC} A_d(C_i) \sum_{m=1} (1 - V(A_{mi}))}{\sum_{i=1}^{TC} A_d(C_i)}, \quad (2.32)$$

$$\sum_{i=1}^{TC} A_d(C_i)$$

где $V(A_{mi})$ – процентное количество классов, которые видят m -е свойство i -го класса из j -го класса, $i = 1, \dots, TC$;

Оценка качества

$A_d(C_i)$ – общее количество свойств, определённых в классе (не учитываются наследованные свойства);

C_i – класс с номером i , $i = 1, \dots, TC$;

TC – количество классов в создаваемой программной системе.

Процентное количество классов, которые видят m -е свойство i -го класса определяется по формуле

$$V(A_{mi}) = \frac{\sum_{i=1}^{TC} \text{is_visible}(A_{mi}, C_j)}{TC - 1}, \quad (2.33)$$

Оценка качества

где **is_visible**(A_{mi} , C_j) – уровень видимости m -го свойства i -го класса из j -го класса, $i = 1, \dots, TC$.

Уровень видимости m -го свойства i -го класса из j -го класса определяется по формуле

$$\text{is_visible}(A_{mi}, C_j) = \begin{cases} 1, & \text{если } j \neq 1 \text{ и } C_j \text{ может вызвать } A_{mi} \\ 0, & \text{в остальных случаях} \end{cases} \quad (2.34)$$

где A_{mi} – m -е свойство i -го класса;

C_j – класс с номером j .

В числителе формулы 2.32 расположена сумма закрытости всех свойств во всех классах, а в знаменателе – общее количество свойств, определённых в системе.

Оценка качества

Если все свойства скрыты и доступны для методов класса, то значение фактора закрытости свойств ANF равно 100%.

3 Фактор наследования метода MIF (Method Inheritance Factor) – показатель, характеризующий процентное количество классов, из которых метод наследован.

Значение фактора наследования метода определяется по формуле

Оценка качества

$$\sum_{i=1}^{TC} M_i (C_i)$$

$$MIF = \frac{\sum_{i=1}^{TC} M_i (C_i)}{\sum_{i=1}^{TC} M_a (C_i)}, \quad (2.35)$$

$$\sum_{i=1}^{TC} M_a (C_i)$$

где $M_i (C_i)$ – количество наследованных и непереопределённых методов в классе C_i ;

$M_a (C_i)$ – общее количество методов, доступное в классе C_i ;

C_i – класс с номером i , $i = 1, \dots, TC$;

Оценка качества

ТС – количество классов в создаваемой программной системе.

Общее количество методов, доступных в классе C_i , определяется по формуле

$$M_a(C_i) = M_d(C_i) + M_i(C_i), \quad (2.36)$$

где $M_i(C_i)$ – количество наследованных и непереопределённых методов в классе C_i ;

$M_d(C_i)$ – количество методов, определённых в классе C_i .

Оценка качества

Количество методов, определённых в классе C_i , определяется по формуле

$$M_d(C_i) = M_n(C_i) + M_o(C_i), \quad (2.37)$$

где $M_n(C_i)$ – количество новых (ненаследованных и непереопределённых) методов в классе C_i ;

$M_o(C_i)$ – количество наследованных и переопределённых методов в классе C_i .

Числитель в формуле 2.35 представляет сумму наследованных и непереопределённых методов во всех классах программного приложения.

Оценка качества

Знаменатель – общее количество доступных методов (локально определённых и наследованных) для всех классов. Если значение фактора наследования метода MIF равно нулю, то в анализируемой программной системе отсутствует эффективное наследование. При увеличении значения фактора наследования метода MIF уменьшается плотность дефектов и сокращаются затраты на исправление ошибок.

- 4 Фактор наследования свойства AIF (Attribute Inheritance Factor) показатель, характеризующий процентное количество классов, из которых свойство наследовано. Значение показателя AIF определяется по формуле

Оценка качества

$$\sum_{i=1}^{TC} A_i (C_i)$$

$$AIF = \frac{\sum_{i=1}^{TC} A_i (C_i)}{\sum_{i=1}^{TC} A_a (C_i)}, \quad (2.38)$$

$$\sum_{i=1}^{TC} A_a (C_i)$$

где $A_i (C_i)$ – количество наследованных и непереопределённых свойств в классе C_i ;

$A_a (C_i)$ – общее количество свойств, доступных в классе C_i ;

C_i – класс с номером i , $i = 1, \dots, TC$;

Оценка качества

ТС – количество классов в программной системе.

