

# МЕТРИКИ И КАЧЕСТВО КОНТРОЛЯ

ВОЛГОГРАД, 2021

- 1 Введение
- 1.1 Понятие качества
- 1.3 Метрики
  - 1.3.1 Основные направления применения метрик
  - 1.3.3 Метрики сложности программ
  - 1.3.4 Объектно-ориентированные метрики
  - 1.3.5 Метрики Холстеда
  - 1.3.6 Метрики цикломатической сложности по Мак-Кейбу
  - 1.3.7 Метрики Чепина
  - 1.3.8 Размерно-ориентированные метрики (показатели оценки объема)
    - 1.3.8.1 SLOC - оценка (Source Lines Of Code)
    - 1.3.8.2 Метрика стилистики и понятности программ
- 1.5 Оценка результата
- 1.7.1 Вычисление метрики SLOC

## Введение

Когда вы можете измерить, то о чем вы говорите, и выразить это в **числах**, вы знаете кое-что об этом; но если вы не можете измерить это и выразить в числах, ваше знание скудно и неудовлетворительно.

*Лорд Кельвин*

Требования к качеству программных средств всё время повышаются. Процессы разработки, приобретения и внедрения сложных систем, к которым относятся в частности программные комплексы, должны находиться под жестким управленческим контролем. В настоящее время практически во всех организациях обеспечивается контроль важнейших характеристик, связанных с производством и использованием программных продуктов, таких как время, финансовые средства, ресурсы и т.п. Однако в большинстве случаев вне пределов сферы контроля оказывается наиболее важная характеристика программных продуктов, ради которой, собственно и осуществляются затраты времени, финансовых средств и ресурсов - это качество продукта, поскольку «невозможно контролировать то, что нельзя измерить»

## Понятие качества

Что такое качество и почему оно должно быть столь глубоко представлено? На протяжении многих лет отдельные авторы и целые организации определяли термин “качество” по-разному. Фил Кросби (Phil Crosby) в 1979 году дал определение качеству как “соответствие пользовательским требованиям” (предполагает, что требования должны быть настолько четко определены, что они не могут быть поняты и интерпретированы некорректно.). Уотс Хемпффри (Watts Humphrey) описывает качество как “достижение отличного уровня пригодности к использованию” (принимает во внимание требования и ожидания конечных пользователей продукта, которые ожидают, что продукт или предоставляемый сервис будет удобным для их нужд). Компания IBM, в свою очередь, ввела в оборот фразу “качество, управляемое рыночными потребностями” (“market-driven quality”). Критерий Бэлдриджа (Baldrige) для организационного качества использует похожую фразу - “качество, задаваемое потребителем” (“customer-driven quality”), рассматривая удовлетворение потребителя в качестве главного соображения в отношении качества. Чаще, понятие качества используется в соответствии с определением системы менеджмента качества ИСО 9001 как “степень соответствия присущих характеристик требованиям”.

## 1.3 Метрики

В настоящее время в программной инженерии еще не сформировалась окончательно система метрик. Действуют разные подходы к определению их набора и методов измерения.

Система измерения включает метрики и модели измерений, которые используются для количественной оценки качества ПО.

При определении требований к ПО задаются соответствующие им внешние характеристики и их атрибуты (подхарактеристики), определяющие разные стороны управления продуктом в заданной среде. Для набора характеристик качества ПО, приведенных в требованиях, определяются соответствующие метрики, модели их оценки и диапазон значений мер для измерения отдельных атрибутов качества.

Согласно стандарту метрики определяются по модели измерения атрибутов ПО на всех этапах ЖЦ (промежуточная, внутренняя метрика) и особенно на этапе функционирования (внешние метрики) продукта.

Метрика качества программ - система измерений качества программ. Эти измерения могут проводиться на уровне критериев качества программ или на уровне отдельных характеристик качества. В первом случае система измерений позволяет непосредственно сравнивать программы по качеству.

