

ОСНОВНЫЕ ЗАДАЧИ ПРОГРАММОМЕТРИКИ

Количественный анализ целесообразности разработки процедур и функций информационной системы в заданной постановке.

Численная оценка основных параметров будущего программного средства на основе постановок задач.

Планирование и управление разработкой программного средства, оценка трудоемкости его создания, технико-экономическое обоснование.

Решение вопросов, связанных с метрологией качества программных средств.



Модуль 3. Оценка качества программного обеспечения

Тема 3.1. Общие сведения о программометрике

В модуле «Оценка качества программного обеспечения» рассмотрены методики оценки характеристик и измерения метрик программного обеспечения. В него включены лексический анализ программ, оценка структурной сложности программ,

процедурно- и объектно-ориентированные метрики. А также надежность программ, оценка характеристик при стандартизации и сертификации программного обеспечения.

Разнообразие методик позволяет применять их на различных стадиях жизненного цикла программ – от начала разработки до сертификации ПО.

Новое научное и прикладное направление – метрическая теория программ, или программометрика, – позволяет с приемлемой для практики точностью решать следующие задачи:

- количественный анализ возможности и целесообразности разработки автоматизированных процедур и функций информационной системы в заданной постановке;
- численная оценка основных параметров будущего программного средства: объема, количества модулей, числа уровней иерархии, надежности в

начальный период эксплуатации на основе постановок задач;

- планирование и управление разработкой программного средства, оценка трудоемкости его создания, технико-экономическое обоснование;
- решение вопросов, связанных с метрологией качества программных средств.

Применение метрик позволяет упорядочить разработку, испытания, эксплуатацию и сопровождение программного средства.

ХАРАКТЕРИСТИКА МЕТОДОВ ОЦЕНКИ КАЧЕСТВА ПРОГРАММ

Методология разработки	<ul style="list-style-type: none">• Процедурно-ориентированная• Объектно-ориентированная
Система измерений	<ul style="list-style-type: none">• Нормативная (стандартная)• Экспериментальная• Иерархическая• Одноранговая
Назначение модели	<ul style="list-style-type: none">• Комплексная оценка качества• Оценка сложности• Оценка надежности• Оценка трудоемкости
Метод определения значений	<ul style="list-style-type: none">• Измерительный• Расчетный• Регистрационный• Экспертный



Метрическую теорию программ представляют различные математические модели определения численных значений характеристик программного обеспечения, в том числе характеристик качества. Каждая такая модель представляет ту или иную метрику программы.

«Метрика качества программного обеспечения определяется как количественный масштаб и метод.

Последние могут быть использованы для определения значения признака, принятого для конкретной программной продукции.

Признаки, или показатели, определяют свойства программной продукции, которые могут быть отнесены к характеристикам качества. Примерами признаков являются длина маршрута, модульность, структура программы.

Существует три типа метрик:

- метрики программного продукта, которые используются при измерении его характеристик;
- метрики процесса, которые используются при измерении свойства процесса жизненного цикла создания продукта;
- метрики использования, которые оценивают результаты эксплуатации программы» [15].

Метрики могут по-разному зависеть от окружения и фаз процесса разработки, в которых они

используются. Метрики, используемые в процессе разработки, должны быть соотнесены с соответствующими метриками пользователя. Обусловлено это тем, что метрики из представления пользователя являются решающими.

Выбор и последующее применение той или иной модели зависят от методологии разработки, назначения программного средства, используемой системы и приемлемых методов оценки или измерений.

ТРЕБОВАНИЯ К МЕТОДИКАМ И МОДЕЛЯМ ОЦЕНКИ ХАРАКТЕРИСТИК ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ



Отмечают некоторые недостатки моделей и метрик, оценивающих производственные и эксплуатационные свойства программного обеспечения:

- стремление к их универсальности;
- неполный учет области применения разрабатываемых программных средств;
- игнорирование этапов жизненного цикла

программного обеспечения.

«Анализ технологического опыта лидеров производства программного обеспечения показывает, насколько дорого обходится несовершенство прогноза результатов и трудозатрат, сложности программ, негибкость контроля и управления их разработкой.

Вышеуказанные причины свидетельствуют об отсутствии сквозной методической поддержки. Также они приводят к несоответствию требованиям пользователя, требуемому стандарту и к последующей трудоемкой переделке программных средств» [1].

