

Метрики качества программного проекта

Введение

**Процессы разработки, приобретения и внедрения
сложных систем**

Жесткий управленческий контроль характеристик

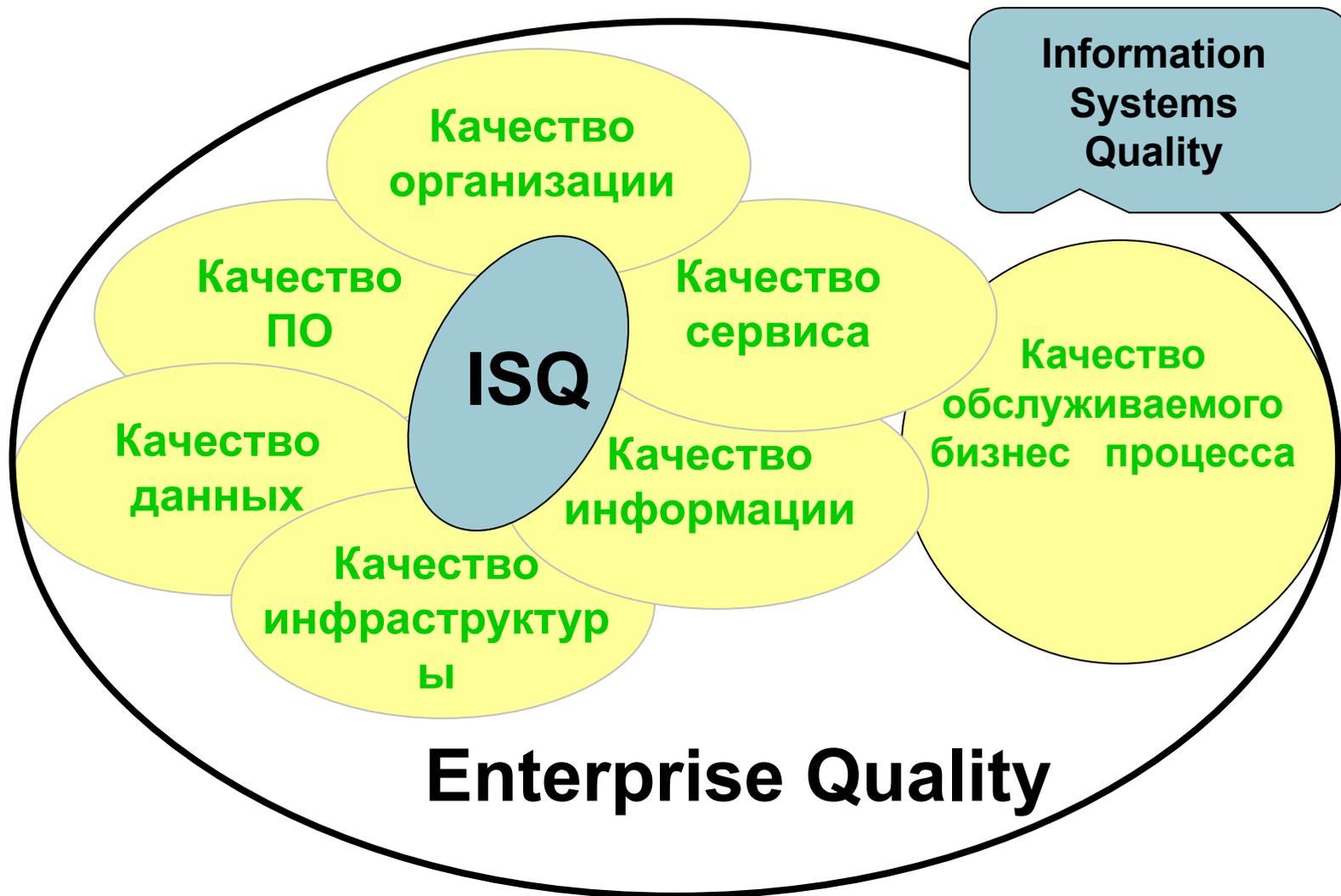
Качество

“You cannot control what you cannot measure”

Метрики качества ПО

- Понятие качества и его многомерность
- Характеристики качества и его цена
- Качество продукта, процесса, его организации
- Метрики качества
- Иерархия метрик
- Статистический анализ

Понятие качества и его многомерность



Понятие качества и его многомерность

- Качество инфраструктуры
- Качество ПО
- Качество данных
- Качество информации
- Качество организации
- Качество сервиса
- Качество процесса

Качество инфраструктуры

Качество аппаратного и поддерживающего программного обеспечения (например, качество операционных систем, компьютерных сетей и т.п.).

Качество ПО

**качество программного
обеспечения информационной
системы.**

Качество данных

**качество данных, использующихся информационной
системой на входе**

Качество информации

**качество информации, продуцируемое информационной
системой**

Качество организации

**качество менеджмента, включая качество
бюджетирования, планирования и календарного
контроля**

Качество сервиса

качество обучения, системной поддержки и т.п.

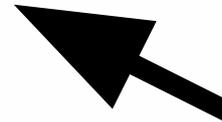
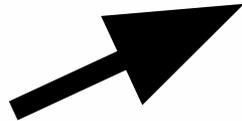
Качество процесса

качество обслуживаемого бизнес процесса

Понятие качества и его многомерность

Анализ

Сферы ответственности заинтересованных сторон



**in-process
stakeholder**

**end-of-process
stakeholders**

Управление качеством будет успешным, если под контролем находятся все измерения качества.

Характеристики качества

НАЧАЛЬНЫЙ ЭТАП ЖЦ

Разработчики

Заказчики

Цель проекта и детализация

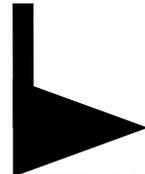
Набор функций

Характеристики качества

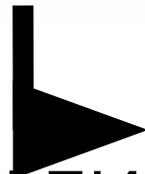


Характеристики качества

Отсутствие характеристики при договоре



Разный учёт или пропуск при испытаниях



КОНФЛИКТ!

Дерево характеристик качества

Исторически сначала были выделены ряд универсальных и неполных метрик на основе следующих шагов

1. Определение множества характеристик, которые, являясь важными для программного обеспечения, допускают несложное измерение и не перекрываются.

Дерево характеристик качества

2. Выделение кандидатов в метрики, которые измеряют степень удовлетворения указанным характеристикам.

Дерево характеристик качества

3. Исследование характеристик и связанных метрик, для определения корреляции, значимости, степени автоматизируемости.

Дерево характеристик качества

4. Исследование корреляции между метриками, степени перекрытия, зависимости и недостатков.

Дерево характеристик качества

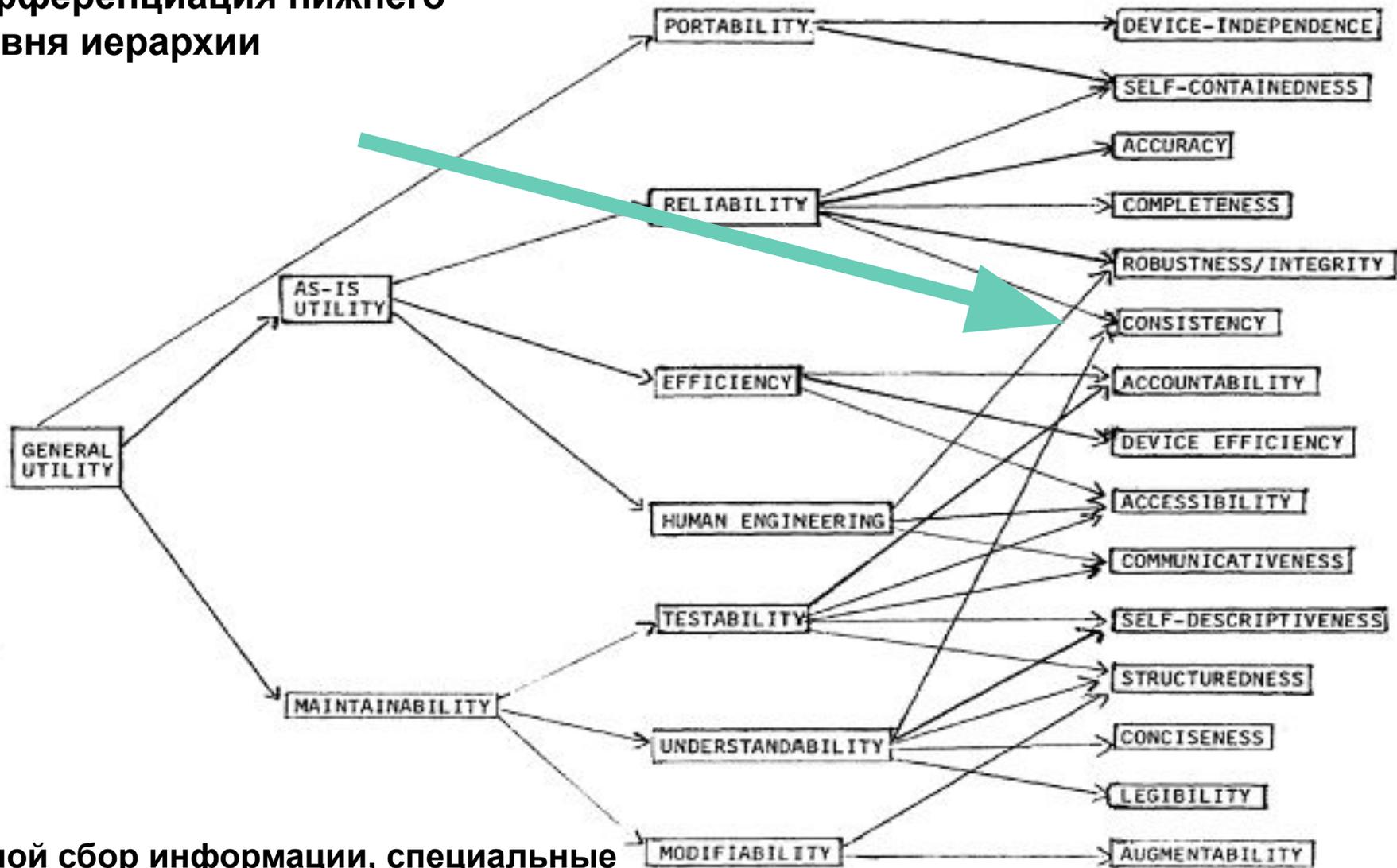
5. Рафинирование множества метрик в целом во множество метрик, которые в совокупности адекватно отражают качество программного обеспечения.

