

Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

В. В. Бахтизин, Л. А. Глухова

***МЕТРОЛОГИЯ, СТАНДАРТИЗАЦИЯ
И СЕРТИФИКАЦИЯ
В ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЯХ***

*Допущено Министерством образования Республики Беларусь
в качестве учебного пособия
для студентов учреждений высшего образования
по направлению образования
«Информатика и вычислительная техника»
и по специальностям
«Автоматизированные системы обработки информации»,
«Информационные технологии и управление в технических системах»*

В двух частях
Часть 2

Минск БГУИР 2016

УДК [006.91+006.1]:004(075.8)
ББК 30.10я73+30ця73+32.973.26-018.2я73
Б30

Рецензенты:

кафедра информационных систем управления
Белорусского государственного университета
(протокол №4 от 22.10.2015);

заведующий кафедрой системного программирования
и компьютерной безопасности
учреждения образования «Гродненский государственный университет
имени Янки Купалы»,
кандидат технических наук, доцент А. М. Кадан

Бахтизин, В. В.

Б30 Метрология, стандартизация и сертификация в информационных технологиях : учеб. пособие. В 2 ч. Ч. 2 / В. В. Бахтизин, Л. А. Глухова. – Минск : БГУИР, 2016. – С. 141–343.
ISBN 978-985-543-260-0 (ч. 2).

Приведены общие сведения о правилах разработки и обозначения международных, межгосударственных, региональных и национальных стандартов. Рассмотрены национальные и международные стандарты в области жизненного цикла ПС и систем. Рассмотрены стандарты в области оценки качества ПС и систем, действующие на территории Республики Беларусь и за рубежом. Приведены основы метрологии ПС и систем. Даны сведения о Национальной системе оценки соответствия Республики Беларусь. Рассмотрены вопросы сертификации ПС.

Издано в двух частях. Часть 2 издания включает части III – V учебного пособия, литературу и основное содержание по двум частям.

Табл. : 19. Ил. : 44. Библиогр. : 82 назв.

УДК [006.91+006.1]:004(075.8)
ББК 30.10я73+30ця73+32.973.26-018.2я73

ISBN 978-985-543-260-0 (ч. 2)
ISBN 978-985-543-205-1

© Бахтизин В. В., Глухова Л. А., 2016
© УО «Белорусский государственный университет информатики и радиоэлектроники», 2016

КРАТКОЕ СОДЕРЖАНИЕ

Часть III. Оценка качества программных средств и систем

9. Качество программных средств и систем. Основные понятия и определения	144
10. Стандартизация качества программных средств в Республике Беларусь	149
11. История развития стандартизации качества программных средств за рубежом.....	176
12. Качество систем и программных средств в серии стандартов SQuaRE	192
Заключение по части III.....	252
Вопросы для самопроверки по части III	254

Часть IV. Метрология программных средств и систем

13. Общие сведения о метрологии.....	258
14. Метрики качества программных средств по стандартам ISO/IEC TR 9126-2-4:2003-2004.....	264
15. Метрология качества систем и программных средств в стандартах серии SQuaRE	279
16. Метрология сложности программных средств	292
Заключение по части IV	310
Вопросы для самопроверки по части IV.....	312

Часть V. Сертификация программного обеспечения

17. Сертификация программных средств.....	316
Заключение по части V.....	330
Вопросы для самопроверки по части V	331
Литература	332
Основное содержание.....	338

ЧАСТЬ III

ОЦЕНКА КАЧЕСТВА ПРОГРАММНЫХ СРЕДСТВ И СИСТЕМ

Библиотека БУМР

9. КАЧЕСТВО ПРОГРАММНЫХ СРЕДСТВ И СИСТЕМ. ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ

В настоящее время компьютеры находят все более широкое применение во всех сферах жизнедеятельности человека. От правильности их работы во многом зависят результаты труда и безопасность тысяч людей. В этой связи в последние годы во всем мире особое внимание уделяется стандартизации управления качеством ПС и систем на протяжении всего их ЖЦ.

С учетом этого во второй редакции стандартов *ISO/IEC 15288:2008* и *ISO/IEC 12207:2008* (см. подразд. 7.2 и 8.2) существенно расширен перечень процессов жизненного цикла, связанных с управлением качеством, по сравнению с первыми редакциями данных стандартов. Например, в *ISO/IEC 12207:1995* (*СТБ ИСО/МЭК 12207–2003*, *ГОСТ ИСО/МЭК 12207–2002*, см. подразд. 6.2) к таким процессам относятся процессы обеспечения качества, верификации, аттестации, совместного анализа и аудита из подгруппы вспомогательных процессов (на рис. 6.3 данные процессы выделены серым прямоугольником). В *ISO/IEC 12207:2008* дополнительно к вышеназванным процессам введен ряд системных процессов, основной направленностью которых является обеспечение заданного уровня качества разрабатываемых продуктов и процессов их жизненного цикла. К таким процессам в первую очередь следует отнести процесс менеджмента качества, процесс измерений, процесс управления и оценки проекта, процесс менеджмента модели жизненного цикла (см. рис. 8.2). Введение в состав жизненного цикла новых процессов, связанных с управлением качеством, способствует разработке высококачественных программных продуктов.

На процесс разработки (реализации) и деятельность по оценке качества ПС оказывают влияние следующие *обобщенные показатели ПС* [68]:

- область применения и назначение ПС;
- тип решаемых задач;
- объем и сложность ПС;
- необходимый состав и требуемые значения характеристик качества ПС и величина допустимого ущерба из-за их недостаточного качества;
- степень связи решаемых задач с реальным масштабом времени или допустимой длительностью ожидания результатов решения задачи;
- прогнозируемые значения длительности эксплуатации и перспектива создания множества версий ПС;
- предполагаемый тираж производства и применения ПС;
- степень необходимой документированности ПС.

Вопросами оценки качества продукции занимается наука квалиметрия. Основные понятия в области квалиметрии определены в стандарте *ГОСТ 15467–79. Управление качеством продукции. Основные понятия. Термины и определения* [6]. Данный стандарт переиздан в качестве межгосударственного стандарта стран СНГ и в октябре 2008 г. – непосредственно в качестве государственного стандарта Республики Беларусь.

В *ГОСТ 15467–79* дано следующее определение квалиметрии. **Квалиметрия** – это область науки, предметом которой являются количественные методы оценки качества продукции.

В настоящее время принято трактовать понятие квалиметрии более широко: **квалиметрия** – научная дисциплина, изучающая методологию и проблематику количественного оценивания качества объектов любой природы.

Части III – IV учебного пособия посвящены квалиметрии и метрологии программных средств. Данным направлениям посвящен ряд международных, межгосударственных стандартов и национальных стандартов Республики Беларусь. Основные понятия в области квалиметрии и метрологии программных средств будут определены по мере изложения материала при рассмотрении данных стандартов.

В этой части учебного пособия используются следующие **основные термины**, соответствующие определениям *ГОСТ 28806–90, СТБ ИСО/МЭК 9126–2003, ISO/IEC 9126–2–4:2003–2004* [10, 15, 60 – 62].

Атрибут (attribute) – внутренне присущее свойство объекта, которое может быть распознано количественно или качественно человеком или автоматизированными средствами. Атрибуты могут быть внешними или внутренними.

Измерение (measurement) – использование метрики для присвоения атрибуту продукта значения (числа или категории) из шкалы.

Индикатор (indicator) – мера, которая может использоваться для оценки или прогнозирования другой меры.

Качество (quality) – совокупность характеристик программного продукта, относящаяся к его способности удовлетворять установленные и подразумеваемые потребности.

Качество внешнее (external quality) – степень, в которой продукт удовлетворяет установленные и подразумеваемые потребности при использовании в заданных условиях.

Качество внутреннее (internal quality) – полный набор атрибутов продукта, определяющих его способность удовлетворять установленные и подразумеваемые потребности при использовании в заданных условиях.

Качество в использовании (quality in use) – степень, в которой программный продукт, используемый заданными пользователями, удовлетворяет их потребности в достижении заданных целей с результативностью, продуктивностью, безопасностью и удовлетворенностью в заданном контексте использования.

Контекст использования (context of use) – пользователи, задания, среда (аппаратное обеспечение, программное обеспечение и материалы), а также физические и социальные среды, в которых продукт используется.

Критерий оценки качества программного средства (software quality assessment criteria) – совокупность принятых в установленном порядке правил и условий, с помощью которых устанавливается приемлемость общего качества программного продукта.

Мера (measure) – число или категория, присваиваемая атрибуту продукта путем измерения.

Мера внешняя (external measure) – косвенная мера продукта, полученная из мер поведения системы, частью которой он является. Внешние меры могут использоваться для оценки атрибутов качества промежуточных продуктов ближе к конечным целям проекта.

Мера внутренняя (internal measure) – собственная мера продукта, прямая или косвенная.

Мера косвенная (indirect measure) – мера атрибута, которая получена из мер одного или большего числа других атрибутов.

Мера прямая (direct measure) – мера атрибута, которая не зависит от меры любого другого атрибута.

Метрика (metric) – определенный метод и шкала измерения. Метрики могут быть внутренними, внешними или метриками качества в использовании; прямыми или косвенными. Метрики включают методы для категоризации качественных данных (данных, которые нельзя измерить количественно).

Модель качества (quality model) – набор характеристик и связей между ними, обеспечивающий основу для определения требований к качеству и для оценки качества.

Отказ (failure) – прекращение способности продукта выполнять требуемую функцию или его неспособность работать в пределах заданных ограничений.

Оценка качества (quality evaluation) – систематическое исследование степени, в которой продукт способен к выполнению указанных требований.

Оценочный модуль (evaluation module) – пакет технологии оценивания для конкретной характеристики или подхарактеристики качества ПС.

Ошибка (fault) – некорректный шаг, процесс или определение данных в программе.

Подразумеваемые потребности (implied needs) – потребности, которые не были заданы, но являются реально существующими потребностями при использовании продукта в конкретных условиях. Подразумеваемые потребности включают потребности не заданные, но подразумеваемые другими заданными потребностями, и те незадаваемые потребности, которые рассматриваются как очевидные. Некоторые потребности становятся очевидными при использовании продукта в конкретных условиях.

Подхарактеристика качества программного средства (software quality subcharacteristic) – характеристика качества программного средства, входящая в состав другой характеристики качества.

Показатель качества программного средства (software quality feature) – признак, определяющий свойство программного средства, которое может быть соотнесено с некоторой характеристикой качества.

Промежуточный программный продукт (intermediate software product) – продукт процесса разработки программного обеспечения, который используется в качестве входных данных для другой стадии процесса разработки программного обеспечения. Частным случаем промежуточного продукта является конечный продукт.

Ранжирование (rating) – действие по отнесению измеренного значения к соответствующему уровню ранжирования.

Уровень качества функционирования (уровень пригодности, level of performance) – степень удовлетворения потребности, представленная конкретным набором значений характеристик качества.

Уровень ранжирования (уровень оценки, rating level) – точка на порядковой шкале, которая используется для категоризации шкалы измерения. Уровень оценки позволяет ранжировать программное обеспечение в соответствии с установленными или подразумеваемыми потребностями. Соответствующие уровни ранжирования могут быть связаны с различными точками зрения на качество, например, пользователей, администраторов или разработчиков.

Характеристика качества программного средства (software quality characteristic) – категория свойств (атрибутов) программного средства, с помощью которых описывается и оценивается его качество. Характеристики качества программных средств могут быть определены с помощью подхарактеристик и в конечном итоге атрибутов качества программного средства.

Шкала (scale) – набор значений с определенными свойствами.

При оценке качества используются следующие *типы шкал*:

- *номинальная* – соответствует набору категорий; классифицирует программы по признаку наличия или отсутствия некоторого свойства без учета градаций (например «да», «нет»);
- *порядковая (упорядоченная)* – соответствует упорядоченному набору делений шкалы; позволяет ранжировать свойства путем сравнения с опорными значениями; имеет небольшое количество делений (например, шкала с четырьмя градациями «отлично», «хорошо», «удовлетворительно», «неудовлетворительно», с двумя градациями «удовлетворительно», «неудовлетворительно»);
- *интервальная* – соответствует упорядоченной шкале с равноудаленными делениями; обычно содержит достаточно большое

количество делений с количественными значениями (например шкала с делениями 0, 1, 2, ..., 10);

- *отношений* – соответствует упорядоченной шкале с равноудаленными делениями, содержащей значение нуля, представляющего полное отсутствие атрибута.

Два первых типа шкал применяются для оценки качественных атрибутов ПС, которые нельзя измерить количественно, и для ранжирования измеренных значений, третий и четвертый типы – для оценки количественных атрибутов. Более подробно данные типы шкал рассмотрены в подразд. 13.2.

В ГОСТ 15467–79 введены следующие **термины и определения** [6].

Показатель качества продукции – количественная характеристика одного или нескольких свойств продукции, входящих в ее качество, рассматриваемая применительно к определенным условиям ее создания и эксплуатации или потребления.

Уровень качества продукции – относительная характеристика качества продукции, основанная на сравнении значений показателей качества оцениваемой продукции с базовыми значениями соответствующих показателей.

Оценка уровня качества продукции – совокупность операций, включающая выбор номенклатуры показателей качества оцениваемой продукции, определение значений этих показателей и сопоставление их с базовыми.

Коэффициент весомости (весовой коэффициент) показателя качества продукции – количественная характеристика значимости данного показателя качества продукции среди других показателей ее качества.

Дифференциальный метод оценки качества продукции – метод оценки качества продукции, основанный на использовании единичных показателей ее качества (определение единичного показателя качества продукции дано в подразд. 13.1).

Комплексный метод оценки качества продукции – метод оценки качества продукции, основанный на использовании комплексных показателей ее качества (определение комплексного показателя качества продукции дано в подразд. 13.1).

Смешанный метод оценки качества продукции – метод оценки качества продукции, основанный на одновременном использовании единичных и комплексных показателей ее качества.

Статистический метод оценки качества продукции – метод оценки качества продукции, при котором значения показателей качества продукции определяют с использованием правил математической статистики.

10. СТАНДАРТИЗАЦИЯ КАЧЕСТВА ПРОГРАММНЫХ СРЕДСТВ В РЕСПУБЛИКЕ БЕЛАРУСЬ

10.1. Общие сведения о стандартах в области качества программных средств, действующих на территории Республики Беларусь

В настоящее время в области качества программных средств на территории Республики Беларусь действуют следующие основные стандарты:

- межгосударственный стандарт стран СНГ *ГОСТ 28806–90. Качество программных средств. Термины и определения* (в 1992 г. введен в действие непосредственно в качестве государственного стандарта Республики Беларусь, переиздан в июне 2011 г.) [10];

- межгосударственный стандарт стран СНГ *ГОСТ 28195–99. Оценка качества программных средств. Общие положения* (в 2000 г. введен в действие непосредственно в качестве государственного стандарта Республики Беларусь) [9];

- национальный стандарт Беларуси *СТБ ИСО/МЭК 9126–2003. Информационные технологии. Оценка программной продукции. Характеристики качества и руководства по их применению* [15];

- национальный стандарт Беларуси *СТБ ISO/IEC 25000–2009. Разработка программного обеспечения. Требования к качеству и оценка программного продукта (SQuaRE). Руководство по SQuaRE* [19];

- национальный стандарт Беларуси *СТБ ISO/IEC 25001–2009. Разработка программного обеспечения. Требования к качеству и оценка программного продукта (SQuaRE). Планирование и управление* [20];

- национальный стандарт Беларуси *СТБ ECSS–E–ST–40C–2014. Космическая техника. Разработка продукции. Общие требования к программному обеспечению* [16];

- национальный стандарт Беларуси *СТБ ECSS–Q–ST–80C–2014. Космическая техника. Обеспечение качества продукции. Гарантия качества программного обеспечения* [17];

- национальный стандарт Беларуси *СТБ ECSS–Q–HB–80–03A–2014. Космическая техника. Обеспечение качества продукции. Надежность и безопасность программного обеспечения* [18].

Стандарты *СТБ ISO/IEC 25000–2009* и *СТБ ISO/IEC 25001–2009* имеют идентичную степень соответствия стандартам *ISO/IEC 25000:2005* и *ISO/IEC 25001:2007*, представляющим собой первые редакции действующих международных стандартов *ISO/IEC 25000:2014* и *ISO/IEC 25001:2014* [38, 40]. Данные стандарты относятся к серии международных стандартов *SQuaRE*, посвященной качеству ПС и систем. Серия *SQuaRE* рассмотрена в разд. 12.

Стандарты *СТБ ECSS–E–ST–40C–2014*, *СТБ ECSS–Q–ST–80C–2014*, *СТБ ECSS–Q–HB–80–03A–2014* идентичны соответственно стандартам на разработку программного обеспечения *ECSS–E–ST–40C*, *ECSS–Q–ST–80C* и *ECSS–Q–HB–80–03A* Европейской организации по стандартизации в области космической деятельности (European Cooperation on Space Standardization, ECSS). Данные стандарты формально распространяются на специальные ПС наземных и орбитальных комплексов, однако представляют интерес для разработки ПС, предназначенных для использования в любых критических областях с высокими рисками, для которых важным является обеспечение высокого уровня качества ПС. Вышеназванные стандарты входят в состав разделов стандартов ECSS «Инженерия космических разработок» и «Обеспечение космических разработок».

В данном разделе рассмотрены основные положения стандартов *ГОСТ 28806–90*, *ГОСТ 28195–99*, *СТБ ИСО/МЭК 9126–2003*.

ГОСТ 28806–90 регламентирует основные термины и определения, принятые в области качества программного обеспечения, определяет характеристики и подхарактеристики качества ПС.

СТБ ИСО/МЭК 9126–2003 представляет собой стандарт, идентичный международному стандарту *ISO/IEC 9126:1991* [58]. В данном стандарте определена трехуровневая иерархическая модель качества ПС и модель метода оценки качества ПС (см. подразд. 10.3).

ГОСТ 28195–99 регламентирует четырехуровневую иерархическую модель качества и ориентированный на нее метод оценки качества ПС (см. подразд. 10.5).

Обеспечение и оценка качества ПС выполняются в течение всего жизненного цикла программных средств и неразрывно связаны с положениями стандартов *СТБ ИСО/МЭК 12207–2003* и *ISO/IEC 12207:2008*, рассмотренных в части II учебного пособия (см. разд. 6 и 8).

10.2. Классификация методов определения показателей качества программных средств и систем

Классификация методов определения показателей качества определена в стандартах *ГОСТ 28195–99* и *ГОСТ 15467–79* [9, 6].

ГОСТ 28195–99 классифицирует методы определения показателей качества программных средств следующим образом:

- по способам получения информации о показателе качества: измерительный; регистрационный; восприятие человеком; расчетный;
- по источникам получения информации о показателе качества: экспертный; традиционный.

Измерительный метод – это метод получения информации о свойствах и характеристиках ПС путем измерений с помощью инструментальных средств (например, так может определяться количество операторов в программе, количество выполненных операторов, количество операндов, время выполнения программы при определенных наборах исходных данных и т. д.).

Регистрационный метод – это метод получения информации о свойствах и характеристиках ПС во время его испытания или функционирования, когда регистрируются некоторые события (например количество сбоев и отказов).

Метод восприятия человеком – это метод получения информации о свойствах и характеристиках ПС, основанный на восприятии человеком. Так могут определяться, например, свойства ПС, связанные с удобством его использования.

Расчетный метод – это метод получения информации о свойствах и характеристиках ПС, основанный на использовании эмпирических и теоретических зависимостей (на ранних этапах разработки), статистических данных, накапливаемых при испытаниях, эксплуатации и сопровождении ПС. Так может определяться, например, точность вычислений.

Экспертный метод – это метод получения информации о свойствах и характеристиках ПС на основании мнений группы экспертов-специалистов, компетентных в решении данной задачи. Экспертный метод применяется в том случае, когда невозможно или слишком трудоемко выполнить оценку показателей качества с помощью других методов. Данным методом рекомендуется определять, например, показатели понимаемости и осваиваемости ПС.

Традиционный метод – это метод получения информации о свойствах и характеристиках ПС на основе непосредственного наблюдения за их функционированием в процессе работы. Так могут определяться, например, некоторые из показателей функциональности и удобства использования.

ГОСТ 15467–79 классифицирует методы определения показателей качества для любых объектов, в том числе для систем и ПС. В данном стандарте определены те же методы, что и в ГОСТ 28195–99. Дополнительно к ним по источникам получения информации о показателе качества определен **социологический метод** – это метод определения значений показателей качества продукции, осуществляемый на основе сбора и анализа мнений ее фактических или возможных пользователей путем обработки анкет.

В ГОСТ 15467–79 метод восприятия человеком называется *органолептическим методом*.

10.3. Иерархическая модель оценки качества программных средств по ГОСТ 28806–90 и СТБ ИСО/МЭК 9126–2003

Стандарты ГОСТ 28806–90, ГОСТ 28195–99, СТБ ИСО/МЭК 9126–2003 [10, 9, 15] регламентируют выполнение оценки качества ПС на основе *иерархической модели качества*. В соответствии с данной моделью совокупность свойств, отражающих качество программного средства, представляется в виде многоуровневой структуры. Характеристики на первом (верхнем) уровне соответствуют основным свойствам ПС. Характеристики каждого уровня оцениваются посредством характеристик последующих уровней.

Стандарты ГОСТ 28806–90, СТБ ИСО/МЭК 9126–2003 базируются на трехуровневой иерархической модели качества. При этом в данных стандартах определены только первые два уровня модели. Номенклатура характеристик первого уровня является *обязательной*, а номенклатура характеристик второго уровня (подхарактеристик) – *рекомендуемой*.

Вышеназванные стандарты регламентируют шесть основных характеристик качества ПС, находящихся на верхнем уровне модели качества.

Следует отметить, что модели качества, определенные в ГОСТ 28806–90 и СТБ ИСО/МЭК 9126–2003, между собой практически совпадают. Данные модели соответствуют принятым в 1991 – 2011 гг. в мировой практике (см. ISO/IEC 9126:1991 и ISO/IEC 9126–1:2001 [58, 59]). Но ГОСТ 28806–90 и СТБ ИСО/МЭК 9126–2003 используют частично разную терминологию для обозначения одних и тех же характеристик и подхарактеристик на русском языке.

На рис. 10.1 и 10.2 приведены два верхних уровня иерархической модели качества, определенной в ГОСТ 28806–90 и СТБ ИСО/МЭК 9126–2003 соответственно. Характеристики и подхарактеристики качества, различающиеся в данных стандартах терминологически, на этих рисунках подчеркнуты.

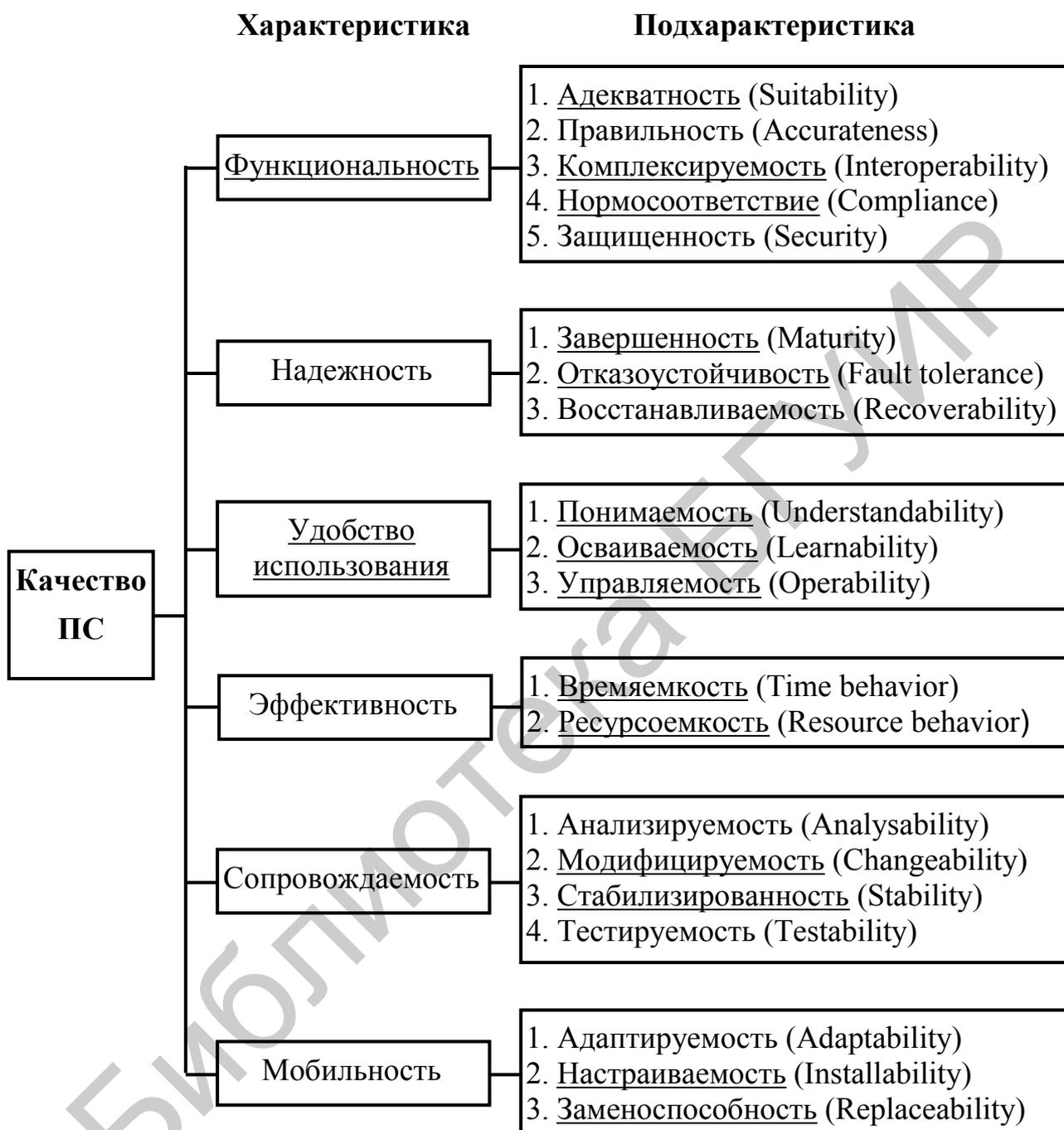


Рис. 10.1. Модель качества по ГОСТ 28806–90

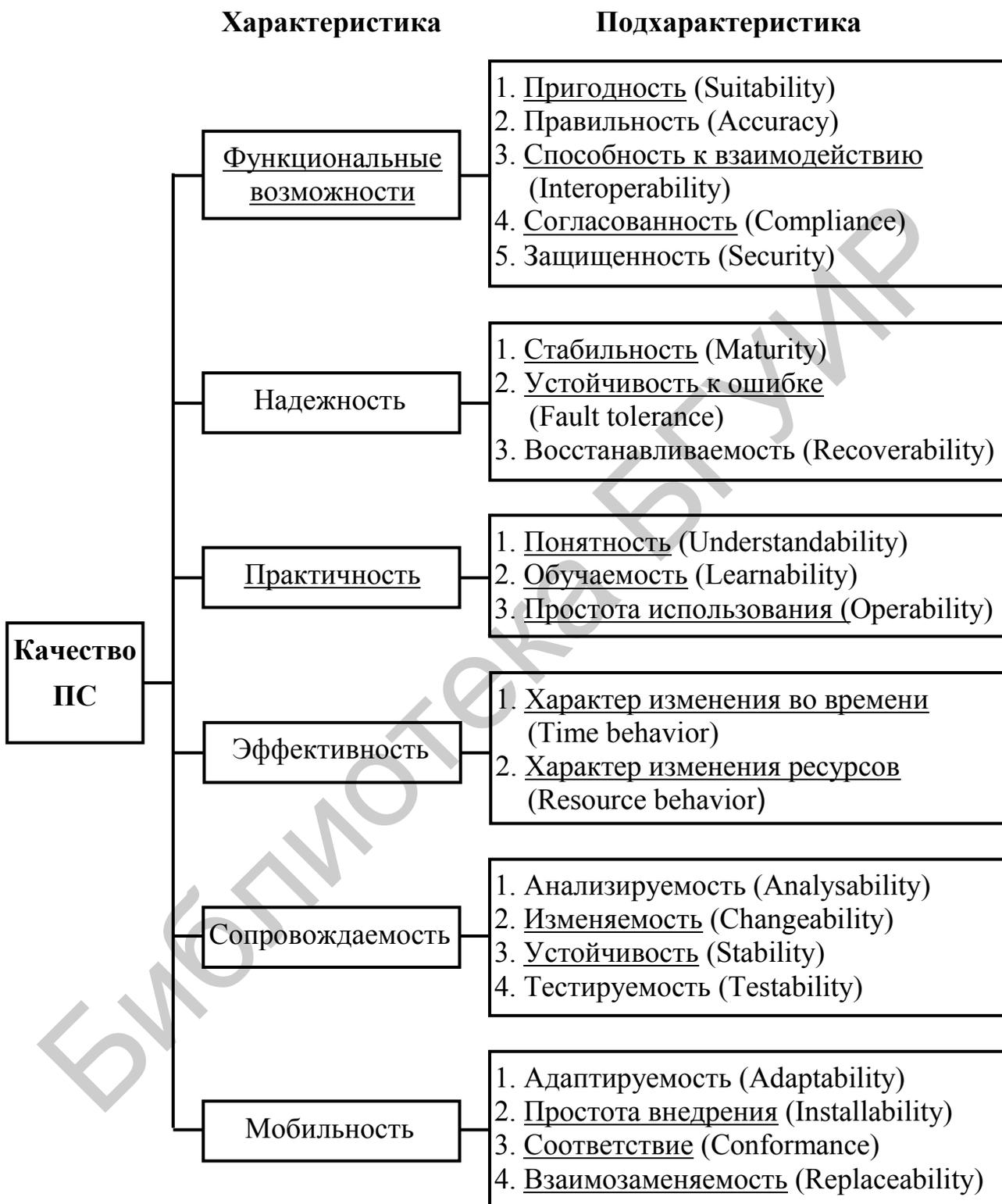


Рис. 10.2. Модель качества по СТБ ИСО/МЭК 9126–2003 (ISO/IEC 9126:1991)

На верхнем уровне модели определены *шесть основных характеристик качества* ПС (характеристики качества верхнего уровня):

- *Функциональность* – в ГОСТ 28806–90 (*Функциональные возможности* – в СТБ ИСО/МЭК 9126–2003);
- *Надежность*;
- *Удобство использования* – в ГОСТ 28806–90 (*Практичность* – в СТБ ИСО/МЭК 9126–2003);
- *Эффективность*;
- *Сопровождаемость*;
- *Мобильность*.

Ниже даны определения характеристик и подхарактеристик рассматриваемой модели качества.

10.3.1. Функциональность

Функциональность (*Функциональные возможности, Functionality*) – способность программного продукта обеспечивать функции, удовлетворяющие установленные и подразумеваемые потребности при применении программного средства в заданных условиях.

В СТБ ИСО/МЭК 9126–2003 отмечено, что характеристика *Функциональность* оценивает то, что делает ПС в соответствии с потребностями. Другие характеристики оценивают, когда и как эти потребности удовлетворяются.

Подхарактеристики *Функциональности* приведены на рис. 10.1, 10.2.

Пригодность (*Адекватность, Suitability*) – способность программного продукта обеспечивать набор функций, соответствующий специфическим задачам и целям пользователей.

Правильность (*Accuracy, Accurateness*) – способность программного продукта обеспечивать правильные или приемлемые результаты или эффекты с необходимой степенью точности.

Способность к взаимодействию (*Комплексируемость, Interoperability*) – способность программного продукта взаимодействовать с одной или несколькими заданными системами.

Согласованность (*Нормосоответствие, Compliance*) – способность программного продукта соответствовать стандартам, соглашениям или нормам законов, связанным с *Функциональностью*.

Защищенность (*Security*) – способность программного продукта защищать информацию и данные так, чтобы несанкционированные пользователи или системы не могли прочитать или модифицировать их, а санкционированные пользователи или системы не могли получить отказа в доступе к ним.

10.3.2. Надежность

Надежность (Reliability) – способность программного продукта поддерживать заданный уровень качества функционирования при его использовании в заданных условиях в течение заданного интервала времени.

Ограничения *Надежности* в процессе эксплуатации вызваны ошибками в требованиях, проектировании и кодировании.

Подхарактеристики *Надежности* приведены на рис. 10.1, 10.2.

Стабильность (Завершенность, Maturity) – способность программного продукта избегать отказов вследствие ошибок в программах.

Устойчивость к ошибке (Отказоустойчивость, Fault tolerance) – способность программного продукта поддерживать заданный уровень качества функционирования в случаях ошибок в программах или нарушения заданного интерфейса ПП.

Восстанавливаемость (Recoverability) – способность программного продукта восстанавливать заданный уровень качества функционирования и данные, поврежденные в случае отказа.

К показателям *Восстанавливаемости* относятся трудоемкость и длительность восстановления.

10.3.3. Практичность

Практичность (Удобство использования, Usability) – способность программного продукта быть понятным, изученным и использованным при применении в заданных условиях. *Практичность* характеризует усилия, необходимые для использования продукта, и индивидуальную оценку результатов его использования заданным или подразумеваемым кругом пользователей.

Очевидно, что *Практичность* зависит от некоторых аспектов таких характеристик качества, как *Функциональность*, *Надежность* и *Эффективность*.

Подхарактеристики *Практичности* приведены на рис. 10.1, 10.2.

Понятность (Понимаемость, Understandability) – способность программного продукта обеспечивать понимание пользователем пригодности и способа использования программного средства для конкретных задач и условий применения.

Обучаемость (Осваиваемость, Learnability) – способность программного продукта обеспечивать изучение пользователем принципов его применения. *Обучаемость* отражает усилия пользователя, затрачиваемые на освоение его применения.

Простота использования (Управляемость, Operability) – способность программного продукта позволять пользователю эксплуатировать его и управлять им. На *Простоту использования* влияют некоторые аспекты таких подхарактеристик, как *Изменяемость*, *Адаптируемость*, *Простота внедрения*, *Устойчивость к ошибке*.