Учитывая недостатки методической поддержки оценки характеристик ПО, необходимо выполнение следующих требований:

- «тщательный отбор методик, моделей, методов оценки качества программного обеспечения;

- учет ограничений пригодности методик и моделей для различных жизненных циклов и в пределах жизненного цикла;
- установление порядка совместного использования методик и моделей;
- применение избыточного разномодельного исследования одних и тех же показателей для повышения достоверности текущих оценок;
- накопление и интеграция разнородной метрической информации» [1].

Все эти мероприятия позволят улучшить методическую поддержку программных средств на всех этапах жизненного цикла.

КЛАССИФИКАЦИЯ МОДЕЛЕЙ ОЦЕНКИ ХАРАКТЕРИСТИК ПРОГРАММНЫХ СРЕДСТВ

По типу получаемой информации о метриках программ

- оценивающие отклонение от нормы;
- прогнозирующие значения характеристик;
- формирующие принятие решений о соответствии программного обеспечения заданным требованиям.

По типу используемой информации о программах

- метрики на основе лексического анализа программ;
- метрики на основе анализа потока управления;
- метрики на основе анализа потока данных;
- метрики на основе анализа внутримодульных и межмодульных связей.

Классификация моделей оценки характеристик программных средств состоит из двух групп. Первая группа – по типу получаемой информации о метриках программ. Вторая – по типу используемой информации о программах.

В моделях метрики используют для прогнозирования характеристик будущего программного продукта, а также для оценки

сложности и корректности создаваемых программ.

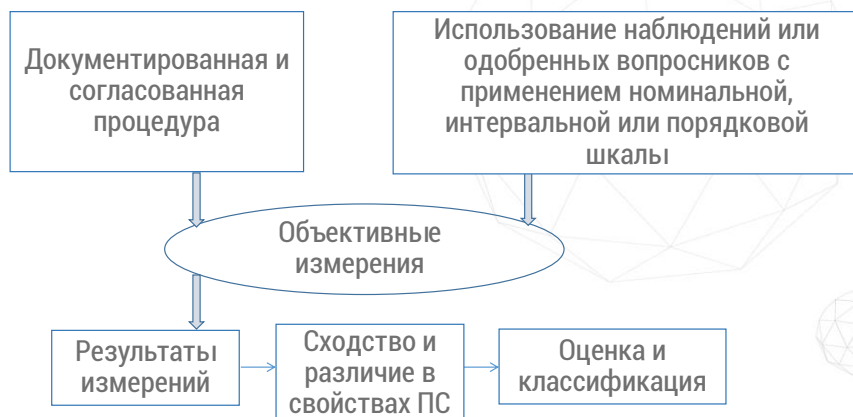
К наиболее используемым метрикам относят метрики Холстеда, Джилба, Маккейба, Майерса, Чепина и некоторые другие.

Метрики на основе анализа внутримодульных и межмодульных связей являются основными характеристиками сложности программного обеспечения в фазе проектирования. Объединяя сведения о потоке управления и потоке данных, метрики внутримодульных и межмодульных связей образуют шкалы функциональной прочности и сцепления модулей.

Исторически первыми появились математические модели, представляющие метрики программ, основанные на анализе лексики и потока управления программ. Типичными представителями таких метрик являются метрики Холстеда и

Маккейба.

ПРОЦЕСС ИЗМЕРЕНИЯ КАЧЕСТВА ПРОГРАММНЫХ СРЕДСТВ



Для оценки качества необходимо разрабатывать показатели качества, то есть выделить измеримые свойства продукции, важные для потребителя. Для измерения качества продукции применяются статистический анализ данных и экспертное оценивание.

Чтобы выбрать характеристики качества программного средства и сравнить их с

требованиями, а также сопоставить их значения для различных программных продуктов, необходимы измерения. Полезно использовать определенные меры и шкалы.

Стандартами рекомендуется измерение каждой характеристики качества программного средства с точностью, достаточной для выполнения сравнений с требованиями. При этом нужно, чтобы эта точность обеспечивалась при измерении.

Система измерений характеристик программного обеспечения – это совокупность измеряемых характеристик, единиц измерения, измерительных шкал и связей, установленных между ними. Если между измеряемыми характеристиками установлены иерархические связи, систему измерений называют иерархической; в противном случае – одноранговой.