Дерево характеристик качества

6. Корректировка каждой метрики в итоговом множестве в контексте зафиксированных множеств характеристик и метрик.

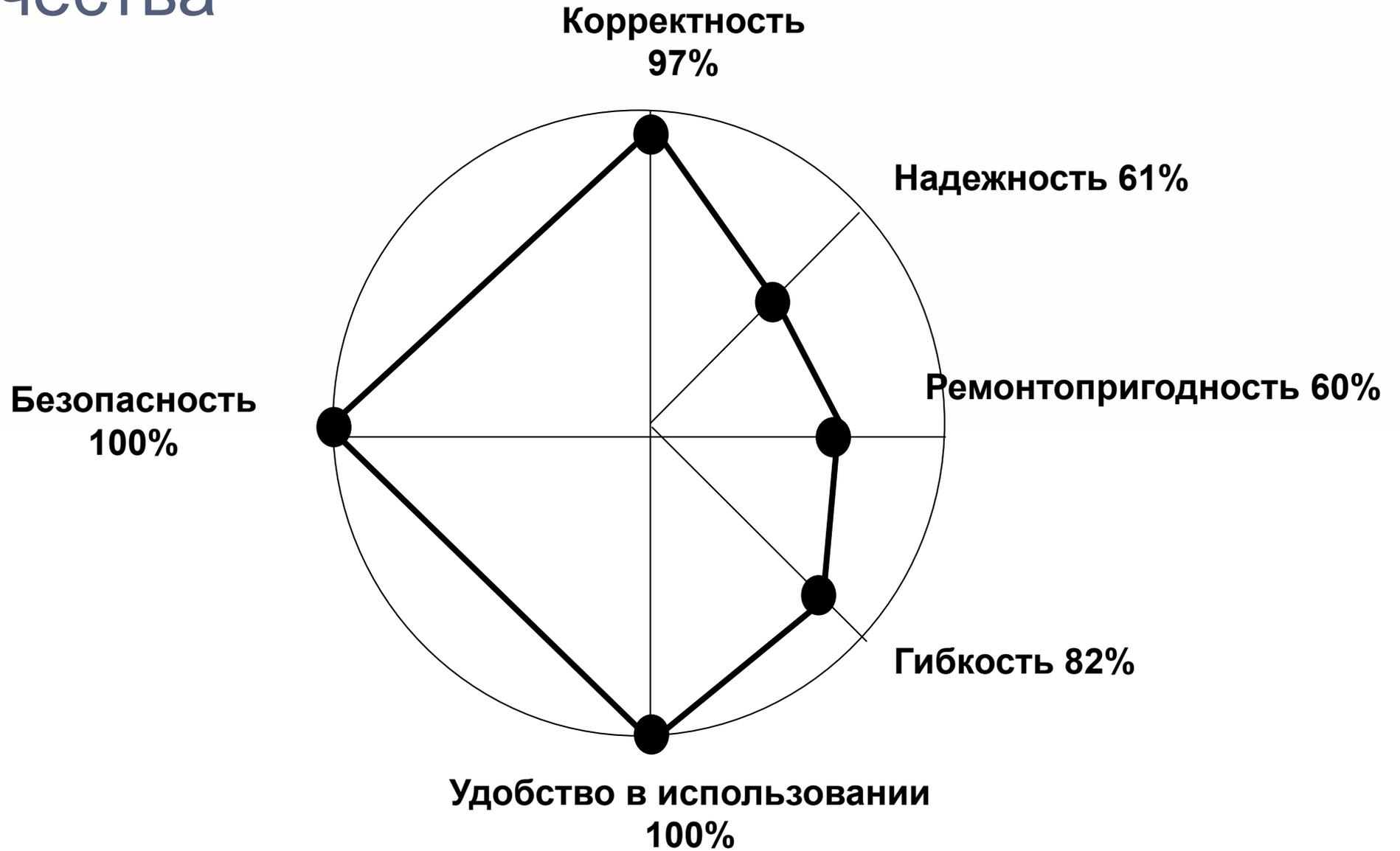
Дерево характеристик качества

Дифференциация нижнего уровня иерархии



Ручной сбор информации, специальные автоматизированные средства или экспертный способ

Пример графического изображения качества



Цена качества

Цена качества - стоимость в составе продукта, которая может быть сэкономлена, если все исполнители работают безупречно.