10.3.4. Эффективность

Эффективность (Efficiency) – способность программного продукта обеспечить соответствующую производительность в зависимости от количества используемых вычислительных ресурсов в заданных условиях.

Ресурсы могут включать другие программные продукты, конфигурацию программных и аппаратных средств системы и материалы.

Подхарактеристики *Эффективности* приведены на рис. 10.1, 10.2.

Характер изменения во времени (Времяемкость, Time behaviour) – способность программного продукта обеспечивать соответствующие времена отклика и обработки, а также пропускную способность при выполнении своих функций в заданных условиях.

Характер изменения ресурсов (Ресурсоемкость, Resource utilisation) – способность программного продукта использовать соответствующее количество всех типов ресурсов при выполнении своих функций в заданных условиях. Данная характеристика определяет объем используемых ресурсов и продолжительность их использования.

10.3.5. Сопровождаемость

Сопровождаемость (Maintainability) – способность программного продукта к модификации. Модификации могут включать исправления, усовершенствования или адаптацию ПС к изменениям в среде применения, в функциональных и нефункциональных требованиях.

Подхарактеристики *Сопровождаемости* представлены на рис. 10.1, 10.2.

Анализируемость (Analysability) – способность программного продукта к диагностике его недостатков или причин отказов или к идентификации его частей, которые должны быть модифицированы.

Изменяемость (Модифицируемость, Changeability) – способность программного продукта к реализации заданной модификации. Реализация модификации включает проектирование, кодирование и изменение документации.

Устойчивость (Стабилизированность, Stability) – способность программного продукта предотвращать непредвиденные эффекты от его модификации.

Тестируемость (Testability) – способность программного продукта к проверке результата модификации. *Тестируемость* отражает усилия, необходимые для проверки программного продукта после любых видоизменений.

10.3.6. Мобильность

Мобильность (Portability) – способность программного продукта к переносу из одной среды функционирования в другую. Среда может включать организационное, аппаратное и программное окружение.

Подхарактеристики *Мобильности* представлены на рис. 10.1, 10.2.

Адаптируемость (Adaptability) – способность программного продукта адаптироваться к различным окружающим средам без применения дополнительных действий или средств.

Адаптируемость включает масштабируемость внутренних возможностей (например, областей экранов, таблиц, форматов отчетов).

Настраиваемость (Простота внедрения, Installability) – способность программного продукта устанавливаться в заданной среде окружения.

Взаимозаменяемость (Заменоспособность, Replaceability) – способность программного продукта к использованию вместо другого (заданного) программного продукта с той же целью и в той же среде.

Например, для пользователя важна взаимозаменяемость новой версии программного продукта с его старой версией. *Взаимозаменяемость* может включать атрибуты таких подхарактеристик, как *Настраиваемость* и *Адаптируемость*.

Соответствие (Conformance) – способность программного продукта соответствовать стандартам или соглашениям, связанным с *Мобильностью*.

Данная подхарактеристика присутствует в *СТБ ИСО/МЭК 9126–2003*, но отсутствует в *ГОСТ 28806–90*.

10.4. Метод оценки качества программных средств по СТБ ИСО/МЭК 9126–2003

СТБ ИСО/МЭК 9126–2003 [15] определяет метод оценки качества программного продукта, основанный на *трехуровневой иерархической модели качества*. На первом уровне модели находятся шесть характеристик качества (см. подразд. 10.3). Второй уровень составляют подхарактеристики и третий – метрики качества.

Модель процесса оценки, положенная в основу рассматриваемого метода, приведена на рис. 10.3. Данная модель отражает основные стадии и этапы, требуемые для оценки качества программного продукта.

Процесс оценки состоит из *трех стадий*: определение требований к качеству программного продукта, подготовка к оцениванию и процедура оценивания. Данный процесс может применяться после любой подходящей работы жизненного цикла для каждого компонента программного продукта.

Стадия 1. Определение требований к качеству

Целью данной стадии является установка требований в терминах характеристик и подхарактеристик качества. Требования выражают потребности внешнего окружения программного продукта и должны быть определены до начала разработки. Так как программный продукт в общем случае разделяется на компоненты, то требования для программного продукта в целом могут отличаться от требований для отдельных компонентов.

Стадия 2. Подготовка к оцениванию

Целью второй стадии является подготовка основы для оценивания. Данная стадия состоит из трех этапов.

Этап 2.1. Выбор метрик качества

С учетом регламентированной в СТБ ИСО/МЭК 9126–2003 иерархической модели качества уровень характеристик качества программного продукта определяется уровнем входящих в них подхарактеристик, а значения подхарактеристик в свою очередь определяются значениями входящих в них метрик.

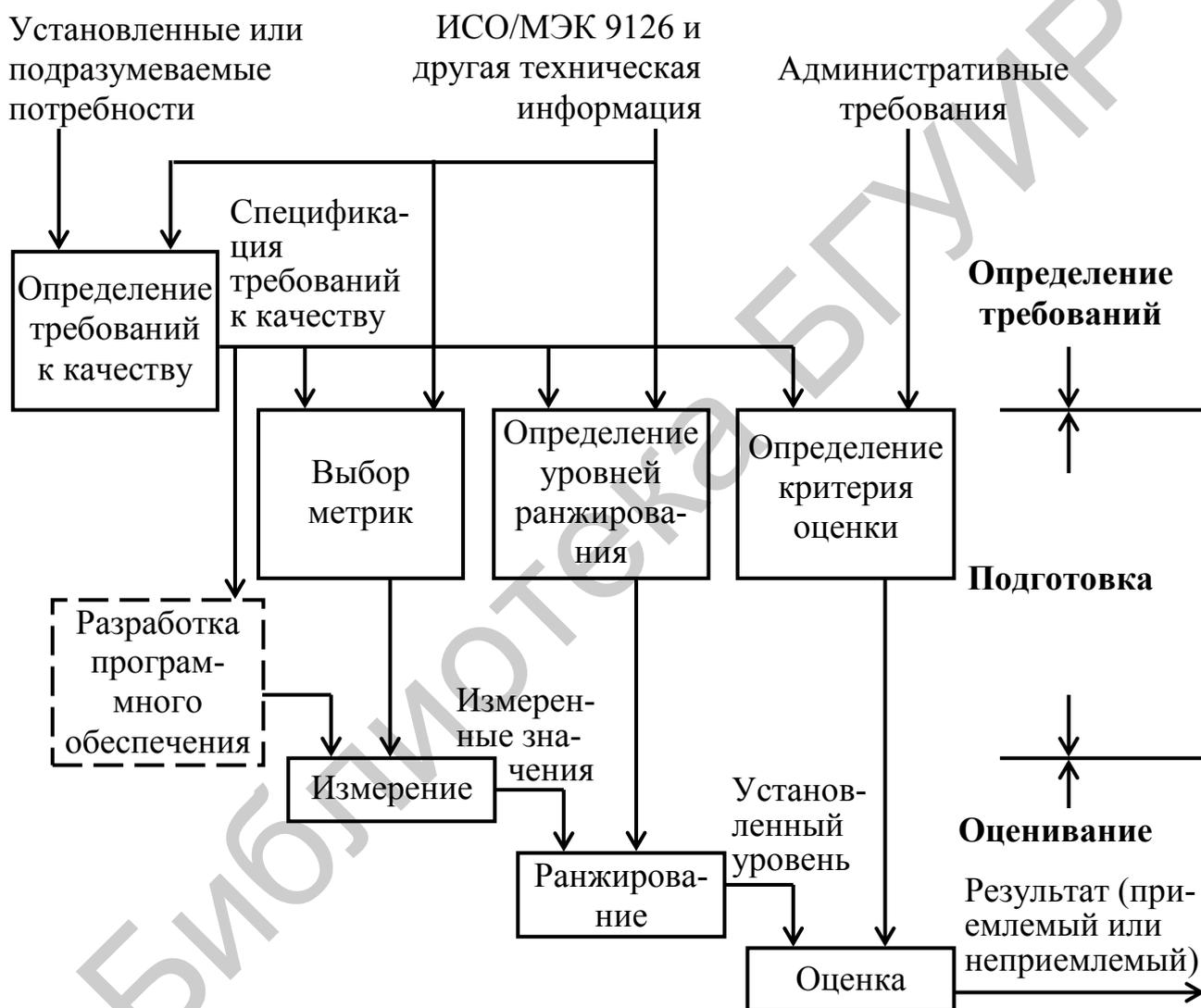


Рис. 10.3. Модель процесса оценки по СТБ ИСО/МЭК 9126–2003

В стандарте СТБ ИСО/МЭК 9126–2003 набор рекомендуемых метрик отсутствует. Поэтому существует потребность в установлении метрик, которые соотносятся с подхарактеристиками, а следовательно, и с характеристиками программного продукта. Каждый количественный признак и каждое количественно оцениваемое взаимодействие программного продукта с его

окружением, которые соотносятся с характеристикой, могут быть приняты в качестве метрики. Метрики, используемые в процессе разработки, должны быть соотнесены с соответствующими метриками пользователя, поскольку метрики пользователя являются решающими.

Этап 2.2. Определение уровней ранжирования

Для измерения количественных признаков программного продукта используются метрики качества. Измеренные значения отображаются на некоторой шкале. Данные значения не показывают уровень удовлетворения требований к качеству программного продукта. Для этой цели шкалы метрик должны быть разделены на диапазоны, соответствующие различным степеням удовлетворения требований.

На рис. 10.4 приведен пример двух возможных диапазонов ранжирования из стандарта *ISO/IEC 14598-1:1999* (см. подразд. 11.1, 11.5) [28]:

1) разделение шкалы на две категории: неудовлетворительно и удовлетворительно;

2) разделение шкалы на четыре категории (отлично, хорошо, удовлетворительно, неудовлетворительно), ограниченные соответственно запланированным уровнем, текущим уровнем для существующего или альтернативного продукта и уровнем худшего случая.



Рис. 10.4. Пример вариантов ранжирования измеренных значений метрик

Текущий уровень определяется для управления тем, чтобы новый программный продукт и система в целом не становились хуже по сравнению с существующими. Запланированный уровень определяет уровень, который считается достижимым при доступных ресурсах. Уровень худшего случая определяет границу принятия пользователем в случае, если продукт не удовлетворяет запланированному уровню (см. рис. 10.4). Так как качество

программного продукта связано с конкретными потребностями, общие уровни ранжирования метрик невозможны и должны определяться для каждого конкретного оценивания.

Этап 2.3. Определение критерия оценки

Для определения общего качества программного продукта должна быть учтена вся совокупность результатов оценивания различных метрик. Оценщик должен подготовить для этого процедуры, используя, например, таблицы решений или средневзвешенные значения (вариант метода оценки, использующего средневзвешенные значения, подробно описан в подразд. 10.6). Обычно при этом могут учитываться и другие аспекты, которые являются косвенными факторами качества программного продукта (такие как продолжительность и стоимость оценки).

Стадия 3. Процедура оценивания

Последняя стадия модели процесса оценки реализуется тремя этапами: «Измерение», «Ранжирование» и «Оценка».

Этап 3.1. Измерение

Для измерения выбранные метрики применяются к оцениваемому программному продукту. Результатом являются конкретные значения, измеренные в шкалах метрик.

Этап 3.2. Ранжирование

На этапе ранжирования устанавливаются уровни ранжирования для измеренных значений метрик (см. рис. 10.4).

Этап 3.3. Оценка

Оценка является последним этапом процесса оценки программного продукта, на котором обобщается множество установленных уровней ранжирования метрик. Результатом является заключение о качестве ПС (приемлемый или неприемлемый уровень качества).

К *основным недостаткам* данного метода оценки качества следует отнести: отсутствие рекомендуемых вариантов метрик; представление метода лишь в общем виде (в виде модели); ориентированность на получение качественного, а не количественного результата оценки (результат приемлемый или неприемлемый). Это затрудняет конкретное использование и снижает точность данного метода оценки программных продуктов.

10.5. Модель качества программных средств по ГОСТ 28195–99

Под *показателем качества продукции* в ГОСТ 28195–99 и ГОСТ 15467–79 [9, 6] подразумевается количественная характеристика одного или нескольких свойств продукции, составляющих ее качество, рассматриваемая применительно к определенным условиям ее создания и эксплуатации или потребления. Показатели качества представляют собой иерархическую

многоуровневую систему, в которой показатели вышестоящих уровней определяются через показатели нижестоящих уровней, и лишь на последнем (нижнем) уровне оценка значений показателей осуществляется на основе информации, относящейся непосредственно к ПС.

ГОСТ 28195–99 определяет *четырёхуровневую иерархическую модель* качества ПС. В данном стандарте предлагается следующая **терминология** для показателей качества каждого уровня:

- *уровень 1* – факторы качества (в терминологии, принятой в международных стандартах, соответствуют характеристикам качества [58, 59]);
- *уровень 2* – критерии качества (в международной терминологии – подхарактеристики качества);
- *уровень 3* – метрики (в международной терминологии в настоящее время используются как термин «метрика», так и термин «мера» [38, 60 – 62]);
- *уровень 4* – оценочные элементы или единичные показатели (в современной международной терминологии в общем случае соответствуют элементам мер качества [40]).

Факторы и критерии качества, определенные в *ГОСТ 28195–99*, приведены на рис. 10.5.

В соответствии с положениями *ГОСТ 28195–99* номенклатура показателей качества двух верхних уровней (факторов и критериев) является *обязательной*, номенклатура показателей третьего и четвертого уровней (метрик и оценочных элементов) – *рекомендуемой*.

Верхний уровень модели качества включает шесть факторов. Данный уровень определяется группой показателей ПС, характеризующих потребителски ориентированные свойства, которые соответствуют потребностям населения, народного хозяйства и экспорта продукции.

Второй уровень (уровень критериев) определяется комплексными показателями качества ПС, характеризующими программно-ориентированные свойства, которые обеспечивают достижение требуемых потребителски ориентированных свойств.

Из рис. 10.5 видно, что совокупность факторов и критериев, регламентированная в *ГОСТ 28195–99*, существенно отличается от двух верхних уровней иерархической модели качества, принятой в 1991 – 2011 гг. в мировой практике и описанной в подразд. 10.3. Рассмотрим данную модель более детально в терминологии *ГОСТ 28195–99*.

10.5.1. Надежность

Надежность характеризует способность ПС в конкретных областях применения выполнять заданные функции в соответствии с программными документами в условиях возникновения отклонений в среде функционирования, вызванных сбоями технических средств, ошибками во входных данных, ошибками обслуживания и другими дестабилизирующими воздействиями.



Рис. 10.5. Факторы и критерии качества программных средств по ГОСТ 28195–99

Подхарактеристики (критерии) *Надежности* приведены на рис. 10.5.

Устойчивость функционирования – способность обеспечивать продолжение работы программы после возникновения отклонений, вызванных сбоями технических средств, ошибками во входных данных и ошибками обслуживания.

Работоспособность – способность программы функционировать в заданных режимах и объемах обрабатываемой информации в соответствии с программными документами при отсутствии сбоев технических средств.

10.5.2. Сопровождаемость

Сопровождаемость характеризует технологические аспекты, обеспечивающие простоту устранения ошибок в программе и программных документах и поддержания ПС в актуальном состоянии.

Подхарактеристики (критерии) *Сопровождаемости* приведены на рис. 10.5.

Структурность – организация всех взаимосвязанных частей программы в единое целое с использованием логических структур «последовательность», «выбор», «повторение».

Простота конструкции – построение модульной структуры программы наиболее рациональным с точки зрения восприятия и понимания образом.

Наглядность – наличие и представление в наиболее легко воспринимаемом виде исходных модулей ПС, полное их описание в соответствующих программных документах.

Повторяемость – степень использования типовых проектных решений или компонентов, входящих в ПС.

Полнота документации – наличие документов, необходимых для понимания всех технических решений, принятых на фазах проектирования, реализации, тестирования и изготовления жизненного цикла ПС.

10.5.3. Удобство использования

Удобство использования характеризует свойства ПС, способствующие быстрому освоению, применению и эксплуатации ПС с минимальными трудозатратами с учетом характера решаемых задач и требований к квалификации обслуживающего персонала.

Подхарактеристики (критерии) *Удобства использования* приведены на рис. 10.5.

Легкость освоения – представление программных документов и программы в виде, способствующем пониманию логики функционирования программы в целом и ее частей.

Доступность программных документов – понятность, наглядность и полнота описания взаимодействия пользователя с программой в программных документах.

Удобство эксплуатации и обслуживания – соответствие процесса обработки данных и форм представления результатов характеру решаемых задач.

10.5.4. Эффективность

Эффективность характеризует степень удовлетворения потребности пользователя в обработке данных с учетом экономических, людских ресурсов и ресурсов системы обработки информации.

Подхарактеристики (критерии) *Эффективности* приведены на рис. 10.5.

Уровень автоматизации – уровень автоматизации функций процесса обработки данных с учетом рациональности функциональной структуры программы с точки зрения взаимодействия с ней пользователя и использования ресурсов системы обработки информации.

Временная эффективность – способность программы выполнять заданные действия в интервале времени, отвечающем заданным требованиям.

Ресурсоемкость – минимально необходимые ресурсы системы обработки информации и число обслуживающего персонала для эксплуатации ПС.

10.5.5. Универсальность

Универсальность характеризует адаптируемость ПС к новым функциональным требованиям, возникающим вследствие изменения условий функционирования.

Подхарактеристики (критерии) *Универсальности* приведены на рис. 10.5.

Гибкость – возможность использования ПС в различных областях применения.

Мобильность – возможность применения ПС без существенных дополнительных трудозатрат на компьютерах аналогичного класса.

Модифицируемость – обеспечение простоты внесения необходимых изменений и доработок в программу в процессе эксплуатации.

10.5.6. Функциональность

Функциональность характеризует степень соответствия ПС требованиям, установленным в техническом задании (ТЗ), требованиям к обработке данных и общесистемным требованиям.

Подхарактеристики (критерии) *Функциональности* приведены на рис. 10.5.

Полнота реализации – полнота реализации заданных функций ПС и достаточность их описания в программной документации.

Согласованность – однозначное, непротиворечивое описание и использование тождественных объектов, функций, терминов, определений, идентификаторов и т. д. в различных частях программных документов и текста программы.

Логическая корректность – функциональное и программное соответствие процесса обработки данных при выполнении задания общесистемным требованиям.

Проверенность – полнота проверки возможных маршрутов выполнения программы в процессе тестирования.

Защищенность – способность программного средства предотвращать несанкционированный доступ к программам и данным, а также степень удобства и полноты обнаружения результатов такого доступа или действий по разрушению программ и данных.

10.6. Метод оценки качества программных средств по ГОСТ 28195–99

ГОСТ 28195–99 [9] определяет *оценку качества* программного средства как совокупность операций, включающих выбор номенклатуры показателей качества оцениваемого программного средства, определение значений этих показателей и сравнение их с базовыми значениями. В соответствии с данным стандартом оценка качества должна проводиться применительно ко всем работам ЖЦ ПС при планировании показателей качества ПС, контроле качества программного продукта в процессе разработки, проверке эффективности модификации ПС в процессе сопровождения.

Основными задачами, решаемыми при оценке качества программного средства, по ГОСТ 28195–99 являются:

- 1) планирование номенклатуры показателей качества;
- 2) планирование уровня качества;
- 3) выбор методов контроля показателей качества;
- 4) контроль значений показателей качества в процессе ЖЦ ПС;
- 5) выбор базовых образцов по подклассам и группам;
- 6) принятие решения о соответствии реальных значений показателей качества установленным требованиям.

Данным стандартом рекомендован метод интегральной оценки качества программных средств, основанный на четырехуровневой иерархической модели качества, описанной в подразд. 10.5.

В соответствии с этим методом выбор номенклатуры показателей качества для конкретного программного средства осуществляется с учетом его назначения и требований области применения в зависимости от принадлежности ПС к тому или иному подклассу, определяемому *общесоюзным классификатором продукции (ОКП)*.

В ОКП предусмотрены следующие *подклассы программных средств*:

- 5011 – операционные системы и средства их расширения;
- 5012 – программные средства управления базами данных;
- 5013 – инструментально-технологические средства программирования;

- 5014 – ПС интерфейса и управления коммуникациями;
- 5015 – программные средства организации вычислительного процесса (например планирования, контроля);
- 5016 – сервисные программы;
- 5017 – ПС обслуживания вычислительной техники;
- 503 – прикладные программы для научных исследований;
- 504 – прикладные программы для проектирования;
- 505 – прикладные программы для управления техническими устройствами и технологическими процессами;
- 506 – прикладные программы для решения экономических задач;
- 509 – прочие программные средства.

Оценка качества ПС производится на всех фазах жизненного цикла.

ГОСТ 28195–99 базируется на следующих *процессах и фазах жизненного цикла ПС*:

1. Процесс разработки:

- фаза анализа;
- фаза проектирования;
- фаза реализации;
- фаза тестирования;
- фаза изготовления.

2. Процесс применения:

- фаза внедрения;
- фаза эксплуатации;
- фаза сопровождения.

Вышеприведенные фазы представляют собой периоды, соответствующие работам, совокупностям работ или процессам ЖЦ ПС, определенным стандартами *СТБ ИСО/МЭК 12207–2003*, *ГОСТ ИСО/МЭК 12207–2002* и *ISO/IEC 12207:2008* и описанным в разд. 6 и 8.

Как было отмечено выше, оценка качества ПС заключается в выборе номенклатуры показателей, их оценке и сопоставлении с базовыми значениями.

Для каждого из выбранных факторов качества составляется четырехуровневая иерархическая модель, отражающая взаимосвязь факторов, критериев, метрик и оценочных элементов. Вид данной модели зависит от фазы ЖЦ ПС.

В качестве примера на рис. 10.6 – 10.8 приведены три верхних уровня иерархической модели фактора *Надежность* для различных фаз жизненного цикла. Номера на данных рисунках соответствуют номерам метрик данного фактора в *ГОСТ 28195–99*.

Выбор оценочных элементов в метрике зависит от функционального назначения ПС и формируется с учетом данных, ранее полученных при проведении испытаний ПС и эксплуатации аналогичных программ. Для выбора оценочных элементов *ГОСТ 28195–99* предлагает перечень таблиц,

содержащих наименование элемента, метод оценки и применяемость элемента для различных подклассов ПС.



Рис. 10.6. Модель надежности для фазы анализа

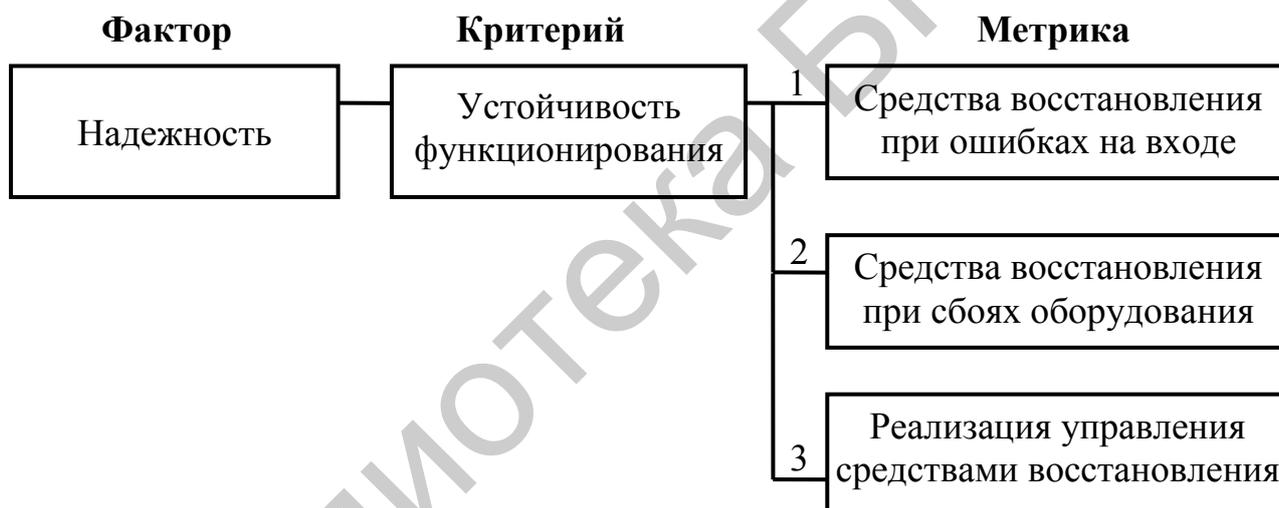


Рис. 10.7. Модель надежности для фазы проектирования

Перечень оценочных элементов для фактора *Надежность* приведен в табл. 10.1. В данной таблице код оценочного элемента состоит из пяти символов. Первый символ (буква) указывает на принадлежность элемента фактору (в *ГОСТ 28195–99* приняты следующие обозначения факторов: Н – *Надежность*; С – *Сопровождаемость*; У – *Удобство использования*; Э – *Эффективность*; Г – *Универсальность*; К – *Функциональность*). Два следующих символа – номер метрики, которой принадлежит оценочный элемент (для фактора *Надежность* номера метрик обозначены на рис. 10.6 – 10.8). Четвертый и пятый символы – порядковый номер данного оценочного элемента в метрике.

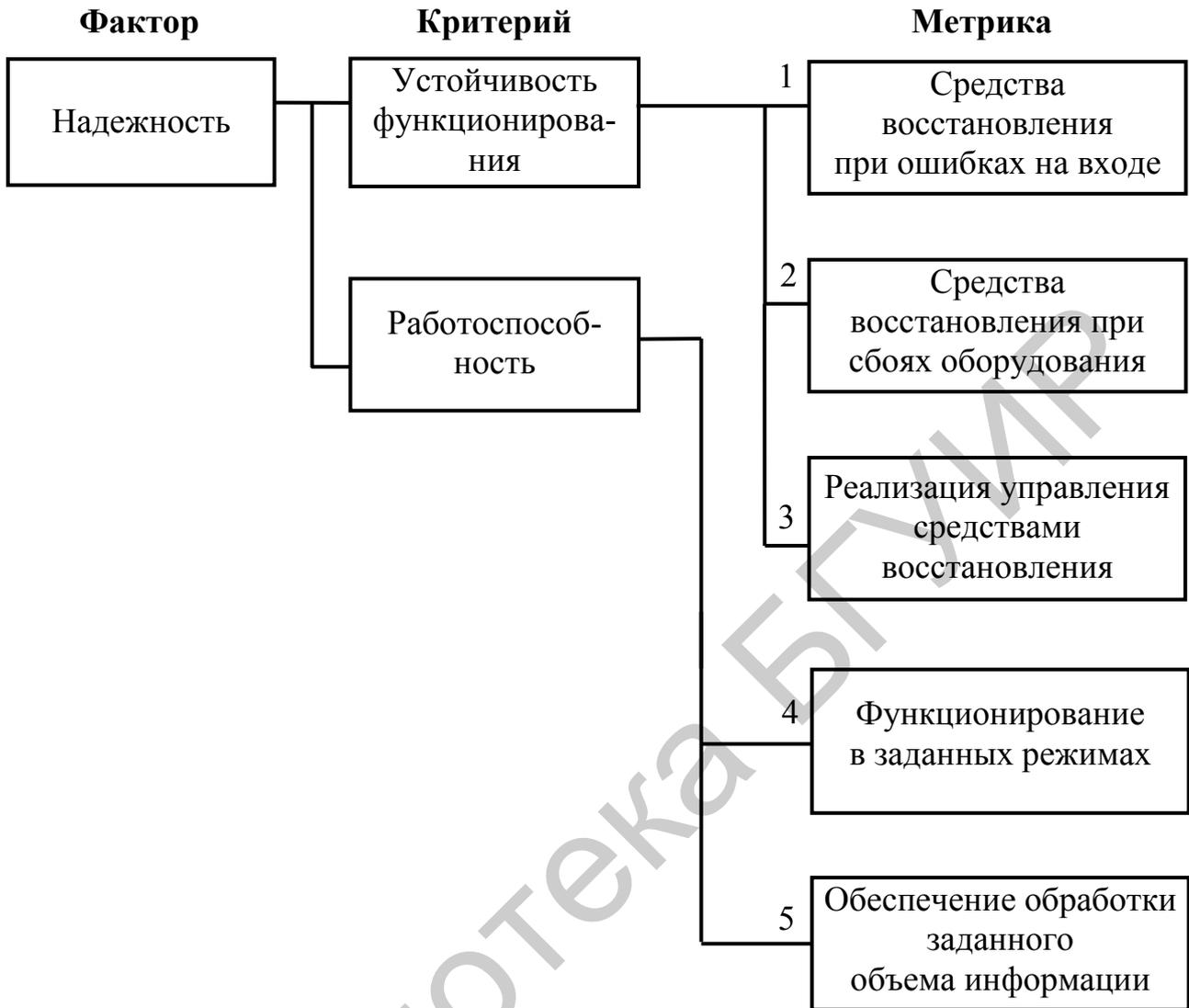


Рис. 10.8. Модель надежности для фаз реализации, тестирования, изготовления и сопровождения

Таблица 10.1

Оценочные элементы фактора *Надежность*

Код элемента	Наименование	Метод оценки	Оценка
H0101	Наличие требований к программе по устойчивости функционирования при наличии ошибок во входных данных	Экспертный	0 – 1
H0102	Возможность обработки ошибочных ситуаций	Экспертный	0 – 1
H0103	Полнота обработки ошибочных ситуаций	То же	0 – 1
H0104	Наличие тестов для проверки допустимых значений входных данных	»	0 – 1
H0105	Наличие системы контроля полноты входных данных	»	0 – 1
H0106	Наличие средств контроля корректности входных данных	»	0 – 1
H0107	Наличие средств контроля непротиворечивости входных данных	»	0 – 1
H0108	Наличие проверки параметров и адресов по диапазону их значений	»	0 – 1
H0109	Наличие обработки граничных результатов	»	0 – 1
H0110	Наличие обработки неопределенностей	»	0 – 1
H0201	Наличие требований к программе по восстановлению процесса выполнения в случае сбоя операционной системы, процессора, внешних устройств	»	0 – 1
H0202	Наличие требований к программе по восстановлению результатов при отказах процессора, операционной системы	»	0 – 1

Код элемента	Наименование	Метод оценки	Оценка
Н0203	Наличие средств восстановления процесса в случае сбоя оборудования	»	0 – 1
Н0204	Наличие возможности разделения по времени выполнения отдельных функций программ	»	0 – 1
Н0205	Наличие возможности повторного старта с точки останова	»	0 – 1
Н0301	Наличие централизованного управления процессами, конкурирующими из-за ресурсов	»	0 – 1
Н0302	Наличие возможности автоматически обходить ошибочные ситуации в процессе вычисления	»	0 – 1
Н0303	Наличие средств, обеспечивающих завершение процесса решения в случаях помех	»	0 – 1
Н0304	Наличие средств, обеспечивающих выполнение программы в сокращенном объеме в случае ошибок или помех	»	0 – 1
Н0305	Показатель устойчивости к искажающим воздействиям $P(Y)$	Регистрационный + Расчетный	$P(Y) = 1 - D/K$, где D – число экспериментов, в которых искажающие воздействия приводят к отказу; K – число экспериментов, в которых имитируются искажающие воздействия; Y – вид искажающего воздействия

Код элемента	Наименование	Метод оценки	Оценка
Н0401	Вероятность безотказной работы P	Регистрационный + Расчетный	$P = 1 - Q/N$, где N – число экспериментов; Q – число зарегистрированных отказов
Н0501	Оценка по среднему времени восстановления Q	Измерительный + Расчетный	$Q = \begin{cases} 1, & \text{если } T_{\text{в}} \leq T_{\text{в доп}}; \\ T_{\text{в доп}} / T_{\text{в}}, & \text{если } T_{\text{в}} > T_{\text{в доп}}, \end{cases}$ <p>где $T_{\text{в}}$ – среднее время восстановления:</p> $T_{\text{в}} = \left(\sum_{i=1}^N T_{\text{в}i} \right) / N;$ <p>N – число восстановлений; $T_{\text{в}i}$ – время восстановления после i-го отказа; $T_{\text{в доп}}$ – допустимое среднее время восстановления</p>
Н0502	Оценка по продолжительности преобразования i -го входного набора данных в выходной Q_{ni}	Измерительный + Расчетный	$Q_{ni} = \begin{cases} 1, & \text{если } T_{ni} \leq T_{ni \text{ доп}}; \\ T_{ni \text{ доп}} / T_{ni}, & \text{если } T_{ni} > T_{ni \text{ доп}}, \end{cases}$ <p>где $T_{ni \text{ доп}}$ – допустимое время преобразования i-го входного набора данных; T_{ni} – фактическая продолжительность преобразования i-го входного набора данных</p>

Оценка качества программного средства проводится в следующей последовательности:

1. На фазе анализа выполняется выбор показателей и их базовых значений.

2. Для показателей качества на всех уровнях принимается единая шкала оценки (от 0 до 1).

3. В процессе оценки качества на каждом уровне (кроме уровня оценочных элементов) проводится вычисление двух величин:

- абсолютного показателя качества P_{ij} ;
- относительного показателя качества R_{ij} ,

где j – порядковый номер показателя данного уровня для i -го показателя вышестоящего уровня.

Относительный показатель качества R_{ij} является функцией показателя P_{ij} и его базового значения P_{ij}^b и определяется по формуле

$$R_{ij} = \frac{P_{ij}}{P_{ij}^b}.$$

ГОСТ 28195–99 содержит таблицу с базовыми значениями для характеристик качества второго уровня (критериев). Данные значения определяются подклассом программного средства в соответствии с ОКП.

Базовые значения для показателей первого и третьего уровней формируются методом экспертного опроса с учетом назначения ПС или на основании показателей существующих аналогов или расчетного эталонного ПС. Значения базовых показателей должны соответствовать значениям показателей, отражающих современный уровень качества и прогнозируемый мировой уровень.

4. Каждый показатель качества второго и третьего уровней характеризуется двумя параметрами:

- количественным значением;
- весовым коэффициентом V_{ij} .

Сумма весовых коэффициентов всех показателей некоторого уровня, относящихся к показателю вышестоящего уровня, постоянна и равна 1:

$$\sum_{j=1}^J V_{ij} = 1,$$

где J – общее количество всех показателей j -го уровня, относящихся к i -му показателю вышестоящего уровня, определенных в стандарте.

