Тестирование программных средств

Сафронов Сергей,
2008 год

Оглавление

- Классификация метрик сложности
- Статические метрики сложности
- Цикломатическая сложность
- Анализ тестового покрытия
  - Различные методы оценки тестового покрытия, их сравнение

Классификация метрик качества

Две группы признаков:

1. метрики производительности, качества продукции и технические характеристики продукта
   - Производительность – результирующий продукт
   - Качество - соответствие явным и подразумеваемым требованиям пользователя, т.е. пригодность изделия к использованию
   - Технические метрики в большей степени относятся к особенностям программного изделия, а не к процессу его разработки (например, логическая сложность изделия, модульность проекта и т.п.)

Классификация метрик качества

2. группы метрик по их ориентации
   - Размеро-ориентированные метрики
     - сбор результатов прямых измерений программного продукта и его качества, а так же процесса разработки
   - Функционально-ориентированные метрики
     - косвенно характеризуют функциональное назначение продукта и особенности его входных и выходных данных
   - Человеко-ориентированные метрики
     - косвенно позволяют судить о том, как персонал (разработчики и пользователи) оценивают эффективность и качество работы программного продукта, удобство взаимодействия с ним, простоту обучения и т.д.

Статические метрики сложности

Объектно-ориентированные показатели:

- число классов;
- максимальная глубина класса в дереве наследования;
- число классов, связанных с данным классом (связь между классами объектов);
- число непосредственных подклассов (число дочерних классов);
- число методов экземпляра;
- число локальных методов (взвешенные методы на класс);
- число методов, включая унаследованные;
- число локальных дружественных методов;
- число локальных общих методов;
- число локальных частных методов;
- число переменных экземпляра;
- число используемых входных данных (параметров, глобальных переменных);
- число непосредственных базовых классов;
- число используемых выходных данных (параметров, глобальных переменных);
- недостаток связности - 100% минус средняя связность для компонентов
- данных класса.

Статические метрики сложности

Показатели сложности:

- цикломатическая сложность;
- «модифицированная» цикломатическая сложность;
- «строгая» цикломатическая сложность;
- средняя цикломатическая сложность для всех вложенных функций или методов;
- средняя «модифицированная» цикломатическая сложность для всех вложенных функций или методов;
- средняя «строгая» цикломатическая сложность для всех вложенных функций или методов;
- максимальная цикломатическая сложность для всех вложенных функций или методов;
- максимальная «модифицированная» цикломатическая сложность для всех вложенных функций или методов;
- максимальная «строгая» цикломатическая сложность для всех вложенных функций или методов;
- максимальный уровень вложенности управляющих конструкций.

Статические метрики сложности

Показатели размера/объема

- число файлов;
- число функций;
- число операторов;
- число декларативных операторов;
- число выполняемых операторов;
- полное число строк;
- число пустых строк;
- число строк, содержащих исходный код;
- число строк, содержащих декларативный исходный код;
- число строк, содержащих выполняемый исходный код;
- число строк, содержащих комментарии;
- число строк комментариев, деленное на число строк кода, умноженное на 100 (процент комментариев);
- среднее число строк для всех вложенных функций или методов;
- среднее число пустых строк для всех вложенных функций или методов;
- среднее число строк, содержащих исходный код, для всех вложенных функций или методов;
- среднее число строк, содержащих комментарии, для всех вложенных функций или методов.

Цикломатическая сложность

Цикломатическая сложность – это показатель сложности структуры модуля (число независимых маршрутов в модуле).

Направления использования ЦС

1. Дает рекомендуемое число тестов для ПО.
2. Используется в течение всех фаз жизненного цикла ПО, начиная с разработки, для обеспечения необходимого уровня надежности, тестируемости и управляемости.

Цикломатическая сложность

графа управляющей логики (control flow graph) модуля ПО:

- вычислительные операторы или выражения изображаются в виде узлов
- передача управления между узлами – в виде дуг

Формула вычисления ЦС:

$$C = e - n + 2,$$

где *e* и *n* – число ребер и число узлов на графе управляющей логики соответственно.

Цикломатическая сложность

Упрощенный метод вычисления ЦС:

- модуль с прямолинейным графом управляющей логики (из каждого узла, за исключением узла выхода, выходит ровно одна дуга) имеет цикломатическую сложность, равную 1
- Каждый оператор двоичного выбора (на графе управляющей логики – узел, из которого выходят ровно 2 дуги; например, if, while) увеличивает цикломатическую сложность на 1

Формула вычисления:

$$C = 1 + P2,$$

где P2 – число операторов двоичного выбора.

Более общий случай:

$$C = 1 + P2 + 2*P3 + 3*P4 + \ldots$$

где P*k* – число операторов выбора из *k* альтернатив (на графе управляющей логики узел, из которого выходят *k* дуг).

Цикломатическая сложность

Вариации метрики ЦС:

- «модифицированная ЦС»

  при подсчете операторы выбора (case) не учитываются; считается, что оператор switch как одно целое увеличивает цикломатическую сложность на 1
- Строгая ЦС

  при подсчете учитываются операторы «&&» и «||» (считается, что каждый из них добавляет 1 к цикломатической сложности)

Цикломатическая сложность

Алгоритм вычисления

1. Мера сложности простого оператора равна 1;
2. M ({F1; F2; …;Fn}) = Σin M(Fi);
3. M (IF P THEN F1 ELSE F2) = 2 MAX (M (F1), M (F2));
4. M (WHILE P DO F) = 2 M(F).