При этом сами измерения не могут быть проведены без субъективных оценок свойств программ. Во втором случае измерения характеристик можно выполнить объективно и достоверно, но оценка качества ПО в целом будет связана с субъективной интерпретацией получаемых оценок. [3, 12]

### 1.3.1 Основные направления применения метрик

В настоящее время в мировой практике используется несколько сотен метрик программ. Существующие качественные оценки программ можно сгруппировать по шести направлениям:

оценки топологической и информационной сложности программ;

оценки надежности программных систем, позволяющие прогнозировать от казовые ситуации;

оценки производительности ПО и повышения его эффективности путем выявления ошибок проектирования;

оценки уровня языковых средств и их применения;

оценки трудности восприятия и понимания программных текстов, ориентированные на психологические факторы, существенные для сопровождения и модификации программ;

оценки производительности труда программистов для прогнозирования сроков разработки программ и планирования работ по созданию программных комплексов

### 1.3.3 Метрики сложности программ

Теория сложности программ ориентирована на управление качеством ПО и контроль ее эталонной сложности в период эксплуатации. В настоящее время многообразие показателей (в той или иной степени описывающих сложность программ) столь велико, что для их употребления требуется предварительное упорядочение. В ряде случаев удовлетворяются тремя категориями метрик сложности. Первая категория определяется как словарная метрика, основанная на метрических соотношениях Холстеда, цикломатических мерах Мак-Кейба и измерениях Тейера. Вторая категория ориентирована на метрики связей, отражающих сложность отношений между компонентами системы - это метрики Уина и Винчестера. Третья категория включает семантические метрики, связанные с архитектурным построением программ и их оформлением.

### 1.3.4 Объектно-ориентированные метрики

В современных условиях большинство программных проектов создается на основе ОО подхода, в связи с чем существует значительное количество метрик, позволяющих получить оценку сложности объектно-ориентированных проектов.

### 1.3.5 Метрики Холстеда

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:

NUOprtr (Number of Unique Operators) — число уникальных операторов программы, включая символы-разделители, имена процедур и знаки операций (словарь операторов);

NUOprnd (Number of Unique Operands) — число уникальных операндов программы (словарь операндов);

Noprtr (Number of Operators) — общее число операторов в программе;

Noprnd (Number of Operands) — общее число операндов в программе.

### 1.3.6 Метрики цикломатической сложности по Мак-Кейбу

Показатель цикломатической сложности является одним из наиболее распространенных показателей оценки сложности программных проектов. Данный показатель был разработан ученым Мак-Кейбом в 1976 г., относится к группе показателей оценки сложности потока управления программой и вычисляется 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 передача управления между узлами - в виде дуг.

Показатель цикломатической сложности позволяет не только произвести оценку трудоемкости реализации отдельных элементов программного проекта и скорректировать общие показатели оценки длительности и стоимости проекта, но и оценить связанные риски и принять необходимые управленческие решения.

### 1.3.7 Метрики Чепина

Существует несколько ее модификаций. Рассмотрим более простой, а с точки зрения практического использования - достаточно эффективный вариант этой метрики.

Суть метода 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» - не используемые в программе (“паразитные”) переменные. Поскольку каждая переменная может выполнять одновременно несколько функций, необходимо учитывать ее в каждой соответствующей функциональной группе.

### 1.3.8 Размерно-ориентированные метрики (показатели оценки объема)

Размерно-ориентированные метрики прямо измеряют программный продукт и процесс его разработки. Самой распространенной метрикой исходного кода ПО, отражающей размер программного проекта, является показатель количества строк кода (Source Lines Of Code, SLOC)

### **1.3.8.1 SLOC - оценка (Source Lines Of Code)**

Первоначально SLOC применялся в условиях, когда языки программирования обладали относительно простой структурой и для них характерным было соответствие одной строки кода одной команде языка. Со временем языки программирования эволюционировали, стали гораздо гибче, и это соответствие для них больше не выполнялось – одна физическая строка кода теперь может содержать несколько команд языка. С учетом этого выделялись и две основные разновидности показателя SLOC:

### **1.3.8.2 Метрика стилистики и понятности программ**

Иногда важно не просто посчитать количество строк комментариев в коде и просто соотнести с логическими строчками кода, а узнать плотность комментариев. То есть код сначала был документирован хорошо, затем - плохо. Или такой вариант: шапка функции или класса документирована и комментируется, а код нет.