Стоимость работ на доработку

Цена качества

Цена качества

сумма,
затраченная на
достижение
качества
продукта

Согласованная

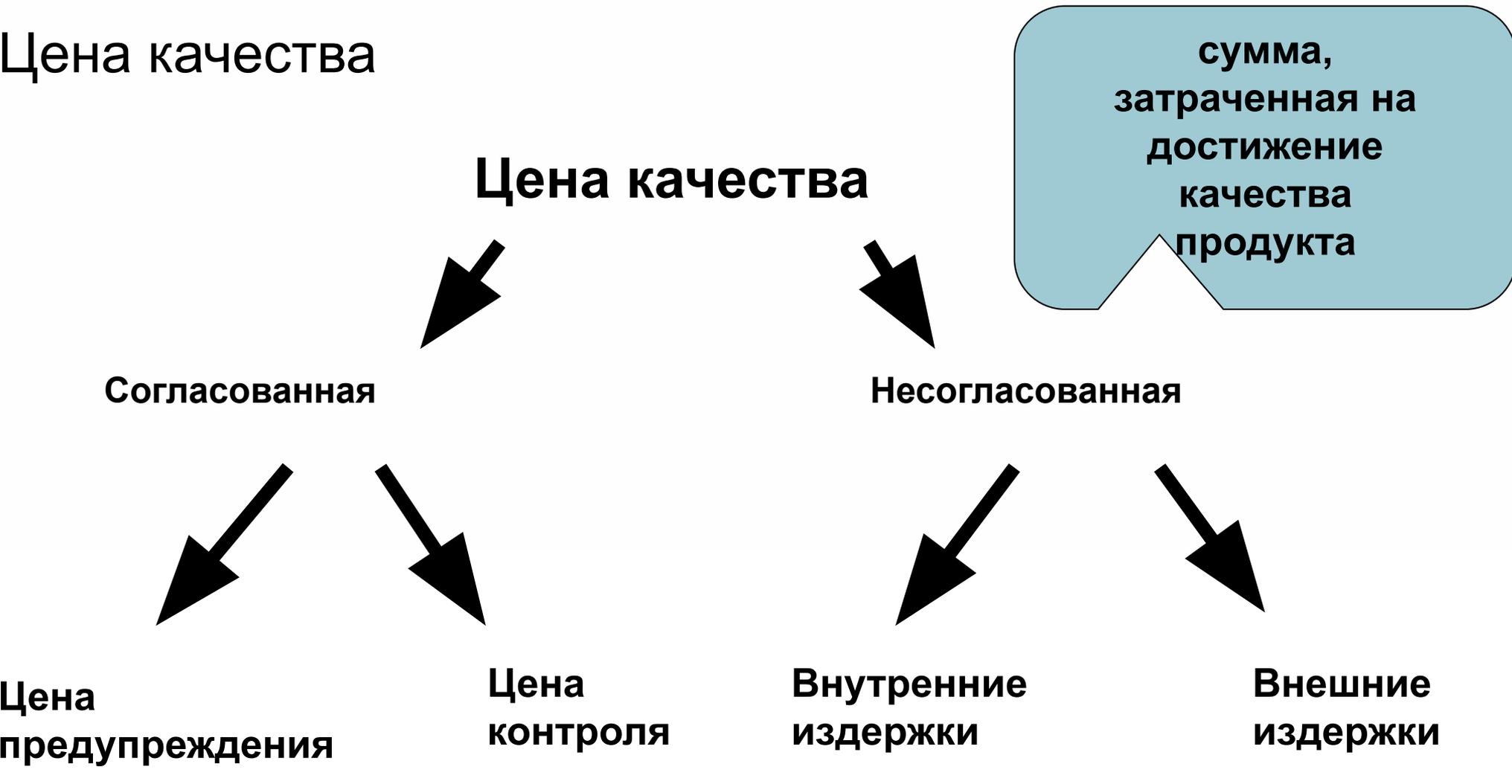
Несогласованная

Цена
предупреждения

Цена
контроля

Внутренние
издержки

Внешние
издержки



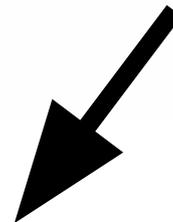
Цена качества

Цена качества

включает все
издержки
понесенные,
вследствие
выявления
недостатков,
возникновения
ошибок и выхода
из строя



Несогласованная



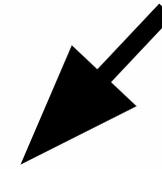
Внутренние
издержки



Внешние
издержки

Цена качества

Согласованная



Цена
предупреждения

Цена
контроля

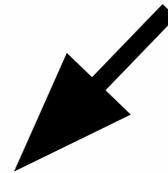


**Предупреждением дефектов
прежде, чем они произойдут**

**(обучение коллектива , переход на современные
технологии)**

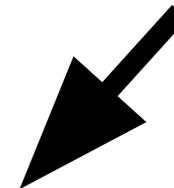
Цена качества

Согласованная



**Цена
предупреждения**

**Цена
контроля**



**Измерение, оценивание
или ревизия продукта**

Цена качества

Несогласованная

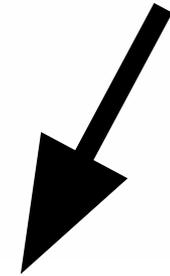


Внутренние
издержки

Внешние
издержки



**Издержки связанные с
проблемами, выявленными
до того, как продукт
отправлен заказчику**



**Затраты связанные с
ошибками,
проявившимися при
эксплуатации продукта**

Цена качества

- Совершенствование процесса разработки и внедрения программного обеспечения значительно уменьшают относительную несогласованную стоимость качества при сохранении согласованной стоимости на прежнем уровне
- Инвестиции в совершенствование процесса разработки ведут к значительному сокращению дефектов и дают высокий положительный экономический эффект

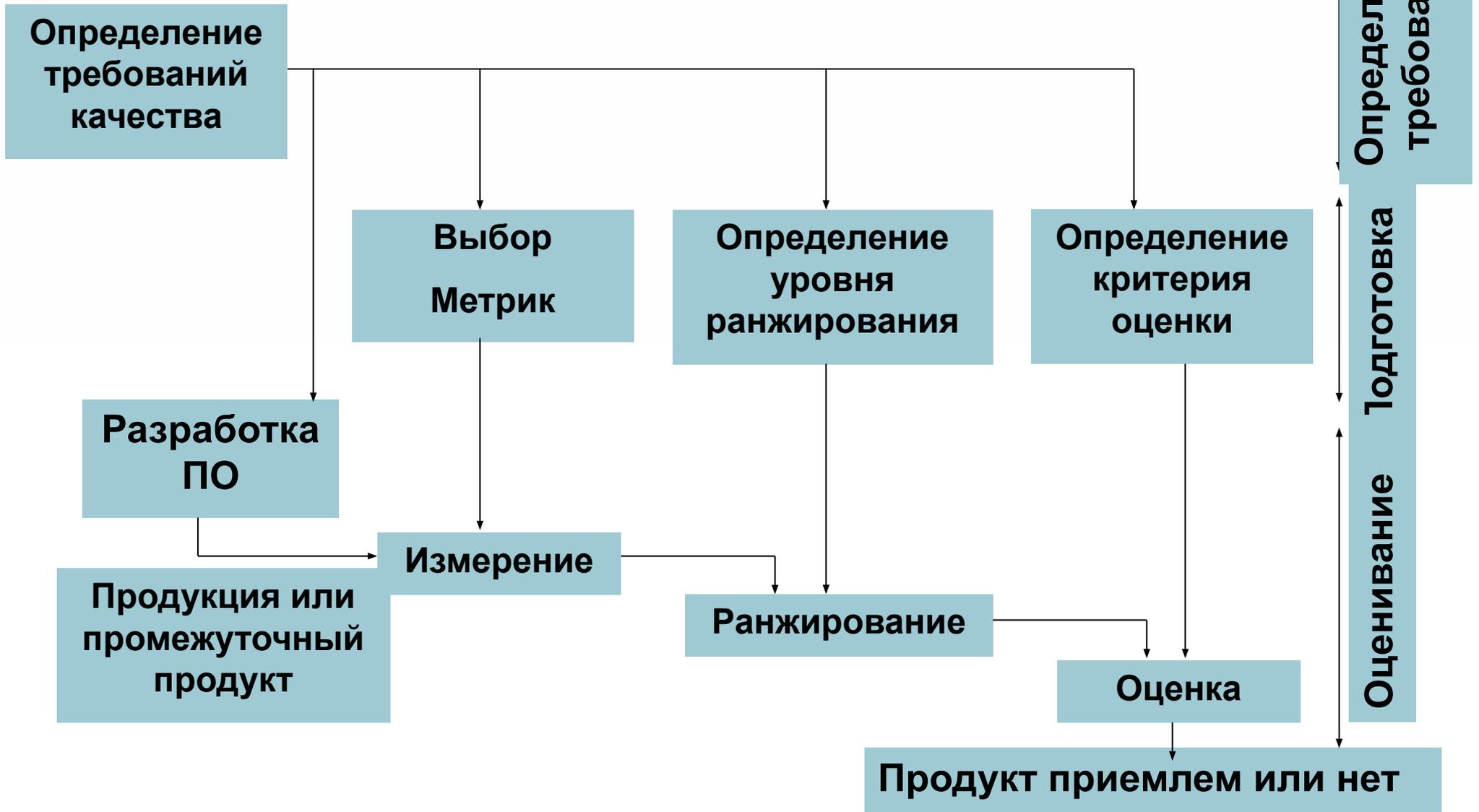
Качество продукта

Какие характеристики важнее?

- **Пользователь**
 - Применение ПО, его производительность, результаты использования.
- **Разработчик**
 - Требования пользователя к конечному продукту
 - Характеристики качества промежуточной продукции
- **Руководитель**
 - Общее качество
 - Коммерческие требования

