

Проведений аналіз застосування основних програмних метрик коду на ранньому етапі проектування програмного забезпечення складних корпоративно-інформаційних систем технологічної підготовки виробництва

Ключові слова: метрики, програмне забезпечення, корпоративно-інформаційні системи технологічної підготовки виробництва

Проведен анализ применения основных программных метрик кода на раннем этапе проектирования программного обеспечения сложных корпоративно-информационных систем технологической подготовки производства

Ключевые слова: метрики, программное обеспечение, корпоративно-информационные системы технологической подготовки производства

The analysis of basic programmatic code birth-certificates application is conducted on the early stage of technological preproduction corporate informative systems software planning

Keywords: birth-certificates, software, technological preproduction corporate informative systems

ПРИМЕНЕНИЕ ПРОГРАММНЫХ МЕТРИК КОДА НА РАННЕМ ЭТАПЕ ЖИЗНЕННОГО ЦИКЛА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

В. В. Евсеев

Кандидат технических наук, доцент
Кафедра технологии и автоматизаций производства
Харьковский национальный университет
радиоэлектроники
пр. Ленина, 14, г. Харьков, 03037
Контактный тел.: (057) 702-14-86
E-mail: evv_crow@mail.ru

1. Введение

При разработке программного обеспечения для корпоративных информационных систем технологической подготовки производства (КИС ТПП), возникает важный вопрос о стоимости, трудоемкости и степени риска разрабатываемого программного обеспечения. В отличие от большинства отраслей материального производства, при разработке КИС ТПП недопустимы стандартные подходы расчета трудоемкости, которые основаны на умножении трудоемкости на среднюю производительность труда. На данный момент времени многие разработчики специализированного программного обеспечения, такие как IBM, Oracle, используют программные метрики для определения стоимости и трудоемкости программного обеспечения. Метрика программного обеспечения (англ. software metric) - это мера, позволяющая получить численное значение некоторого свойства программного обеспечения или его спецификаций. Проведем анализ основных метрик программного обеспечения и их возможности применения на раннем этапе жизненного цикла разработки технического задания для оценки предварительной стоимости и трудоемкости разработки программного обеспечения корпоративных информационных систем технологической подготовки производства.

2. Анализ основных метрик программного обеспечения

Метрики сложности программного обеспечения принято разделять на три основные группы [1,2]:

- метрика размера программ;
- метрика сложности потока управления программ;
- метрика сложности потока данных программ.

Метрики размера программ определяются на количественных характеристиках, таких как: размер программы LOC (Lines of Code) - количество исходных строк программного кода, число операторов, этот вид метрик косвенно измеряет программное обеспечение и процесс его разработки. Наиболее часто используемыми в практике для разработки программного обеспечения невысокой сложности получили так называемые размерно-ориентированные метрики, которые регистрируют такие показатели [3]:

- общая трудоемкость (в человеко-месяцах, человеко-часах);
- объем программного обеспечения (в тысячах строк исходного кода);
- стоимость разработки;
- объем документации.

В зависимости от метода, которым учитывается код программного обеспечения, принято использовать два основных показателя:

– количество «физических» строк кода LOC, SLOC, KLOC, KSLOC, DSLOC (общее число строк исходного кода + комментарий + пустые строки) вследствие чего вводится ограничение, учитывающее число «пустых» строк, которое не превышает 25% общего числа строк в измеряемом блоке;

- количество «логических» строк кода LSI, DSI, KDSI (SI-source instructions) определяется как количество команд в зависимости от языка программирования (если язык программирования не допускает размещение нескольких команд на одной строке, то количество «логических» строк будет равно «физическим», за исключение комментариев и пустых строк).

Метрики данной группы ориентированы на анализ исходного текста программного обеспечения, что не позволяет их использовать для оценки трудоемкости и стоимости программного обеспечения на этапе проектирования технического задания на разработку корпоративно информационных систем технологической подготовки производства.

Метрики сложности потока управления программ основаны на анализе управляющего графа программного обеспечения (построенного на основе алгоритмов модулей) вследствие чего эти метрики применяются для оценки сложности промежуточных продуктов разработки. Метрики сложности потока данных программ основаны на оценке использования, конфигурации и размещения данных о программном обеспечении (глобальные переменные). Вследствие чего данный вид метрик можно поделить на следующие классы:

- объектно-ориентируемые метрики;
- метрики Холстеда;
- метрики цикломатической сложности по Мак-Кейбу;
- метрики Чепина.

На современном этапе развития разработки программного обеспечения большинство проектов создаются на основе объектно-ориентируемого подхода; в связи с этим существует много метрик, позволяющих оценить сложность объектно-ориентированных проектов, основные из данных метрик приведены в таблице 1 [4].

Метрика Холстеда относится к метрикам, вычисляемым на основании анализа числа строк и синтаксических элементов исходного кода программы.

Основу метрики Холстеда составляют четыре измеряемые характеристики программы:

- Number of Unique Operators - число уникальных операторов программы, включая символы-разделители, имена процедур и знаки операций (словарь операторов);
- Number of Unique Operands - число уникальных операндов программы (словарь операндов);
- Number of Operators - общее число операторов в программе;
- Number of Operands - общее число операндов в программе.

На основании этих характеристик рассчитываются оценки: словарь программы (Halstead Program Vocabulary), длина программы (Halstead Program Length), объем программы (Halstead Program Volume), сложность программы (Halstead Difficulty).