ГОСТ 28195–99 содержит таблицы, содержащие перечни весовых коэффициентов для характеристик второго и третьего уровней (критериев и метрик). Количественные величины весовых коэффициентов зависят от фазы ЖЦ ПС и подкласса ПС в соответствии с ОКП.

5. Определение усредненной оценки m_{kq} оценочного элемента по нескольким его значениям (измерениям) m_{qt} осуществляется по следующей формуле (формула для вычисления значений показателей качества 4-го уровня):

$$m_{kq} = \frac{1}{T} \sum_{t=1}^T m_{qt},$$

где k – порядковый номер метрики; q – порядковый номер оценочного элемента; T – число значений (измерений) оценочного элемента; t – номер значения оценочного элемента.

6. Итоговая оценка k -й метрики j -го критерия определяется по формуле (формула для вычисления значений показателей качества 3-го уровня):

$$P_{jk}^M = \frac{1}{Q} \sum_{q=1}^Q m_{kq},$$

где M – признак метрики; Q – число оценочных элементов, реально используемых при оценке k -й метрики.

7. Абсолютные показатели j -го критерия i -го фактора вычисляются по формуле (формула для вычисления значений показателей качества 2-го уровня):

$$P_{ij}^K = \frac{\sum_{k=1}^n (P_{jk}^M \cdot V_{jk}^M)}{\sum_{k=1}^n V_{jk}^M},$$

где n – число метрик, относящихся к j -му критерию, реально используемых при оценке; K – признак критерия.

8. Относительные значения R_{ij}^K j -го критерия i -го фактора P_{ij}^K по отношению к базовому значению P_{ij}^{Kb} определяются по формуле

$$R_{ij}^K = \frac{P_{ij}^K}{P_{ij}^{Kb}}.$$

9. Абсолютные и относительные значения i -го фактора качества определяются по формулам (формулы для вычисления значений показателей качества 1-го уровня)

$$P_i^F = \frac{\sum_{j=1}^N (P_{ij}^K \cdot V_{ij}^K)}{\sum_{j=1}^N V_{ij}^K}, \quad R_i^F = \frac{\sum_{j=1}^N (R_{ij}^K \cdot V_{ij}^K)}{\sum_{j=1}^N V_{ij}^K},$$

где F – признак фактора; N – число критериев качества, относящихся к i -му фактору, реально используемых при оценке.

10. Общая оценка качества в целом формируется экспертами по набору полученных значений факторов качества.

К *достоинствам* метода оценки качества, определенного в *ГОСТ 28195–99*, можно отнести:

1. Метод позволяет накапливать статистический материал о состоянии различных подклассов ПС в отношении значений метрик и оценочных элементов. Это создает предпосылки для определения их нормативных (базовых) значений по подклассам ПС и может служить основой для деятельности по стандартизации в области программного обеспечения.

2. Списки значений метрик и оценочных элементов являются основой для деятельности по управлению качеством в процессе разработки ПС.

3. Возможно создание инструментальных средств с целью автоматизации оценки качества ПС.

Библиотека БГУИР

11. ИСТОРИЯ РАЗВИТИЯ СТАНДАРТИЗАЦИИ КАЧЕСТВА ПРОГРАММНЫХ СРЕДСТВ ЗА РУБЕЖОМ

11.1. Стандарты ISO/IEC в области оценки качества программных средств

В течение десяти лет (с 1991 по 2001 г.) основой регламентирования характеристик качества ПС за рубежом являлся международный стандарт *ISO/IEC 9126:1991. Информационная технология. Оценка программного продукта. Характеристики качества и руководства по их применению* [58]. В подразд. 10.3, 10.4 приведены положения стандарта *СТБ ИСО/МЭК 9126–2003* [15], являющегося аутентичным переводом вышеназванного международного стандарта.

В 2001–2004 гг. стандарт *ISO/IEC 9126:1991* был заменен на две взаимосвязанные серии стандартов: *ISO/IEC 9126–1–4:2001–2004* и *ISO/IEC 14598–1–6:1998–2001* [59 – 62, 28 – 33].

Как и в *ISO/IEC 9126:1991*, в серии стандартов *ISO/IEC 9126–1–4:2001–2004* регламентирована иерархическая модель качества программных средств. На верхнем уровне модели находятся характеристики. Характеристики разделяются на подхарактеристики. Подхарактеристики определяются метриками. Метрики измеряют атрибуты (свойства) ПС.

Данная серия стандартов состоит из четырех частей под общим названием *Программная инженерия. Качество продукта* [59 – 62]:

- *ISO/IEC 9126–1:2001. Часть 1: Модель качества;*
- *ISO/IEC TR 9126–2:2003. Часть 2: Внешние метрики;*
- *ISO/IEC TR 9126–3:2003. Часть 3: Внутренние метрики;*
- *ISO/IEC TR 9126–4:2004. Часть 4: Метрики качества в использовании.*

Первая часть *ISO/IEC 9126–1:2001* по существу является пересмотренной редакцией стандарта *ISO/IEC 9126:1991*. В данной части определены два верхних уровня (характеристики и подхарактеристики) иерархической модели качества, приведены общие требования к метрикам качества, даны рекомендации по их выбору. При этом сохранена та же номенклатура из шести базовых характеристик качества ПС, что и в *ISO/IEC 9126:1991* (см. подразд. 10.3), к подхарактеристикам качества добавлено несколько новых подхарактеристик.

Однако в отличие от *ISO/IEC 9126:1991* подхарактеристики второго уровня в *ISO/IEC 9126–1:2001* стали нормативными, а не рекомендуемыми, определены две части модели качества (модель внешнего и внутреннего

качества и модель качества в использовании) и исключен процесс оценки качества (он перенесен в серию стандартов *ISO/IEC 14598-1-6:1998-2001*). Модель внутреннего и внешнего качества и модель качества в использовании рассмотрены в подразд. 11.3, 11.4.

В *ISO/IEC 9126-1:2001* регламентированы следующие **виды метрик**:

- внутренние метрики;
- внешние метрики;
- метрики качества в использовании.

В 2011 г. *ISO/IEC 9126-1:2001* заменен на стандарт *ISO/IEC 25010:2011* серии *SQuaRE* [41]. Серия *SQuaRE* описана в разд. 12.

Вторая – четвертая части *ISO/IEC TR 9126-2-4:2003-2004* опубликованы в виде технических отчетов (TR). Совокупности метрик, перечисленные в данных частях, являются рекомендуемыми, их набор не является исчерпывающим. Метрики могут модифицироваться. Возможно применение метрик, не включенных в данные части. В этих частях стандарта содержатся желательные свойства и критерии обоснованности метрик, пояснения к их применению, к типам шкал метрик и типам измерений, примеры метрик для каждой подхарактеристики, примеры применения метрик на протяжении жизненного цикла ПС.

Части *ISO/IEC TR 9126-2-4:2003-2004* в настоящее время являются действующими.

Во второй части *ISO/IEC TR 9126-2:2003* определены метрики внешнего качества ПС. **Внешние метрики** – это метрики, предназначенные для измерения качества программного продукта путем измерения поведения системы, частью которой является данный продукт. Внешние метрики могут использоваться в процессе эксплуатации и на стадиях тестирования или испытаний в процессах разработки и сопровождения ПС, когда уже созданы исполнимые коды программного продукта. Примеры внешних метрик из *ISO/IEC TR 9126-2:2003* приведены в подразд. 14.3.

В третьей части *ISO/IEC TR 9126-3:2003* определены метрики внутреннего качества ПС. **Внутренние метрики** – это метрики, измеряющие собственные свойства ПС. Они измеряются в процессе разработки программного продукта на основе спецификации требований, результатов проектирования, исходного кода или другой документации ПС. Внутренние метрики дают возможность оценить качество промежуточных программных продуктов разработки, предсказывая качество конечного программного средства. Примеры внутренних метрик из *ISO/IEC TR 9126-3:2003* приведены в подразд. 14.2.

В настоящее время на основе *ISO/IEC TR 9126-2:2003* и *ISO/IEC TR 9126-3:2003* подкомитетом SC7 объединенного комитета JTC1 организаций ISO и IEC ведется разработка (стадия 60.00 на 31.03.2016 г.) стандарта *ISO/IEC 25023* серии *SQuaRE* (см. подразд. 1.1, 1.2, 12.4) [46].

В четвертой части *ISO/IEC TR 9126–4:2004* определены метрики качества в использовании. **Метрики качества в использовании** – это метрики, измеряющие соответствие продукта потребностям заданных пользователей в достижении заданных целей с результативностью, продуктивностью, безопасностью и удовлетворением в заданных контекстах использования. Данные метрики могут использоваться только в процессе эксплуатации ПС в реальной среде окружения. Метрики качества в использовании основаны на измерении поведения типичных пользователей и системы, содержащей данное программное средство. Примеры метрик качества в использовании приведены в подразд. 14.4.

В настоящее время на основе *ISO/IEC TR 9126–4:2004* подкомитетом SC7 ведется разработка (стадия 60.00 на 31.03.2016 г.) стандарта *ISO/IEC 25022* серии *SQuaRE* (см. подразд. 1.1, 1.2, 12.4) [45].

В серии стандартов *ISO/IEC 14598–1–6:1998–2001* определены процессы оценки качества программного продукта, содержатся руководство и требования к оценке. Данная серия предназначена для применения при разработке, приобретении и независимой оценке программного средства. Серия состоит из шести частей [28 – 33]:

- *ISO/IEC 14598–1:1999. Информационная технология. Оценка программного продукта. Часть 1: Общий обзор;*

- *ISO/IEC 14598–2:2000. Программная инженерия. Оценка продукта. Часть 2: Планирование и управление;*

- *ISO/IEC 14598–3:2000. Программная инженерия. Оценка продукта. Часть 3: Процесс для разработчиков;*

- *ISO/IEC 14598–4:1999. Программная инженерия. Оценка продукта. Часть 4: Процесс для заказчиков;*

- *ISO/IEC 14598–5:1998. Информационная технология. Оценка программного продукта. Часть 5: Процесс для оценщиков;*

- *ISO/IEC 14598–6:2001. Программная инженерия. Оценка продукта. Часть 6: Документация модулей оценки.*

В первой части *ISO/IEC 14598–1:1999* приведен обзор остальных частей, определена связь стандартов серии *ISO/IEC 14598* со стандартами серии *ISO/IEC 9126* и стандартом *ISO/IEC 12207:1995*. В данной части содержатся общие требования к спецификации и оценке качества, разъяснены концепции оценки. Установлены требования к методам измерений и оценки программных продуктов. Определен общий процесс оценки качества программного продукта. Основой для данного процесса оценки явился процесс оценки качества ПС из *ISO/IEC 9126:1991* (данный процесс описан в подразд. 10.4).

В 2011 г. *ISO/IEC 14598–1:1999* заменен на стандарт *ISO/IEC 25040:2011* серии *SQuaRE* (см. подразд. 12.6) [49].

Вторая часть *ISO/IEC 14598–2:2000* содержит требования и руководство по поддержке оценки. В данной части приведены концепции планирования и управления процессом оценки качества программного

продукта, рассмотрено содержание плана количественной оценки качества. Эта часть стандарта предназначена для применения на уровне организации или ее подразделений.

В 2007 г. *ISO/IEC 14598–2:2000* заменен на первую редакцию стандарта *ISO/IEC 25001:2007* серии *SQuaRE*, которая в свою очередь в 2014 г. заменена второй редакцией *ISO/IEC 25001:2014* (см. п. 12.2.3) [39, 40].

Третья часть ISO/IEC 14598–3:2000 предназначена для организаций – разработчиков ПС. В ней приведены концепции оценки и требования к процессу оценки. Данная часть ориентирована на выполнение оценки программного продукта, используя собственный технический персонал. Внимание в данной части сконцентрировано на оценках, которые могут предсказать качество конечного программного продукта на основе измерений промежуточных продуктов процесса разработки.

Четвертая часть ISO/IEC 14598–4:1999 предназначена для организаций, которые планируют приобретать готовый или разрабатываемый программный продукт. В ней определена связь работ процесса заказа из стандарта *ISO/IEC 12207:1995* с работами, выполняемыми при оценке программного продукта. Рассмотрены особенности выполнения оценки для имеющегося в наличии готового программного продукта, для приемки заказного программного продукта и для выбора из альтернативного числа готовых программных продуктов.

Пятая часть ISO/IEC 14598–5:1998 предназначена для использования оценщиком, выполняющим независимую оценку программного продукта. Как правило, персонал оценки работает в независимой организации. В данной части приведены концепции оценки и требования к процессу оценки. Рассмотрена структура отчета об оценке. Приведены рекомендации по выбору уровней ранжирования при проведении измерений. Оценка качества программного продукта оценщиком может выполняться по запросу разработчика, заказчика (покупателя) или другой стороны.

В 2012 г. части *ISO/IEC 14598–3–5:1998–2000* заменены на стандарт *ISO/IEC 25041:2012* серии *SQuaRE* (см. подразд. 12.6) [50].

Шестая часть ISO/IEC 14598–6:2001 предназначена для поддержки оценки программного продукта и содержит руководство по документированию модулей оценки. **Модуль оценки** представляет собой полностью укомплектованную информацию, необходимую для проведения процесса оценки некоторой характеристики или подхарактеристики качества. Модуль содержит спецификацию соответствующей модели качества (характеристика, подхарактеристики, метрики качества), методики и процедуры оценки, входные данные для оценки, информацию о запланированном применении модели и о ее фактическом применении, структуру типового отчета о результатах выполненной оценки. Рассмотрен ряд примеров модулей оценки. Для каждого процесса оценки должны выбираться соответствующие модули оценки. Данная часть стандарта предназначена для использования организациями,

разрабатывающими модули оценки и производящими оценку программных продуктов.

Часть *ISO/IEC 14598-6:2001* в настоящее время находится в стадии пересмотра (стадия 90.92, см. подразд. 1.1, 1.2).

11.2. Связь качества программного средства с его жизненным циклом

В течение жизненного цикла программного средства его качество изменяется. Существует несколько точек зрения на качество ПС в течение его ЖЦ. От их выбора зависит как оценка качества ПС, так и управление качеством на каждой стадии жизненного цикла.

С учетом этого для различных стадий ЖЦ в стандарте *ISO/IEC 9126-1:2001* определены следующие *виды качества программных средств* [59]:

1. *Потребности пользователя в качестве* определяются как требования к качеству, выраженные в терминах метрик качества в использовании, внешних и иногда внутренних метрик. Эти требования должны применяться как критерии при аттестации продукта. Получение программного продукта, удовлетворяющего потребностям пользователя, обычно требует итеративного подхода к разработке программного средства с постоянной обратной связью с потенциальным пользователем.

2. *Требования к внешнему качеству* определяют требуемый уровень качества с внешней точки зрения. Они включают требования, вытекающие из потребностей пользователя в качестве, включая требования к качеству в использовании. Требования к внешнему качеству применяются как цель при аттестации продукта на различных стадиях разработки. Данные требования для всех характеристик качества, определенных в *ISO/IEC 9126-1:2001*, во-первых, должны быть выражены в спецификации требований к качеству, используя внешние метрики, во-вторых, должны быть преобразованы в требования к внутреннему качеству и, в-третьих, должны использоваться как критерии при оценке продукта.

3. *Требования к внутреннему качеству* определяют требуемый уровень качества с внутренней точки зрения на программный продукт. Они используются для определения свойств промежуточных продуктов разработки. Промежуточные продукты могут включать статические и динамические модели, другие документы и исходный код программного продукта. Требования к внутреннему качеству могут использоваться как цель при аттестации продукта на различных стадиях разработки. Они могут использоваться для определения стратегий разработки, а также критериев оценки и верификации в течение разработки. Требования к внутреннему качеству должны определяться количественно, используя внутренние метрики.

4. *Внутреннее качество* – совокупность характеристик программного продукта с внутренней точки зрения. Внутреннее качество измеряется с помощью внутренних метрик и оценивается по отношению к требованиям к внутреннему качеству. Отдельные элементы качества программного продукта могут улучшаться при реализации кода, проверке или тестировании, но фундаментальная основа качества программного продукта, представленная внутренним качеством, остается неизменной до повторного проектирования.

5. *Оценочное (или прогнозируемое) внешнее качество* – оцененное или предсказанное качество конечного программного продукта на каждой стадии процесса разработки для каждой характеристики качества, основанное на знании внутреннего качества.

6. *Внешнее качество* – совокупность характеристик программного продукта с внешней точки зрения. Это качество, измеряемое и оцениваемое на основе внешних метрик при выполнении программного продукта во время тестирования (испытаний) в моделируемой среде с моделируемыми данными или во время эксплуатации.

7. *Оценочное (или прогнозируемое) качество в использовании* – оцененное или предсказанное качество конечного программного продукта на каждой стадии процесса разработки для каждой характеристики качества в использовании, основанное на знании внутреннего и внешнего качества.

8. *Качество в использовании* – качество программного продукта, применяемого в заданной среде и заданном контексте использования, с точки зрения пользователя. Оно оценивается на основе метрик качества в использовании и в первую очередь измеряет степень достижения пользователем своих целей в конкретной среде, а не свойства самого программного продукта. Пользователь оценивает только те атрибуты программного продукта, которые он применяет в своих задачах.

Изменение и взаимосвязь различных видов качества в жизненном цикле программных средств показаны на рис. 11.1 [28].

11.3. Модель внешнего и внутреннего качества программных средств ISO/IEC 9126–1:2001

Как уже было отмечено, в стандарте *ISO/IEC 9126–1:2001* регламентированы две части модели качества программных средств: модель внешнего и внутреннего качества и модель качества в использовании. Данные модели различаются в зависимости от представления качества в ЖЦ ПС (см. рис. 11.1). Эти модели могут быть использованы, например, в следующих случаях:

- проверка полноты определения требований;

- определение требований к ПС;
- определение целей проектирования ПС;
- определение целей испытаний ПС;
- определение критериев обеспечения качества;
- определение критериев приемки завершеного ПС.

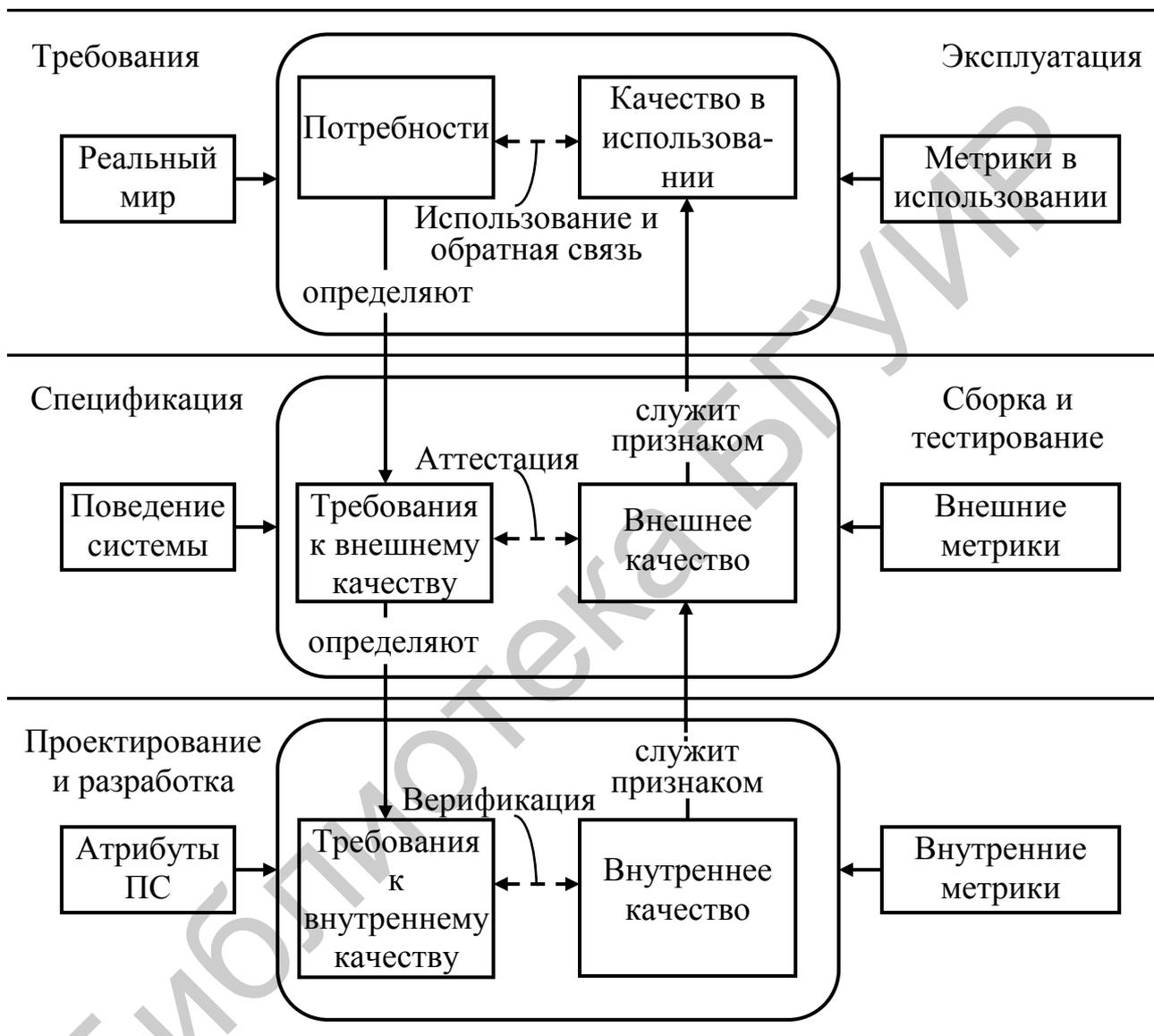


Рис. 11.1. Качество в жизненном цикле программных средств по ISO/IEC 14598–1:1999

На рис. 11.2 приведена модель внешнего и внутреннего качества программных средств, регламентированная стандартом *ISO/IEC 9126–1:2001*. На данном рисунке подчеркнуты подхарактеристики, новые или отличающиеся по названию от модели качества *ISO/IEC 9126:1991*.

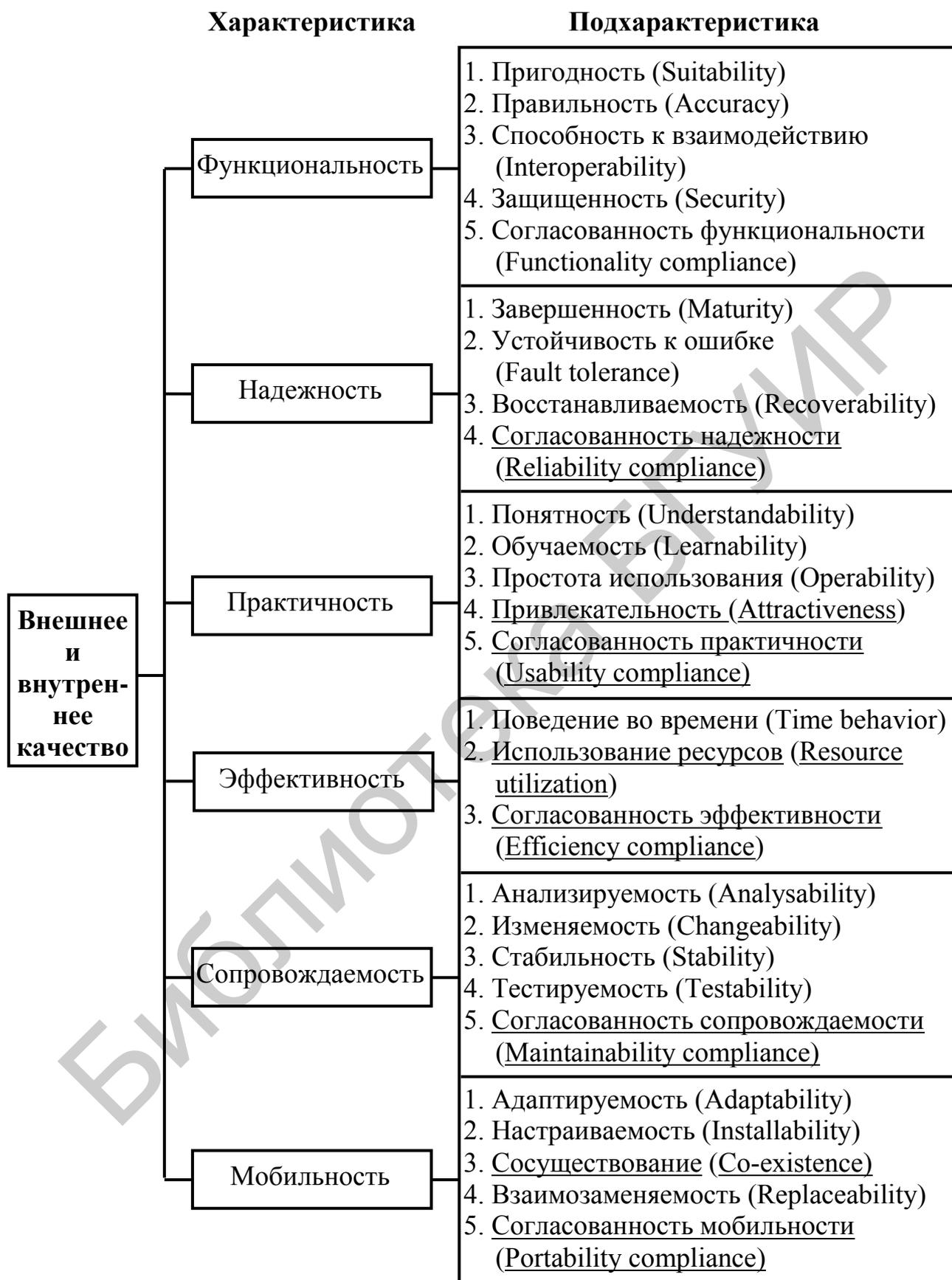


Рис. 11.2. Модель внешнего и внутреннего качества по ISO/IEC 9126–1:2001

Как видно из рис. 11.2 и 10.2, в модели внешнего и внутреннего качества стандарта *ISO/IEC 9126-1:2001* сохранена номенклатура характеристик качества верхнего уровня из стандарта *ISO/IEC 9126:1991*. Это *Функциональность*, *Надежность*, *Практичность*, *Эффективность*, *Сопровождаемость*, *Мобильность*.

Подхарактеристики, находящиеся на втором уровне, в *ISO/IEC 9126-1:2001* из разряда рекомендуемых перешли в разряд обязательных. Они могут измеряться внутренними или внешними метриками.

Сравнение уровня подхарактеристик модели внешнего и внутреннего качества *ISO/IEC 9126-1:2001* и модели качества *ISO/IEC 9126:1991* показывает, что изменения на данном уровне минимальны и заключаются в следующем.

Отличие 1

Во все характеристики качества стандарта *ISO/IEC 9126-1:2001* введены подхарактеристики *Согласованность (Согласованность функциональности (Functionality compliance), Согласованность надежности (Reliability compliance), Согласованность практичности (Usability compliance), Согласованность эффективности (Efficiency compliance), Согласованность сопровождаемости (Maintainability compliance), Согласованность мобильности (Portability compliance))*.

Согласованность характеристики – это способность программного продукта соответствовать стандартам или соглашениям, связанным с данной характеристикой.

В стандарте *ISO/IEC 9126:1991* характеристика *Согласованность* присутствовала только в характеристиках *Функциональность* и *Мобильность*. При этом в характеристике *Мобильность* данная подхарактеристика имела название *Соответствие (Conformance, см. рис. 10.2)*.

Отличие 2

В характеристике *Практичность* введена новая подхарактеристика *Привлекательность*.

Привлекательность (Attractiveness) – способность программного продукта нравиться пользователю. Данная подхарактеристика связана со свойствами оформления программного продукта (например, использование цветов, графики и т. п.).

Отличие 3

Изменено название одной из подхарактеристик характеристики *Эффективность*.

В стандарте *ISO/IEC 9126:1991* данная характеристика называлась *Resource behavior (Характер изменения ресурсов в СТБ ISO/IEC 9126-2003, см. рис. 10.2)*.

В *ISO/IEC 9126-1:2001* для обозначения этой характеристики применяется название *Resource utilization (Использование ресурсов, см. рис. 11.2)*.

Отличие 4

В характеристике *Мобильность* введена новая подхарактеристика *Сосуществование*.

Сосуществование (Co-existence) – способность программного продукта к сосуществованию с другими независимыми программными средствами в общей среде, разделяя общие ресурсы.

Остальные подхарактеристики внешнего и внутреннего качества стандарта *ISO/IEC 9126–1:2001* по названию и по сути соответствуют подхарактеристикам качества из *ISO/IEC 9126–1:2001* (см. подразд. 10.3).

11.4. Модель качества в использовании ISO/IEC 9126–1:2001

Второй частью модели качества, определенной в стандарте *ISO/IEC 9126–1:2001* [59], является модель качества в использовании. Достижение качества в использовании зависит от достижения необходимого внешнего качества, которое, в свою очередь, зависит от достижения необходимого внутреннего качества (см. рис. 11.1).

Качество в использовании (quality in use) – это способность программного продукта позволять заданным пользователям достигать заданные цели с результативностью, продуктивностью, безопасностью и удовлетворением в заданном контексте использования.

Качество в использовании – это восприятие пользователем качества окружающей среды, содержащей программный продукт. Оно измеряется больше на основе результатов использования программного средства в среде, чем на основе собственных внутренних свойств программного продукта.

Модель качества в использовании является иерархической. На ее верхнем уровне находятся четыре характеристики. Характеристики определяются непосредственно метриками (второй уровень модели). В отличие от модели внешнего и внутреннего качества (см. рис. 11.2) уровень подхарактеристик в модели качества в использовании отсутствует.

На рис. 11.3 приведен верхний уровень (уровень характеристик) модели качества в использовании. В соответствии с данной моделью качество в использовании разделяется на четыре характеристики: *Результативность*, *Продуктивность*, *Безопасность*, *Удовлетворенность*.

Результативность (Effectiveness) – это способность программного продукта обеспечивать достижение пользователями заданных целей с точностью и полнотой в заданном контексте использования.

Продуктивность (Productivity) – это способность программного продукта обеспечивать расходование пользователями количества ресурсов, соответствующего результативности, достигаемой в заданном контексте

использования. Ресурсы могут включать время выполнения задачи, усилия пользователя, материалы, стоимость использования.



Рис. 11.3. Модель качества в использовании по ISO/IEC 9126–1:2001

Безопасность (Safety) – это способность программного продукта достигать приемлемых уровней риска причинения вреда людям, бизнесу, программному обеспечению, имуществу или окружающей среде в заданном контексте использования. Обычно *риски* – это результат оставшихся дефектов в функциональности (включая защищенность), надежности, практичности и сопровождаемости.

Удовлетворенность (Satisfaction) – это способность программного продукта удовлетворять пользователя в заданном контексте использования. Удовлетворенность определяется реакцией пользователя на взаимодействие с программным продуктом и включает отношение к применению данного продукта.

11.5. Метод оценки качества программных средств по ISO/IEC 14598–1:1999

Стандарт *ISO/IEC 14598–1:1999* [28] регламентирует метод оценки качества программных средств, который основан на иерархических моделях качества, определенных в *ISO/IEC 9126–1:2001* (см. подразд. 11.3, 11.4).

Данный метод оценки качества является развитием метода оценки качества программных средств, регламентированного в *ISO/IEC 9126:1991* и описанного в подразд. 10.4, и имеет с последним много общих черт. В обоих методах процесс оценки качества разделяется на стадии и этапы.

Процесс оценки качества программных средств, определенный в *ISO/IEC 14598–1:1999*, приведен на рис. 11.4 [28]. Процесс оценки состоит из *четырёх*

стадий: установка требований к оценке, определение оценки, проектирование оценки и выполнение оценки. Данный процесс может применяться после любой подходящей работы жизненного цикла для промежуточных или конечного продукта разработки (сопровождения).

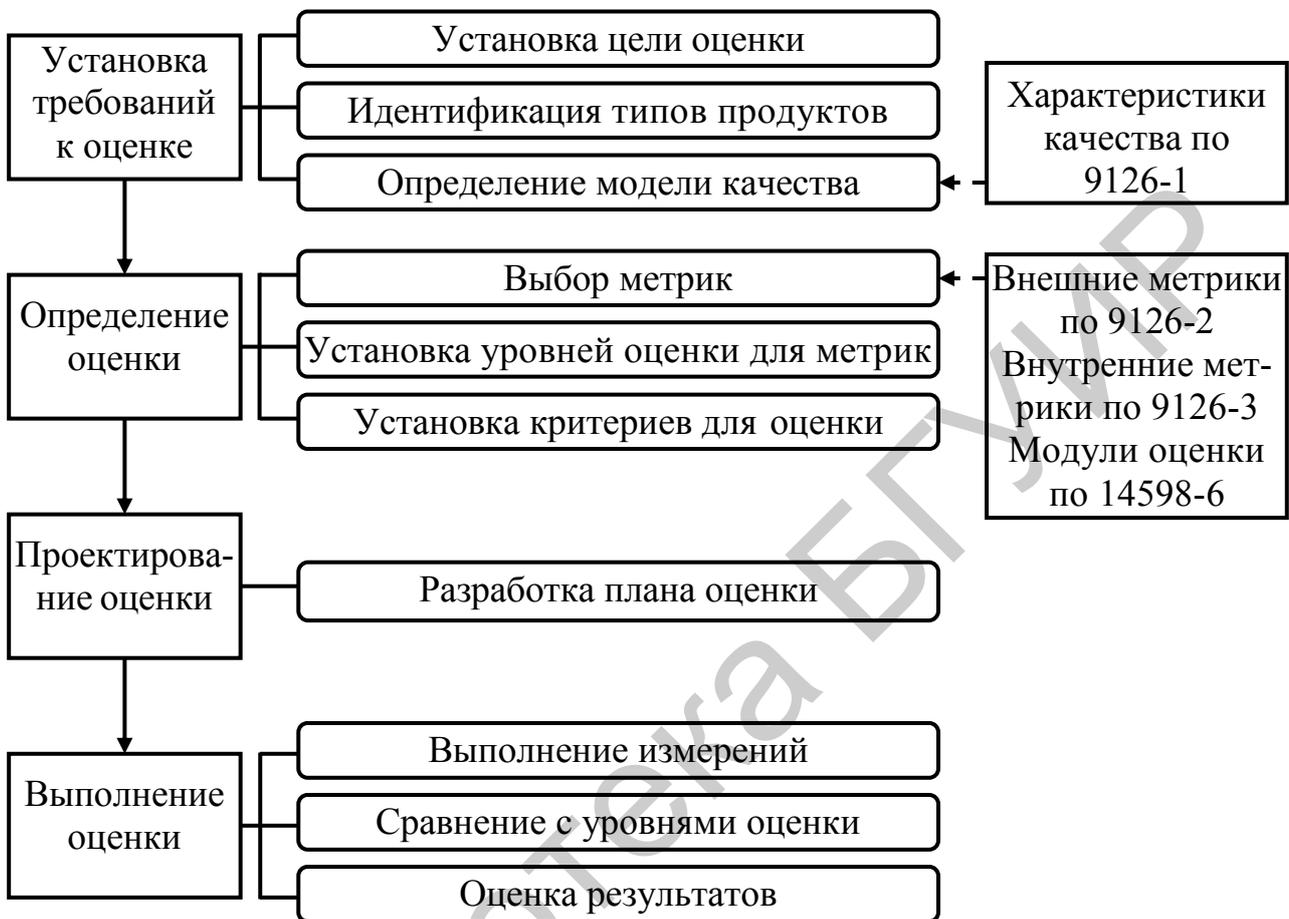


Рис. 11.4. Процесс оценки качества по ISO/IEC 14598–1:1999

Стадия 1. Установка требований к оценке

Данная стадия состоит из трех этапов.