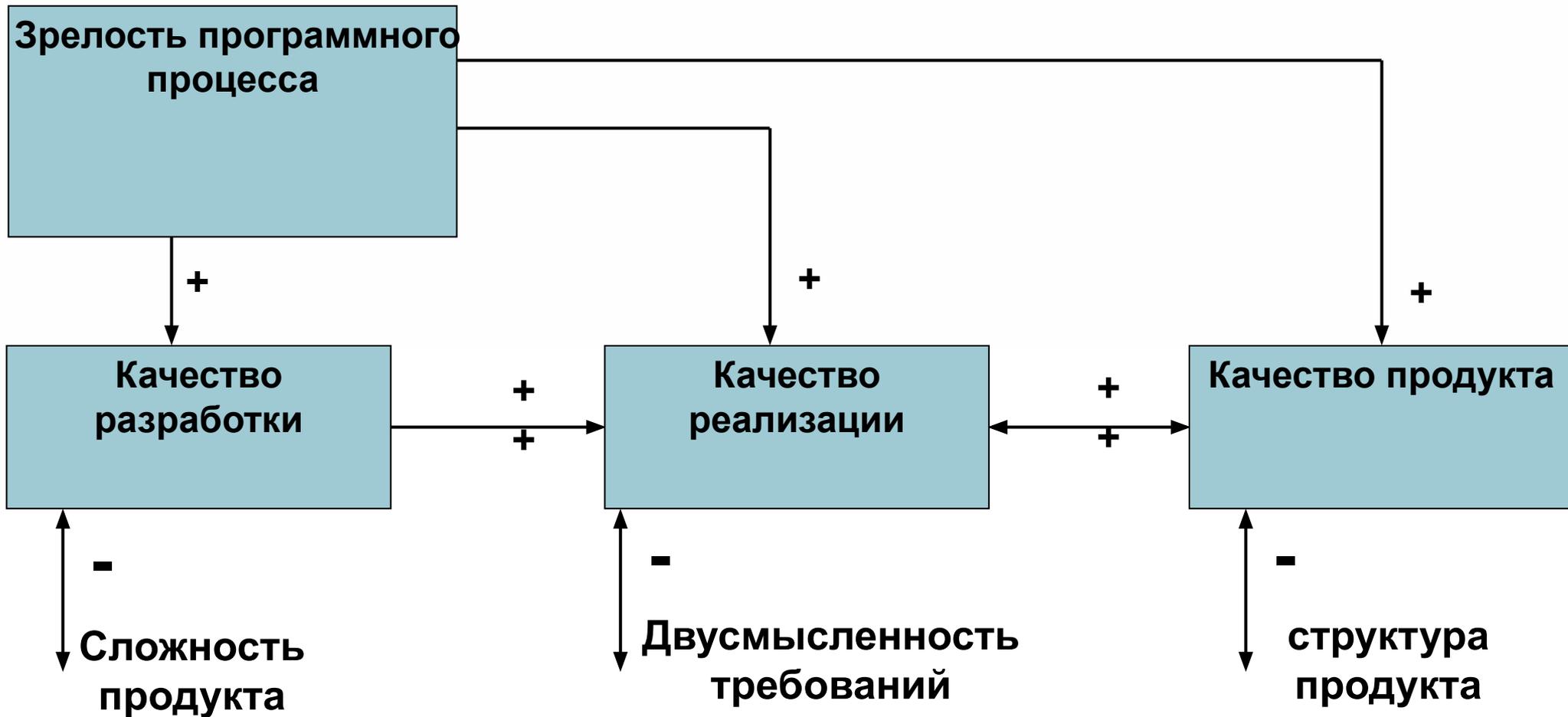
Качество продукта

Оценка качества программного продукта



Качество процесса, его организация

Модель качества процесса разработки



Качество процесса, его организация

Следствия принятой модели:

- **Качество накапливается в продукте при сложном производстве кумулятивным образом, причем, вклад в качество, осуществленный на ранних стадиях, имеет более сильное влияние на конечный продукт, чем на более поздних стадиях.**
- **Тестирование и измерение качества должно происходить на всех стадиях жизненного цикла.**

Качество процесса, его организация

Подход тотального управления качеством
(TQM – Total Quality Management)

Стандарты:

- **ISO 9001 -проектирование в процессе производства**
- **ISO 9000-3, формулирует требования модели качества ISO 9001 к организации процесса разработки программного обеспечения**

Качество процесса, его организация

- **Наличие процесса разработки программного обеспечения, удовлетворяющего высокому уровню качества, не гарантирует выпуска продукта высокого качества.**
- **Отсутствие информации о качестве процесса означает, что качество разрабатываемого продукта является непредсказуемым.**

Метрики качества

При выборе метрик главными показателями являются :

- **Адекватность метрик целям качества**
- **Прозрачность и четкость интерпретации**
- **Экономическая эффективность получения**

Метрики качества

Метрики менеджмента:

- **Цена (Cost)**
- **Время разработки
(Time-to-market)**
- **Среда разработки
(Software Engineering Environment)**
- **Использование системных ресурсов
(System Resource Utilization)**

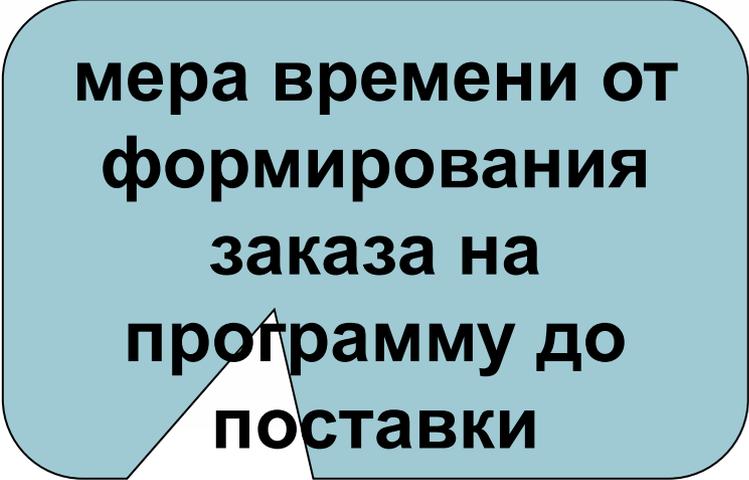


**расходы на
приобретение /
разработку**

Метрики качества

Метрики менеджмента:

- **Цена (Cost)**
- **Время разработки (Time-to-market)**
- **Среда разработки (Software Engineering Environment)**
- **Использование системных ресурсов (System Resource Utilization)**



**мера времени от
формирования
заказа на
программу до
поставки**

Метрики качества

Метрики менеджмента:

- **Цена (Cost)**
- **Время разработки (Time-to-market)**
- **Среда разработки (Software Engineering Environment)**
- **Использование системных ресурсов (System Resource Utilization)**

**процент целевых
компьютерных
ресурсов,
используемых
системой**

Метрики качества

Метрики менеджмента:

- **Цена (Cost)**
- **Время разработки (Time-to-market)**
- **Среда разработки (Software Engineering Environment)**
- **Использование системных ресурсов (System Resource Utilization)**

**мера
способности
производителя
разрабатывать
программное
обеспечение
ВЫСОКОГО
качества**

Метрики качества

Метрики требований:

- **Соответствие требованиям
(requirement conformance)**
- **Стабильность требований
(requirement stability)**

**дают
возможность
контролировать
спецификации,
изменение
требований, а
также степень их
удовлетворения**

Метрики качества

Метрики качества:

- **Адаптируемость (adaptibility)**
- **Сложность интерфейсов и интеграции (complexity of interfaces and integration)**
- **Тестовое покрытие (test coverage)**
- **Надежность (reliability)**
- **Профили ошибок (fault profiles)**
- **Степень удовлетворения потребностей заказчика (customer satisfaction)**



**мера гибкости
системы**

Метрики качества

Метрики качества:

- Адаптируемость (adaptability)
- Сложность интерфейсов и интеграции (complexity of interfaces and integration)
- Тестовое покрытие (test coverage)
- Надежность (reliability)
- Профили ошибок (fault profiles)
- Степень удовлетворения потребностей заказчика (customer satisfaction)

**метрика,
измеряющая
степень
сложности
интерфейса**

или

**дополнительно
го**

**программирова
ния требуемого
для интеграции**

**компоненты в
систему**

Метрики качества

Метрики качества:

- Адаптируемость (adaptability)
- Сложность интерфейсов и интеграции (complexity of interfaces and integration)
- Тестовое покрытие (test coverage)
- Надежность (reliability)
- Профили ошибок (fault profiles)
- Степень удовлетворения потребностей заказчика (customer satisfaction)

**степень полноты
различных типов
тестирования**

Метрики качества

Метрики качества:

- **Адаптируемость (adaptability)**
- **Сложность интерфейсов и интеграции (complexity of interfaces and integration)**
- **Тестовое покрытие (test coverage)**
- **Надежность (reliability)**
- **Профили ошибок (fault profiles)**
- **Степень удовлетворения потребностей заказчика (customer satisfaction)**

**вероятность
работы системы
без отказов**

Метрики качества

Метрики качества:

- Адаптируемость (adaptability)
- Сложность интерфейсов и интеграции (complexity of interfaces and integration)
- Тестовое покрытие (test coverage)
- Надежность (reliability)
- Профили ошибок (fault profiles)
- Степень удовлетворения потребностей заказчика (customer satisfaction)

**кумулятивное
число
обнаруженных
ошибок**

Метрики качества

Метрики качества:

- **Адаптируемость (adaptability)**
- **Сложность интерфейсов и интеграции (complexity of interfaces and integration)**
- **Тестовое покрытие (test coverage)**
- **Надежность (reliability)**
- **Профили ошибок (fault profiles)**
- **Степень удовлетворения потребностей заказчика (customer satisfaction)**

**степень
соответствия
программного
обеспечения
ожиданиям и
требованиям
заказчика**

АДАПТИРУЕМОСТЬ

- «Adaptability» - мера гибкости системы, оценивает способность системы адаптироваться к изменениям требований либо перепроектированием системы, либо интеграцией приложений.

