

# ИСПОЛЬЗОВАНИЕ ОБЪЕКТНО-ОРИЕНТИРОВАННЫХ МЕТРИК В CASE-СИСТЕМАХ ДЛЯ ПРЕДВАРИТЕЛЬНОГО ТЕСТИРОВАНИЯ

С. В. АФАНАСЬЕВ

Санкт-Петербургский институт информатики и автоматизации РАН

СПИИРАН, 14-я линия ВО, д. 39, Санкт-Петербург, 199178

<serg@sqlab.nw.ru>

---

УДК 004.05

Афанасьев С. В. **Использование объектно-ориентированных метрик в CASE-системах для предварительного тестирования.** // Труды СПИИРАН. Вып. 5 — СПб.: Наука, 2007..

**Аннотация.** *Метамоделирование - важная часть процесса проектирования и разработки программного продукта на всех этапах. Тестирование, в свою очередь, трудоемкий, но необходимый процесс на конечной стадии разработки программного продукта. В статье проводится анализ существующих метрик и предлагается выборка для включения в CASE – системы на примере Together-Borland для предварительного тестирования программного обеспечения на стадии разработки.* — Библ. 6 назв.

UDC 004.05

Afanasyev S.V. **Object-oriented metrics using in CASE-systems for pretesting.** // SPIIRAS Proceedings. Issue 5. — SPb.: Nauka, 2007.

**Abstract.** *Metamodeling is very important part of design developing software in the all steps. Testing, in one's turn, is a hard but necessary part in completing of developing.. This article describes analysis of object-oriented metrics, some of them are suggested to include in Together Borland CASE-system to reduce general (end) testing.* — Bibl. 6 items

---

## 1. Введение

Функциональное тестирование (functional testing) — процесс верификации соответствия функционирования продукта его начальным спецификациям. По мере сложности программных систем, комбинации различных входных параметров и поддерживаемых операционных систем нередко исчисляются десятками и сотнями [2]. В такой ситуации функциональное тестирование требует значительного объема ресурсов; естественным выходом является автоматизация этого процесса. Наиболее подходящим можно считать использование тестирования на основе метамodelей [3,4]. *Объектом* настоящего исследования является особый вид математических моделей фрагментов знаний (ФЗ) и баз фрагментов знаний (БФЗ) с вероятностной неопределенностью — идеалы цепочек конъюнкций с оценками вероятностей истинности своих элементов и, соответственно, сети таких идеалов, которые называются алгебраическим байесовскими сетями (АБС).

## 2. Метрики, аудит тестирование

Функциональное тестирование (functional testing) — процесс верификации соответствия функционирования продукта его начальным спецификациям. По мере сложности программных систем, комбинации различных входных параметров и поддерживаемых операционных систем нередко исчисляются десятками и сотнями [2]. В такой ситуации функциональное тестирование требует значительного объема ресурсов; естественным выходом является автоматиза-

ция этого процесса. Наиболее подходящим можно считать использование тестирования на основе метамоделей [3,4].

Преимущества тестирования на основе метамоделей с использованием CASE-систем заключаются в том, что:

- тесты на основе спецификации функциональных требований более эффективны, так как они в большей степени нацелены на проверку функциональности, чем тесты, построенные только на знании реализации;
- на основе формальных спецификаций можно создавать самопроверяющие (self-checking) тесты, так как из формальных спецификаций часто можно извлечь критерии проверки результатов целевой системы.

Модели обычно проще реализации, поэтому можно предположить, что тесты, хорошо «покрывающие» модель, слишком бедны для покрытия реальных систем. Поэтому требуется некоторое количество широких экспериментов в реальных проектах для определения качества тестового покрытия.

Модель — некоторое отражение структуры и поведения системы. Модель может описываться в терминах состояния системы, входных воздействий на нее, конечных состояний, потоков данных и потоков управления, возвращаемых системой результатов и т.д. Для отражения разных аспектов системы применяются и различные наборы терминов. Формальная спецификация представляет собой законченное описание модели системы и требований к ее поведению в терминах того или иного формального метода. Для описания характеристик системы можно воспользоваться несколькими моделями в рамках нескольких формализмов, например разных типов UML диаграмм. Обычно, чем более общей является нотация моделирования, тем больше трудностей возникает при тестировании программы на основе модели/спецификации, описанной в этой нотации. Одни нотации и языки больше ориентированы на доступность и прозрачность описания, другие — на последующий анализ и трансляцию, в частности, трансляцию спецификации в тестовый сценарий. Адаптировав к данной ситуации методы построения тестового сценария по метамоделю, мы получим алгоритмы (метрики) проверки для конкретных функциональных моделей.

Для полноты определения качества передачи функциональности необходимо учесть ряд немаловажных метрик, от которых напрямую зависит качество программных продуктов в целом. Такими метриками являются: циклическая сложность, размер класса, отклик для класса, недостаток единства, связь между объектами классов, глубина дерева наследований, число потомков.

Метрики для общей оценки разработанного ПО достаточно хорошо разработаны (Холстед, Джилб, Хур и др.), эти метрики безусловно важны для общей оценки программного продукта (метрики 1, 2, 3 в таблице 1). В тоже время есть ряд метрик применяемых для объектно-ориентированного проектирования и разработки программ, причем среди них существенно выделить ряд метрик, как необходимых для использования на стадии проектирования и разработки, интегрированных в CASE-системе, что позволяет сделать предварительное тестирование разработки ПО (метрики 4 - 9 в таблице 1).