Метрики цикломатической сложности по Мак-Кейбу являются одним из наиболее распространенных по-

казателей оценки сложности программных проектов, относятся к группе показателей оценки сложности потока управления программой и вычисляются на основе графа управляющей логики программы (control flow graph). Данный граф строится в виде ориентированного графа, в котором вычислительные операторы или выражения представляются в виде узлов, а задача управления между узлами - в виде дуг.

Таблица 1

Метрика	Описание
Взвешенная насыщенность класса 1 (Weighted Methods Per Class (WMC))	Отражает относительную меру сложности класса на основе цикломатической сложности каждого его метода. Класс с более сложными методами и большим количеством методов считается более сложным. При вычислении метрики родительские классы не учитываются.
Взвешенная насыщенность класса 2 (Weighted Methods Per Class (WMC2))	Мера сложности класса, основанная на том, что класс с большим числом методов, является более сложным, и что метод с большим количеством параметров также является более сложным. При вычислении метрики родительские классы не учитываются
Глубина дерева наследования (Depth of inheritance tree)	Длина самого длинного пути наследования, заканчивающегося на данном модуле. Чем глубже дерево наследования модуля, тем может оказаться сложнее предсказать его поведение. С другой стороны, увеличение глубины даёт больший потенциал повторного использования данным модулем поведения, определённого для классов-предков
Количество детей (Number of children)	Число модулей, непосредственно наследующих данный модуль. Большие значения этой метрики указывают на широкие возможности повторного использования; при этом слишком большое значение может свидетельствовать о плохо выбранной абстракции.
Связность объектов (Coupling between objects)	Количество модулей, связанных с данным модулем в роли клиента или поставщика. Чрезмерная связность говорит о слабости модульной инкапсуляции и может препятствовать повторному использованию кода.
Отклик на класс (Response For Class)	Количество методов, которые могут вызываться экземплярами класса; вычисляется как сумма количества локальных методов, так и количества удаленных методов

Упрощенная формула вычисления цикломатической сложности представляется следующим образом:

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

где:

e - число ребер,

n - число узлов на графе управляющей логики.

В процессе автоматизированного вычисления показателя цикломатической сложности, как правило, применяется упрощенный подход, в соответствии с которым построение графа не осуществляется, а вычисление показателя производится на основании подсчета числа операторов управляющей логики (if, switch и т.д.) и возможного количества путей исполнения программы. Цикломатическое число Мак-Кейба пока-

зывает требуемое количество проходов для покрытия всех контуров сильносвязанного графа или количества тестовых прогонов программы, необходимых для исчерпывающего тестирования по принципу «работает каждая ветвь». Показатель цикломатической сложности может быть рассчитан для модуля, метода и других структурных единиц программы.

Существует значительное количество модификаций показателя цикломатической сложности:

- «модифицированная» цикломатическая сложность - рассматривает не каждое ветвление оператора множественного выбора (switch), а весь оператор как единое целое.

- «строгая» цикломатическая сложность - включает логические операторы.

- «упрощенное» вычисление цикломатической сложности - предусматривает вычисление не на основе графа, а на основе подсчета управляющих операторов.

Метрика Чепина состоит в оценке информационной прочности отдельно взятого программного модуля с помощью анализа характера использования переменных из списка ввода-вывода.

Все множество переменных, составляющих список ввода-вывода, разбивается на четыре функциональные группы:

- множество «Р» - вводимые переменные для расчетов и для обеспечения вывода. Примером может служить используемая в программах лексического анализатора переменная, содержащая строку исходного текста программы, то есть сама переменная не модифицируется, а только содержит исходную информацию.

- множество «М» - модифицируемые или создаваемые внутри программы переменные.

- множество «С» - переменные, участвующие в управлении работой программного модуля (управляющие переменные).

- множество «Т» - неиспользуемые в программе («паразитные») переменные. Поскольку каждая переменная может выполнять одновременно несколько

функций, необходимо учитывать ее в каждой соответствующей функциональной группе.

Далее вводится значение метрики Чепина:

$$Q = a1P + a2M + a3C + a4T,$$

где

$a1, a2, a3, a4$ – весовые коэффициенты.

Весовые коэффициенты использованы для отражения различного влияния на сложность программы каждой функциональной группы. По мнению автора метрики наибольший вес, равный трем, имеет функциональная группа С, так как она влияет на поток управления программы. Весовые коэффициенты остальных групп распределяются следующим образом: $a1=1$; $a2=2$; $a4=0.5$. Весовой коэффициент группы Т не равен нулю, поскольку «паразитные» переменные не увеличивают сложности потока данных программы, но иногда затрудняют ее понимание.

3. Выводы

Проведенный анализ некоторых программных метрик показал невозможность их применения на ранней стадии составления технического задания для оценки трудоемкости, стоимости и степени риска разработки сложных корпоративно-информационных систем технологической подготовки производства. Это объясняется отсутствием на данном этапе программного кода, алгоритма работы программного обеспечения, а также в данных метриках не учитывается специфика и сложность разработки корпоративно-информационных систем технологической подготовки производства, которые содержат в себе не только программные коды, но и специфические решения необходимые для решения задач технологической подготовки производства, которые выдвигают требования точности, адекватности и рациональности технологических процессов, необходимых на производстве.

Литература

1. Технологии разработки программного обеспечения: Учебник / С. Орлов. - СПб.: Питер, 2002. - 464 с.
2. Ерик Дж. Брауде Технология разработки программного обеспечения. - СПб.: Питер, 2004. - 655 с.
3. А.В. Рудаков Технология разработки программных продуктов.- Академия.- Москва.-2006.-208 с.
4. Метрики кода и их практическая реализация в IBM Rational ClearCase /<http://www.ibm.com/developerworks/ru/edu/0108novich/section2.html>.