Complexity of interfaces and integration

- «Complexity of interfaces and integration» - метрика, измеряющая степень сложности интерфейса или дополнительного программирования требуемого для интеграции компоненты в систему, которые требуются для тестирования, отладки и сопровождения, компенсирующего потерю качества.

Покрытие тестами

- Метрики «test coverage» указывают степень полноты различных типов тестирования.

Надежность

- «Reliability»- метрика, оценивающая вероятность работы системы без отказов. Данная метрика может быть получена в рамках традиционного подхода.

Профили ошибок

- «Fault profiles» - метрика, измеряющая кумулятивное число обнаруженных ошибок.

Удовлетворенность пользователя

- «Customer satisfaction» - метрика, оценивающая степень соответствия программного обеспечения ожиданиям и требованиям заказчика. Данная метрика может быть оценена перед поставкой на этапе опытной эксплуатации на основе прогнозирующих параметров.

Качество программного кода

- Единственным доступным механизмом определения «ожиданий заказчика» являются требования (**software requirement specifications**). Требования Технического задания определяют функции программного обеспечения и нефункциональные требования, такие как производительность, надежность и т.п. Нетехнические требования, такие как цена, сроки поставки утверждаются в контрактных документах.

Метрики качества, выводимые из требований

- чрезвычайно важны для анализа качества продукта
- создаются на начальных этапах разработки, когда степень неопределенности и риск, связанный с разработкой и внедрением новых программных продуктов велики.
- Для эволюционного процесса разработки должны быть приняты к рассмотрению метрики качества программ, используемые в процессе реализации циклов разработки.

Метрики качества, выводимые из требований (2)

- Гибкость (flexability), которая аккумулирует ряд свойств:
 - Модульность (Modularity)
 - Изменяемость (Changeability)
 - Сопровождаемость (Maintainability)

Метрики качества, выводимые из требований (3)

- Адаптивность (adaptability), которая подразумевает:
 - Настраиваемость (customizability)
 - Переносимость (Portability)
 - Способность к взаимодействию (Interoperability)

Метрики качества, выводимые из требований (4)

- Оценка качества по приведенным выше метрикам, как правило, не проводится.
- Однако уже через короткое время обычно происходит снижение уровня качества программного обеспечения, связанное с расхождением текущих требований заказчика к системе.
- Обычно причиной этого является высокая стоимость исправлений или изменений в программной системе.

Исправления

- Исправления программного обеспечения может быть инициировано по следующим причинам:
 - исправление программы с недостаточным уровнем качества (bug fixing),
 - изменение программы для повышения уровня качества (enhancement),
 - изменение программы для удовлетворения изменения в требованиях.

Качество технического проекта

- Измерение качества проектирования является очень важной составляющей частью в процессе обеспечения качества программного продукта.
- Особую важность это приобретает при объектно–ориентированном проектировании программных систем.
- Объектно–ориентированная разработка программ вводит новые факторы качества, связанные с повторным использованием ранее выполненных работ и выполнением модификаций.

Уровни повторного использования

- повторное использование и модификация может быть произведена на нескольких уровнях:
 - уровень исходного кода,
 - физический дизайн,
 - логический дизайн,
 - требования
 - ~~в рамках системы,~~
 - в рамках проекта,
 - в рамках компании.
-

Процесс модификации

- Процесс модификации программной системы включает три главных фазы:
 - реструктуризация (создание логически эквивалентной системы);
 - обратный инжиниринг (reverse engineering, анализ системы с целью выделения системных компонент, их функций и взаимосвязей, включая спецификации верхнего уровня системы)
 - прямой инжиниринг (forward engineering, разработка системы от спецификаций до кодирования и внедрения).

Процесс модификации (2)

- В настоящее время выполняется большое количество проектов, связанных с переводом использующихся унаследованных систем в объектно-ориентированные системы для увеличения их срока жизни и функционального развития.
- Критичным в таких проектах является контроль качества, на который ложится задача гарантировать обеспечение роста или неуменьшения уровня потребительских характеристик программных систем

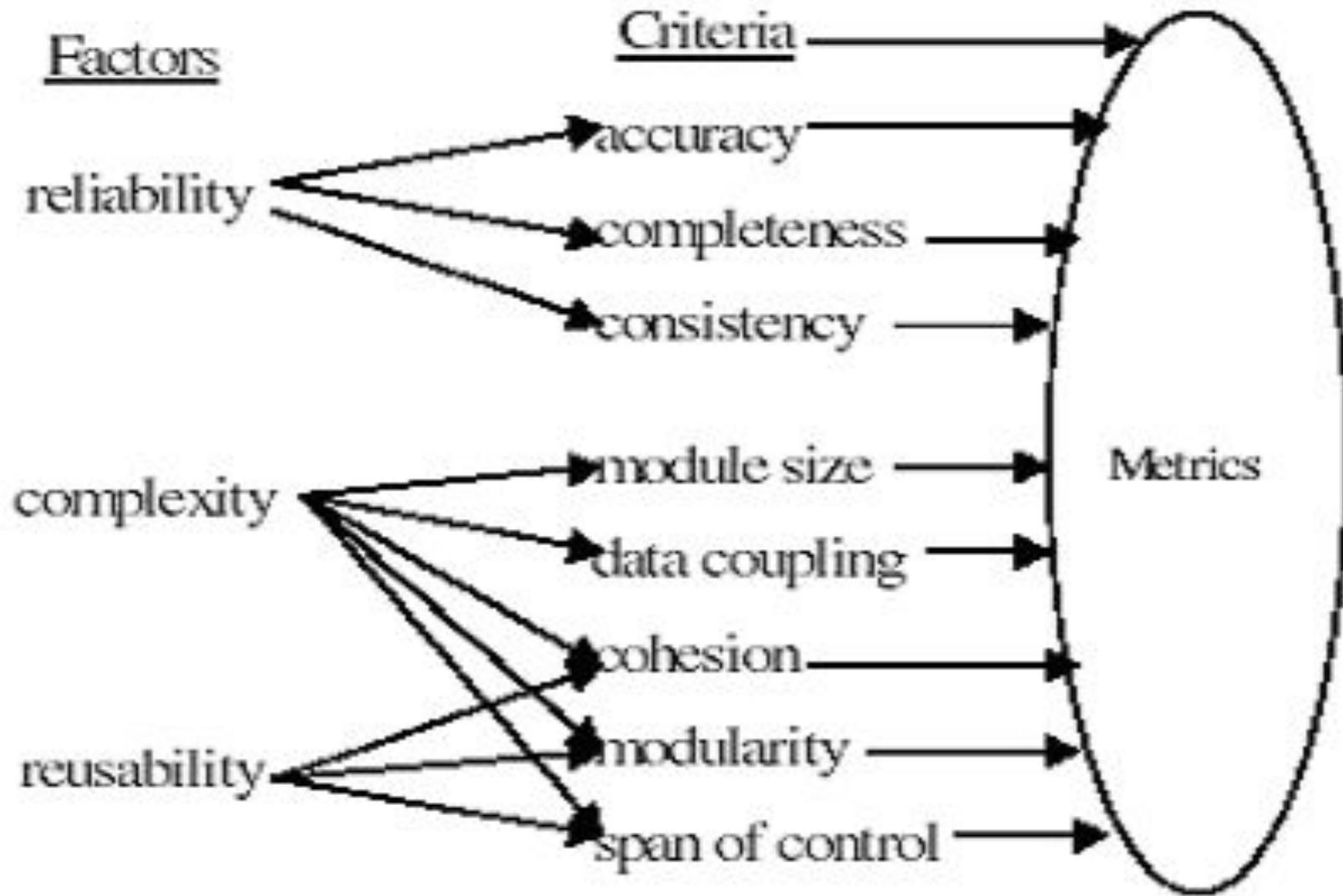
Процесс модификации (3)

- В процессе инжиниринга программных систем в дополнение к классическим метрикам должны быть включены в число наиболее важных такие метрики качества объектно-ориентированного дизайна как: надежность (reliability), сложность (complexity) и возможность повторного использования (reusability).
- Измерение качества проектирования является принципиально важной частью процесса разработки, поскольку, как показывает статистика, стоимость ошибки проектирования в среднем на два порядка выше стоимости ошибки кодирования

Модель факторов качества

- При измерении факторов качества широко используется модель: фактор - критерий - метрика (factor a criteria a measurement). Установка связи фактор а критерий требует анализа составляющих факторов качества.