Этап 1.1. Установка цели оценки

Общей целью оценки качества программного средства является поддержка разработки и приобретения программного средства, удовлетворяющего заявленные и подразумеваемые потребности пользователей. *Конечная цель* состоит в том, чтобы гарантировать, что продукт обеспечивает требуемое качество.

Целью оценки качества промежуточного продукта может быть:

- решение о принятии промежуточного продукта от субподрядчика;
- решение о завершении процесса и передаче продукта следующему процессу;
- прогноз или предварительная оценка качества конечного продукта;

- сбор информации о промежуточных продуктах для контроля и управления процессом.

Целью оценки качества конечного продукта может быть:

- решение о принятии продукта;
- решение о выпуске продукта;
- сравнение продукта с конкурентными продуктами;
- выбор продукта из числа альтернативных продуктов;
- оценка положительного и отрицательного результата использования продукта;
- решение о сроках улучшения или замены продукта.

Качество программного средства может оцениваться в процессах жизненного цикла, определенных в *СТБ ИСО/МЭК 12207–2003 (ISO/IEC 12207:2008)*, см. разд. 6).

В процессе заказа программного продукта заказчик должен установить требования к внешнему качеству программного продукта, определить требования к поставщику и оценить качество разработанного программного продукта по этим требованиям перед его приемкой. Оценка качества программного средства заказчиком детально рассмотрена в *ISO/IEC 14598–4:1999*, замененном в настоящее время стандартом *ISO/IEC 25041:2012* серии *SQuaRE* [50].

При покупке готового программного продукта оценка может использоваться для сравнения альтернативных программных продуктов и гарантии, что выбранный программный продукт удовлетворяет требованиям к качеству. Оценка качества программного средства оценщиком детально рассмотрена в стандарте *ISO/IEC 14598–5:1998*, также замененном в настоящее время стандартом *ISO/IEC 25041:2012* серии *SQuaRE* [50].

В процессе поставки поставщик может использовать результаты оценки программного продукта, чтобы гарантировать, что программный продукт удовлетворяет требуемым критериям качества как в сравнении с другими программными продуктами, так и установленным заказчиком.

В процессе разработки на основании требований к внешнему качеству программного продукта, установленных заказчиком, разработчик для каждого компонента программного продукта должен определить спецификацию требований к внешнему качеству в терминах характеристик и подхарактеристик (см. подразд. 10.3, 11.3).

На базе спецификации требований к внешнему качеству разработчик должен определить требования к внутреннему качеству программного продукта. Эти требования должны использоваться на начальных этапах процесса разработки (когда еще нет исполнимых кодов программного продукта или его компонентов) для проверки качества промежуточных продуктов с целью прогноза и предварительной оценки качества конечного программного продукта. На последующих этапах процесса разработки (когда уже имеются исполнимые коды модулей, компонентов или всего программного продукта)

должны использоваться требования к внешнему качеству. На их основе должна выполняться оценка внешнего качества промежуточных продуктов и конечного программного продукта (с целью прогноза итогового внешнего качества в среде эксплуатации). С этой целью обычно используется моделируемая среда с моделируемыми данными. Оценка качества программного продукта разработчиком рассмотрена в стандарте *ISO/IEC 14598-3:2000*, также замененном в настоящее время стандартом *ISO/IEC 25041:2012* серии *SQuaRE* [50].

В процессе эксплуатации оценка качества программного средства может использоваться для подтверждения того, что требования к качеству удовлетворяются при различных условиях эксплуатации.

В процессе сопровождения оценка программного средства может использоваться для подтверждения того, что требования к качеству по-прежнему удовлетворяются и требования по сопровождаемости и мобильности достигаются.

Этап 1.2. Идентификация типов продуктов

Тип оцениваемого промежуточного или конечного программного продукта зависит от цели оценки и стадии жизненного цикла программных средств (см. рис. 11.1). Например, в процессе разработки интерес представляет оценка промежуточного продукта. На ранних этапах процесса разработки это может быть спецификация требований, архитектура программного средства, технический проект программного средства, исходные коды модулей. Для них выполняется оценка внутреннего качества с целью прогноза внешнего качества.

На последующих этапах процесса разработки промежуточными продуктами являются исполнимые коды модулей, компонентов и других промежуточных продуктов сборки, а также конечный программный продукт. Для них выполняется оценка внешнего качества в моделируемой среде с моделируемыми данными с целью прогноза внешнего качества программного средства в среде эксплуатации.

В процессе эксплуатации оценивается система, частью которой является программный продукт. В этом случае может выполняться, во-первых, оценка внешнего качества при использовании программного средства в окружающей среде с целью подтверждения соответствия требованиям к внешнему качеству и прогноза качества программного средства в использовании и, во-вторых, оценка качества в использовании для подтверждения удовлетворения потребностей пользователя в выполнении заданных задач в заданных аппаратных и операционных средах.

Этап 1.3. Определение модели качества

На данном этапе, исходя из типов оцениваемых продуктов, выбирается соответствующая модель качества (внутреннего, внешнего или качества в использовании). Основу модели составляет модель качества из стандарта *ISO/IEC 9126-1:2001* (см. подразд. 11.3, 11.4). Выбранная модель должна быть адаптирована с учетом целей оценки и конкретных требований к качеству оцениваемого продукта. В процессе адаптации из модели качества стандарта

ISO/IEC 9126–1:2001 должны быть выбраны соответствующие характеристики и подхарактеристики, которые будут оцениваться.

Стадия 2. Определение оценки

Данная стадия также состоит из трех этапов.

Этап 2.1. Выбор метрик

На данном этапе, исходя из разработанной модели качества программного средства, выбираются соответствующие метрики качества. Каждое измеримое внутреннее или внешнее свойство продукта, влияющее на значение характеристики или подхарактеристики качества, может быть установлено как метрическое. При выборе метрик следует учитывать простоту и экономические факторы их использования. Примеры метрик качества приведены в стандартах *ISO/IEC TR 9126–2:2003*, *ISO/IEC TR 9126–3:2003*, *ISO/IEC TR 9126–4:2004* [60 – 62]. Метрики качества описаны в разд. 13.

На выбор метрик оказывает влияние также тип требуемых измерений, который зависит от цели оценки. Если целью оценки является исправление недостатков разработки, то для контроля достаточно выполнить несколько измерений продукта. Данные измерения могут быть дополнены метриками (мерами) из имеющихся контрольных списков или мнениями экспертов.

Если же целью оценки является сравнение готовых программных продуктов или анализ соответствия продукта требованиям к качеству, то необходимо использовать строгие метрики с достаточной точностью измерений. При этом должны быть учтены ошибки в измерениях, вызванные человеческим фактором или инструментом измерения.

Кроме того, для осуществления прогноза качества программного средства важно, чтобы выбранные внутренние метрики коррелировали с некоторыми аспектами внешнего качества, а выбранные внешние метрики – с аспектами качества в использовании.

Этап 2.2. Установка уровней оценки (ранжирования) для метрик

Значение, измеренное с помощью метрики, имеет некоторую величину, которая сама по себе не отражает степень удовлетворения результатом измерения. Поэтому шкала измерений должна быть разделена на диапазоны, соответствующие различным степеням удовлетворения требований. Примеры уровней ранжирования рассмотрены в подразд. 10.4 (см. рис. 10.4).

Этап 2.3. Установка критериев для оценки

Чтобы оценить качество продукта, необходимо некоторым образом объединить результаты оценки его различных характеристик, подхарактеристик и метрик. С этой целью должна быть разработана процедура, включающая критерии для показателей качества различных уровней.

Например, каждая из характеристик может быть определена в терминах единственной подхарактеристики или средневзвешенной комбинации подхарактеристик. Каждая из подхарактеристик должна быть некоторым образом определена на основе совокупности выбранных на предыдущем этапе метрик.

Процедура установки критериев для оценки обычно включает и другие аспекты, такие как продолжительность и стоимость оценки, которые вносят вклад в оценку качества программного продукта в конкретной среде окружения.

Стадия 3. Проектирование оценки

Данная стадия состоит из одного этапа *разработки плана оценки*.

В плане оценки должны быть описаны методы оценки и график действий по оценке. Действия по оценке для разработчика, заказчика и оценщика описаны в стандартах *ISO/IEC 14598–3:2000*, *ISO/IEC 14598–4:1999* и *ISO/IEC 14598–5:1998* (в настоящее время данные стандарты заменены на *ISO/IEC 25041:2012* серии *SQuaRE*) [30 – 32, 50].

План оценки должен быть согласован с планом количественной оценки качества, регламентированным в *ISO/IEC 14598–2:2000* (заменен на стандарт *ISO/IEC 25001:2007* серии *SQuaRE*) [29, 39].

Стадия 4. Выполнение оценки

Данная стадия состоит из трех этапов.

Этап 4.1. Выполнение измерений

Для измерения выбранные метрики применяются к программному продукту. Результатом являются значения на шкалах метрик.

Этап 4.2. Сравнение с уровнями оценки (ранжирование)

На данном этапе измеренные значения сравниваются с уровнями ранжирования (например, как показано на рис. 10.4).

Этап 4.3. Оценка результатов

Оценка результатов – заключительный этап процесса оценки программного продукта. На данном этапе с учетом решений, принятых при выполнении этапа 2.3, суммируются оцененные уровни метрик. Результатом является заключение о степени удовлетворения продуктом требований к качеству. Полученное в итоге качество оценивается с учетом других аспектов процесса оценки, таких как продолжительность и стоимость оценки.

На основании результатов оценки принимается административное решение относительно принятия или отклонения, выпуска или невыпуска программного продукта.

12. КАЧЕСТВО СИСТЕМ И ПРОГРАММНЫХ СРЕДСТВ В СЕРИИ СТАНДАРТОВ SQUARE

12.1. Структура серии стандартов SQaRE

Стандарты в области оценки качества, рассмотренные в разд. 10 и 11, ориентировались только на оценку качества программных средств. Область действия серии стандартов *SQaRE* распространена на качество систем и программных средств.

Поскольку в настоящее время компьютеры используются во все более разнообразных областях жизнедеятельности человека, то их корректная работа часто является критически важной для успешного ведения бизнеса и безопасности людей. С учетом этого делом первостепенной важности является разработка или выбор высококачественных программных продуктов и систем. Ключевым фактором в обеспечении надлежащего качества является комплексная спецификация и оценка качества программного продукта. Это может быть достигнуто путем определения соответствующих характеристик качества с учетом предполагаемого использования программного продукта. Важно, чтобы каждая характеристика качества программного продукта была специфицирована и оценена по возможности с помощью проверенных или широко используемых мер качества.

С учетом вышеизложенного подкомитетом SC 7 Объединенного технического комитета JTC1 организаций ISO и IEC (см. подразд. 1.1) активно ведутся работы по разработке серии стандартов под общим названием *Системная и программная инженерия. Требования к качеству и оценка систем и программных средств (Systems and software engineering. Systems and software Quality Requirements and Evaluation (SQaRE))*.

Данная серия призвана заменить собой серии стандартов *ISO/IEC 9126* и *ISO/IEC 14598* (см. разд. 11) [59 – 62, 28 – 33].

Генеральной целью создания серии международных стандартов *SQaRE* является переход к логически организованной, усовершенствованной и единой серии, охватывающей два основных процесса:

- спецификацию требований к качеству программных средств и систем;
- оценку качества программных средств и систем, поддерживаемую процессом измерения качества.

Серия *SQaRE* устанавливает критерии для спецификации требований к качеству программного продукта и системы, их измерения и оценки. Серия содержит рекомендуемые меры для атрибутов качества программного продукта

и системы, которые могут быть использованы разработчиками, покупателями и оценщиками.

Основными преимуществами серии стандартов *SQuaRE* являются:

- координация руководств по измерению и оценке качества программных продуктов и систем;
- наличие руководства по спецификации требований к качеству программного продукта и системы;
- гармонизация со стандартом *ISO/IEC 15939:2007. Программная инженерия. Процесс измерения программных средств* [35] в форме эталонной модели измерений качества.

На рис. 12.1 приведена организация серии стандартов *SQuaRE* [38].

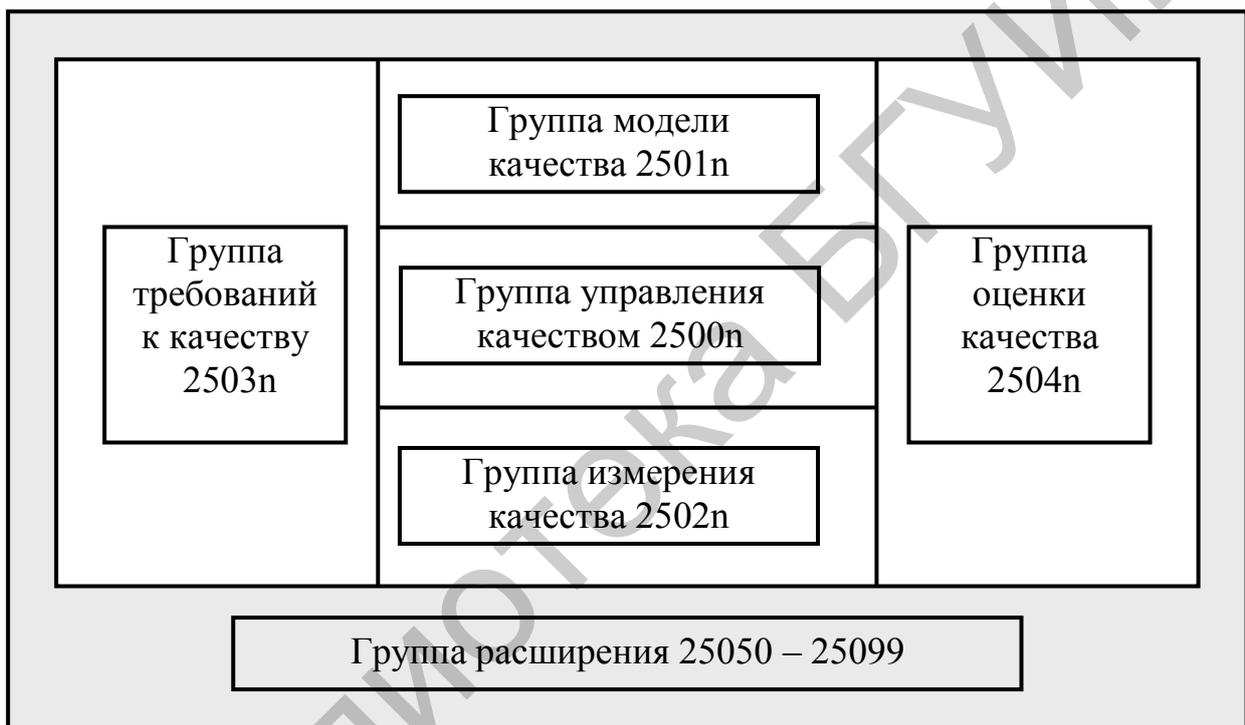


Рис. 12.1. Организация серии международных стандартов *SQuaRE*

Серия стандартов *SQuaRE* разделена на следующие группы (разделы):

1. *ISO/IEC 2500n* – группа управления качеством. Стандарты из данной группы определяют общие модели *SQuaRE*, регламентируют термины и определения, которые используются в остальных стандартах серии *SQuaRE*, определяют принципы планирования и управления качеством систем и программных средств. Данная группа стандартов содержит руководство по использованию стандартов серии *SQuaRE*.

2. *ISO/IEC 2501n* – группа модели качества. В стандартах данной группы представлены модели качества компьютерных систем, программных продуктов и данных. Данная группа стандартов содержит также практическое руководство по использованию регламентированных моделей качества.

3. *ISO/IEC 2502n* – группа измерения качества. Стандарты данной группы включают эталонную модель измерений качества систем и программных средств, математические определения мер качества и практическое руководство по их применению. Даны примеры внутренних и внешних мер качества и мер качества в использовании. Определены и представлены элементы мер качества, являющиеся основой этих мер.

4. *ISO/IEC 2503n* – группа требований к качеству. Стандарты данной группы помогают определить требования к качеству, основываясь на моделях и мерах качества. Эти требования к качеству могут использоваться в процессе выявления требований к качеству разрабатываемых систем и программных продуктов или как входные данные для процесса оценки.

5. *ISO/IEC 2504n* – группа оценки качества. Стандарты данной группы содержат требования, рекомендации и руководства по оценке систем и программных средств оценщиками, заказчиками или разработчиками. Здесь также представлены правила документирования мер в виде модуля оценки.

6. Группа расширения *ISO/IEC 25050 – 25099*. Стандарты данной группы включают требования к качеству коммерческих готовых программных продуктов (Commercial Off-The-Shelf Software – COTS) и общие промышленные форматы для отчетов по практичности.

На рис. 12.2 приведены стандарты серии *SQuaRE* с учетом их распределения по группам.

12.2. Группа стандартов ISO/IEC 2500n

В настоящее время в данную группу входят следующие стандарты:

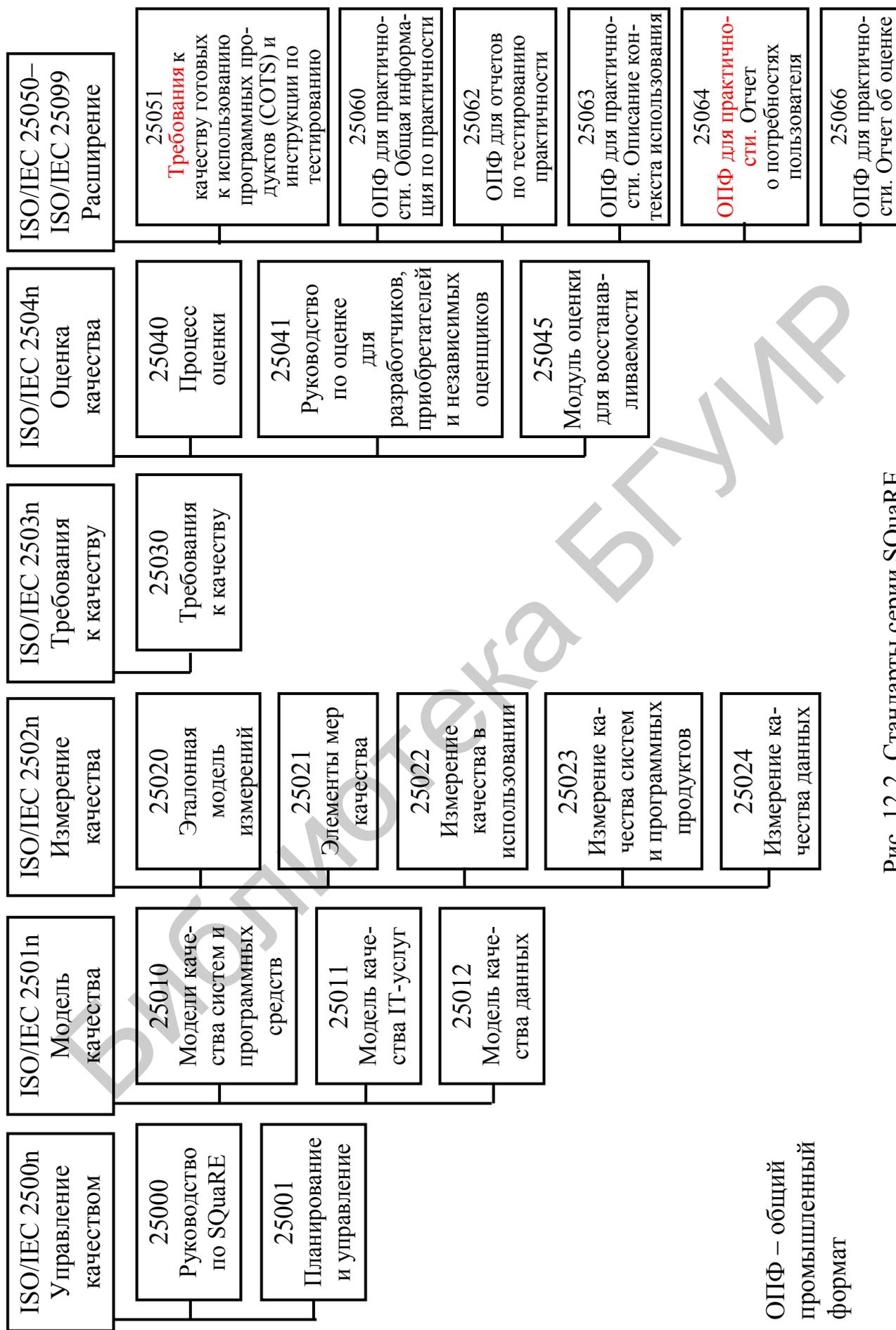
- *ISO/IEC 25000:2014. Системная и программная инженерия. Требования к качеству и оценка систем и программных средств (SQuaRE). Руководство по SQuaRE* [38];

- *ISO/IEC 25001:2014. Системная и программная инженерия. Требования к качеству и оценка систем и программных средств (SQuaRE). Планирование и управление* [40].

Вышеназванные стандарты представляют собой вторые редакции соответствующих стандартов. Первые редакции данных стандартов *ISO/IEC 25000:2005* и *ISO/IEC 25001:2007* [36, 39] приняты в качестве национальных стандартов Республики Беларусь *СТБ ISO/IEC 25000–2009* и *СТБ ISO/IEC 25001–2009* [19, 20].

12.2.1. Основные понятия и определения в серии стандартов SQuaRE

ISO/IEC 25000:2014 регламентирует понятия и определения, используемые во всех стандартах серии *SQuaRE*. Ниже даны определения, введенные в *SQuaRE*, новые или отличающиеся от приведенных в разд. 9.



ОПФ – общий промышленный формат

Рис. 12.2. Стандарты серии SQuaRE

Атрибут для меры качества (attribute for quality measure) – атрибут, относящийся к самому программному продукту, к его применению или к процессу его разработки и используемый для получения элементов мер качества. В стандартах серии *SQuaRE* понятие атрибута применяется аналогично понятию свойства.

Воспроизводимость (результатов измерений) (reproducibility (of result of measurement)) – близость между результатами измерений одной и той же величины, выполняемых в изменяющихся условиях.

Данные (data) – различным образом интерпретируемое представление информации в формализованном виде, пригодном для передачи, толкования или обработки.

Измерение (measurement) – набор операций для определения значения меры.

Индикатор (указатель, indicator) – мера, обеспечивающая прогноз или оценку заданных атрибутов на основе модели с учетом определенных информационных потребностей.

Информационная потребность (information need) – познания, необходимые для управления целями, задачами, рисками, проблемами.

Информационная система (information system) – одна или несколько компьютерных и коммуникационных систем (включая соответствующие организационные ресурсы, такие как люди, технические и финансовые ресурсы), предоставляющих и распространяющих информацию.

Информационный продукт (information product) – один или более индикаторов и их интерпретаций, которые относятся к информационной потребности.

Информация (information) – знания об объектах, таких как факты, события, предметы, процессы, идеи, концепции, которые в конкретном контексте имеют определенное значение. Информация обязательно имеет некоторую форму представления, чтобы обеспечить ее передачу. При этом первостепенное значение имеет соответствующая интерпретация этого представления.

Качество в использовании (quality in use) – степень применимости программного продукта или системы заданными пользователями для удовлетворения их потребностей в достижении заданных целей с результативностью, эффективностью, свободой от риска и удовлетворенностью в заданных контекстах использования.

Качество данных (data quality) – степень удовлетворения характеристиками данных заданных и подразумеваемых потребностей при использовании в заданных условиях.

Качество программного средства (software quality) – степень удовлетворения программным продуктом заданных и подразумеваемых потребностей при его использовании в заданных условиях.

Качество продукта (product quality) – собирательный термин для обозначения качества системы, качества программного продукта, качества данных или качества системной услуги.

Компонент (component) – объект с дискретной структурой, такой как собранная группа модулей или программный модуль, выделяемый на некотором уровне структурной организации системы или программного средства.

Компьютерная система (computer system) – система, содержащая один или более компонентов. Компоненты включают компьютеры (оборудование), соответствующее программное обеспечение и данные.

Критерий принятия решения (decision criteria) – пороговые значения, плановые значения или конкретные наборы, используемые для определения потребности в действии, дальнейшем исследовании или для описания степени доверия к данному результату.

Мера (measure) – переменная, которой присваивается значение в результате измерений. Термин «мера» используется собирательно для указания базовых мер, производных мер и индикаторов (указателей). Понятие «мера» в серии *SQuaRE* эквивалентно понятию «метрика» в сериях ISO/IEC 9126 и ISO/IEC 14598.

Мера базовая (base measure) – мера, определенная в терминах атрибута и метода для ее количественной оценки. Базовая мера функционально не зависит от других мер.

В стандартах серии *SQuaRE* понятие базовой меры соответствует понятию *элемента меры качества*.

Мера качества (quality measure) – производная мера, определяемая как результат функции измерения двух или более величин элементов меры качества.

Мера качества программного средства внешняя (external measure of software quality) – мера обеспечения программным продуктом поведения системы (включая программное обеспечение), удовлетворяющего заданным и подразумеваемым потребностям, при использовании программного продукта в заданных условиях.

Например, количество отказов, обнаруженных в течение тестирования, – это внешняя мера качества программного средства, связанная с количеством ошибок в нем. Две меры количества отказов, измеренные в разное время, не обязательно будут идентичны, поскольку тестирование может не найти всех ошибок, а ошибка может вызвать рост других отказов в других условиях.

Мера качества программного средства внутренняя (internal measure of software quality) – мера удовлетворения набором статических атрибутов программного продукта заданных и подразумеваемых потребностей в данном продукте при его использовании в заданных условиях. Статические атрибуты – это атрибуты, связанные со спецификацией требований к ПС, архитектурой ПС, структурой его компонентов и их кодами.

Мера производная (derived measure) – мера, определяемая как функция двух или более значений базовых мер.

Базовая мера, преобразованная с помощью математической функции, также может рассматриваться как производная мера.

Например, меры сложности, а также количество, серьезность и частота найденных при анализе ошибок являются внутренними мерами качества ПС.

Метаданные (metadata) – данные, которые описывают другие данные.

Метод измерения (measurement method) – логическая последовательность действий, описанная с общих позиций, используемая при измерении.

Модель анализа (analysis model) – алгоритм или вычисление, связывающее одну или более базовых и/или производных мер с соответствующим критерием принятия решения.

Модель качества данных (data quality model) – определенный набор характеристик, обеспечивающий основу для специфицирования требований к качеству данных и оценки качества данных.

Повторяемость (результатов измерений) (repeatability (of result of measurement)) – близость между результатами последовательных измерений одной и той же величины, выполняемых в одних и тех же условиях.

Пользователь конечный (end user) – лицо, которое в конечном счете получает выгоду от выходных результатов системы. Конечным пользователем может быть владелец продукта или случайный пользователь.

Пользователь косвенный (indirect user) – лицо, получающее выходные данные от системы, но не взаимодействующее с ней.

Пользователь непосредственный (direct user) – лицо, взаимодействующее с продуктом.

Правообладатель (stakeholder) – лицо или организация, имеющие права, долю, претензии или интерес к системе или ее характеристикам, обеспечивающим их потребности и ожидания.

Программный продукт (software product) – набор компьютерных программ, процедур, возможно связанных с ними документации и данных. В стандартах серии *SQuaRE* понятие качества программных средств (программного обеспечения) имеет тот же смысл, что и качество программного продукта.

Продукт – разработанный артефакт, являющийся полным элементом или компонентом другого элемента, который может быть количественно оценен.

В информационных технологиях обычно принято выделять четыре согласованных категории продуктов: аппаратное обеспечение, программное обеспечение, услуги и данные.

Промежуточный системный или программный продукт – продукт процесса разработки системы или программного средства, который может использоваться как вход для другой стадии процесса их разработки.

Промежуточные программные продукты могут включать статические и динамические модели, другие документы, исходные коды.

Процедура измерения (measurement procedure) – логическая последовательность действий, применяемая конкретно, используемая при выполнении необходимых вычислений в соответствии с данным методом измерений. Процедура измерений обычно оформляется в виде документа, в котором она должна быть описана достаточно подробно, чтобы позволить выполнять измерение без дополнительной информации.

Свойство для количественной оценки (property to quantify) – свойство целевого объекта, которое связано с элементом меры качества и которое может быть оценено количественно с помощью метода измерений. В стандартах серии *SQuaRE* понятие свойства используется аналогично понятию атрибута.

Свойство качества (quality property) – измеримый компонент качества.

Система (system) – совокупность взаимодействующих элементов, организованная для достижения одной или более заданных целей.

Функция измерения (measurement function) – алгоритм или вычисления, выполняемые для объединения двух или более элементов мер качества.

Характеристика качества данных (data quality characteristic) – категория атрибутов качества данных, которая является носителем качества данных.

Целевой объект (target entity) – фундаментальный объект интереса для пользователя, о котором собирается информация и который должен быть измерен. Целевым объектом могут быть разрабатываемые рабочие (промежуточные) продукты, а также поведение системы, программного средства или правообладателей, таких как пользователи, операторы, разработчики, тестировщики или персонал сопровождения.

Элемент меры качества (quality measure element) – мера, определяемая в терминах свойства и метода измерения для ее количественной оценки, возможно включая преобразование с помощью математической функции. Элемент меры качества может являться как базовой, так и производной мерой.

12.2.2. Основные положения ISO/IEC 25000:2014

ISO/IEC 25000:2014 представляет собой незначительно измененную редакцию первого издания стандарта *ISO/IEC 25000:2005 (СТБ ISO/IEC 25000–2009)* [38, 36, 19]. Стандарт содержит руководство по структуре и применению серии *SQuaRE*.

В *ISO/IEC 25000:2014* регламентированы общие термины и определения, которые используются в остальных стандартах серии *SQuaRE*. Данные термины и определения приведены в п. 12.2.1.

ISO/IEC 25000:2014 содержит описание организации серии *SQuaRE* (см. подразд. 12.1), обзор содержания основных стандартов данной серии.

Даны общие модели *SQuaRE*, в том числе:

- общая эталонная модель серии *SQuaRE*;
- модель качества в жизненном цикле программных средств;
- иерархическая структура моделей качества, регламентированных в серии *SQuaRE*.

Общая эталонная модель серии *SQuaRE* приведена на рис. 12.3. Данная модель разработана для оказания помощи пользователям серии в ориентации в стандартах данной серии. Выбор того или иного стандарта из серии зависит от роли пользователя и его информационных потребностей. *ISO/IEC 25000:2014* рекомендует ознакомиться вначале с общими руководствами, приведенными в данном стандарте, а затем переходить к использованию отдельных стандартов серии в зависимости от целей применения серии *SQuaRE*.

Модель качества в жизненном цикле программных средств отражает виды качества в ЖЦ ПС (рис. 12.4). Данная модель построена на основе модели качества в ЖЦ ПС, определенной в стандартах *ISO/IEC 9126-1:2001* и *ISO/IEC 14598-1:1999* и описанной в подразд. 11.2. Модель выделяет следующие основные виды качества в жизненном цикле программных средств:

- требования к качеству в использовании (потребности пользователя в качестве);
- требования к внешнему качеству;
- требования к внутреннему качеству;
- внутреннее качество;
- внешнее качество;
- качество в использовании.

Зависимость и влияние видов качества программных средств в их жизненном цикле отражены на рис. 12.5.

Иерархическая структура моделей качества, регламентированных в серии *SQuaRE*, определяет, что модель качества состоит из характеристик, которые далее в общем случае делятся на подхарактеристики, определяемые атрибутами (свойствами) качества (рис. 12.6).

Серия *SQuaRE* определяет несколько моделей качества:

- модели качества систем и программных средств, регламентируемые с помощью модели качества в использовании и модели качества продукта;
- модель качества данных;
- модель качества IT-услуг.

Верхние уровни данных моделей (уровни характеристик и подхарактеристик) регламентированы в группе стандартов *ISO/IEC 2501n*.

Для измерения значений свойств (атрибутов) качества, находящихся на нижнем уровне иерархической структуры моделей качества, используются меры качества (см. рис. 12.5, 12.6). Измерению значений свойств (атрибутов) качества с помощью мер качества посвящена группа стандартов *ISO/IEC 2502n*.

В *ISO/IEC 25000:2014* определена связь между стандартами серии *SQuaRE* и другими международными стандартами, в том числе *ISO/IEC 12207:2008*, *ISO/IEC 15288:2008*, *ISO/IEC 15939:2007*, сериями стандартов *ISO/IEC 15504* и *ISO/IEC 29119*, семейством стандартов *ISO 9000*. Определены связи между стандартами серий *ISO/IEC 9126*, *ISO/IEC 14598* и *SQuaRE*.