Измерительная шкала устанавливает границы и точность измерений характеристик свойств в

установленных единицах. Результаты измерений в избранной измерительной шкале позволяют обнаружить сходство и различие в свойствах программного обеспечения. Это важно для последующей оценки и классификации.

Чтобы измерения были объективными, должна существовать согласованная процедура присвоения числового значения, свойства или категории каждому атрибуту программного продукта.

Процедуры измерений с приемлемой устойчивостью должны давать в результате одинаковые меры, получаемые различными объектами при выполнении одних и тех же измерений характеристик программных средств в различных случаях.

ИЗМЕРИТЕЛЬНЫЕ ШКАЛЫ

Статистическая шкала	Эмпирическая значимость
Номинальная	Нет
Порядковая	Порядок чисел
Интервальная	Разность чисел
Шкала отношений	Отношение чисел

Рассмотрим типы измерительных шкал, применяемых при эмпирических исследованиях качества программных средств.

Возможности обработки переменных, относящихся к номинальной шкале, очень ограничены. Переменные, относящиеся к номинальной шкале, часто используются для группировки. Это позволяет совокупную выборку

разбить по категориям этих переменных.

В частичных выборках проводятся одинаковые статистические тесты, результаты которых затем сравниваются друг с другом.

Такие переменные, для которых используются численные значения, соответствующие постепенному изменению эмпирической значимости, относятся к порядковой шкале. Однако эмпирическая значимость этих переменных не зависит от разницы между соседними численными значениями.

Кроме частотного анализа переменные с порядковой шкалой допускают также вычисление определенных статистических характеристик, таких как медианы. В некоторых случаях возможно вычисление среднего значения.

Если должна быть установлена корреляция с другими переменными такого рода, для этой цели можно использовать коэффициент ранговой

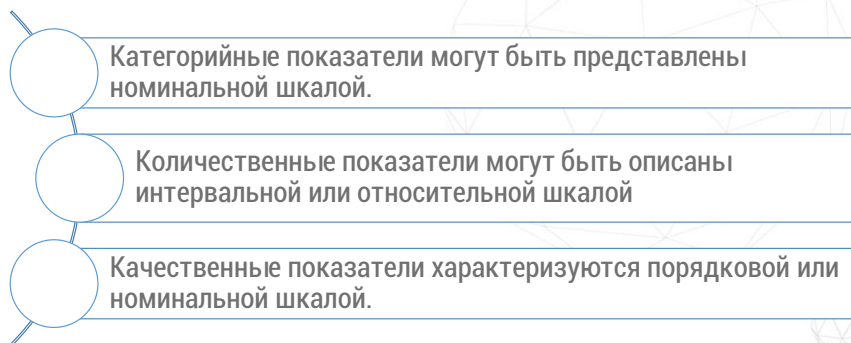
корреляции.

Переменные, у которых разность между двумя значениями имеет эмпирическую значимость, относятся к интервальной шкале. Они могут обрабатываться любыми статистическими методами без ограничений.

Все интервальные переменные, которые имеют абсолютную нулевую точку, относятся к шкале отношений. Поэтому переменные, относящиеся к интервальной шкале, как правило, имеют и шкалу отношений.

Таким образом, существуют четыре вида измерительных статистических шкал, на которых могут сравниваться численные значения.

КЛАССИФИКАЦИЯ ПОКАЗАТЕЛЕЙ ИЗМЕРЕНИЯ



С позиции возможности и точности измерения характеристики, субхарактеристики и атрибуты качества программных средств можно классифицировать в три группы показателей. Их особенности уточняются при их выборе.

К группе категориальных относятся показатели качества, которые характеризуются наибольшим разнообразием значений свойств программ и наборов

данных. Они охватывают весь спектр классов, назначений и функций программных средств. Эти свойства сравниваются только в пределах однотипных программных средств.

К этой группе относится функциональная пригодность, являющаяся важной характеристикой программных средств.

К группе количественных показателей качества относятся достоверно и объективно измеряемые численные характеристики программных средств. Значения этих характеристик обычно в наибольшей степени влияют на функциональную пригодность и метрики использования программных средств. Такими характеристиками являются надежность и эффективность комплексов программ.

К группе качественных относятся показатели качества, которые выбираются и оцениваются в значительной степени субъективно и экспертно.