фактор - критерий - метрика



Модель факторов качества (2)

- конкретные метрики выводятся в соответствии с особенностями проекта из критериев качества:
 - accuracy (точность),
 - completeness (полнота),
 - consistency (согласованность),
 - module size (размер модулей),
 - data coupling (связь модулей по данным),
 - cohesion (связность),
 - modularity (модульность),
 - span of control (норма управляемости).

Измерение качества на основе сопровождения продукта

- Одной из главных путей способов повышения качества является путь анализа практического опыта использования данного продукта или процесса и использования полученных данных для его совершенствования.
- Речь идет о так называемой парадигме QIP – Quality Improvement Paradigm

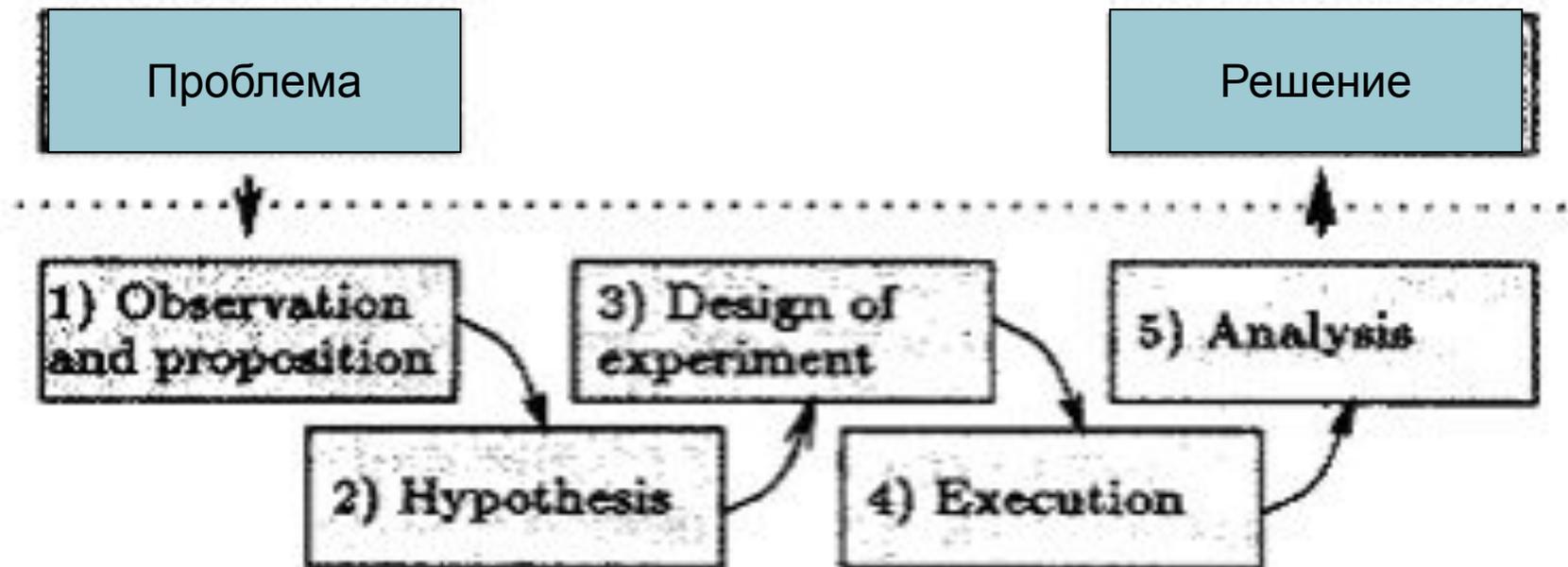
Quality Improvement Paradigm

- Парадигма совершенствования качества дает систематический подход к организации процессов сбора практического опыта, систематизации накопленной информации и подготовке результирующих выводов для улучшения качества.
- Одним из главных процессов в дисциплине разработки программ, где этот опыт должен накапливаться, является процесс сопровождения.
- Потеря значимой информации на этапе сопровождения или несистематический недостаточно формальный процесс сбора информации неизбежно дает недостаточно объективную или искаженную основу для принятия технических решений о путях развития и совершенствования программной системы

Quality Improvement Paradigm (2)

- Наиболее применимым способом реализации парадигмы улучшения качества является иерархический подход формирования информации: от целей к вытекающим из них вопросам, от вопросов - к метрикам.
- Другой частью метода является стандартный научный подход к выработке решений: формулирование проблемы, сбор данных и планирование, формирование гипотез, подготовка к проверке, исполнение проверки, анализ результатов, формулировка решения

Вывод решений



Quality Improvement Paradigm (3)

- Итогом процесса является накопление знаний о качестве.
- Для больших систем, находящихся длительное время в эксплуатации, в особенности для распределенных систем необходимым условием здесь является регулярность структур базы данных о качестве.
- Одним из наиболее эффективных решений является организация информации в виде так называемых фреймов качества