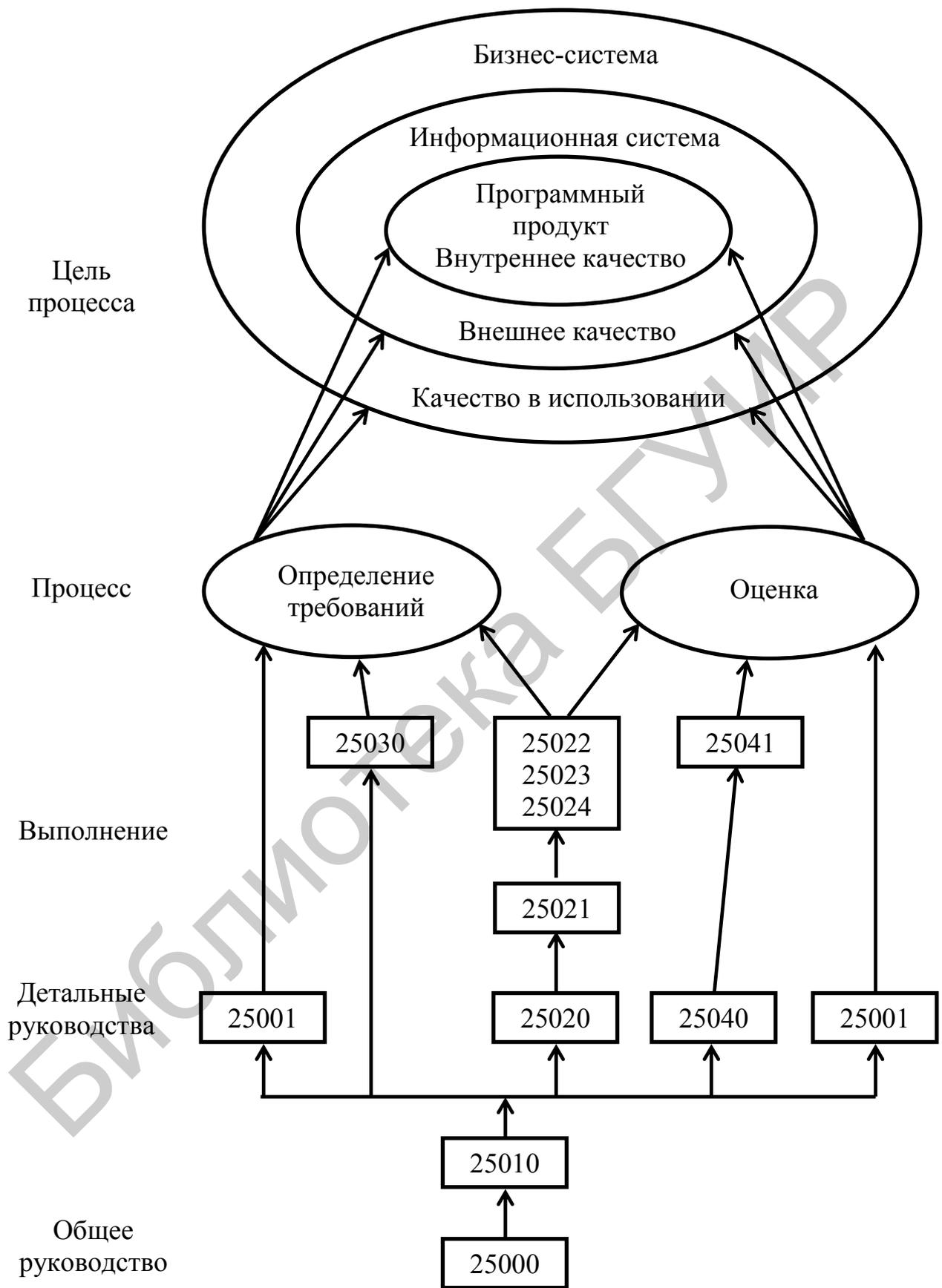


Рис. 12.3. Общая эталонная модель серии SQaRE

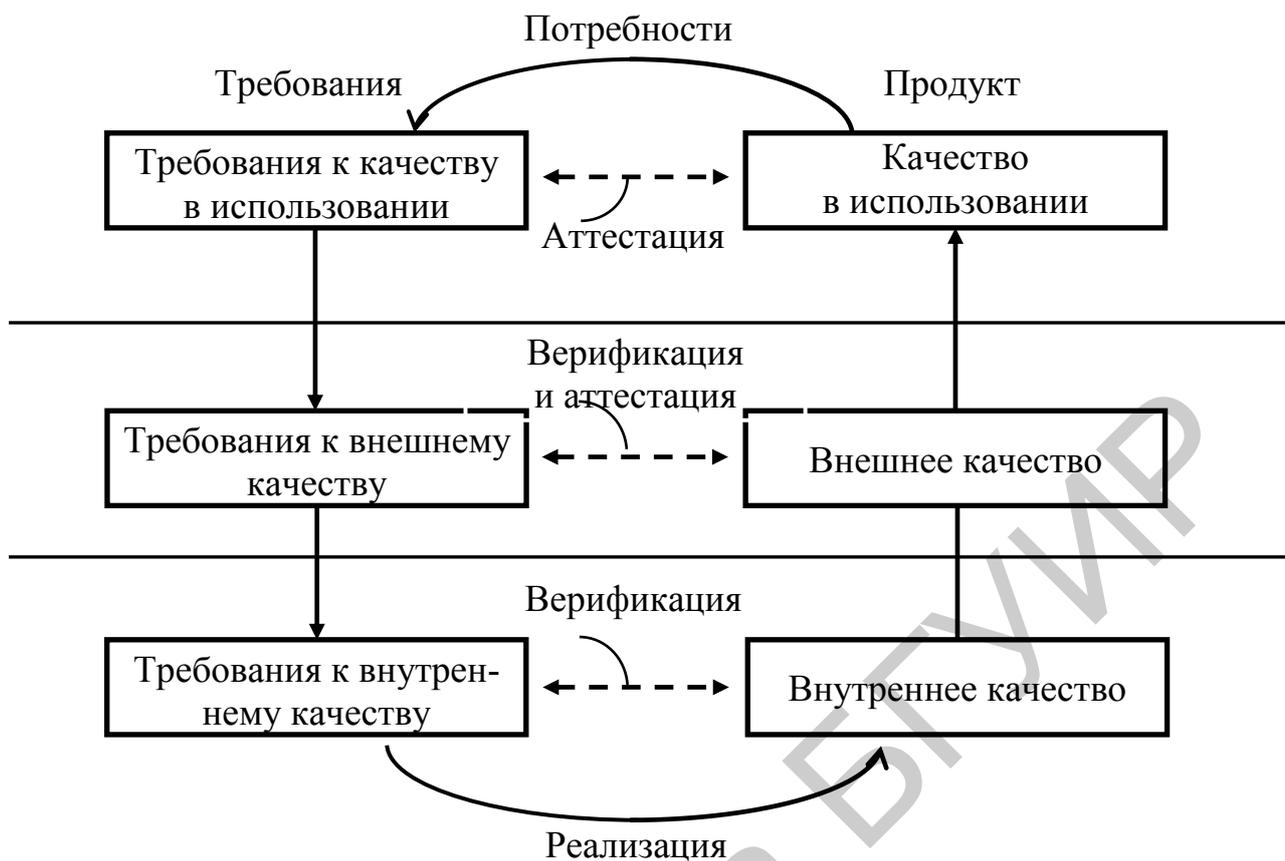


Рис. 12.4. Качество в жизненном цикле программных средств

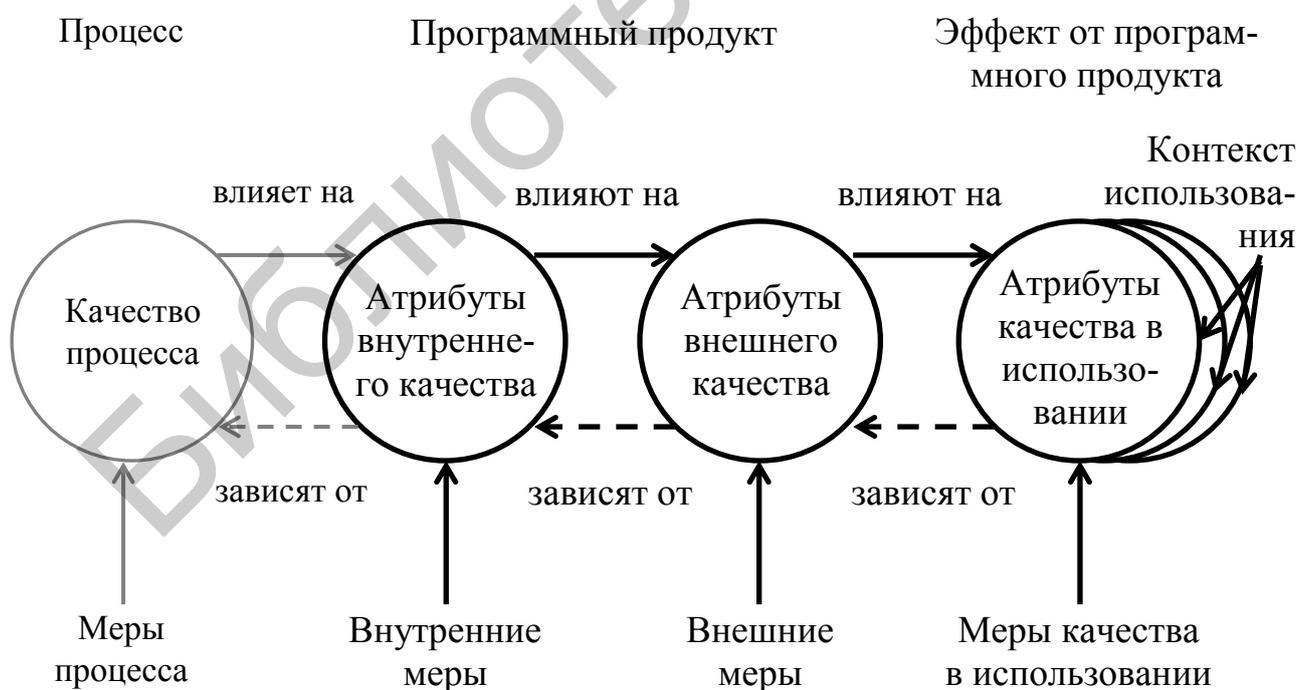


Рис. 12.5. Зависимость и влияние видов качества программных средств в их жизненном цикле

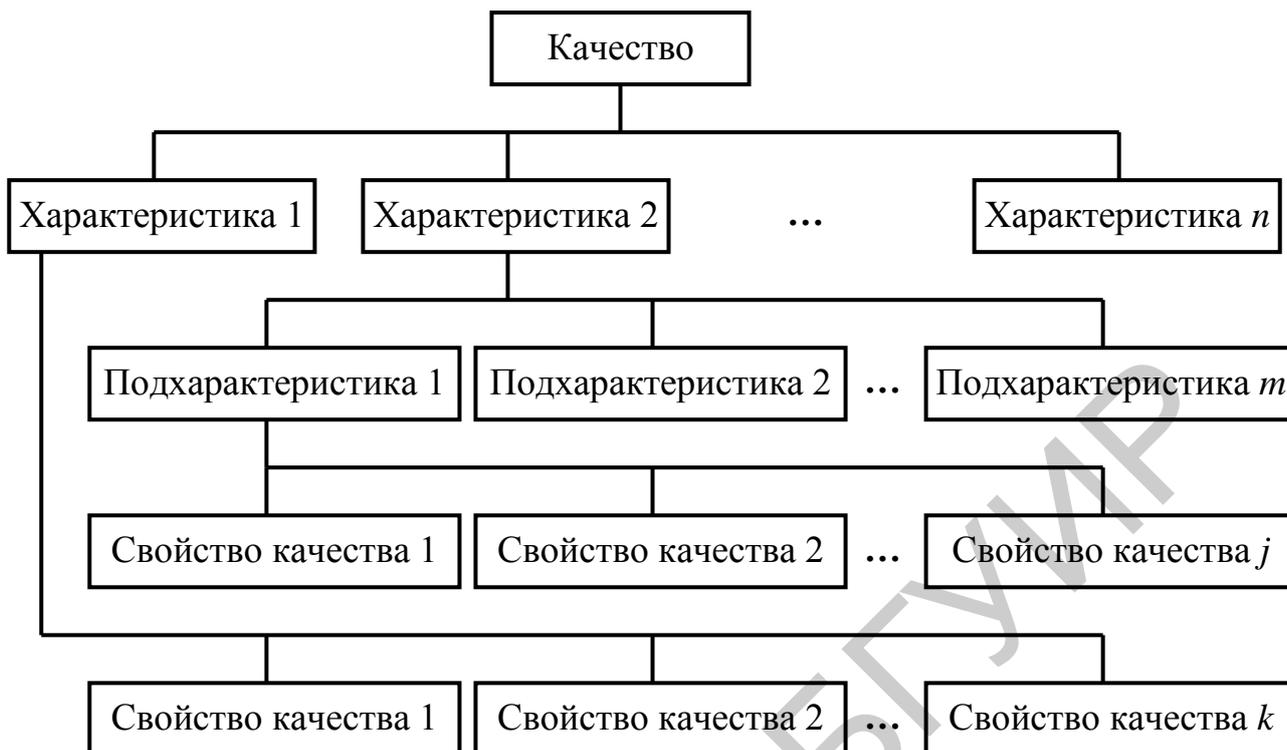


Рис. 12.6. Структура моделей качества по стандартам ISO/IEC 25000:2014 и ISO/IEC 25010:2011

12.2.3. Основные положения ISO/IEC 25001:2014

Стандарт *ISO/IEC 25001:2014* представляет собой незначительно измененную вторую редакцию стандарта *ISO/IEC 25001:2007* (СТБ *ISO/IEC 25001–2009*) [40, 39, 20].

ISO/IEC 25001:2014 содержит требования и рекомендации для организации, ответственной за реализацию и управление, во-первых, разработкой спецификации требований к качеству продуктов и, во-вторых, деятельностью по оценке качества посредством предоставления технологий, инструментов, опыта и навыков управления.

Предполагаемыми пользователями стандарта *ISO/IEC 25001:2014* являются лица, ответственные за:

- управление технологиями, используемыми для разработки спецификации требований и выполнения оценки;
- разработку спецификации требований к качеству продуктов;
- поддержку оценки качества продуктов;
- управление организацией разработки программных средств и систем;
- реализацию функций обеспечения качества.

Положения стандарта должны учитываться группами оценки, осуществляющими в рамках организации поддержку всех проектов по

разработке и приобретению программных средств, а также оценку проектов других организаций.

Главными обязанностями группы оценки являются:

- руководство и управление действиями, связанными с оценкой качества программных средств;
- руководство идентификацией и определением требований к качеству;
- выполнение планов по определению требований к качеству и оценке качества;
- разработка критериев для эталонных значений, используемых при оценке;
- сбор и анализ результатов действий группы оценки;
- распространение результатов действий группы оценки внутри организации;
- приобретение необходимой технической информации;
- приобретение технологии оценки;
- разработка специфических для организации стандартов и инструментов оценки;
- оценка эффективности и качества приобретаемых и разрабатываемых программных средств;
- содействие переходу на новые технологии оценки.

В стандарте *ISO/IEC 25001:2014* определено, что организация должна разработать политику и планы для спецификации требований к качеству и действий по оценке качества.

В *плане оценки качества проекта* должны быть установлены и описаны действия, выполняемые на следующих шагах:

- специфицирование требований к качеству программных средств и систем;
- определение целей оценки качества программных средств и систем;
- определение требований к оценке;
- специфицирование оценки;
- проектирование оценки;
- выполнение оценки;
- анализ результатов оценки.

Любая организация, связанная с разработкой, заказом, приобретением или оценкой программных продуктов, должна разработать свою политику в части спецификации требований к качеству и действий по оценке качества. Данная политика должна включать три основных компонента:

- действия на уровне организации;
- действия на уровне управления проектами;
- анализ и использование результатов оценки.

Действия на уровне организации должны включать следующие разделы:

- управление организационной средой;

- управление ресурсами;
- планирование использования и усовершенствования технологий специфицирования требований к качеству и оценки качества;
- реализация технологии оценки;
- передача технологии, используемой для оценки;
- оценка технологии для спецификации требований к качеству и оценки качества;
- управление результатами и опытом оценки.

Действия на уровне управления проектами должны включать поддержку планирования оценки качества программных продуктов для конкретных проектов, выполняемых в организации.

Анализ и использование результатов оценки определяет сбор результатов оценок в конце каждой оценки проекта, анализ данных результатов и их эффективное использование. С этой целью должны быть выполнены:

- верификация качества собранных данных (например, значимость, представительность, корректность, статистическая обоснованность);
- определение соответствующих методов объединения и анализа данных;
- определение соответствующих методов интерпретации данных;
- пересмотр плановых значений показателей качества для каждой оценки проекта;
- соответствующее обучение, если это необходимо.

Для усовершенствования технологии оценки должны быть проанализированы результаты оценки, методы оценки, плановые значения показателей качества для каждой оценки проекта. По результатам анализа должны быть оформлены отчеты.

В стандарте *ISO/IEC 25001:2014* приведен *шаблон плана оценки качества проекта*. Данный шаблон содержит следующие основные разделы:

- цели оценки;
- требования к качеству программных средств и систем и применяемые характеристики качества;
- список приоритетов;
- цели в области качества;
- определение обязанностей;
- проектирование оценки;
- используемые и анализируемые данные;
- планирование и выполнение оценки.

12.3. Группа стандартов ISO/IEC 2501n

В данную группу в настоящее время входят следующие стандарты:

- *ISO/IEC 25010:2011. Системная и программная инженерия. Требования к качеству и оценка систем и программных средств (SQuaRE). Модели качества систем и программных средств* [41];

- *ISO/IEC PDS 25011. Информационная технология. Требования к качеству и оценка систем и программных средств (SQuaRE). Модели качества IT-услуг;*

- *ISO/IEC 25012:2008. Программная инженерия. Требования к качеству и оценка программных продуктов (SQuaRE). Модель качества данных* [42].

В 2016 г. введен в действие стандарт России *ГОСТ Р ИСО/МЭК 25010–2015* с идентичной степенью соответствия стандарту *ISO/IEC 25010:2011*.

Стандарт *ISO/IEC PDS 25011* в настоящее время находится в разработке (стадия 30.60 на 06.05.2015 г., см. подразд. 1.2).

12.3.1. Модели качества систем и программных средств по стандарту ISO/IEC 25010:2011

Стандарт *ISO/IEC 25010:2011* введен взамен *ISO/IEC 9126–1:2001* (см. разд. 11) [41, 59].

Качество системы – это степень удовлетворения системой заданных и подразумеваемых потребностей своих правообладателей. Эти потребности представляются моделями качества, приведенными в *ISO/IEC 25010:2011*. Данные модели имеют иерархическую структуру. На верхнем уровне моделей находятся характеристики. Большинство из характеристик подразделяются на подхарактеристики.

Измеряемые свойства, связанные с качеством системы или программного продукта, представляют собой *свойства (атрибуты) качества*. Свойства качества ассоциируются с мерами качества. Чтобы получить меры характеристики или подхарактеристики качества без их непосредственного измерения, необходимо определить совокупность свойств, которые вместе покрывают характеристику или подхарактеристику, получить меры качества для каждого из свойств и объединить их с помощью некоторых вычислений для получения результирующей меры качества, связанной с характеристикой или подхарактеристикой качества.

Иерархическая связь между характеристиками, подхарактеристиками и свойствами (атрибутами) качества представлена на рис. 12.6.

В *ISO/IEC 25010:2011* определены *две модели качества*:

1. Модель качества в использовании, состоящая из пяти характеристик, имеющих отношение к результатам взаимодействия с продуктом при его

применении в заданном контексте использования. Модель качества в использовании представляет собой модель качества, применимую к полным человеко-компьютерным системам, включая как используемые компьютерные системы, так и используемые программные продукты.

2. Модель качества продукта, состоящая из восьми характеристик, имеющих отношение к статическим свойствам программного продукта и динамическим свойствам компьютерной системы. Эта модель применима как к компьютерным системам, так и к программным продуктам.

В *ISO/IEC 25010:2011* под продуктом подразумевается программный продукт или система.

12.3.2. Модель качества в использовании по стандарту ISO/IEC 25010:2011

Качество в использовании – это степень применимости программного продукта или системы заданными пользователями для удовлетворения их потребностей в достижении заданных целей с результативностью, эффективностью, свободой от риска и удовлетворенностью в заданных контекстах использования.

Понятие *контекста использования* свойственно как для качества в использовании, так и для качества продукта. В последнем случае оно определяется как *заданные условия*.

Качество в использовании характеризует влияние, которое продукт (система или программный продукт) оказывает на правообладателей. Оно определяется качеством программного обеспечения, аппаратных средств и эксплуатационной среды, а также характеристиками пользователей, задач и социального окружения. Все эти факторы вносят вклад в качество применения системы.

Модель качества в использовании приведена на рис. 12.7. Данная модель состоит из пяти характеристик, связанных с результатами взаимодействия с системой: *Результативности, Эффективности, Удовлетворенности, Свободы от рисков, Покрытия контекста*.

Каждая из характеристик может быть связана с различной деятельностью правообладателей. Например, это может быть взаимодействие с системой для эксплуатирующего персонала или сопровождение для разработчиков.

Отличия данной модели от предыдущей версии модели качества в использовании, определенной в *ISO/IEC 9126–1:2001* (см. подразд. 11.4), на рис. 12.7 подчеркнуты. Новой характеристикой качества в использовании является характеристика *Покрытие контекста*. Кроме того, отличием данной модели от предыдущей версии является разделение большинства ее характеристик на подхарактеристики.



Рис. 12.7. Модель качества в использовании по стандарту ISO/IEC 25010:2011

12.3.2.1. Результативность и Эффективность

Результативность (Effectiveness) – точность и полнота, с которой пользователи достигают заданных целей.

Эффективность (Efficiency) – ресурсы, затрачиваемые в зависимости от точности и полноты, с которыми пользователь достигает целей.

Характеристике *Эффективность* стандарта *ISO/IEC 25010:2011* в *ISO/IEC 9126-1:2001* соответствует характеристика качества в использовании *Продуктивность (Productivity)*, см. подразд. 11.4).

Как видно из рис. 12.7, данные характеристики не разделяются на подхарактеристики. Их значения определяются непосредственно значениями соответствующих мер качества, вычисляемых на основе свойств качества. Информация по мерам качества приведена в разд. 15.

12.3.2.2. Удовлетворенность

Удовлетворенность (Satisfaction) – степень удовлетворения потребностей пользователя при применении программного продукта или системы в заданном контексте использования.

Подхарактеристиками *Удовлетворенности* являются *Применимость, Доверие, Удовольствие, Комфорт* (см. рис. 12.7).

Применимость (Usefulness) – степень удовлетворения пользователя своими осознанными достижениями прагматических целей, включая результаты и последствия использования.

Доверие (Trust) – степень уверенности пользователя или другого правообладателя в том, что программный продукт или система будет работать в соответствии с предназначением.

Удовольствие (Pleasure) – степень получения пользователем удовольствия от реализации своих персональных потребностей. Персональные потребности могут включать потребности в приобретении новых знаний и умений, в общении с конкретными людьми и т. п.

Комфорт (Comfort) – степень удовлетворения пользователя физическим комфортом.

12.3.2.3. Свобода от риска

Свобода от риска (Freedom from risk) – степень предотвращения программным продуктом или системой потенциального риска для экономического статуса, человеческой жизни, здоровья или окружающей среды.

Данной характеристике в *ISO/IEC 9126–1:2001* соответствует характеристика качества в использовании *Безопасность (Safety)*, см. подразд. 11.4).

Риск представляет собой функцию вероятности наступления события, связанного с некоторой угрозой, и потенциальных неблагоприятных последствий наступления данной угрозы.

Подхарактеристиками характеристики *Свобода от риска* являются *Минимизация экономического риска, Минимизация риска для здоровья и безопасности, Минимизация риска для окружающей среды* (см. рис. 12.7).

Минимизация экономического риска (Economic risk mitigation) – степень ограничения программным продуктом или системой потенциального риска для финансового статуса, эффективной деятельности, коммерческой собственности, репутации или других ресурсов в заданном контексте использования.

Минимизация риска для здоровья и безопасности (Health and safety risk mitigation) – степень ограничения программным продуктом или системой потенциального риска для людей в заданном контексте использования.

Минимизация риска для окружающей среды (Environmental risk mitigation) – степень ограничения программным продуктом или системой

потенциального риска для имущества или окружающей среды в заданном контексте использования.

12.3.2.4. Покрытие контекста

Покрытие контекста (Context coverage) – степень применимости программного продукта или системы с результативностью, эффективностью, свободой от риска и удовлетворенностью как в заданных контекстах использования, так и вне их.

Данная характеристика является новой по отношению к модели качества в использовании, определенной в *ISO/IEC 9126–1:2001* (см. подразд. 11.4).

Подхарактеристиками *Покрытия контекста* являются *Полнота контекста, Гибкость* (см. рис. 12.7).

Полнота контекста (Context completeness) – степень применимости программного продукта или системы с результативностью, эффективностью, свободой от риска и удовлетворенностью во всех заданных контекстах использования.

Гибкость (Flexibility) – степень применимости программного продукта или системы с результативностью, эффективностью, свободой от риска и удовлетворенностью в контекстах использования, не входящих в состав изначально определенных в требованиях.

12.3.3. Модель качества продукта по стандарту ISO/IEC 25010:2011

На рис. 12.8 приведена модель качества продукта. Данная модель подразделяет свойства качества системы или программного продукта на восемь характеристик: *Функциональная пригодность, Эффективность функционирования, Совместимость, Практичность, Надежность, Защищенность, Сопровождаемость, Мобильность*.

Новые характеристики и подхарактеристики, отличающие данную модель от модели внешнего и внутреннего качества, которая определена в *ISO/IEC 9126–1:2001* (см. подразд. 11.3), на рис. 12.8 подчеркнуты.

Как видно из данного рисунка, в состав модели качества продукта введены две новых характеристики – *Совместимость* и *Защищенность*.

Помимо этого, изменились названия и определения некоторых характеристик и подхарактеристик. Из всех характеристик качества исключены подхарактеристики согласованности (*Согласованность функциональности, Согласованность надежности, Согласованность практичности, Согласованность эффективности, Согласованность сопровождаемости, Согласованность мобильности*).

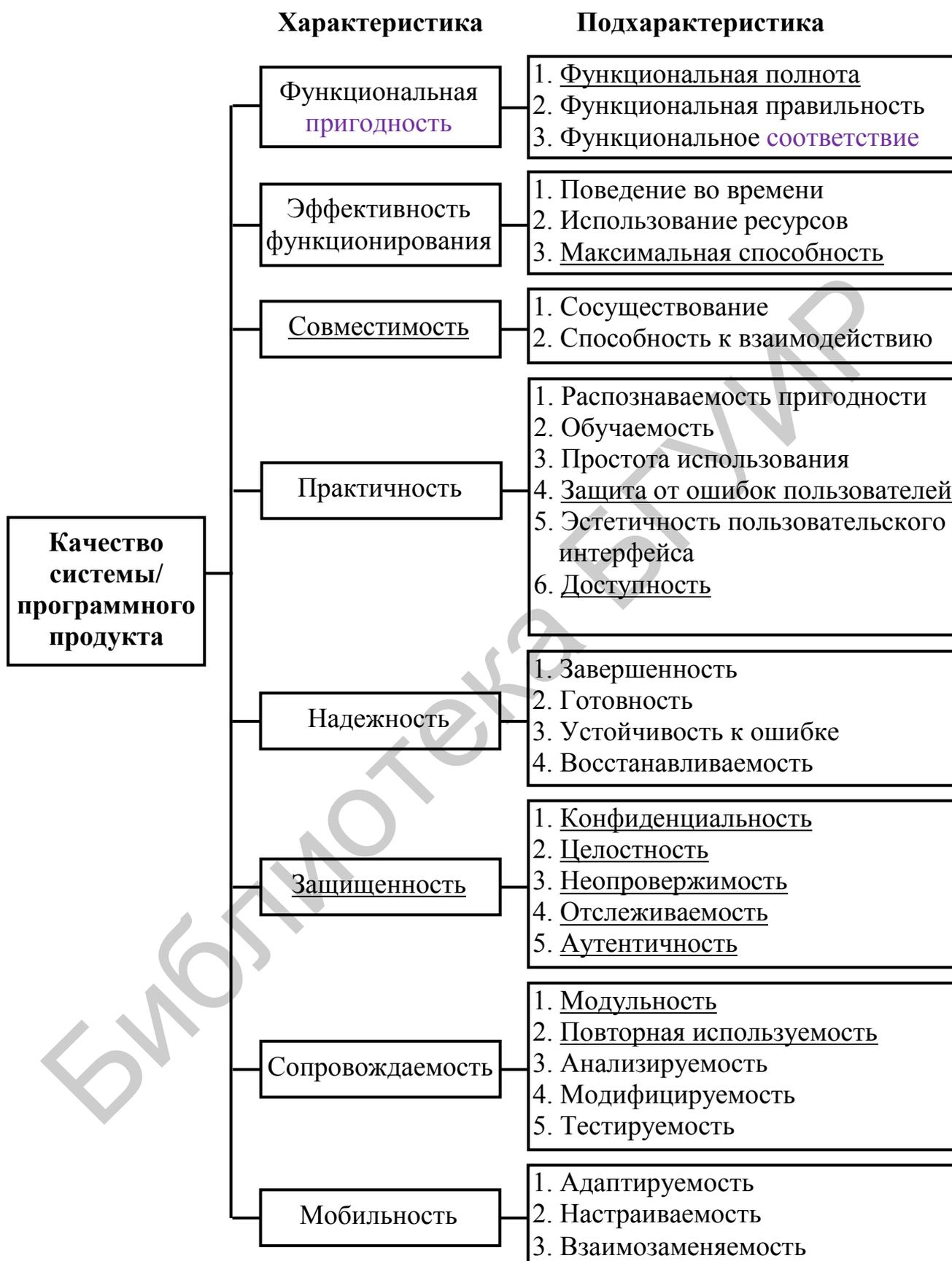


Рис. 12.8. Модель качества продукта по стандарту ISO/IEC 25010:2011

12.3.3.1. Функциональная пригодность

Функциональная пригодность (*Functional suitability*) – степень обеспечения программным продуктом или системой функций, отвечающих заданным или подразумеваемым потребностям при использовании в заданных условиях.

В модели внешнего и внутреннего качества, определенной в *ISO/IEC 9126–1:2001*, данной характеристике соответствует характеристика *Функциональность* (см. подразд. 11.3).

Функциональная пригодность касается соответствия функций программного продукта или системы именно заданным и подразумеваемым потребностям, а не функциональной спецификации.

Подхарактеристиками *Функциональной пригодности* являются *Функциональная полнота*, *Функциональная правильность*, *Функциональная пригодность* (см. рис. 12.8).

Функциональная полнота (*Functional completeness*) – степень покрытия набором функций программного продукта или системы всех заданных задач и целей пользователя.

Данная подхарактеристика является новой по отношению к модели внешнего и внутреннего качества, определенной в *ISO/IEC 9126–1:2001*.

Функциональная правильность (*Functional correctness*) – степень обеспечения программным продуктом или системой правильных результатов с необходимой степенью точности.

В модели внешнего и внутреннего качества, определенной в стандарте *ISO/IEC 9126–1:2001*, данная подхарактеристика имела название *Правильность* (*Корректность*, *Assigasy*, см. подразд. 11.3).

Функциональное соответствие (*Functional appropriateness*) – степень обеспечения функциями продукта или системы выполнения заданных задач и целей.

В модели внешнего и внутреннего качества, определенной в стандарте *ISO/IEC 9126–1:2001*, данная подхарактеристика имела название *Пригодность* (*Suitability*, см. подразд. 11.3).

12.3.3.2. Эффективность функционирования

Эффективность функционирования (*Performance efficiency*) – зависимость функционирования от количества ресурсов, используемых в заданных условиях. Ресурсы могут включать другие программные продукты, программную и аппаратную конфигурацию системы, материалы.

В модели внешнего и внутреннего качества, определенной в стандарте *ISO/IEC 9126–1:2001*, данной характеристике соответствует характеристика *Эффективность* (см. подразд. 11.3).

Подхарактеристиками *Эффективности функционирования* являются *Поведение во времени*, *Использование ресурсов*, *Максимальная способность* (см. рис. 12.8).

Поведение во времени (Time behavior) – степень соответствия требованиям значений времен отклика и обработки, а также пропускной способности программного продукта или системы при выполнении ими своих функций.

Использование ресурсов (Resource utilization) – степень соответствия требованиям количества и типов ресурсов, используемых программным продуктом или системой при выполнении своих функций.

Максимальная способность (Capacity) – степень соответствия требованиям максимальных значений параметров программного продукта или системы. Параметры могут включать количество элементов, которые должны сохраняться, количество одновременно работающих пользователей, ширину полосы пропускания, пропускную способность транзакций, размер базы данных и т. д.

Данная подхарактеристика является новой по отношению к модели внешнего и внутреннего качества, определенной в *ISO/IEC 9126–1:2001* (см. подразд. 11.3).

12.3.3.3. Совместимость

Совместимость (Compatibility) – степень возможности программного продукта, системы или компонента обмениваться информацией с другими программными продуктами, системами или компонентами и/или выполнять свои заданные функции при совместном использовании одной и той же аппаратной или программной среды.

В модели внешнего и внутреннего качества, определенной в стандарте *ISO/IEC 9126–1:2001*, данная характеристика отсутствует (см. подразд. 11.3).

Подхарактеристиками *Совместимости* являются *Сосуществование* и *Способность к взаимодействию* (см. рис. 12.8).

Сосуществование (Co-existence) – степень эффективности выполнения продуктом требуемых функций при разделении общей среды и ресурсов с другими продуктами без вредного влияния на другой продукт.