- 
- ▶ 1.5 Оценка результата
  - ▶ Оценка результата программных проектов - задача весьма сложная. Зачастую какие-то показатели нужны еще до начала работ, чтобы спланировать будущие расходы, другие используются для определения размера вознаграждения за уже выполненный труд. Наука не стоит на месте и предлагает все новые подходы к решению этой задачи, однако до сих пор нет надежной универсальной методики, дающей гарантированный результат. Поэтому в большинстве случаев приходится опробовать различные методы.

### 1.7.1 Вычисление метрики SLOC

**Locmetrics** - очень простой бесплатный продукт с минималистским интерфейсом. В числе поддерживаемых языков - C/C++, C#, Java, SQL - возможно вычисление не только метрики SLOC и ее разновидностей, но и цикломатической сложности. Отсутствие документации не является препятствием для использования программы, поскольку разобраться в интерфейсе из двух кнопок и двух полей ввода совсем несложно. Хуже, что нет даже описания методики расчета метрик. К недостаткам также следует отнести отсутствие хотя бы самых простых средств построения отчетности или экспорта данных в один из популярных форматов.

**USC Codecount** - бесплатный продукт с открытыми исходными кодами на языке ANSI C, разработанный Университетом Южной Калифорнии (University of Southern California, USC) - той же организацией, в которой были созданы COCOMO/COCOMO II. По этой причине USC Codecount является официальным инструментом для подсчета метрики SLOC при использовании указанных моделей. В число поддерживаемых языков входят C/C++, C#, Java, JavaScript, SQL, Perl, XML, в документации указано, что методика расчета соответствует принятой SEI для моделей CMM/CMMI.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- Андон Ф.И., Суслов В.Ю., Коваль Г.И., Коротун Т.М. Основы качества программных систем.–Киев, Академперіодика.– 2002.–502с.
- Бабенко Л.П., Лаврищева Е.М Основы программной инженерии. Учебник Киев: Знание, 2001. – 269 с
- Боэм Б.У. Инженерное проектирование программного обеспечения. Пер. с англ. / Под ред. А.А. Красилова. – М.: Изд-во Радио и связь, 1985. – 512 с.
- Боэм Б.У., Браун Дж., Каспар Х. и др. Характеристики качества программного обеспечения. М. Мир, 1981.
- Воробьев В. И., Копыльцов А. В., Пальчун Б. П., Юсупов Р. Методы и модели оценивания качества программного обеспечения. М. С-Пб.: СПИИРАН.1992.-33с.
- Колдовский В. Разработка ПО: оценка результата. Компьютерное обозрение №34 (553) 2006
- Кулаков А.Ю. Оценка качества программ ЭВМ.–Киев: Техніка.–1984.–167с.
- Липаев В. Качество программного обеспечения. - М.: Финансы и статистика, 1983.
- Липаев В.В. Методы обеспечения качества крупномасштабных программных систем. – М.: СИНТЕГ.– 2003.–510 с.
- Липаев В.В. Обеспечение качества программных средств. Методы и стандарты. – М.: Синтег, 2001. – 380 с.
- Орлов С. Технологии разработки программного обеспечения: Учебник/ - СПб.: Питер, 2002. - 464 с.: ил.
- Соммервил И. Инженерия программного обеспечения. 6 -издание.–Москва–Санкт– Петербург–Киев, 2002.–623 с.
- Фокс Дж. Программное обеспечение и его разработка М.: "Мир", 1982.
- Холстед М.Х. Начало науки о программах. - М.: Финансы и Статистика, 1981.
- Boehm B.W. The COCOMO 2.0 Software Cost Estimation Model. – American Programmer. – 2000. – 586 p.
- ISO/IEC 9126 Software engineering — Product quality — Part 1: Quality model, 2001
- ISO/IEC 9126 Software engineering — Product quality — Part 2: External metrics, 2001
- ISO/IEC 9126 Software engineering — Product quality — Part 3: Internal metrics, 2001
- ISO/IEC 9126 Software engineering — Product quality — Part 4: Quality in use metrics, 2001