Общее количество свойств, доступных в классе C_i , определяется по формуле

$$A_a(C_i) = A_d(C_i) + A_i(C_i), \quad (2.39)$$

где $A_d(C_i)$ – количество свойств, определённых в классе C_i , $i = 1, \dots, TC$;

$A_i(C_i)$ – количество наследованных и непереоопределённых свойств в классе C_i .

Оценка качества

Количество свойств, определённых в классе C_i , рассчитывается по формуле

$$A_d(C_i) = A_n(C_i) + A_o(C_i), \quad (2.40)$$

где $A_n(C_i)$ – количество новых (ненаследованных и переопределённых) свойств в классе C_i , $i = 1, \dots, TC$;

$A_o(C_i)$ – количество наследованных и переопределённых свойств в классе C_i .

Числитель в формуле 2.38 представляет сумму наследованных и непереопределённых свойств во всех классах программной системы.

Оценка качества

Знаменатель – общее количество доступных свойств (локально определённых и наследованных) для всех классов.

5 Фактор полиморфизма POF (Polymorphism Factor) – показатель, представляющий отношение реального количества возможных полиморфных ситуаций к максимальному количеству возможных полиморфных ситуаций для класса.

Значение фактора полиморфизма POF рассчитывается по формуле

Оценка качества

$$\sum_{i=1}^{TC} M_0 (C_i)$$

$$POF = \frac{\sum_{i=1}^{TC} M_0 (C_i)}{\sum_{i=1}^{TC} [M_n (C_i) * DC(C_i)]}, \quad (2.41)$$

$$\sum_{i=1}^{TC} [M_n (C_i) * DC(C_i)]$$

где $M_0 (C_i)$ – количество наследованных и переопределённых методов в классе C_i ;

$DC(C_i)$ – количество потомков класса C_i ;

$M_n (C_i)$ – количество новых методов, определённых в классе C_i .

Оценка качества

Количество методов, определённых в классе C_i , определяется по формуле

$$M_d(C_i) = M_n(C_i) + M_o(C_i), \quad (2.42)$$

где $M_n(C_i)$ – количество новых (ненаследованных и переопределённых) методов в классе C_i ;

$M_o(C_i)$ – количество наследованных и переопределённых методов в классе C_i , $i = 1, \dots, TC$;

TC – количество классов в программной системе.

Использование полиморфизма уменьшает плотность дефектов и затраты на доработку программных средств.

Оценка качества

Если значение фактора полиморфизма $POF > 10\%$, то количество ошибок в программе и затраты могут увеличиться.

6 Фактор сцепления COF (Coupling Factor) – показатель, характеризующий наличие отношения между классами: класс-клиент содержит по меньшей мере одну ненаследованную ссылку на свойство или метод класса-поставщика.

Значение фактора сцепления COF определяется по формуле

Оценка качества

$$\sum_{i=1}^{TC} \left[\sum_{j=1}^{TC} \text{is_client} (C_i, C_j) \right]$$

$$\text{COF} = \frac{\sum_{i=1}^{TC} \left[\sum_{j=1}^{TC} \text{is_client} (C_i, C_j) \right]}{TC^2 - TC}, \quad (2.43)$$

где **is_client** (C_i, C_j) – бинарная переменная, указывающая на наличие отношения «клиент-поставщик» и принимающая значение 1 или 0;
 TC – количество классов в программной системе.

Бинарная переменная, указывающая на наличие отношения «клиент-поставщик», определяется по формуле

Оценка качества

$$\text{is_client} (C_c, C_s) = \quad (2.44)$$

1, если $C_c \rightarrow C_s \cap C_c \neq C_s$

0, в остальных случаях

где $C_c \rightarrow C_s$ – отношение «клиент-поставщик», означающее, что класс-клиент содержит по меньшей мере одну ненаследованную ссылку на свойство или метод класса-поставщика.

Числитель формулы расчёта COF содержит реальное количество сцеплений, не относящихся к наследованию. Знаменатель формулы расчёта COF соответствует максимально возможному количеству сцеплений в программной системе с классами.

Оценка качества

С увеличением количества сцеплений плотность дефектов и уровень затрат на доработку возрастают. Сцепления отрицательно влияют на качество программного средства и увеличивают сложность программного обеспечения, уменьшают инкапсуляцию и возможность повторного использования программных средств, затрудняет понимание и усложняет сопровождение программ.