В модели внешнего и внутреннего качества *ISO/IEC 9126–1:2001* данная подхарактеристика относилась к характеристике *Мобильность*.

Способность к взаимодействию (Interoperability) – степень, в которой два или более продукта (системы, программных продукта или компонента) могут обмениваться информацией и использовать информацию, которой они обменялись.

В модели внешнего и внутреннего качества *ISO/IEC 9126–1:2001* данная подхарактеристика относилась к характеристике *Функциональность*.

12.3.3.4. Практичность

Практичность (Usability) – степень применимости программного продукта или системы заданными пользователями для достижения заданных целей с результативностью, эффективностью и удовлетворенностью в заданном контексте использования.

Практичность может быть задана или измерена как характеристика качества продукта в терминах соответствующих подхарактеристик или измерена экспериментально с помощью мер, относящихся к качеству в использовании.

Подхарактеристиками *Практичности* являются *Распознаваемость пригодности*, *Обучаемость*, *Простота использования*, *Защита от ошибок пользователей*, *Эстетичность пользовательского интерфейса*, *Доступность* (см. рис. 12.8).

Распознаваемость пригодности (Appropriateness recognizability) – степень возможности распознавания пользователями пригодности программного продукта или системы для их потребностей.

Распознаваемость пригодности зависит от первоначальных впечатлений от продукта или системы и от соответствующей документации. Информация, предоставляемая программным продуктом или системой, может включать демонстрации, учебные пособия, документацию или, для веб-сайтов, информацию на домашней странице.

В модели внешнего и внутреннего качества, определенной в *ISO/IEC 9126–1:2001*, данной подхарактеристике соответствует подхарактеристика *Понятность (Understandability)*, см. подразд. 11.3).

Обучаемость (Learnability) – степень применимости программного продукта или системы заданными пользователями для достижения заданных целей изучения применения данного продукта или системы с результативностью, эффективностью, свободой от риска и удовлетворенностью в заданном контексте использования.

Простота использования (Operability) – степень наличия в программном продукте или системе свойств (атрибутов), позволяющих легко применять данный продукт или систему и управлять ими.

Защита от ошибок пользователей (User error protection) – степень защиты программным продуктом или системой пользователей от совершения ошибок.

Данная подхарактеристика является новой по отношению к модели внешнего и внутреннего качества, определенной в *ISO/IEC 9126–1:2001*.

Эстетичность пользовательского интерфейса (User interface aesthetics) – степень, в которой пользовательский интерфейс делает взаимодействие с пользователем приятным и удовлетворяющим его.

На данную подхарактеристику влияют, например, используемые цвета и графический дизайн интерфейса.

В модели внешнего и внутреннего качества, определенной в *ISO/IEC 9126–1:2001*, данной подхарактеристике соответствует подхарактеристика *Привлекательность (Attractiveness)*, см. подразд. 11.3).

Доступность (Accessibility) – степень применимости программного продукта или системы людьми самого широкого диапазона характеристик и возможностей для достижения заданных целей в заданном контексте использования.

Диапазон возможностей пользователя включает в том числе ограниченные возможности, связанные с возрастом, инвалидностью, болезнями и т. п.

Данная подхарактеристика является новой по отношению к модели внешнего и внутреннего качества, определенной в *ISO/IEC 9126-1:2001*.

12.3.3.5. Надежность

Надежность (*Reliability*) – степень выполнения системой, программным продуктом или компонентом заданных функций в заданных условиях в течение заданного периода времени.

Для программных средств ограничения *Надежности* могут быть связаны с ошибками в требованиях, проекте, кодах или с изменениями контекста использования.

Подхарактеристиками *Надежности* являются *Завершенность*, *Готовность*, *Устойчивость к ошибке*, *Восстанавливаемость* (см. рис. 12.8).

Завершенность (*Стабильность, Maturity*) – степень соответствия системы, программного продукта или компонента потребностям в надежности при нормальной эксплуатации. *Завершенность* зависит от количества ошибок, оставшихся в системе, программном продукте или компоненте, и определяет возможность их безотказной работы.

Такая концепция завершенности также может применяться к другим характеристикам качества для отражения степени их соответствия требуемым потребностям при нормальной эксплуатации.

Готовность (*Availability*) – степень работоспособности и доступности системы, программного продукта или компонента тогда, когда требуется их использование.

Готовность может быть оценена соотношением времен, в течение которых система, программный продукт или компонент находится в работоспособном и неработоспособном состоянии. Поэтому *Готовность* представляет собой комбинацию *Завершенности* (обуславливающей частоту отказов), *Устойчивости к ошибке* и *Восстанавливаемости* (определяющей продолжительность времени восстановления после каждого отказа).

Подхарактеристика *Готовность* была определена в стандарте *ISO/IEC 9126-1:2001*. Однако с учетом зависимости *Готовности* от других подхарактеристик *Надежности* данная подхарактеристика не была включена в модель внешнего и внутреннего качества *ISO/IEC 9126-1:2001* в виде отдельной подхарактеристики (см. подразд. 11.3).

Устойчивость к ошибке (*Fault tolerance*) – степень функционирования системы, программного продукта или компонента в соответствии с предназначением, несмотря на наличие сбоев аппаратного обеспечения или ошибок в программном обеспечении программного продукта.

Восстанавливаемость (*Recoverability*) – степень восстановления поврежденных данных и переустановления требуемого состояния системы или программного продукта в случае их прерывания или отказа.

В случае отказа компьютерная система будет некоторое время находиться в неработоспособном состоянии. Продолжительность этого времени определяется ее *Восстанавливаемостью*.

12.3.3.6. Защищенность

Защищенность (Security) – степень защиты программным продуктом или системой информации и данных так, чтобы люди, другие программные продукты или системы имели степень доступа к данным, соответствующую типам и уровням их авторизации. Защищенность распространяется как на хранимые данные, так и на передаваемые данные.

В модели внешнего и внутреннего качества, определенной в стандарте *ISO/IEC 9126-1:2001*, *Защищенность* являлась подхарактеристикой характеристики *Функциональность* (см. подразд. 11.3).

Подхарактеристиками *Защищенности* являются *Конфиденциальность*, *Целостность*, *Неопровержимость*, *Отслеживаемость*, *Аутентичность* (см. рис. 12.8).

Конфиденциальность (Confidentiality) – степень обеспечения программным продуктом или системой доступности данных только для авторизованных субъектов.

Целостность (Integrity) – степень предотвращения системой, программным продуктом или компонентом неавторизованного доступа или неавторизованной модификации компьютерных программ или данных.

Неопровержимость (Non-repudiation) – степень доказательности того, что некоторые действия или события имели место, так что в дальнейшем от них нельзя будет отказаться.

Отслеживаемость (Accountability) – степень однозначной отслеживаемости действий некоторого объекта до самого объекта.

Аутентичность (Authenticity) – степень обеспечения идентификации субъекта или ресурса с целью предъявления требований.

12.3.3.7. Сопровождаемость

Сопровождаемость (Maintainability) – степень результативности и эффективности модификаций программного продукта или системы запланированным персоналом сопровождения. Модификации могут включать исправления, улучшения или адаптацию программного средства к изменениям в окружающей среде, требованиях и функциональных спецификациях.

Сопровождаемость может быть интерпретирована либо как внутренняя способность программного продукта или системы облегчать работы по сопровождению, либо как качество в использовании. В последнем случае *Сопровождаемость* оценивается экспериментально персоналом сопровождения с целевой установкой сопровождения программного продукта или системы.

Подхарактеристиками *Сопровождаемости* являются *Модульность*, *Повторная используемость*, *Анализируемость*, *Модифицируемость*, *Тестируемость* (см. рис. 12.8).

Модульность (Modularity) – степень компоновки системы или компьютерной программы из таких дискретных компонентов, что изменения в одном компоненте имеют минимальное влияние на другие компоненты.

Данная подхарактеристика является новой по отношению к модели внешнего и внутреннего качества, определенной в *ISO/IEC 9126–1:2001* (см. подразд. 11.3).

Повторная используемость (Reusability) – степень применимости объекта более чем в одной системе или при разработке других средств.

В модели внешнего и внутреннего качества, определенной в стандарте *ISO/IEC 9126–1:2001*, данная подхарактеристика отсутствует (см. подразд. 11.3).

Анализируемость (Analysability) – степень результативности и эффективности оценки влияния запланированного изменения программного продукта или системы на одну или более их частей, диагностирования продукта на отсутствие или причины отказов, а также идентификации частей, которые должны быть модифицированы.

Модифицируемость (Modifiability) – степень результативности и эффективности модификаций программного продукта или системы без внесения дефектов или ухудшения качества существующего продукта.

Реализация модификаций включает проектирование, кодирование, документирование и верификацию изменений. *Модифицируемость* представляет собой комбинацию способности к изменениям и стабильности. На модифицируемость могут оказывать влияние *Модульность* и *Анализируемость*.

В модели внешнего и внутреннего качества, определенной в *ISO/IEC 9126–1:2001*, данной подхарактеристике соответствует подхарактеристика *Изменяемость (Changeability)*, см. подразд. 11.3).

Тестируемость (Testability) – степень результативности и эффективности установления критериев тестов для системы, программного продукта или компонента и выполнения данных тестов для определения, удовлетворены ли установленные критерии.

12.3.3.8. Мобильность

Мобильность (Portability) – степень результативности и эффективности переноса системы, программного продукта или компонента из одной аппаратной, программной или иной эксплуатационной или используемой среды в другую.

Мобильность может быть интерпретирована либо как внутренняя способность программного продукта или системы облегчать работы по переносу, либо как качество в использовании. В последнем случае *Мобильность* оценивается экспериментально с целевой установкой переноса программного продукта или системы.

Подхарактеристиками *Мобильности* являются *Адаптируемость*, *Настраиваемость*, *Взаимозаменяемость* (см. рис. 12.8).

Адаптируемость (Adaptability) – степень результативности и эффективности адаптации программного продукта или системы к другим существующим или разворачивающимся аппаратным, программным или иным эксплуатационным или используемым средам. *Адаптируемость* включает масштабирование внутренних свойств, таких как поля экрана, таблицы, величины транзакций, форматы отчетов и т. д.

Адаптации включают соответствующие действия, выполняемые специализированным персоналом поддержки, бизнес-персоналом, эксплуатирующим персоналом или конечными пользователями.

Настраиваемость (Простота внедрения, Инсталлируемость, Installability) – степень результативности и эффективности инсталляции и/или деинсталляции программного продукта или системы в заданной среде окружения.

Если программный продукт или система должны инсталлироваться конечным пользователем, то настраиваемость может влиять на такие подхарактеристики качества, как *Функциональное соответствие* и *Простоту использования*.

Взаимозаменяемость (Replaceability) – степень замещения программным продуктом другого заданного программного продукта для той же цели в той же среде окружения.

Например, взаимозаменяемость новой версии важна для пользователя при замене старой версии программного продукта на новую.

12.3.4. Модель качества данных по стандарту ISO/IEC 25012:2008

В настоящее время во всем мире возрастает количество данных и информации, обрабатываемых компьютерными системами. Как правило, жизненный цикл данных является существенно более длительным, чем жизненный цикл программных средств. С учетом этого качество данных является ключевым компонентом качества и полезности информации, получаемой из этих данных. Подавляющее большинство бизнес-процессов зависит от качества данных.

В этой связи общим предварительным условием для всех проектов, выполняемых в области информационных технологий, является качество данных, передаваемых, обрабатываемых и используемых компьютерными системами и пользователями этих систем.

Стандарт *ISO/IEC 25012:2008* ориентирован на качество данных, являющихся частью компьютерной системы, и определяет характеристики качества целевых данных, применяемых людьми и системами.

Под **целевыми данными** подразумеваются данные, которые организация решает подвергнуть анализу и аттестации (валидации) с помощью модели качества.

Стандарт *ISO/IEC 25012:2008* определяет двухуровневую иерархическую модель качества данных. На верхнем уровне модели находится пятнадцать характеристик качества данных. Характеристики определяются непосредственно мерами качества данных. Примеры мер качества данных приведены в подразд. 15.3.

Характеристики качества данных рассматриваются с двух точек зрения – собственное качество данных и системно-зависимое качество данных.

Собственное качество данных определяется степенью, в которой характеристики качества данных обладают внутренними возможностями удовлетворять заданные и подразумеваемые потребности при использовании данных в заданных условиях.

Собственное качество данных относится к самим данным, в частности к значениям области данных и возможным ограничениям (например, ограничениям, определяемым нормативами предметной области, требующими соответствующего уровня характеристик качества данных в конкретном приложении), отношениям между значениями данных (например их согласованности), метаданным.

Системно-зависимое качество данных определяется степенью, в которой качество данных достигается и сохраняется в компьютерной системе при использовании данных в заданных условиях.

Системно-зависимое качество данных зависит от технологической области, в которой используются данные. Оно достигается за счет возможностей компонентов компьютерной системы, таких как аппаратные средства (например средства, обеспечивающие доступность данных или получение требуемой точности), программные средства компьютерной системы (например, программное средство резервного копирования для достижения восстанавливаемости данных), другие программные средства (например, средства миграции для достижения мобильности данных).

Характеристики качества данных приведены на рис. 12.9. Как видно из данного рисунка, некоторые характеристики значимы только для собственного качества данных, некоторые – только для системно-зависимого качества данных. Отдельные характеристики имеют значимость как для собственного, так и для системно-зависимого качества данных.

Для разных правообладателей характеристики качества данных имеют различную важность и приоритет.

При определении ряда характеристик и мер качества данных в стандарте *ISO/IEC 25012:2008* используются понятия сущности и экземпляра сущности.

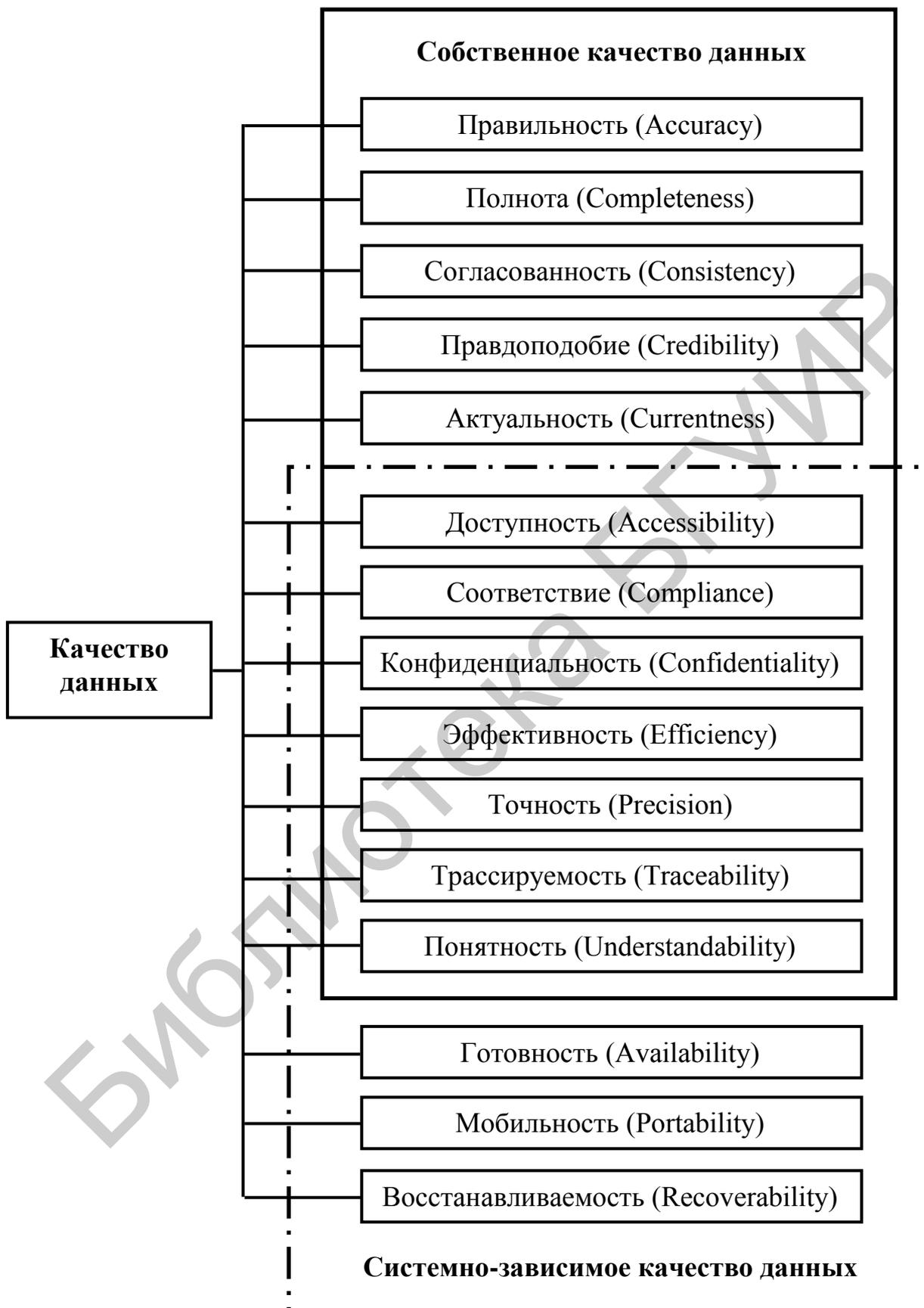


Рис. 12.9. Модель качества данных по ISO/IEC 25012:2008

Сущность (entity) – представление множества реальных или абстрактных объектов предметной области одного типа, для которых свойственны одни и те же характеристики и которые подчинены одному и тому же набору правил и линий поведения и участвуют в одних и тех же связях. Объектами предметной области могут быть, например, люди, предметы, места, идеи, события, документы и т. п.

Экземпляр сущности (entity instance) – один элемент из множества физических или абстрактных объектов, представляемых сущностью.

Во всех определениях характеристик и мер качества данных понятие «атрибут» эквивалентно понятию «свойство».

12.3.5. Характеристики качества данных с собственной точки зрения по стандарту ISO/IEC 25012:2008

12.3.5.1. Правильность

Правильность (Accuracy) – степень обеспечения атрибутами данных корректности представления истинных значений соответствующих свойств некоторых понятий или результатов в заданном контексте использования.

Существует два основных аспекта характеристики **Правильность** – синтаксическая правильность и семантическая правильность.

Синтаксическая правильность определяется как точность попадания значений данных в набор значений, относящихся к синтаксически корректной области определения.

Примером низкой степени синтаксической правильности является сохранение слова *Mary* в виде *Marj* (при написании имени *Mary* сделана синтаксическая ошибка).

Семантическая правильность определяется как точность попадания значений данных в набор значений, относящихся к семантически корректной области определения.

Примером низкой степени семантической правильности является сохранение вместо имени *John* имени *George*. Синтаксически эти имена написаны правильно, но смысл у этих имен разный, поскольку они относятся к разным людям.

Пример собственной меры **Правильности** данных приведен в подразд. 15.3.

12.3.5.2. Полнота

Полнота (Completeness) – степень, в которой данные, относящиеся к некоторой сущности, имеют значения для всех полагаемых атрибутов и экземпляров данной сущности в заданном контексте использования.

Например, для базы данных сотрудников значение характеристики *Полнота* данных снижается, если в записях некоторых сотрудников отсутствует информация об их номере телефона или дате рождения.

Пример собственной меры *Полноты* данных приведен в подразд. 15.3.

12.3.5.3. Согласованность

Согласованность (Consistency) – степень обеспечения атрибутами данных непротиворечивости и согласованности данных с другими данными в заданном контексте использования. Это может относиться как к данным одной сущности, так и к соответствующим данным сравниваемых сущностей.

Примером несогласованности данных является дата рождения сотрудника, более поздняя, чем дата его приема на работу.

Пример собственной меры *Согласованности* данных приведен в подразд. 15.3.

12.3.5.4. Правдоподобие

Правдоподобие (Credibility) – степень наличия у данных таких атрибутов (свойств), которые пользователями считаются истинными и вызывающими доверие в заданном контексте использования.

Правдоподобие включает концепцию аутентичности (идентификацию источников, авторства, действий).

Например, данные, сертифицированные независимой и вызывающей доверие организацией, рассматриваются как правдоподобные.

Пример собственной меры *Правдоподобия* данных приведен в подразд. 15.3.

12.3.5.5. Актуальность

Актуальность (Currentness) – степень, в которой атрибуты (свойства) данных являются неустаревшими в заданном контексте использования.

Например, чтобы быть актуальным, расписание движения поездов на железнодорожной станции должно своевременно обновляться при изменении запланированного времени или платформы отправления.

Пример собственной меры *Актуальности* данных приведен в подразд. 15.3.

12.3.6. Характеристики качества данных с собственной и системно-зависимой точек зрения по стандарту ISO/IEC 25012:2008

12.3.6.1. Доступность

Доступность (Accessibility) – степень доступности данных в заданном контексте использования, в частности, для людей, нуждающихся в поддерживающей технологии или специальной конфигурации в связи с их ограниченными возможностями.

Например, звуковые данные со считывающего устройства не могут быть сохранены в виде изображения. Это ограничивает степень доступности таких данных для неслышающих людей. Наоборот, использование текстовых неозвученных данных ограничивает степень доступности таких данных для невидящих людей.

Примеры собственной и системно-зависимой мер *Доступности* данных приведены в подразд. 15.3.

12.3.6.2. Соответствие

Соответствие (Compliance) – степень соответствия атрибутов данных стандартам, соглашениям, действующим постановлениям и другим правилам, связанным с качеством данных в заданном контексте использования.

Например, данные по кредитным рискам банка должны соответствовать конкретным законам и стандартам.

Примеры собственной и системно-зависимой мер *Соответствия* данных приведены в подразд. 15.3.

12.3.6.3. Конфиденциальность

Конфиденциальность (Confidentiality) – степень обеспечения атрибутами данных их доступности и интерпретируемости (понятности) только для авторизованных пользователей в заданном контексте использования.

Например, данные, которые относятся к персональной или конфиденциальной информации (такой как здоровье, прибыль, сумма вклада), должны быть доступны только для авторизованных пользователей или должны быть записаны с помощью секретного кода.

Примеры собственной и системно-зависимой мер *Конфиденциальности* данных приведены в подразд. 15.3.

12.3.6.4. Эффективность

Эффективность (Efficiency) – степень, в которой атрибуты данных позволяют выполнить обработку данных и обеспечить ожидаемый уровень производительности за счет использования соответствующего количества и типов ресурсов в заданном контексте использования.

Например, использование большего, чем необходимо, места для хранения данных может явиться причиной ненужных затрат хранилища данных, памяти и времени.

Примеры собственной и системно-зависимой мер *Эффективности* данных приведены в подразд. 15.3.

12.3.6.5. Точность

Точность (Precision) – степень обеспечения атрибутами данных строгого соответствия или распознавания в заданном контексте использования.

Например, представление данных с точностью в пять десятичных разрядов дает возможность реализовать различные функции на более высоком уровне, чем представление данных с точностью в два десятичных разряда.

Примеры собственной и системно-зависимой мер *Точности* данных приведены в подразд. 15.3.

12.3.6.6. Трассируемость

Трассируемость (Traceability) – степень, в которой атрибуты данных обеспечивают проверку пути доступа к данным и проверку пути любых изменений, производимых над данными, в заданном контексте использования.

Например, государственные органы должны хранить информацию о доступе, выполненном пользователями, для сбора сведений о том, кто читал или записывал конфиденциальные данные.

Примеры собственной и системно-зависимой мер *Трассируемости* данных приведены в подразд. 15.3.

12.3.6.7. Понятность

Понятность (Understandability) – степень наличия у данных атрибутов, позволяющих данным быть прочитанными и интерпретированными (истолкованными) пользователями, а также быть выраженными с помощью соответствующих языков, символов и единиц в заданном контексте использования.

Некоторая информация о *Понятности* данных обеспечивается метаданными.

Например, для обозначения страны более понятным является использование стандартной аббревиатуры (РБ, РФ, КНР, КНДР, США), чем какого-либо цифрового кода; аналогичные рассуждения справедливы для обозначения университетов (БГУИР, БГУ, БНТУ).

Примеры собственной и системно-зависимой мер *Понятности* данных приведены в подразд. 15.3.

12.3.7. Характеристики качества данных с системно-зависимой точки зрения по стандарту ISO/IEC 25012:2008

12.3.7.1. Готовность

Готовность (Availability) – степень, в которой атрибуты данных позволяют извлекать эти данные разрешенным пользователям или приложениям в заданном контексте использования.

Частным случаем *Готовности* является одновременный доступ (как для чтения, так и для обновления данных) более чем для одного пользователя и/или приложения.

Примером *Готовности* данных является способность данных быть используемыми в течение заданного периода времени. Данные должны также позволять их использовать во время выполнения управляющих операций (таких как резервное копирование).

Пример системно-зависимой меры *Готовности* данных приведен в подразд. 15.3.

12.3.7.2. Мобильность

Мобильность (Portability) – степень, в которой атрибуты данных позволяют данным быть установленными, замененными или перенесенными из одной системы в другую с сохранением существующего качества в заданном контексте использования.

Пример системно-зависимой меры *Мобильности* данных приведен в подразд. 15.3.

12.3.7.3. Восстанавливаемость

Восстанавливаемость (Recoverability) – степень, в которой атрибуты данных позволяют данным поддерживать и сохранять заданный уровень функционирования и качества даже в случае отказа в заданном контексте использования.

Восстанавливаемость данных может быть обеспечена такими функциями, как фиксация/точка синхронизации, откат (характеризует устойчивость к ошибке) или механизмами резервного копирования/восстановления.

Пример системно-зависимой меры *Восстанавливаемости* данных приведен в подразд. 15.3.

12.4. Группа стандартов ISO/IEC 2502n

Как уже отмечалось, численные значения свойств (атрибутов) качества системы, программного продукта или данных, находящиеся на нижнем уровне иерархической структуры моделей качества (см. рис. 12.6), определяются с помощью мер качества. Измерению свойств качества посвящена группа стандартов измерения качества *ISO/IEC 2502n*.

В данную группу стандартов в настоящее время входят следующие стандарты:

- *ISO/IEC 25020:2007. Программная инженерия. Требования к качеству и оценка программных продуктов (SQuaRE). Эталонная модель измерений* [43];
- *ISO/IEC 25021:2012. Системная и программная инженерия. Требования к качеству и оценка систем и программных средств (SQuaRE). Элементы мер качества* [44];

- **ISO/IEC 25022. Системная и программная инженерия. Требования к качеству и оценка систем и программных средств (SQuaRE). Измерение качества в использовании** (данный стандарт является ревизией *ISO/IEC TR 9126–4:2004*; в настоящее время находится в разработке – стадия 60.00 на 31.03.2016 г., см. подразд. 1.2) [45];

- **ISO/IEC 25023. Системная и программная инженерия. Требования к качеству и оценка систем и программных средств (SQuaRE). Измерение качества систем и программных продуктов** (данный стандарт является ревизией стандартов *ISO/IEC TR 9126–2:2003* и *ISO/IEC TR 9126–3:2003*; в настоящее время находится в разработке – стадия 60.00 на 31.03.2016 г., см. подразд. 1.2) [46];

- **ISO/IEC 25024:2015. Системная и программная инженерия. Требования к качеству и оценка систем и программных средств (SQuaRE). Измерение качества данных** [47].

На рис. 12.10 приведена структура группы стандартов измерения качества [44].

Основные положения данных стандартов приведены в разд. 15.

12.5. Группа стандартов ISO/IEC 2503n

В данную группу стандартов в настоящее время входит один стандарт: **ISO/IEC 25030:2007. Программная инженерия. Требования к качеству и оценка программных продуктов (SQuaRE). Требования к качеству** [48].

12.5.1. Место требований к программным средствам в системных требованиях по стандарту ISO/IEC 25030:2007

В стандарте *ISO/IEC 25030:2007* определены требования к качеству программного продукта и рекомендации по их выявлению и определению. Стандарт предназначен для организаций, приобретающих или поставляющих программные средства.

Основой для классификации и количественного определения требований к качеству программных средств в терминах мер качества является модель качества из стандарта *ISO/IEC 25010:2011*.

Основное внимание в стандарте *ISO/IEC 25030:2007* уделяется программным средствам. Однако программные средства обычно являются частью некоторой большой системы. В этой связи полезно и в ряде случаев необходимо рассматривать требования к качеству программных средств с системной точки зрения.

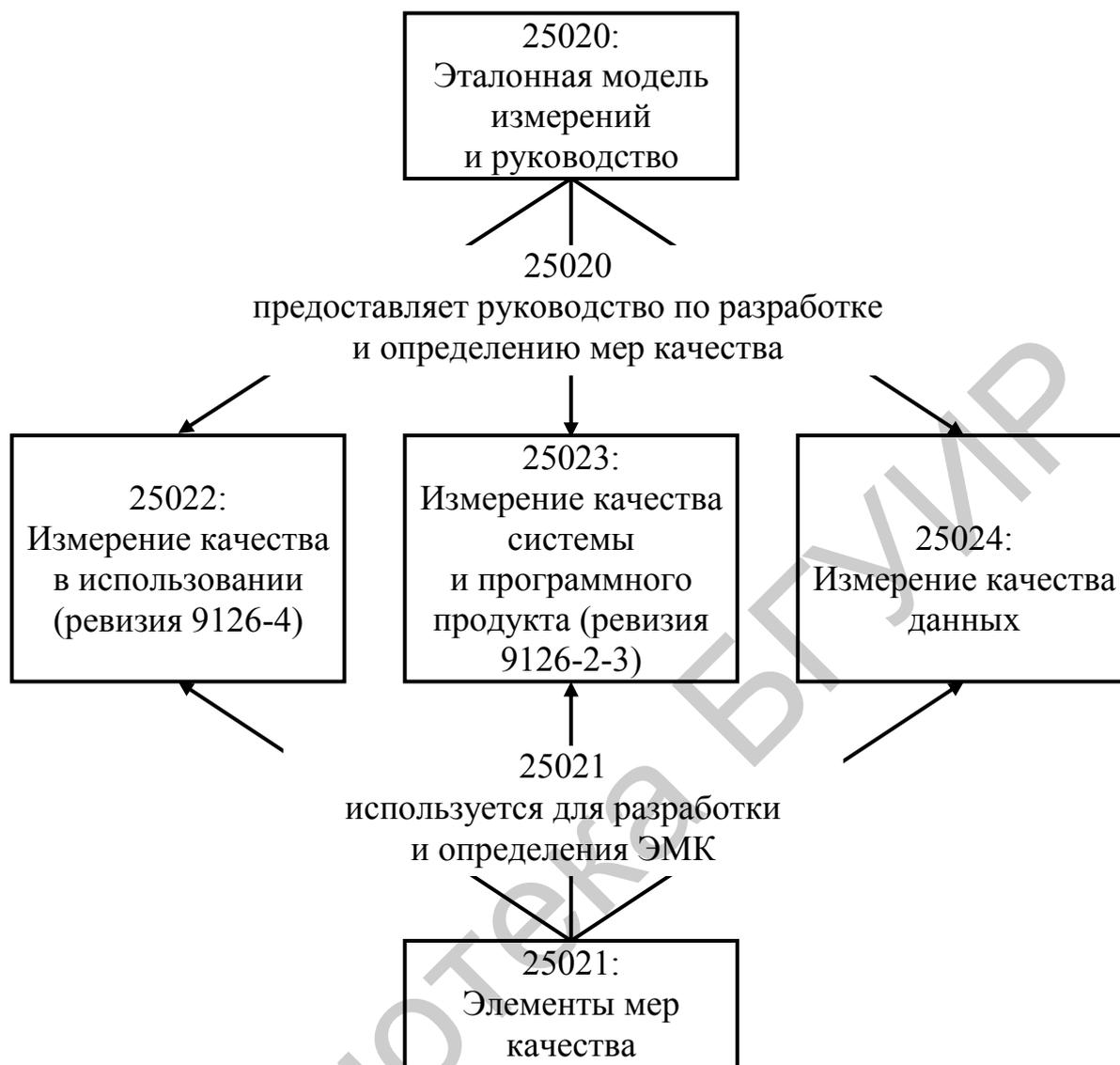


Рис. 12.10. Структура группы измерения качества

На рис. 12.11 приведен пример модели корпоративной системы, отражающий иерархию систем. Модель включает информационную систему, механическую систему, бизнес-процессы человека и связи между ними [48].

В течение жизненного цикла у системы имеются различные правообладатели. В общем случае к ним относятся физические лица (например конечные пользователи), организации (например организации-пользователи, организации-разработчики) и органы (например нормативно-регулирующие органы). Правообладатели имеют различные требования к системе и ожидания от нее. И эти требования и ожидания могут изменяться в течение жизненного цикла.

Потребности правообладателей могут быть точно установлены или только подразумеваться. В ряде случаев правообладатели не могут четко сформулировать все свои потребности или не знают о них. Для выявления всех

потребностей могут использоваться, например, методы моделирования предметной области и прототипирования.