Фрейм качества

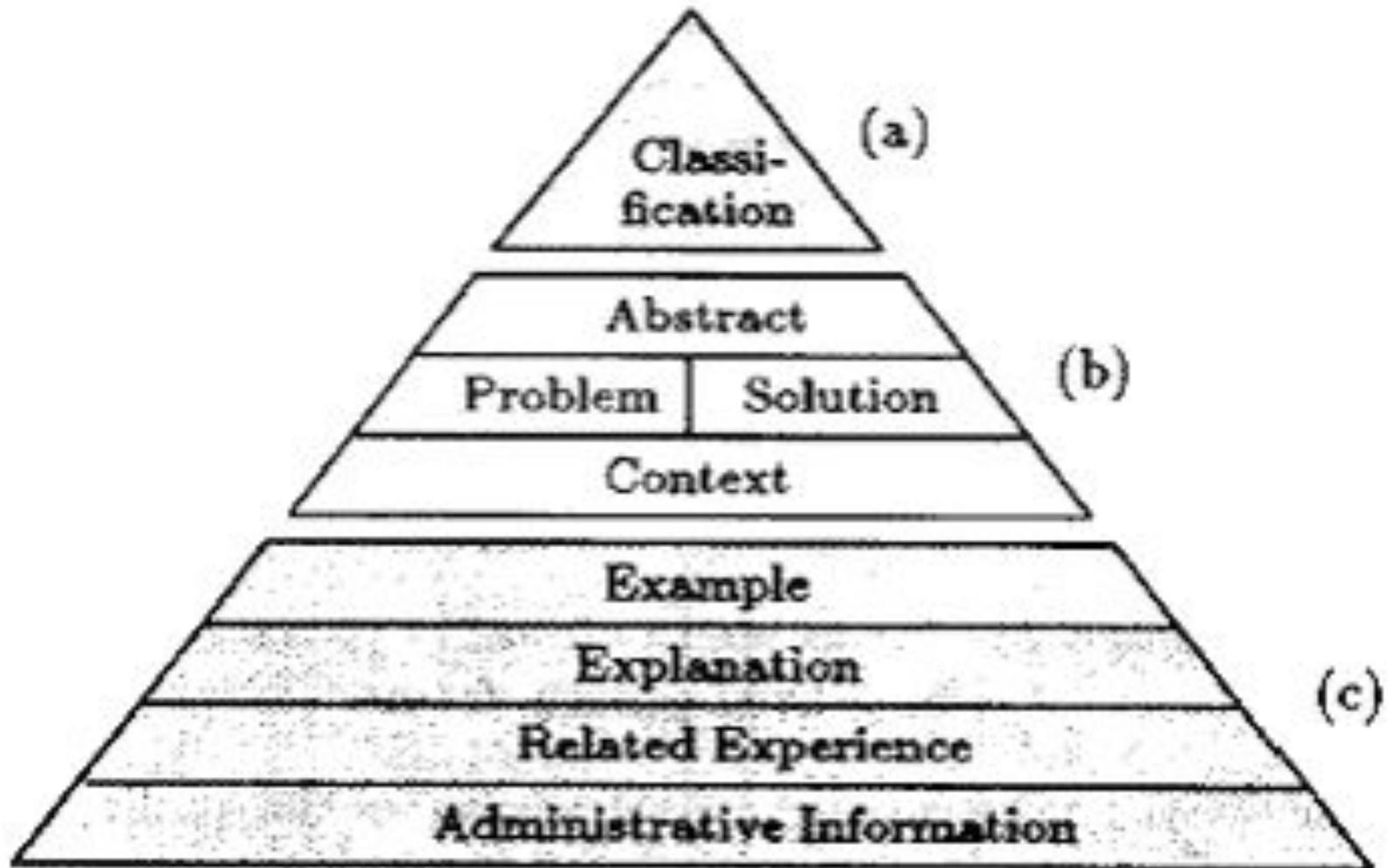
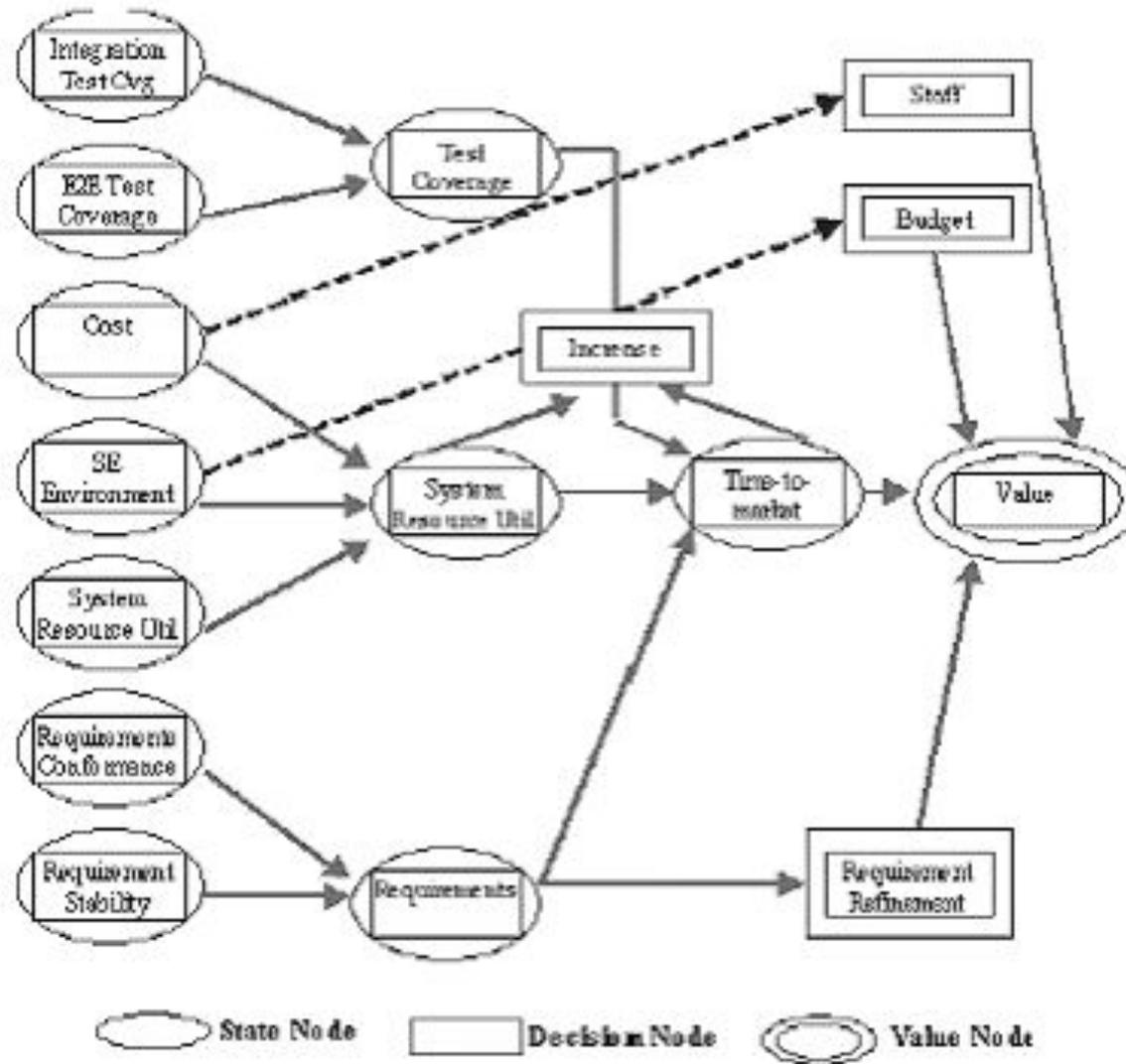


Диаграмма влияния для подмножества метрик качества



Интегральная оценка качества

- Интегральная оценка качества производится на основе анализа взаимовлияния различных параметров характеризующих качество продукта
- Традиционные метрики не учитывают качество обрабатываемых данных и взаимодействие данных с потоком вычислений.

Интегральная оценка качества (2)

- Для обеспечения полноты измерения качества требуется на ранних стадиях проекта на основе анализа целей проекта, области применения, ограничений и характеристик разработать
 - проектно-ориентированные (design-oriented)
 - или структурные метрики (structural metrics) качества

Интегральная оценка качества (3)

- Термин «проектно-ориентированный» в данном контексте означает, что метрики разрабатываются в виде стандарта качества проекта на его ранних стадиях и представляют собой правила или нормы (guideline), которым должен удовлетворять промежуточный или конечный продукт.
- Термин структурный означает, что метрики разрабатываются структурным методом сверху - вниз (top – down) для обеспечения целостности и полноты.

методология создания метрик качества

- Измерение качества в соответствии с данными метриками состоит в вычислении отклонения фактических характеристик продукта от норм и правил.
- Методология создания метрик качества указанным способом утверждена в стандарте IEEE 1061

методология создания метрик качества (2)

- Первый шаг (верхний уровень иерархии):
Определение нетехнического уровня (то есть уровня предназначенного для менеджеров, пользователей, заказчика):
 - Формирование требований качества
 - Выбор свойств (атрибутов), установка приоритетов и связи с требованиями
 - Присвоение атрибутов факторам качества, которые отражают представление заказчика на качество.
 - Установка измерений для факторов качества.
Определение допустимых коридоров для величин качества.

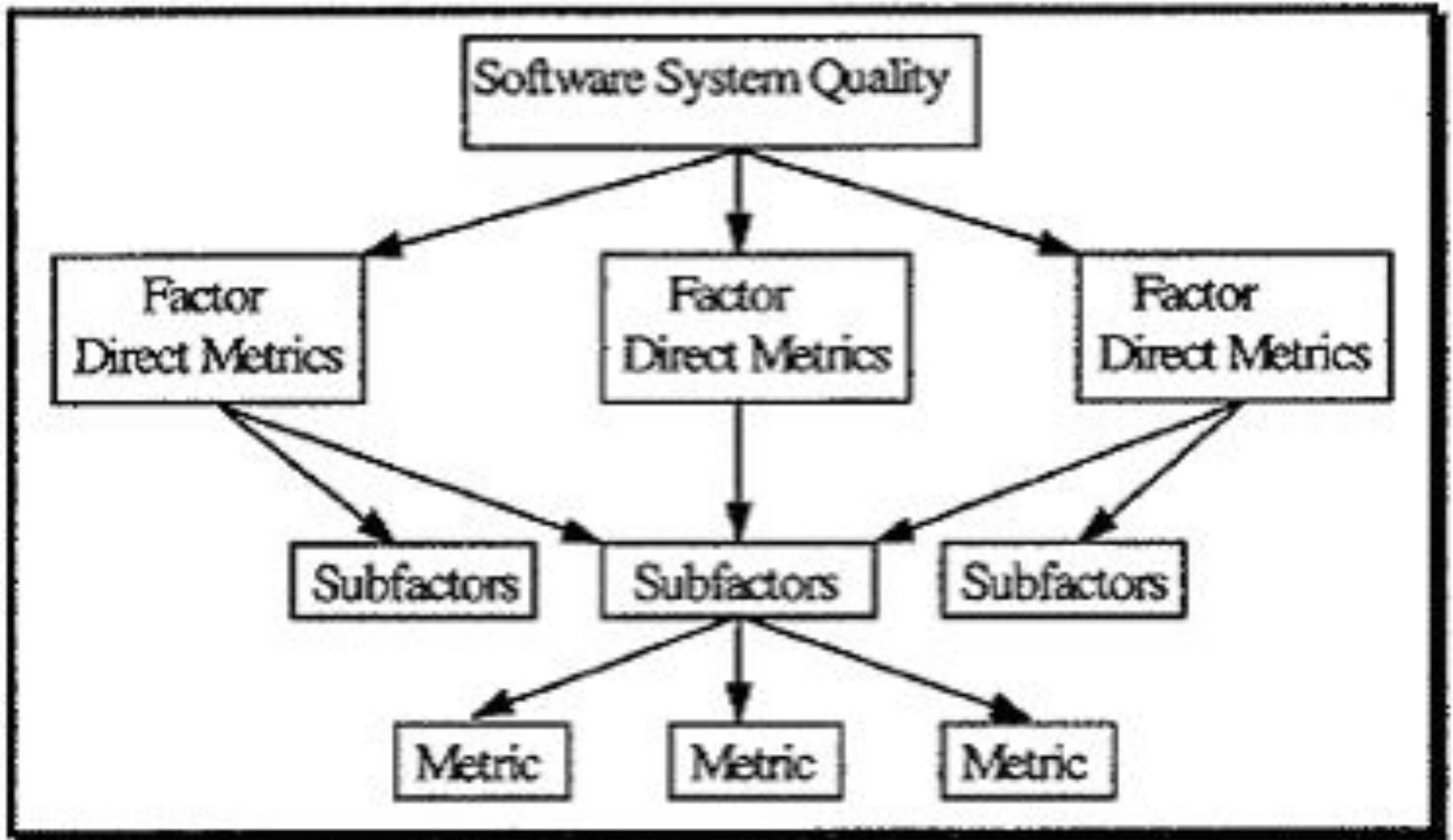
методология создания метрик качества (3)

- Второй шаг (средний уровень иерархии):
 - Определение технического уровня (то есть уровня предназначенного для аналитиков, конструкторов, разработчиков):
 - Осуществление декомпозиции факторов качества в измеряемые характеристики программного обеспечения, определяемые как суб-факторы.