Например, практичность имеет ряд качественных атрибутов. Они могут выбираться и оцениваться экспертно с учетом функционального назначения программного средства, а также надежности и ресурсной эффективности комплекса программ.

ПОНЯТИЕ АЛГОРИТМИЧЕСКОЙ СЛОЖНОСТИ

Пусть φ – произвольная частично-рекурсивная функция. Тогда мерой сложности последовательности x называют величину $K_\varphi =$

$$\begin{cases} \min l(p), \text{ если } \varphi(p) = x; \\ \infty, \text{ если } \forall p \in S, \text{ при которых } \varphi(p) \neq x, \end{cases}$$

где p – программа (код), по которой функция φ восстанавливает искомую последовательность x ; $l(p)$ – длина такой программы (количество двоичных разрядов, обеспечивающих полноценное описание последовательности; S – множество всех допустимых программ.



Понятие алгоритмической сложности является базовым в информатике. Обычное понятие сложности имеет много определений. Если посмотреть на это понятие с точки зрения описания объекта, то становится понятно: чем сложнее объект, тем более длинным текстом его придется описывать. При этом длина описания во многом зависит от способа изложения сути вопроса.

Если в качестве объектов рассматривать произвольные последовательности символов некоторого алфавита, то наиболее экономичным способом их описания будет алгоритмический. Это позволяет ввести следующее определение сложности последовательности.

За меру сложности произвольной последовательности символов некоторого фиксированного алфавита принимается длина – количество двоичных разрядов – самой короткой программы, генерирующей эту последовательность.

«При работе программы многообразие ее поведения и разнообразие связей между ее входными и результирующими данными в значительной степени определяются набором маршрутов, по которым исполняется программа.

Сложность программных модулей связана не столько с размером программы, определяющей

количество выполняемых команд, сколько с числом маршрутов ее исполнения и их сложностью. При создании качественной программы основную сложность ее разработки определяют маршруты возможной обработки данных, которые должны быть тщательно проверены.

Такую метрику сложности, связанную с анализом маршрутов, можно использовать для оценки трудоемкости тестирования и сопровождения модуля. А также для оценки потенциальной надежности его функционирования» [1].

СВОЙСТВА АЛГОРИТМИЧЕСКОЙ СЛОЖНОСТИ

Свойство 1. Сложность последовательности x не превосходит её длины:

$$K_\varphi \leq l(x). \quad (1)$$

Свойство 2. Сложность последовательности в общем случае неограниченно растёт с увеличением её длины:

$$\lim_{x \rightarrow \infty} K_\varphi = \infty. \quad (2)$$

Свойство 3. Доля последовательностей, допускающих сжатие до k бит, будет составлять

$$\frac{2^{N-k+1}}{2^N} \approx 2^{-k} \text{ при } N \gg 1. \quad (3)$$

Свойство 4. Верхняя оценка уровня алгоритмической сложности позволяет определить наибольшее значение алгоритмической сложности K_φ для определенного количества символов в алфавите s .

$$K_\varphi \leq s \cdot \log_2^2 s. \quad (4)$$



Из определения понятия алгоритмической сложности следуют четыре свойства, которым соответствуют формулы 1–4.

Первое свойство: сложность последовательности не превосходит ее длины.

Второе свойство: сложность последовательности в общем случае неограниченно растёт с увеличением ее длины.

Представить любую последовательность в виде программы можно только в том случае, если она содержит какие-либо закономерности: частотные, семантические или иные. Если же закономерности отсутствуют, то не может существовать и более короткой программы, чем сама эта последовательность.

Отсутствие закономерностей означает наличие случайности в чередовании букв алфавита, используемых в рассматриваемой последовательности.

Значит, можно утверждать, что все последовательности, для которых алгоритмическая сложность близка к длине последовательности, являются случайными.

Третье свойство: доля последовательностей, допускающих сжатие до k [ка] бит, будет приблизительно равна 2^{-k} [двум в степени минус ка].

Не существует алгоритма определения алгоритмической сложности для произвольной символьной последовательности. Однако существует способ хорошей оценки значения этой величины сверху.

Четвертое свойство: верхняя оценка уровня алгоритмической сложности позволяет определить наибольшее значение алгоритмической сложности для определенного количества символов в алфавите.