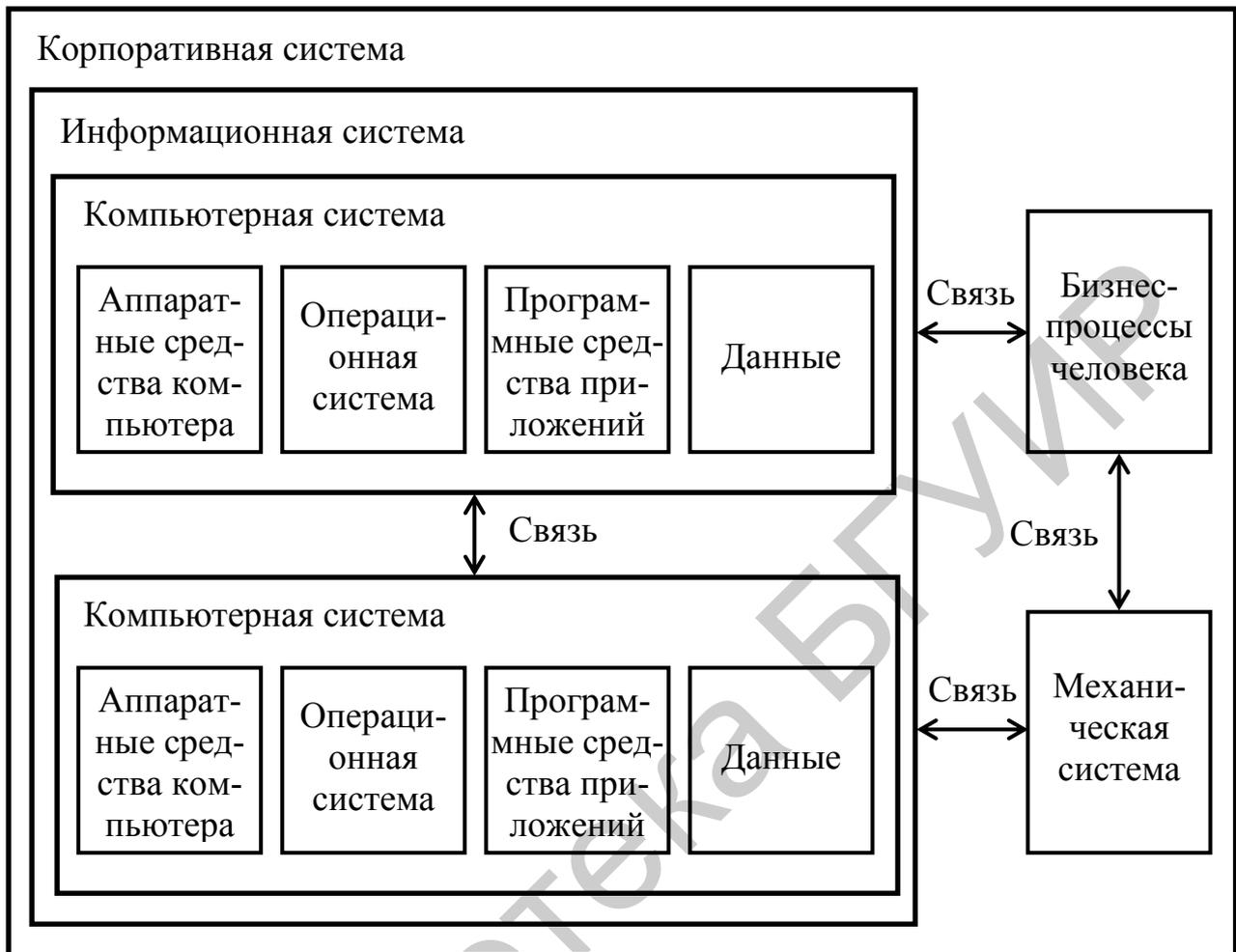


Рис. 12.11. Пример модели корпоративной системы по ISO/IEC 25030:2007

Потребности и ожидания правообладателей в жизненном цикле систем идентифицируются с помощью процессов определения и анализа требований в соответствии с рис. 12.12 [48].

Процесс определения требований правообладателей и процесс анализа требований представляют собой процессы жизненного цикла, структура которых регламентирована положениями международных стандартов *ISO/IEC 15288:2008* (см. разд. 7, рис. 7.2) и *ISO/IEC 12207:2008* (в данном стандарте процесс анализа требований называется процессом анализа системных требований, см. п. 8.3.4, рис. 8.2).

В процессе определения требований правообладателей выявляются все значимые правообладатели программного средства, их потребности и пожелания. Определяются ограничения, налагаемые приобретателями. На основе выявленных потребностей, пожеланий и ограничений формулируются в общем виде требования правообладателей.

В процессе анализа требований требования правообладателей преобразуются в системные требования, представляющие собой формализованные с технической точки зрения требования, пригодные для реализации разрабатываемой системы.



Рис. 12.12. Определение и анализ требований правообладателей по ISO/IEC 25030:2007

Системные требования могут верифицироваться и являются основой для удовлетворения требований правообладателей.

Как видно из рис. 12.11, в общем случае система состоит из различных элементов. У каждого из них имеются свои собственные характеристики и назначение в системе. В связи с этим на основе системных требований должны быть сформулированы требования ко всем элементам системы.

Чаще всего при разработке системы требования правообладателей не включают требований к конкретным программным средствам. Обычно требования к программным средствам формулируются после проектирования архитектуры системы (см. п. 8.3.4, рис. 8.2).

На рис. 12.13 отражена иерархия требований, основанная на результатах проектирования архитектуры системы, модель которой представлена на рис. 12.11. Как видно из рис. 12.13, требования к программным средствам и, в частности, к их качеству, находятся на нижнем уровне иерархии системных требований.

12.5.2. Связь видов качества программных средств с архитектурой системы по стандарту ISO/IEC 25030:2007

Из рис. 12.13 следует, что качество системы определяется качеством элементов системы и их взаимодействия. В стандарте *ISO/IEC 25030:2007* и в

серии *SQuaRE* в целом основное внимание уделяется качеству программных средств как части системы.

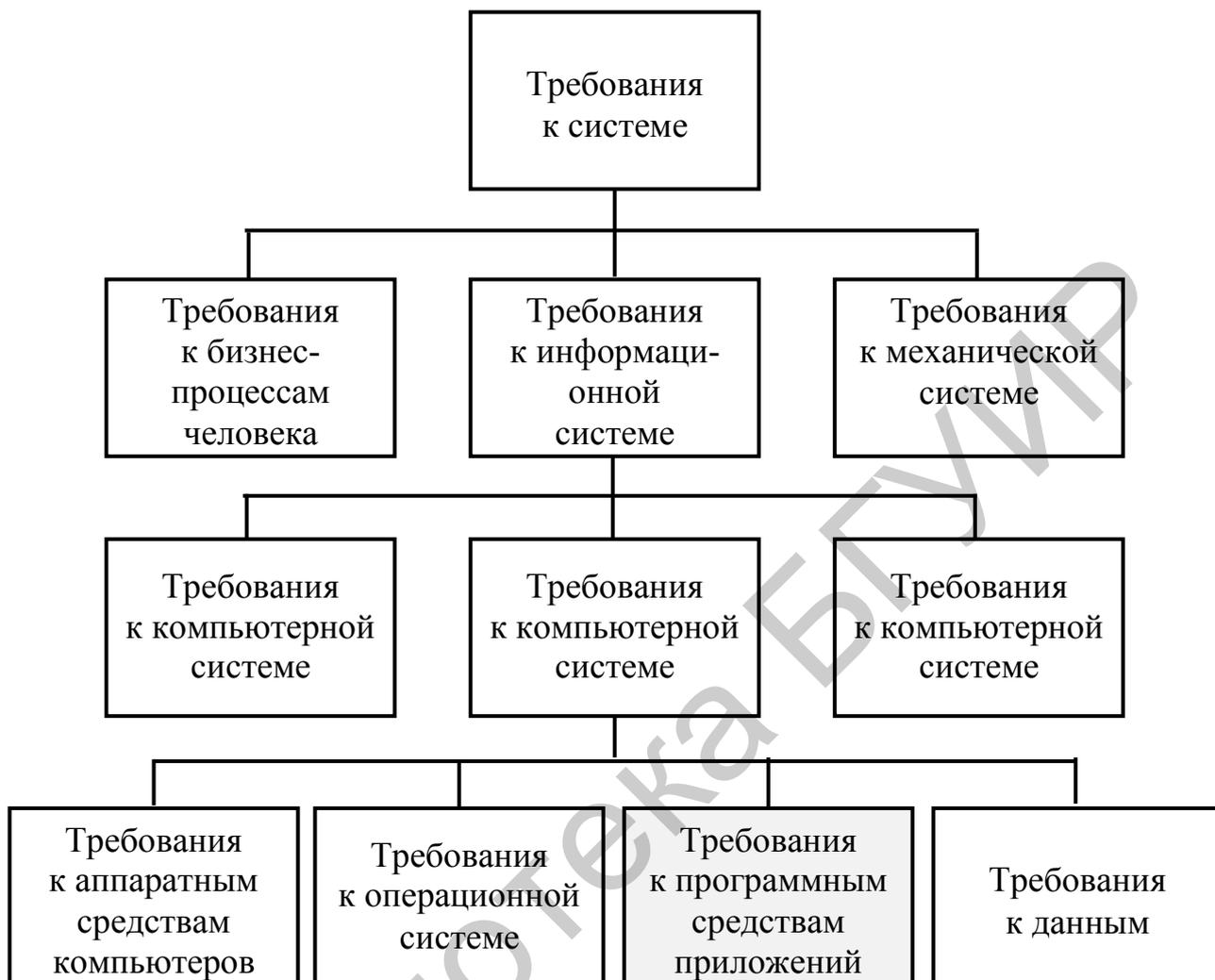


Рис. 12.13. Иерархия требований к системе по ISO/IEC 25030:2007

С учетом этого в *ISO/IEC 25030:2007* определены *три точки зрения на качество программных средств*:

- качество программных средств в использовании;
- внешнее качество программных средств;
- внутреннее качество программных средств.

Точка зрения *качества программного средства в использовании* связана с его применением в эксплуатационной среде для решения заданных задач заданными пользователями.

Внешнее качество программного средства представляет собой точку зрения на программное средство как на «черный ящик» и определяет динамические свойства программного средства, связанные с его работой в операционной системе на аппаратных средствах компьютера.

Внутреннее качество программного средства представляет собой точку зрения на программное средство как на «белый ящик» и определяет статические свойства программного продукта в процессе его разработки.

Данные виды качества программных средств определяются с помощью соответствующих моделей качества, регламентированных стандартом *ISO/IEC 25010:2011*, и подробно рассмотрены в подразд. 12.3.

На рис. 12.14 представлена связь различных точек зрения на качество программных средств с моделью системы.

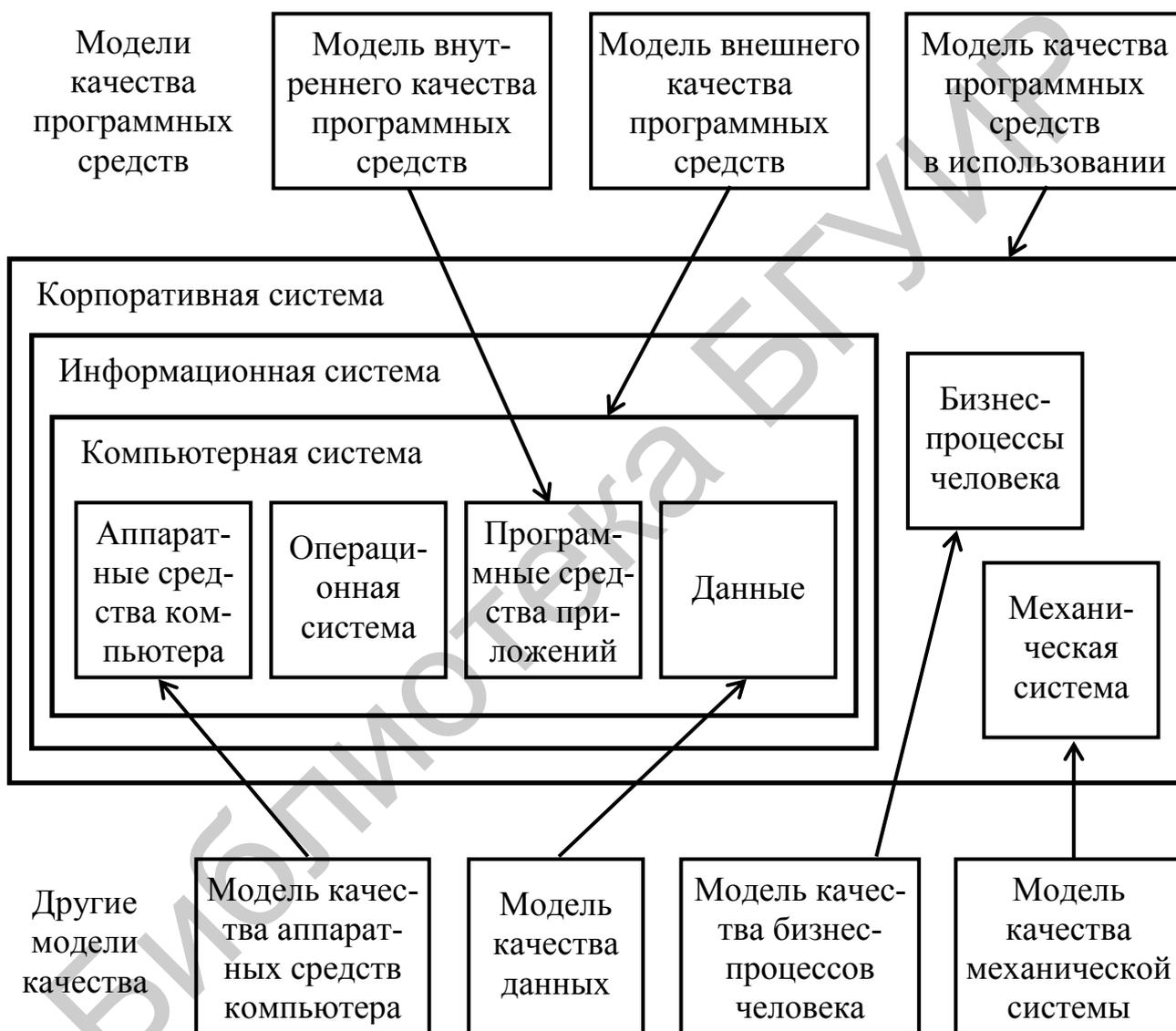


Рис. 12.14. Связь видов качества программных средств с моделью системы

12.5.3. Требования к качеству программных средств и систем по стандарту ISO/IEC 25030:2007

В процессе определения требований правообладателей (см. п. 12.5.1, рис. 12.12) среди других потребностей выявляются и потребности правообладателей в качестве. На выходе данного процесса формируются требования правообладателей к качеству. Формирование этих требований базируется на модели качества из стандарта *ISO/IEC 25010:2011*.

В процессе анализа требований на основе требований правообладателей к качеству разрабатываются формализованные требования к качеству систем.

В последующем процессе проектирования архитектуры системы (см. рис. 7.2) принимаются архитектурные решения, выделяющие объекты системы, в том числе программные. При этом требования к системе распределяются между ее объектами. Это в полной мере касается и требований к качеству.

Классификация требований к системе представлена на рис. 12.15. В соответствии с данным рисунком системные требования подразделяются на требования к программным средствам и другие требования к системе. В свою очередь требования к программным средствам делятся на требования к разработке программных средств и требования к программному продукту.

Требования к программному продукту базируются на **классификации свойств программных средств**.

В соответствии с данной классификацией свойства программных средств подразделяются на собственные и назначенные. Собственные свойства делятся на функциональные и свойства качества (внутреннего, внешнего и качества в использовании).

В *ISO/IEC 25030:2007* подчеркнуто, что понятие *функциональные свойства* не является аналогом свойств, связанных с характеристикой *Функциональная пригодность* из модели качества продукта *ISO/IEC 25010:2011* (см. п. 12.3.3). Функциональные свойства определяют, **что** программное средство способно делать, и вытекают из функциональной спецификации требований. Функциональные свойства не относятся к свойствам качества, хотя и находятся в тесной взаимосвязи с последними.

Свойства качества (в том числе и связанные с характеристикой *Функциональная пригодность*) определяют, **насколько хорошо** программное средство выполняет свои заданные спецификацией функции.

Назначенные свойства не рассматриваются как свойства качества программного средства, поскольку они могут быть изменены без изменения свойств программного средства. Назначенные свойства состоят их управленческих свойств.

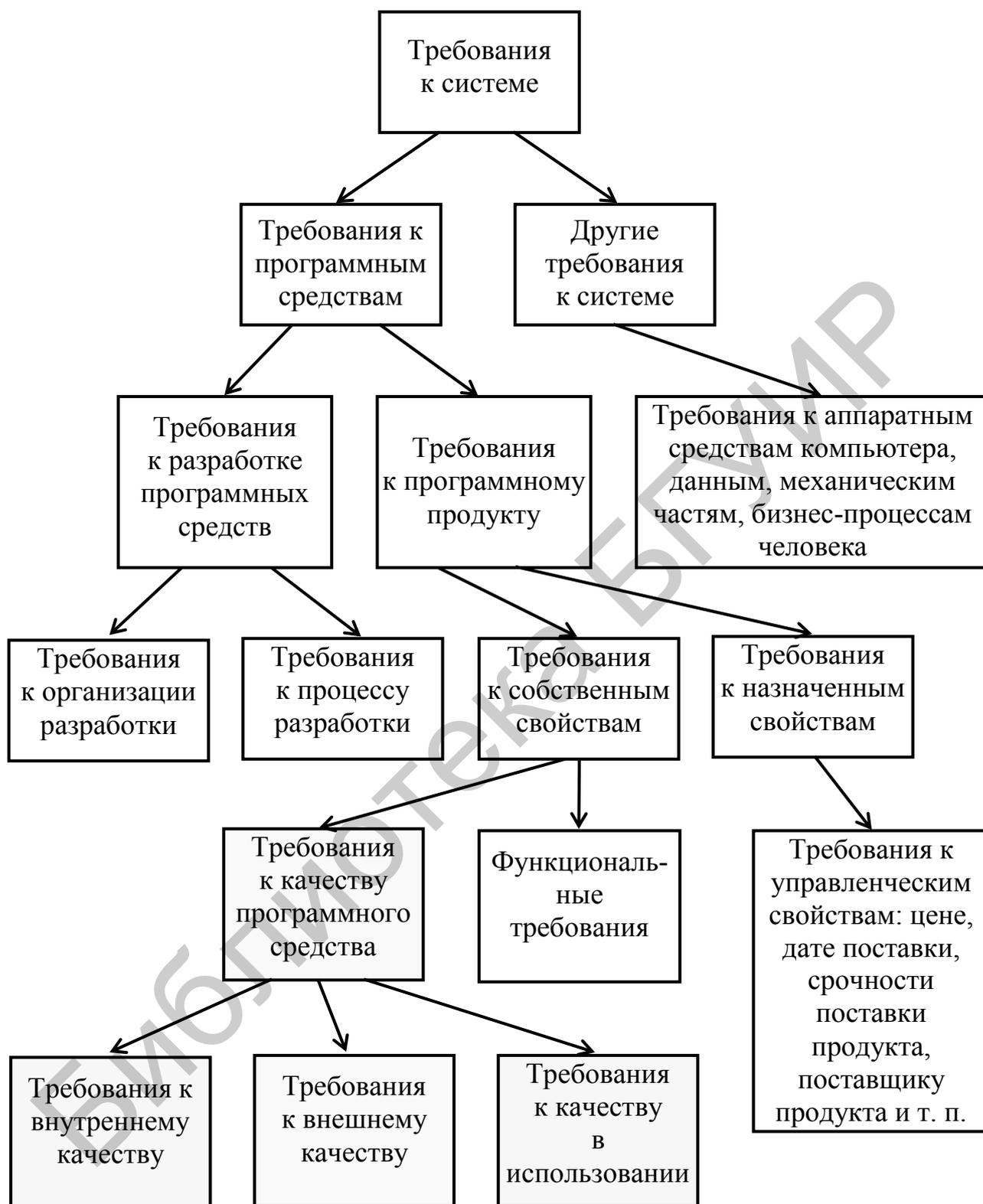


Рис. 12.15. Классификация требований к системе по ISO/IEC 25030:2007

Измеримые на количественном или качественном уровне собственные свойства качества программных средств называются атрибутами качества. Для измерения атрибутов качества используются меры качества. Правила измерения атрибутов с помощью мер изложены в группе стандартов *ISO/IEC 2502n* (см. разд. 15).

С учетом изложенного требования к качеству программных средств должны быть выражены в терминах мер качества. При этом следует учитывать жизненный цикл требований к качеству.

Жизненный цикл требований к качеству определяется наличием трех точек зрения на качество программных средств в системе. С учетом этого требования к качеству программных средств разделяются на требования к качеству в использовании, требования к внешнему качеству и требования к внутреннему качеству (см. рис. 12.15). Место данных видов требований к качеству в их жизненном цикле и взаимозависимость видов требований приведены на рис. 11.1 и 12.4 (см. подразд. 11.2 и п. 12.2.2).

12.5.4. Требования к требованиям к качеству по ISO/IEC 25030:2007

В стандарте *ISO/IEC 25030:2007* определены требования и рекомендации, относящиеся к требованиям к качеству программных продуктов.

При определении требований правообладателей должно быть документально оформлено целевое назначение системы. Документация должна включать использование, преимущества, заданный срок службы системы, критичность системы, зависимости от рисков (например, вопросы безопасности или свободы от рисков), системные ограничения.

Должны быть задокументированы цели определения требований правообладателей к качеству. Для этого должны быть перечислены все существенные правообладатели, их роли и интересы. Должно быть задокументировано, какие конкретно правообладатели будут приниматься во внимание при определении требований правообладателей к качеству.

Должны быть выявлены и документально оформлены все требования правообладателей к качеству, в том числе установленные, предполагаемые и неизвестные правообладателям.

Должна быть обеспечена трассируемость от каждого требования правообладателей к качеству до конкретного правообладателя или класса правообладателей.

Требования правообладателей к качеству должны быть аттестованы и утверждены.

Поскольку программные средства являются частью некоторой системы, то архитектурные решения, принятые на верхних уровнях иерархической структуры системы, определяют границы и интерфейсы программных средств.

После разработки системной архитектуры должно быть документально оформлено целевое назначение программных средств в системе, выделены

требования правообладателей (в том числе и требования к качеству), влияющие на программные средства. На основе выделенных требований правообладателей должны быть идентифицированы требования к качеству программных средств. При этом должна быть обеспечена трассируемость данных требований с требованиями всех важных правообладателей.

В качестве основы для модели качества программных средств должна использоваться модель качества из стандарта *ISO/IEC 25010:2011*. Применяемая модель качества должна быть оформлена документально.

Требования к качеству программных средств должны быть согласованы с характеристиками и подхарактеристиками качества утвержденной модели качества программных средств.

В соответствии с моделью качества требования к качеству должны быть разделены на требования к качеству в использовании, требования к внешнему качеству и требования к внутреннему качеству (см. рис. 12.15).

Требования к качеству должны быть заданы в терминах мер качества программных средств и связанных с ними базовых значений. При задании базовых значений должны быть определены и их допустимые отклонения.

Меры качества, их базовые значения и допустимые отклонения должны быть документально оформлены в соответствии с положениями группы стандартов *ISO/IEC 2502n* (см. подразд. 12.4). Основные положения данной группы стандартов рассмотрены в разд. 15.

Должны быть также документированы критерии, применяемые при выборе мер качества программных средств. Критерии могут включать значимость, возможность и простоту сбора данных, защиту от проникновения в частную жизнь, простоту интерпретации, применимость на стадии жизненного цикла.

Требования к качеству программных средств должны быть верифицированы. Предварительно должны быть документированы специальные средства, технологии и другие ресурсы, необходимые для верификации требований.

Если при верификации выявлены конфликты между требованиями к качеству или конфликты между требованиями к качеству и ограничениями реализации, они должны быть документально оформлены и устранены. При этом должна быть обеспечена трассируемость первоначальных требований к качеству программных средств с дополнительными или измененными требованиями, позволившими устранить выявленные в требованиях проблемы. Требования к качеству программных средств, которые были заменены или удалены, должны быть отмечены.

Требования к качеству программных средств должны быть оценены и утверждены с регистрацией фамилий лиц, выполняющих оценку и утверждение.

Требования к качеству программных средств должны быть документированы в таком формате, который позволяет управлять ими в соответствии с системой управления конфигурацией и изменениями.

12.6. Группа стандартов ISO/IEC 2504n

Группа стандартов *ISO/IEC 2504n* посвящена оценке качества систем и программных средств.

В данную группу стандартов в настоящее время входят следующие стандарты:

- *ISO/IEC 25040:2011. Системная и программная инженерия. Требования к качеству и оценка систем и программных средств (SQuaRE). Процесс оценки* [49];

- *ISO/IEC 25041:2012. Системная и программная инженерия. Требования к качеству и оценка систем и программных средств (SQuaRE). Руководство по оценке для разработчиков, приобретателей и независимых оценщиков* [50];

- *ISO/IEC 25045:2010. Системная и программная инженерия. Требования к качеству и оценка систем и программных средств (SQuaRE). Модуль оценки для восстанавливаемости* [51].

12.6.1. Общие сведения о стандарте ISO/IEC 25040:2011

Стандарт *ISO/IEC 25040:2011* заменяет собой *ISO/IEC 14598–1:1999*.

Стандарт *ISO/IEC 25040:2011* предназначен для использования разработчиками, приобретателями и независимыми оценщиками систем и программных средств.

Данный стандарт содержит общую концепцию и требования к проведению оценки качества систем и программных средств. Процесс оценки, регламентированный в *ISO/IEC 25040:2011*, может использоваться для оценки качества разрабатываемых промежуточных программных продуктов, разработанных программных средств, программных средств в составе целевой системы, готовых к использованию коммерческих программных средств и др. Стандарт не предназначен для оценки других аспектов программных продуктов, таких как функциональные требования, бизнес-требования, требования к процессам и т. п.

В данном стандарте используются следующие определения.

Оценка (*evaluation*) – систематическое определение степени, в которой объект соответствует заданным критериям. Относительная важность различных характеристик качества программных средств зависит от предполагаемого использования и целей системы, частью которой являются данные программные средства. Чтобы определить, удовлетворяют ли соответствующие характеристики качества требованиям к системе, следует оценить программные продукты.

Покрывание оценки (*evaluation coverage*) – степень, в которой оценка покрывает заданные требования к качеству программного продукта.

Уровень оценки (evaluation level) – строгость оценки, определяющая глубину и тщательность оценки в терминах применяемых методик и достигаемых результатов оценки.

Модуль оценки (evaluation module) – пакет технологий оценки для измерения характеристик, подхарактеристик или атрибутов качества программных средств. Этот пакет включает методы и методики оценки, входную информацию для оценки, данные, которые должны быть оценены и собраны, поддерживающие процедуры и средства.

Строгость оценки (evaluation stringency) – степень удовлетворения характеристиками и подхарактеристиками качества программного продукта критических режимов ожидаемого использования программного продукта.

12.6.2. Обобщенная эталонная модель процесса оценки качества программного продукта по стандарту ISO/IEC 25040:2011

В стандарте *ISO/IEC 25040:2011* определена обобщенная эталонная модель процесса оценки качества программного продукта. Данная модель приведена на рис. 12.16. Модель представлена в виде функциональной диаграммы, отражающей входную информацию, результаты, ограничения и ресурсы процесса оценки качества программных продуктов. На функциональной модели входная информация представляется справа от процесса оценки, результаты оценки – слева, ограничения оценки – сверху, ресурсы для оценки – снизу.

Входная информация и результаты оценки описаны в п. 12.6.3.

Ограничения, налагаемые на процесс оценки качества программного продукта, могут включать:

- потребности конкретного пользователя;
- имеющиеся в наличии ресурсы;
- календарные графики оценки;
- стоимость оценки;
- среду оценки;
- инструментальные средства и методологии оценки;
- отчетность.

Ресурсы для процесса оценки качества программного продукта могут включать:

- применяемые инструментальные средства и методологии оценки, включая модули оценки;
- применяемые стандарты *SQuaRE (ISO/IEC 25001, ISO/IEC 25010, ISO/IEC 2502n, ISO/IEC 25030, ISO/IEC 25041* и др.);
- человеческие ресурсы;

- экономические ресурсы;
- информационную систему для оценки качества программного продукта;
- базу знаний для оценки качества программного продукта.

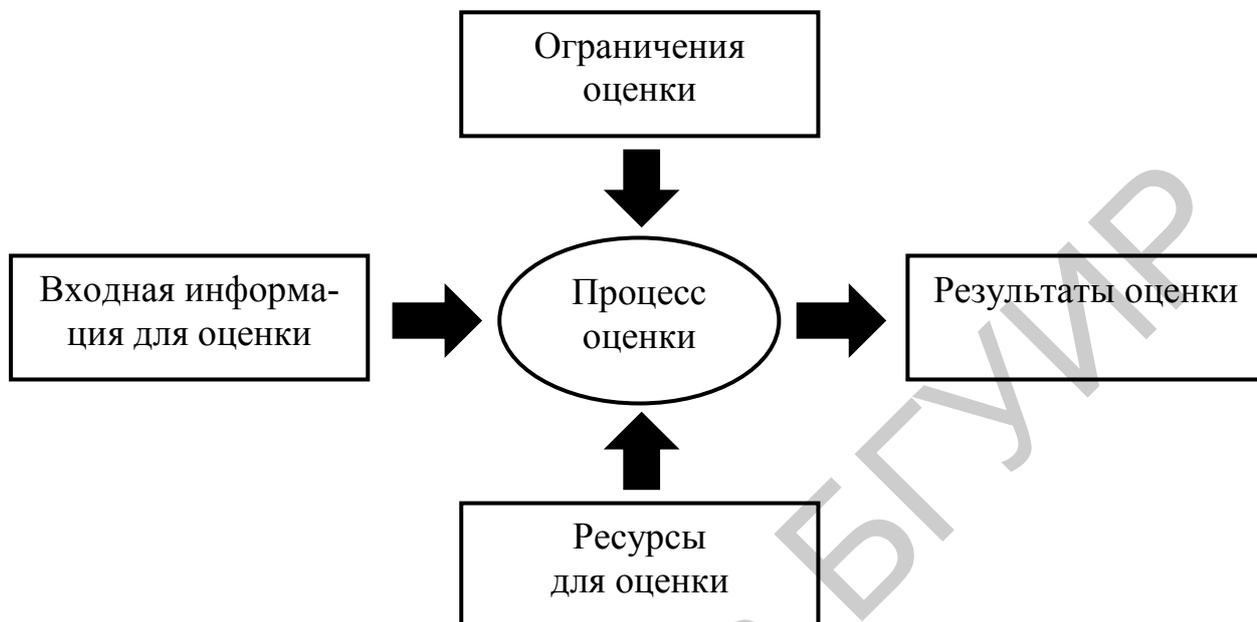


Рис. 12.16. Обобщенная эталонная модель оценки качества программного продукта по ISO/IEC 25040:2011

Эталонная модель процесса оценки качества программного продукта базируется на спецификации требований к качеству программного продукта, разработанной в соответствии с положениями стандарта *ISO/IEC 25030:2007*. В процессе оценки используются модели качества из стандарта *ISO/IEC 25010:2011* и меры качества из группы стандартов *ISO/IEC 2502n*.

12.6.3. Процесс оценки качества программного продукта по стандарту ISO/IEC 25040:2011

Процесс оценки качества программного продукта имеет структуру, соответствующую структуре процессов, определенной в стандартах *ISO/IEC 15288:2008* и *ISO/IEC 12207:2008* (см. рис. 7.1). В соответствии с положениями данных стандартов процесс определяется с помощью *цели* и *выходных результатов*, с учетом которых процесс разделяется на набор *работ*. Каждая работа разделена на набор *задач*.

Целью процесса оценки качества программного продукта является проведение оценки качества заданного промежуточного или конечного программного продукта. **Выходные результаты** процесса оценки качества программного продукта определены посредством результатов оценки каждой работы данного процесса.

Эталонная модель процесса оценки качества программного продукта, отражающая работы и задачи процесса оценки, представлена на рис. 12.17.

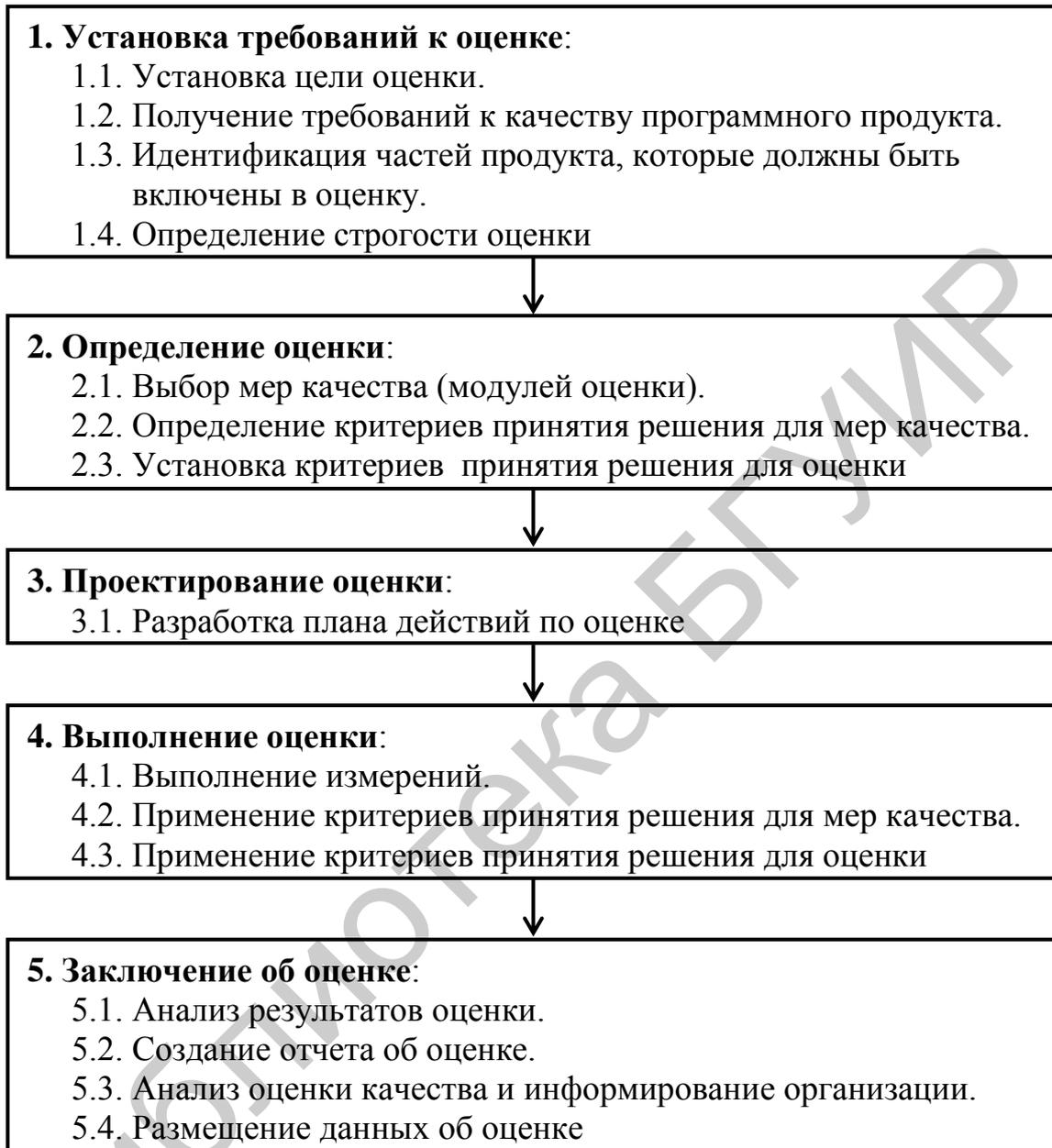


Рис. 12.17. Процесс оценки качества программного продукта по ISO/IEC 25040:2011

Данная модель может быть использована как руководство по оценке качества программного продукта организациями-разработчиками, организациями-приобретателями или независимыми оценщиками в течение или после окончания процесса разработки или процесса приобретения.