Объектно-ориентированный подход в проектировании и разработке сложных систем стал очень популярным в современном программировании. Объектно-ориентированная разработка требует не только другого подхода к проектированию и программированию, но и другого подхода к метрикам, используемым при программировании. Так как в объектно-ориентированной технологии используются объекты, а не алгоритмы, как фундаментальные стандартные блоки, подход к программным метрикам для объектно-ориентированных про-

грамм должен отличаться от стандартного набора метрик. Некоторые метрики, типа количества строк программы и циклической сложности, стала принятой как "стандарт" для традиционного функционально-процедурного программирования, но для объектно-ориентированного в литературе предложены много объектно-ориентированных метрик.

Подход к определению набора объектно-ориентированных метрик состоит в том, чтобы выделить первичные критические конструкции объектно-ориентированного проекта и выбрать метрики, оценивающие эти области [5,6]. Нацеленность метрик на внутренний объект отражает сложность каждого индивидуального объекта (например, метода или класса) на внешней сложности. Метрики измеряют вычислительную сложность этого воздействия на эффективность алгоритма и использования машинных ресурсов и психологические факторы сложности, которые затрагивают способность программиста к созданию, постижению, изменению и поддержке программного обеспечения. Основные метрики, используемые при разработке объектно-ориентированных программных систем представлены в Таблице 1.

Объектно-ориентированные методы разработки имеют собственную терминологию для отображения новых структурных концепций. Объектно-ориентированная система начинается с определения класса (Class A), имеющего связанные и подобные атрибуты и операции (некоторые операции являются методами). Классы используются как основа для объектов (Object A1). Дочерний класс наследует атрибуты и операции от родительского класса в дополнение к наличию собственных атрибутов и операций. Дочерний класс также может стать родительским для другого класса, формируя новые ветви в иерархической древовидной структуре. Когда объект создан, чтобы содержать данные или информацию, это – реализация класса. Классы взаимодействуют друг с другом, посылая сообщения [1]. Методы – операции, реализованные над объектом.

Таблица 1

Метрики для ООП

Категория метрики	Метрика	Конструкция
Обычная	Циклическая сложность (CC)	Метод
Обычная	Строки кода (LOC)	Метод
Обычная	Процент комментариев (CP)	Метод
Объектно-ориентированная	Взвешенные методы на класс (WMC)	Класс/Метод
Объектно-ориентированная	Отклик для класса (RFC)	Класс/Сообщение
Объектно-ориентированная	Недостаток единства Методов (LCOM)	Класс/Единство
Объектно-ориентированная	Связь между объектами (CBO)	Связь
Объектно-ориентированная	Глубина дерева наследований (DIT)	Наследование
Объектно-ориентированная	Число потомков (NOC)	Наследование

Таблица 2

### Краткое описание объектно-ориентированных структур

Атрибут	Определяет структурные свойства класса, уникальные в пределах класса, существительное
Класс	Набор объектов, имеющих общую структуру и общее поведение, описанное набором методов; служит как шаблон, по которому может быть создан объект
Единство	Степень, в которой методы класса связаны друг с другом
Связь	Объект X связан с объектом Y тогда и только тогда, когда X посылает сообщение Y.
Наследование	Отношение между классами, при котором объект класса приобретает характеристики одного или более других классов
Реализация	Процесс создания образца объекта и закрепления или добавления определённых данных
Сообщение	Запрос, который объект получает от другого объекта, чтобы выполнить действие
Метод	Операция над объектом, определённым как часть описания класса
Объект	Реализация некоторого класса, которая способна сохранять состояние (информацию) и которая имеет несколько операций, чтобы исследовать или изменять это состояние
Операция	Действие, выполненное объектом или над объектом, доступное всем образцам класса, не должно быть уникальным

### 3. Заключение

Ценность метрик заключается в их применении к программам. Как они могут помочь разработчикам улучшить качество программ? В то время как существует множество рекомендаций относительно того, как интерпретировать метрики, не имеется достаточного количества статистических данных, доказывающих, что значение 8 для одной метрики вдвойне сложнее или вдвойне «хуже», чем 4. Необходимо сравнить значения, чтобы определить, чем они отличаются от других модулей кода. Это не индикация «вредности», а индикатор различия, которое должно быть исследовано.

Включение (интеграция) по крайней мере перечисленных объектно-ориентированных метрик в CASE-систему существенно сократит и упростит процесс окончательного тестирования. Разработка методов тестирования предполагает исследование и использование результатов, достигнутых в области функционального тестирования программных систем. Должны быть рассмотрены основные стандарты оценки качества программ и проанализированы существующие метрики качества. Обязательным условием решения основной задачи является изучение реализованного в программном комплексе метода трансляции между моделями, что позволит рассмотреть вопрос с точки зрения понимания внутренних процессов трансляции.

### Литература

1. Афанасьев С.В., Воробьев В.И. Метрики для объектно-ориентированного проектирования сложных систем. Вестник гражданских инженеров. СПб, издательство СГБГАСУ, 2005, N4. С. 108–114.
2. Канер С., Фолк Д., Нгуен Е.К., Тестирование программного обеспечения. М: ДиаСофт, 2000. 544 с.
3. Петренко А., Бритвина Е., Грошев С., Монахов А, Петренко О. Тестирование на основе моделей [Электронный ресурс]: <<http://www.citforum.ru/SE/testing/model/>>.

4. *Подъячев , Афанасьев С.В.* Тестирование трансляции формальных моделей. / Труды СПИИРАН, СПб, Наука, 2006, Вып 3, т. 2. С. 156–161.
5. *Booch G.*, Object Oriented Analysis and Design with Applications, The Benjamin/Cummings Publishing Company, Inc., 1994. 463 p.
6. *Rosenberg L. H.*, Metrics for Object Oriented Environments, EFAITP/AIE Third Annual Software Metrics Conference, December, 1997. P. 156–169.