методология создания метрик качества (3)

- Третий шаг (нижний уровень иерархии):
 - Декомпозиция суб-факторов в метрики, которые могут быть применены непосредственно к программному продукту или процессу разработки. Данные метрики служат как не прямые меры (индикаторы) прямых измерений факторов качества на верхних уровнях иерархии. Иными словами это уровень разработанных правил и норм, которым должен удовлетворять продукт или процесс с тем, чтобы были выполнены факторы качества

Схема вывода метрик качества



Факторы качества программной системы (пример):

- Переносимость (portability) – усилия, требуемые для переноса системы с одной платформы на другую.
- Надежность (reliability) – ожидаемая степень корректного выполнения системой требуемых функций
- Тестируемость (testability) – усилия, требуемые для тестирования функций программы

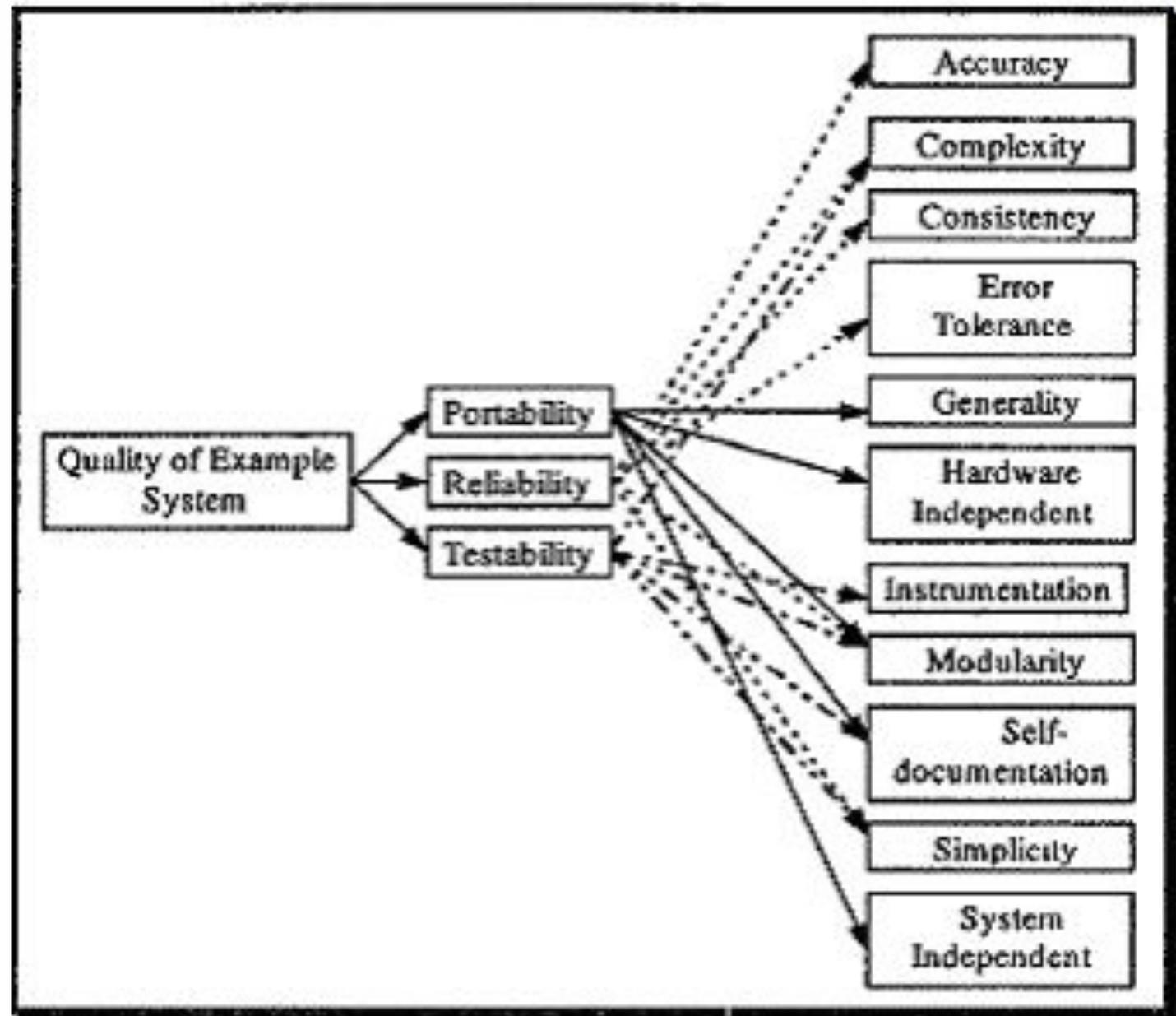
Факторы качества программной системы (пример): (2)

- Прямые измерения факторов качества:
 - Переносимость (portability) – трудоемкость - количество чел.-час., требуемое для переноса программы с платформы X на платформу Y. Допустимый порог: 1 чел.-час. на 1К строк исходного кода .
 - Надежность (reliability) – среднее время наработки на отказ. Допустимый порог: 120 операционных дней.
 - Тестируемость (testability) – трудоемкость – количество чел.-час., требуемое для полного тестирования 90% всех модулей. Допустимый порог 10 чел.-час. на 1К строк исходного кода.

Факторы качества программной системы (пример): (3)

- Следующий шаг – проведение декомпозиции приведенных факторов на суб-факторы

Пример структуры факторов качества



Примеры требуемых определений по Артуру (Arthur L.A.)

- Точность (accuracy) – правильность вычислений и контроля;
- Сложность (complexity) – трудность разработки и модификации;
- Совместимость (consistency) – использование унифицированной технологии проектирования и разработки на протяжении всего цикла разработки;
- Устойчивость к ошибкам (error tolerance) – степень ущерба от возникающих ошибок;
- Универсальность (generality) – широта потенциального использования;
- Аппаратная независимость (hardware independence) – степень применимости программы на другом аппаратном обеспечении;

Примеры требуемых определений по Артуру (Arthur L.A.) (2)

- Оснащенность средствами контроля (instrumentation) – степень контроля программы собственного выполнения и идентификации возникающих ошибок;
- Модульность (modularity) – степень функциональной независимости программных компонент;
- Удобочитаемость (readability) – уровень смыслового наполнения комментирования, соответствие стандартам кодирования и именования;
- Простота (simplicity) – легкость понимания программы;
- Системная независимость (system independence) степень независимости от нестандартных характеристик системного окружения и ограничений.

Уточнение метрик

- Сложность (complexity): – использование метрики цикломатической сложности Мак Каба (McCabe's cyclomatic complexity metric [60])
- Устойчивость к ошибкам (error tolerance): использование правила (нормы): «все модули должны содержать обработчики predetermined исключительных ситуаций. Все обрабатываемые исключительные ситуации должны быть либо распространены на внешний уровень, либо разрешены».

Ключевые моменты

- Опыт управления качеством показывает, что финансовые затраты, произведенные для улучшения качества продукта, являются безусловно целесообразными и дают в итоге высокий экономический эффект.
- Причина, по которой многие организации воздерживаются от таких расходов, состоит, прежде всего, в трудностях связанных с планированием и оценкой результатов повышения качества.
- Частой является ситуация, когда реализуется решение о повышении качества, основываясь на неформальных, интуитивных способах оценки качества. Это неизбежно ведет к неэффективному расходованию ресурсов и фактически увеличивает реальную цену качества.

Дерево характеристик качества

Не существует единственной метрики



Спектр проектно-зависимых метрик



Метрики качества - изначально неочевидная категория