Классический подход к оценке результатов рассчета цикломатической сложности:

Сложность программы	Вероятность ошибок
1 – 10	Простая функция, ошибки маловероятны
11 – 20	Более сложная, средняя вероятность ошибок
21 – 50	Сложная, велика вероятность ошибок
51 и более	Не тестируемая (риск очень велик)

Анализ тестового покрытия

Оценивает не только тестируемую программу, но и набор тестов

Совмещает тестирование «черного» и «стеклянного» ящика

Решает следующие проблемы:

- Локализация кода, не имеющего тестового покрытия;
- Определение модулей, требующих дополнительное тестирование;
- Определение тестовых наборов, которые покрывают наибольший и наименьший набор исходных тесктов; определение пересечений тестовых пакетов по проверяемому коду;
- Определение численной меры покрытия, которая является косвенной характеристикой качества продукта

Анализ тестового покрытия

Определяется как отношением исполненных хоть раз единиц структурных единиц (блоков/узлов/дуг/…) к их общему количеству

Оценки для разных структурных единиц взаимосвязаны, требуется четко определить какую из них использовать?

Рассмотрим с точки зрения следующих критериев:

- Автоматизация
- Достижимость
- Понятность
- Изменяемость
- Тщательность

Покрытие строк

Покрытие строк = s/S
где:

s - число строк, выполненных по крайней мере однажды.

S - общее количество выполнимых строк.

Характеристики:

- **Автоматизация 5**
- **Достижимость 5**
- **Постижимость 5**
- **Изменяемость 5**
- **Тщательность 1**

Покрытие дуг

- **Покрытие дуг** = d/D

где:

d - число дуг, выполненных по крайней мере однажды.

D - общее количество дуг.

Характеристика:

- **Автоматизация 5**
- **Достижимость 5**
- **Постижимость 5**
- **Изменяемость 5**
- **Тщательность 2**

Покрытие линейных блоков

Под линейным блоком мы будем понимать непрерывную линейную последовательность строк:

- A. которая начинается или в начале программы или в точке, к которой управление может перейти
- B. который заканчивается или в конце программы или в точке, от которой управление может куда-либо перейти
- C. и точку, к которой будет сделан переход после данной последовательности команд.

Покрытие линейных блоков = l/L

где:

l - число линейных блоков, выполненных по крайней мере однажды.

L - общее количество линейных блоков.

Характеристика:

- **Автоматизация 4**
- **Достижимость 1**
- **Постижимость 1**
- **Изменяемость 2**
- **Тщательность 3**

Покрытие путей выполнения

Покрытие путей выполнения = p/P

где:

p - число путей, выполненных по крайней мере однажды.

P - общее количество путей.

Покрытие путей исполнения рассматривает полные пути исполнения для всей программы. Например, если модуль содержит цикл, тогда существуют пути исполнения модуля для одной итерации, для двух итераций и так далее до n итераций цикла.

Только небольшое число путей исполнения в программе выполнимо.

Для того чтобы сделать охват путей исполнения **достижимым**, метрику следует ограничить покрытием выполнимых путей исполнения.

Покрытие выполнимых путей исполнения

Покрытие выполнимых путей исполнения = f/F

где:

f = число путей, выполненных по крайней мере однажды.

F = общее количество выполнимых путей

Характеристика:

- **Автоматизация 1**
- **Достижимость 1 (выполнимые 3)**
- **Постижимость 2**
- **Изменяемость 2 (выполнимые 1)**
- **Тщательность 4**

Покрытие условий

Покрытие условий = c/C

где:

c = число условий, выполненных по крайней мере однажды.

C = общее количество условий.

Охват условий уязвим к набору флагов вне условного оператора. А так как использование булевых выражений с флагами для упрощения сложных условных операторов является распространенной практикой в программировании, **тщательность** покрытия условий оказывается не такой высокой, как могла бы быть.

Тщательность может быть улучшена, если включить все булевы выражений в охват условий.

Покрытие Булевых условий = e/E

где:

e = число значений булевых условий, выполненных по крайней мере однажды.

E = общее количество булевых условий

Характеристика:

- **Автоматизация 4**
- **Достижимость 5**
- **Постижимость 5**
- **Изменяемость 5**
- **Тщательность 2 (Булевые 3)**

Покрытие условных операторов

Покрытие условных операторов = o/O

где:

o - число комбинаций условия, выполненных по крайней мере однажды.

O - общее количество комбинаций условных операторов

- **Автоматизация 4**
- **Достижимость 4**
- **Постижимость 4**
- **Изменяемость 5**
- **Тщательность 3**

Покрытие эффективности булевых операндов

Покрытие эффективности булевых операндов = b/B

где:

b = число булевых операндов, независимо влияющих на результат выражения.

B = общее количество булевых операндов.

Характеристика:

- **Автоматизация 3**
- **Достижимость 5**
- **Постижимость 5**
- **Изменяемость 5**
- **Тщательность 4**

Сводная таблица методов

Метрика покрытия	Критерий оценки				
	Автомати-зируемость	Дости-жимость	Понятность	Изменя-емость	Тщатель-ность
Покрытие строк	5	5	5	5	1
Покрытие дуг	5	5	5	5	2
Покрытие линейных блоков	4	1	1	2	3
Покрытие путей исполнения	1	1	2	2	4
Покрытие выполнимых путей исполнения	1	3	2	1	4
Покрытие условий	4	5	5	5	2
Покрытие булевых условий	4	5	5	5	3
Покрытие условных операторов	4	4	4	5	3
Покрытие булевых операторов	4	4	4	5	4
Покрытие эффективности булевых операторов	3	5	5	5	4