Как видно из рис. 12.17, процесс оценки качества программного продукта состоит из **пяти работ**:

- установка требований к оценке;

- определение оценки;
- проектирование оценки;
- выполнение оценки;
- заключение об оценке.

Каждая из данных работ в свою очередь делится на задачи.

Для обеспечения повторяемости, воспроизводимости, беспристрастности и объективности результатов оценки оценщик должен действовать в организационных условиях, гарантирующих достаточное качество его деятельности.

Каждая работа в процессе оценки программного продукта должна быть задокументирована. Отчеты должны включать детальное описание действий, выполненных оценщиком при реализации плана оценки качества программного продукта.

12.6.3.1. Установка требований к оценке

Первая работа *Установка требований к оценке* процесса оценки качества программного продукта состоит из четырех задач (см. рис. 12.17). Входная информация, выходные результаты, ограничения и ресурсы данной работы представлены на функциональной диаграмме, приведенной на рис. 12.18.

Задача 1.1. Установка цели оценки

Оценка качества может быть выполнена как применительно к промежуточному программному продукту, так и к конечному программному продукту.

Целью оценки качества промежуточного программного продукта может быть:

- обеспечение качества продукта;
- принятие решения о приемке промежуточного продукта от субподрядчика;
- определение целесообразности продолжения разработки проекта;
- принятие решения об окончании стадии жизненного цикла и передаче продуктов на следующую стадию;
- предсказание или приблизительная оценка качества конечного программного продукта;
- сбор информации о промежуточных программных продуктах для контроля и управления процессами жизненного цикла и в первую очередь процессом разработки.

Целью оценки качества конечного программного продукта может быть:

- принятие решения о приемке продукта;
- принятие решения о сроке выпуска продукта;
- сравнение продукта с конкурентными продуктами;
- выбор продукта из альтернативных продуктов;

- оценка положительного и отрицательного эффектов использования продукта;
 - определение причины отказа при работе продукта;
 - принятие решения об усовершенствовании или замене продукта.
- Цель оценки качества программного продукта должна быть задокументирована как основание для дальнейших работ и задач оценки.

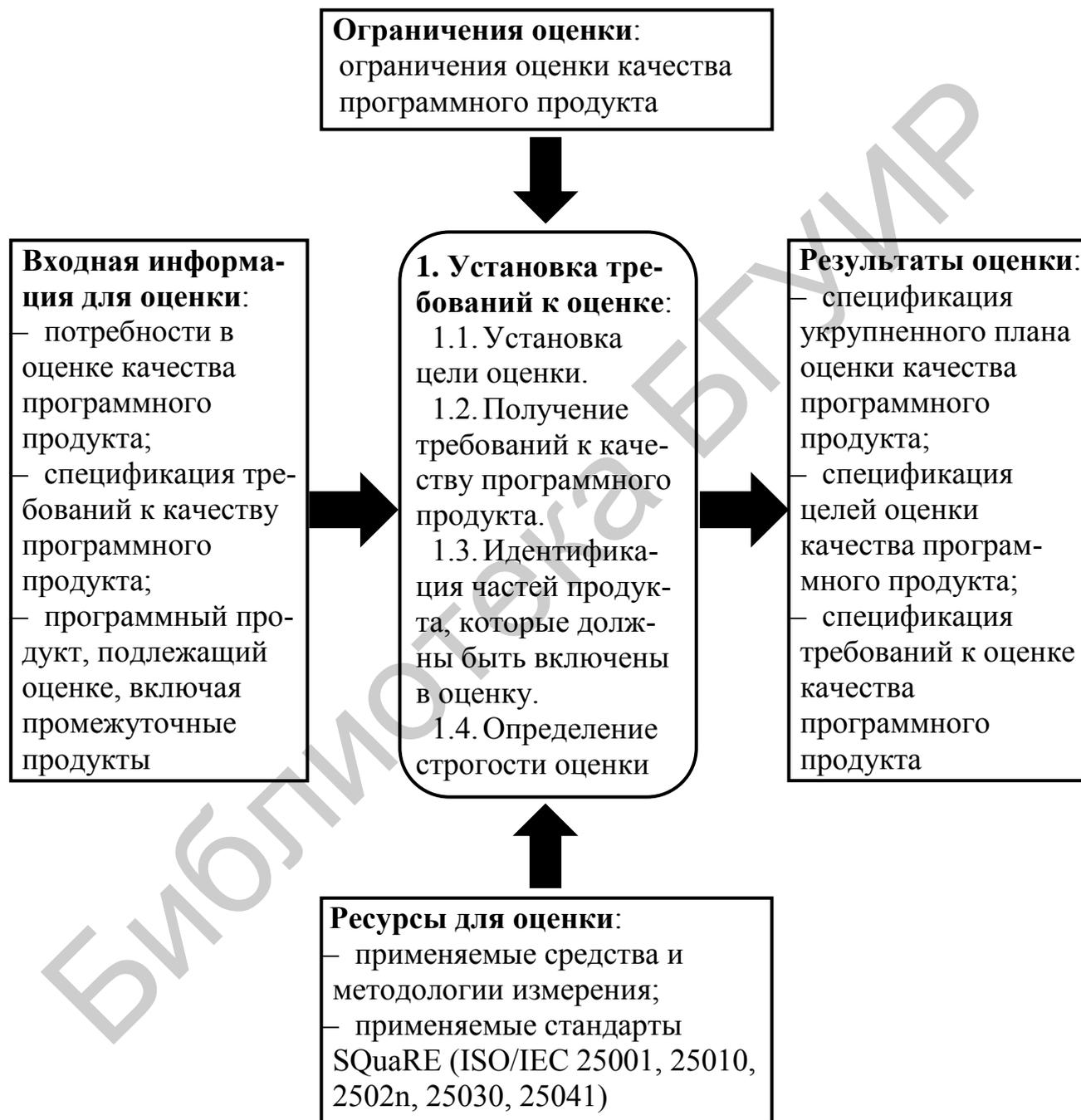


Рис. 12.18. Функциональная диаграмма установки требований к оценке по ISO/IEC 25040:2011

Задача 1.2. Получение требований к качеству программного продукта

Должны быть определены все заинтересованные в программном продукте стороны (правообладатели).

Выделяются *два вида правообладателей*. Первый вид – такие правообладатели программного продукта, как разработчики, приобретатели, независимые оценщики, пользователи, персонал эксплуатации, получатели результатов работы программного средства, персонал сопровождения, поставщики. Второй вид – сторона, которой необходима информация о качестве программного средства, которая запрашивает оценку, спонсирует оценку и нуждается в отчете об оценке.

Требования правообладателей к качеству программного продукта должны быть определены и учтены в процессе оценки.

Если спецификация требований к качеству программного продукта уже существует, она может быть использована, пересмотрена и переработана с учетом цели оценки.

Задача 1.3. Идентификация частей продукта, которые должны быть включены в оценку

Выбор типа промежуточного или конечного продукта, который будет оцениваться, зависит от стадии жизненного цикла и цели оценки.

Если оценивается промежуточный программный продукт, то это может быть, например, спецификация требований, архитектура продукта, технический проект, коды, документация тестирования и т. п.

Если целью оценки является, например, выбор продукта среди альтернативных продуктов, то продуктами для оценки являются обычно конечные продукты или их компоненты.

Задача 1.4. Определение строгости оценки

Определение строгости оценки позволяет обеспечить уверенность в качестве программного продукта в соответствии с назначением его использования и целью оценки.

Строгость оценки должна быть связана с набором характеристик и подхарактеристик. Она определяет ожидаемые уровни оценки данных характеристик и подхарактеристик. От уровней оценки зависят применяемые методы оценки и достигаемые результаты оценки.

Например, для оценки характеристики *Функциональная пригодность* в соответствии с требованиями различных уровней оценки должны применяться следующие методы оценки (от менее строгого уровня к более строгому):

- функциональное тестирование или тестирование методом «черного ящика»;
- проверка документации разработки, руководствуясь контрольными перечнями;
- юнит-тестирование по критериям тестового покрытия.

12.6.3.2. Определение оценки

Вторая работа *Определение оценки* процесса оценки качества программного продукта состоит из трех задач (см. рис. 12.17). Входная информация, выходные результаты, ограничения и ресурсы данной работы представлены на функциональной диаграмме, приведенной на рис. 12.19.



Рис. 12.19. Функциональная диаграмма определения оценки по ISO/IEC 25040:2011

Задача 2.1. Выбор мер качества (модулей оценки)

Оценщик должен выбрать меры качества (модули оценки) таким образом, чтобы покрыть все требования к оценке качества программного средства.

Требования к оценке качества программного продукта должны быть соотнесены с компонентами программного продукта таким образом, чтобы была возможность определить каждую соответствующую меру качества из используемых для оценки качества программного продукта.

Спецификация выбранных мер качества, сформированная на выходе данной работы, должна включать:

- назначение оценки применительно к компонентам продукта, идентифицированным в описании продукта;
- перекрестные ссылки между информацией, необходимой для выполнения оценки, компонентами продукта и другими соответствующими документами, перечисленными в описании продукта;
- спецификацию измерений и верификаций, которые должны быть выполнены, со ссылками на компоненты продукта, к которым они должны быть применены;
- схему связи между спецификацией измерений и верификаций и требованиями к оценке, совместно со ссылками на стандарты или обоснования для каждого перечисленного измерения или верификации.

Задача 2.2. Определение критериев принятия решения для мер качества

Для каждой из выбранных мер должны быть определены критерии принятия решения.

В качестве критериев принятия решения могут использоваться, например, численные уровни ранжирования (см. рис. 10.4), пороговые значения, эталонные значения, статистические контрольные ограничения, исторические данные, требования заказчика или другие критерии.

Критерии принятия решения используются в дальнейшем (после оценки качества программного продукта) для определения потребности в действиях, дополнительных исследованиях или для описания уровня доверия к полученному результату. Например, если результат оценки некоторого дефекта превысит приемлемый порог, следует выполнить дополнительные действия по обнаружению и устранению данного дефекта (информация по действиям, которые необходимо предпринять при неудовлетворительных значениях измеренных мер качества определяется в соответствующей документации).

Задача 2.3. Установка критериев принятия решения для оценки

Оценщик должен подготовить процедуру для дальнейшего обобщения оценки, базирующуюся на отдельных критериях для различных характеристик качества. Каждая из характеристик качества может быть выражена в терминах собственных подхарактеристик и мер качества или как средневзвешенная комбинация подхарактеристик и мер качества.

Для оценки интегрального качества программного продукта должны быть некоторым образом обобщены результаты оценки его различных характеристик.

Критерии принятия решения для оценки могут быть использованы в дальнейшем для принятия управленческих решений. Например, интегральное качество продукта может быть соотнесено с другими аспектами, такими как продолжительность разработки или стоимость приобретения программного продукта.

12.6.3.3. Проектирование оценки

Третья работа *Проектирование оценки* процесса оценки качества программного продукта состоит из одной задачи (см. рис. 12.17). Входная информация, выходные результаты, ограничения и ресурсы данной работы представлены на функциональной диаграмме, приведенной на рис. 12.20.

Задача 3.1. Разработка плана действий по оценке

Идентифицированные в предыдущих работах действия по оценке качества программного продукта должны быть включены в план оценки. При этом должна быть учтена доступность ресурсов, таких как персонал, программные инструментальные средства и компьютеры.

План оценки должен включать следующее:

- цель оценки качества программного продукта;
- организации, вовлеченные в оценку, такие как независимая организация по оценке, разработчики программного продукта и организации-приобретатели;
- бюджет оценки;
- информационные продукты, ожидаемые от оценки;
- график основных этапов оценки;
- обязанности сторон, вовлеченных в оценку;
- среда оценки;
- методы и инструменты оценки;
- критерии принятия решения для мер качества программного продукта;
- критерии принятия решения для оценки качества программного продукта;
- принятые стандарты;
- мероприятия по оценке.

12.6.3.4. Выполнение оценки

Четвертая работа *Выполнение оценки* процесса оценки качества программного продукта состоит из трех задач (см. рис. 12.17). Входная информация, выходные результаты, ограничения и ресурсы данной работы представлены на функциональной диаграмме, приведенной на рис. 12.21.



Рис. 12.20. Функциональная диаграмма проектирования оценки по ISO/IEC 25040:2011

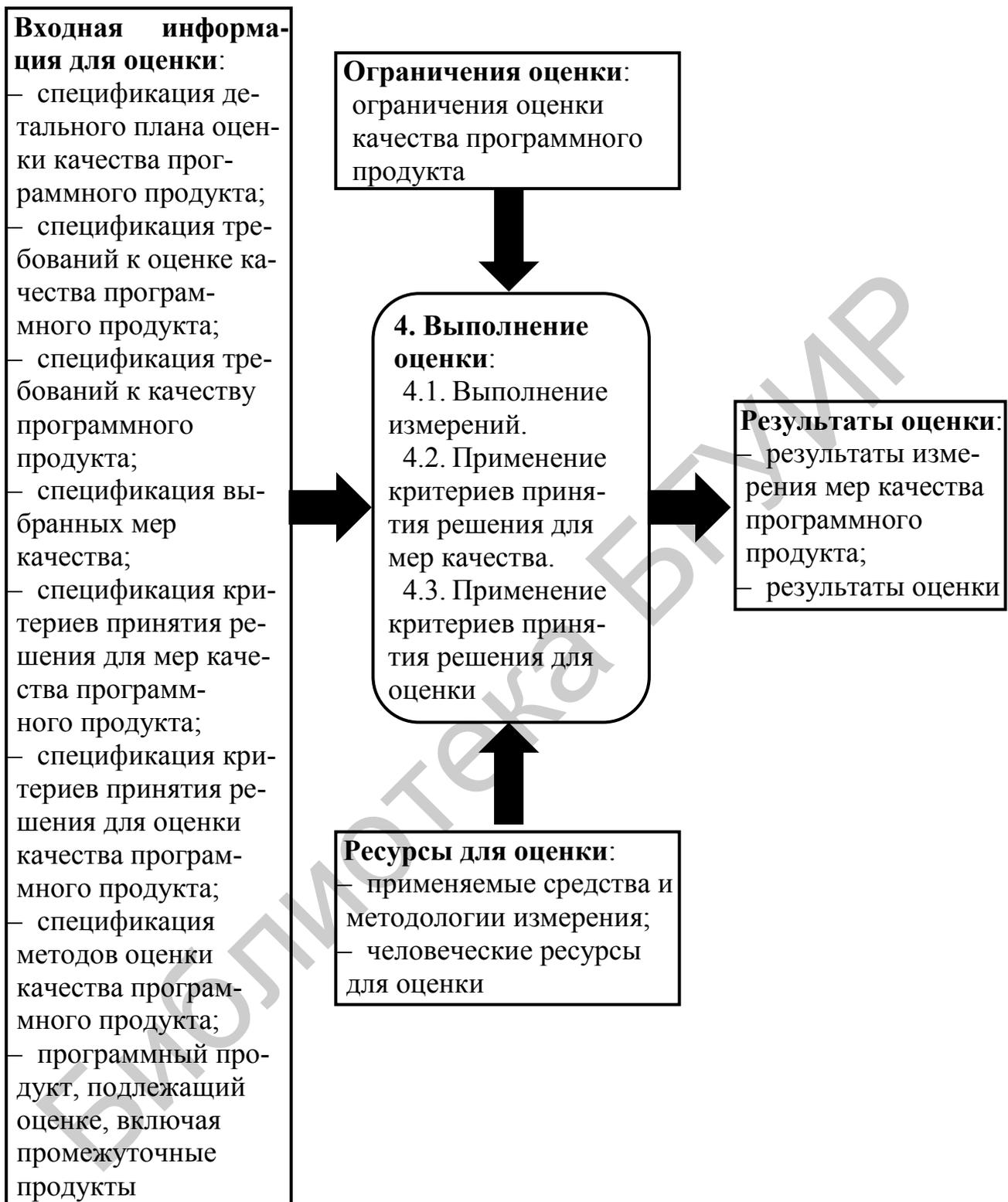


Рис. 12.21. Функциональная диаграмма выполнения оценки по ISO/IEC 25040:2011

Задача 4.1. Выполнение измерений

В соответствии с планом оценки выбранные меры качества программного продукта должны быть применены к программному продукту и его компонентам. Результаты измерений мер должны быть представлены на шкалах измерений.

Задача 4.2. Применение критериев принятия решения для мер качества

Для измеренных значений мер должны быть применены критерии принятия решения для мер качества программного продукта.

Задача 4.3. Применение критериев принятия решения для оценки

Должна быть выполнена интегральная оценка качества программного продукта. Для этого совокупность критериев принятия решения для мер качества должна быть приведена к уровням подхарактеристик и характеристик. В итоге результаты оценки представляются как степень, в которой программный продукт соответствует требованиям качества.

Результаты оценки должны:

- обеспечить соответствующую степень уверенности в том, что программный продукт способен удовлетворять требованиям оценки;
- идентифицировать любые конкретные недостатки относительно требований оценки и любые дополнительные оценки, необходимые для определения влияния этих недостатков;
- идентифицировать любые конкретные ограничения или условия, налагаемые на использование программного продукта;
- идентифицировать любые слабые места или упущения в самой оценке и определить необходимость любой дополнительной оценки;
- идентифицировать любые варианты использования программного продукта, не покрытые оценкой.

12.6.3.5. Заключение об оценке

Пятая работа *Заключение об оценке* процесса оценки качества программного продукта состоит из четырех задач (см. рис. 12.17). Входная информация, выходные результаты, ограничения и ресурсы данной работы представлены на функциональной диаграмме, приведенной на рис. 12.22.

Задача 5.1. Анализ результатов оценки

Оценщик и запросившая оценку сторона должны провести совместный анализ результатов оценки.

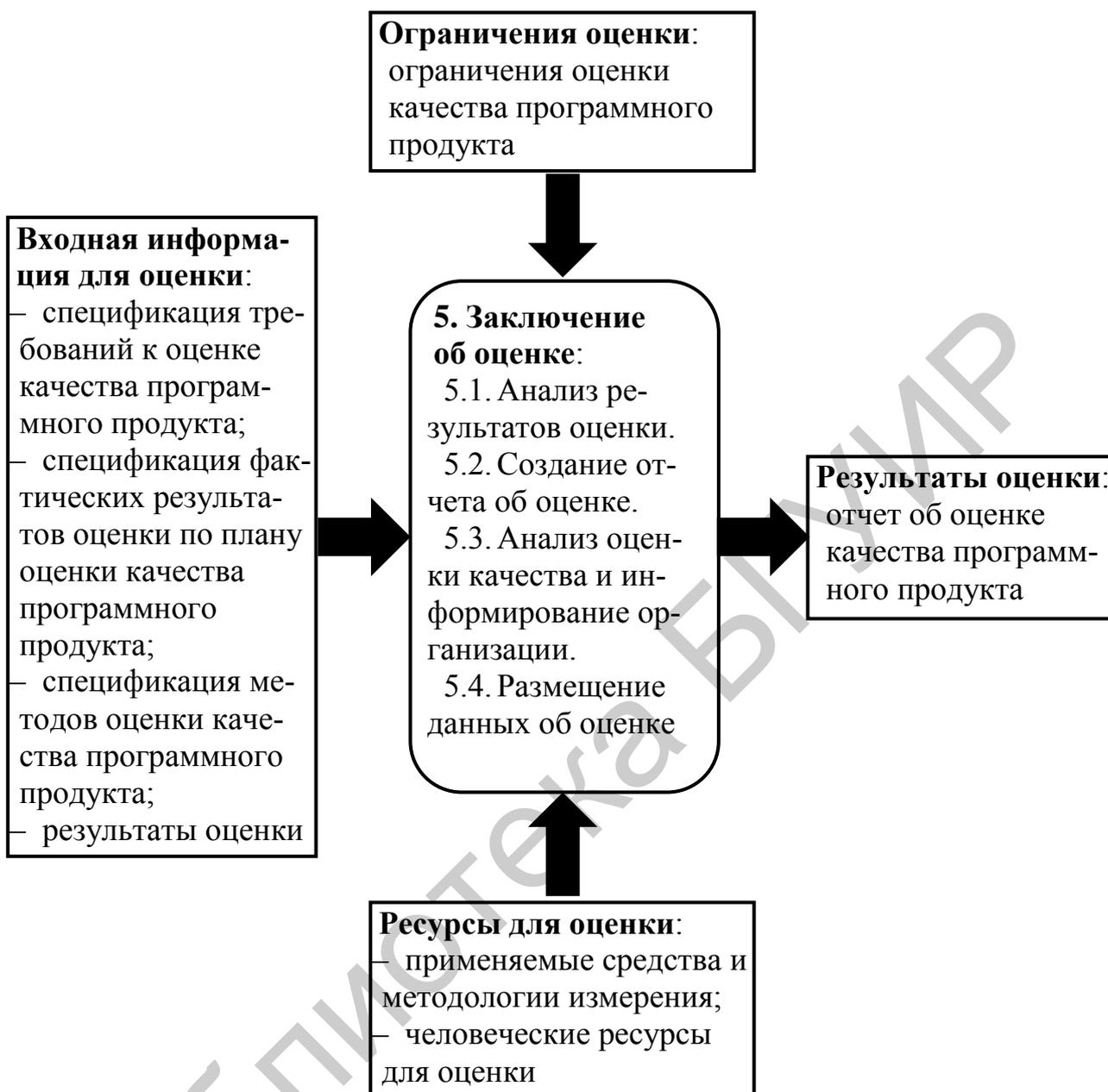


Рис. 12.22. Функциональная диаграмма заключения об оценке по ISO/IEC 25040:2011

Задача 5.2. Создание отчета об оценке

В зависимости от назначения отчет об оценке должен включать следующие пункты:

- требования к оценке качества программного продукта;
- требования к качеству программного продукта;
- план оценки качества программного продукта;
- результаты выполненных измерений и анализов;
- промежуточные результаты или решения по их интерпретации, если это определено планом оценки;

- любые ограничения, недостатки или исключения в действии оценки, включая их влияние в дальнейшем на использование, конфигурацию, модификацию или общее сопровождение программного продукта;

- оценщики и их квалификация;

- любые расхождения между оцененными версиями продукта и соответствующей входной информацией оценки (документацией, линиями поведения и т. п.);

- способы разрешения или обходные пути в случае возникновения недостатков;

- любая другая информация, необходимая для повторения или воспроизведения оценки;

- результат оценки.

На основе анализа действий по оценке, выполненных в задаче 5.1, отчет должен идентифицировать:

- по каждому недостатку соответствующий анализ и как недостаток был разрешен;

- любые дополнительные оценки, выполненные для разрешения всех найденных недостатков;

- случаи, когда необходимо ограничивать или контролировать использование программного продукта;

- любые исключения из области произведенной оценки и/или ограничения на результаты каждой оценки (например, «данная оценка не включает детализированного анализа функциональности продукта»);

- интегрированные результаты всех действий по оценке, позволяющие сделать общее заключение по оценке программного продукта.

Задача 5.3. Анализ оценки качества и информирование организации

Оценщик должен проанализировать результаты оценки, а также обоснованность процесса оценки, используемых индикаторов и мер. Результаты анализа должны быть использованы для усовершенствования процесса оценки и методов оценки (модулей оценки).

Задача 5.4. Размещение данных об оценке

После завершения оценки данные по оценке должны быть размещены в соответствии с требованиями стороны, запросившей оценку. В зависимости от типа данных об оценке они могут быть переданы запрашивающей стороне, заархивированы на определенный период времени или уничтожены.

12.7. Группа расширения ISO/IEC 2505n – 2509n

В данную группу стандартов в настоящее время входят следующие стандарты:

- *ISO/IEC 25051:2014 – Программная инженерия – Требования к качеству и оценка систем и программных средств (SQuaRE) – Требования к качеству готовых к использованию программных продуктов (RUSP) и инструкции по тестированию [52];*

- *ISO/IEC TR 25060:2010 – Системная и программная инженерия – Требования к качеству и оценка систем и программных продуктов (SQuaRE) – Общий промышленный формат (CIF) для практичности. Общая информация, связанная с практичностью [53];*

- *ISO/IEC 25062:2006 – Программная инженерия – Требования к качеству и оценка программных продуктов (SQuaRE) – Общий промышленный формат (CIF) для отчетов по тестированию практичности [54];*

- *ISO/IEC 25063:2014 – Системная и программная инженерия – Требования к качеству и оценка систем и программных продуктов (SQuaRE) – Общий промышленный формат (CIF) для практичности. Описание контекста использования [55];*

- *ISO/IEC 25064:2013 – Системная и программная инженерия – Требования к качеству и оценка программных продуктов (SQuaRE) – Общий промышленный формат (CIF) для практичности. Отчет о потребностях пользователя [56];*

- *ISO/IEC 25066 – Системная и программная инженерия – Требования к качеству и оценка систем и программных средств (SQuaRE) – Общий промышленный формат (CIF) для практичности. Отчет об оценке (стандарт находится в разработке – стадия 60.00 на 19.04.2016 г.) [57].*

ЗАКЛЮЧЕНИЕ ПО ЧАСТИ III

Качество – совокупность характеристик продукта, относящаяся к его способности удовлетворять установленные и подразумеваемые потребности.

В настоящее время в области качества программных средств на территории Республики Беларусь действуют следующие основные стандарты: *ГОСТ 28806–90* (переиздан в июне 2011 г.), *ГОСТ 28195–99*, *СТБ ИСО/МЭК 9126–2003*, *СТБ ISO/IEC 25000–2009*, *СТБ ISO/IEC 25001–2009*.

Модель качества программных средств в *ГОСТ 28806–90* и *СТБ ИСО/МЭК 9126–2003* имеет трехуровневую иерархическую структуру (уровни характеристик, подхарактеристик и метрик). Стандарты регламентируют *шесть характеристик качества* программных средств: *Функциональность* (*Функциональные возможности*), *Надежность*, *Удобство использования* (*Практичность*), *Эффективность*, *Сопровождаемость* и *Мобильность*.

Процесс оценки из *СТБ ИСО/МЭК 9126–2003* состоит из *трех стадий*: определение требований к качеству программного продукта, подготовка к оцениванию и процедура оценивания. Каждая стадия состоит из этапов.

ГОСТ 28195–99 регламентирует четырехуровневую иерархическую модель качества программных средств, состоящую из уровней факторов, критериев, метрик и оценочных элементов. На верхнем уровне находится *шесть факторов качества*: *Надежность*, *Сопровождаемость*, *Удобство использования*, *Эффективность*, *Универсальность*, *Функциональность*.

В *ГОСТ 28195–99* определен метод оценки интегрального качества программных средств, базирующийся на вычислении средневзвешенных сумм значений показателей качества каждого уровня, относящихся к показателю качества вышестоящего уровня.

С 1991 по 2001 г. основой регламентирования качества программных средств и систем за рубежом являлся стандарт *ISO/IEC 9126:1991*. В 2001–2004 гг. данный стандарт был заменен на две взаимосвязанные серии стандартов: *ISO/IEC 9126–1–4:2001–2004* и *ISO/IEC 14598–1–6:1998–2001*.

В стандарте *ISO/IEC 9126–1:2001* регламентированы *две части модели качества систем и программных средств*: модель внешнего и внутреннего качества и модель качества в использовании. *Модель внешнего и внутреннего качества* имеет трехуровневую иерархическую структуру, состоящую из уровней характеристик, подхарактеристик и метрик. Данная модель состоит из *шести характеристик качества*: *Функциональность*, *Надежность*, *Практичность*, *Эффективность*, *Сопровождаемость*, *Мобильность*. *Модель качества в использовании* является двухуровневой. На ее верхнем уровне находятся *четыре характеристики*: *Результативность*, *Продуктивность*, *Безопасность*, *Удовлетворенность*. Характеристики определяются непосредственно метриками. Метрики подразделяются на внутренние, внешние и метрики качества в использовании.

В серии стандартов *ISO/IEC 14598* определены процессы оценки качества программного продукта. Регламентированный в данной серии метод оценки качества является развитием метода оценки качества программных средств из *ISO/IEC 9126:1991 (СТБ ИСО/МЭК 9126–2003)*.

В настоящее время разрабатывается серия стандартов *SQuaRE*, заменяющая собой серии стандартов *ISO/IEC 9126* и *ISO/IEC 14598*. Серия *SQuaRE* разделена на следующие группы: *ISO/IEC 2500n* – группа управления качеством; *ISO/IEC 2501n* – группа модели качества; *ISO/IEC 2502n* – группа измерения качества; *ISO/IEC 2503n* – группа требований к качеству; *ISO/IEC 2504n* – группа оценки качества; *ISO/IEC 25050* – *ISO/IEC 25099* – группа расширения.

В *ISO/IEC 25000:2014 (СТБ ISO/IEC 25000–2009)* регламентированы общие термины и определения, используемые в серии *SQuaRE*, описана организация серии *SQuaRE*, даны общие модели *SQuaRE*.

ISO/IEC 25001:2014 (СТБ ISO/IEC 25001–2009) содержит требования и рекомендации для организаций, занимающихся разработкой спецификации требований к качеству программных средств и систем и оценкой их качества.

В *ISO/IEC 25010:2011* определены две модели качества – модель качества программных средств и систем и модель качества данных. Модель качества программных средств и систем состоит из двух частей – модель качества продукта и модель качества в использовании. Каждая из моделей качества имеет трехуровневую структуру (характеристика – подхарактеристика – мера).

Модель качества в использовании состоит из пяти характеристик, связанных с результатами взаимодействия с системой: *Результативности, Эффективности, Удовлетворенности, Свободы от рисков, Покрытия контекста*. *Модель качества продукта* подразделяет свойства качества системы или программного продукта на восемь характеристик: *Функциональная пригодность, Эффективность функционирования, Совместимость, Практичность, Надежность, Защищенность, Сопровождаемость, Мобильность*.

Стандарт *ISO/IEC 25012:2008* определяет двухуровневую иерархическую модель качества данных. На ее верхнем уровне находится 15 характеристик качества данных. Выделяется: 5 характеристик качества данных с собственной точки зрения: *Правильность, Полнота, Согласованность, Правдоподобие, Актуальность*; 7 характеристик качества с собственной и системно-зависимой точек зрения: *Доступность, Соответствие, Конфиденциальность, Эффективность, Точность, Трассируемость, Понятность*; 3 характеристики качества с системно-зависимой точки зрения: *Готовность, Мобильность, Восстанавливаемость*.

ISO/IEC 25040:2011 определяет процесс оценки качества программного продукта, состоящий из пяти работ: установка требований к оценке, определение оценки, проектирование оценки, выполнение оценки, заключение об оценке. Каждая из данных работ в свою очередь делится на задачи.

ВОПРОСЫ ДЛЯ САМОПРОВЕРКИ ПО ЧАСТИ III

Раздел 9

1. Назовите обобщенные показатели программных средств, существенные при их разработке и оценке качества.
2. Дайте определение качества программного продукта.
3. Дайте определение квалиметрии. В каком стандарте определены основные понятия в области квалиметрии?
4. В чем различия внутреннего, внешнего качества и качества в использовании?
5. Что такое контекст использования?
6. Дайте понятие метрики, назовите и опишите виды метрик.
7. Дайте определение оценки и модели качества.
8. Определите понятие ранжирования и уровня ранжирования.
9. Что такое подразумеваемые потребности?
10. Что представляет собой промежуточный программный продукт?
11. Дайте определение уровня качества функционирования.
12. Дайте определение шкалы, перечислите и опишите типы шкал.

Раздел 10

1. Назовите стандарты в области оценки качества, действующие в Республике Беларусь.
2. Классифицируйте и опишите методы определения показателей качества ПС, определенные в *ГОСТ 28195–99*.
3. Опишите различия иерархических моделей качества программных средств, определенных в *ГОСТ 28806–90* и *СТБ ИСО/МЭК 9126–2003*.
4. Опишите различия иерархических моделей качества программных средств, определенных в *ГОСТ 28195–99* и *СТБ ИСО/МЭК 9126–2003*.
5. Перечислите характеристики и подхарактеристики модели качества программных средств, определенной в *СТБ ИСО/МЭК 9126–2003*.
6. Опишите метод оценки качества программных средств, регламентированный в *СТБ ИСО/МЭК 9126–2003*.
7. Перечислите факторы и критерии качества ПС по *ГОСТ 28195–99*.
8. Опишите модель *Надежности*, регламентированную в *ГОСТ 28195–99*.
9. Опишите метод оценки качества ПС, рекомендованный в *ГОСТ 28195–99*.
10. Приведите примеры диапазонов ранжирования измеренных значений.

Раздел 11

1. Опишите содержание каждой из частей серии стандартов *ISO /IEC 9126–1–4:2001–2004*.
2. Опишите содержание каждой части серии стандартов *ISO/IEC 14598–1–6:1998–2001*.
3. Опишите и представьте графически связь качества ПС с его ЖЦ. Назовите виды качества программных средств для различных стадий ЖЦ.
4. Назовите части модели качества ПС, определенные в *ISO/IEC 9126–1:2001*.
5. Назовите уровни иерархической модели качества по *ISO/IEC 9126–1:2001*.
6. Определите отличия модели внутреннего и внешнего качества программных средств, определенной в *ISO/IEC 9126–1:2001*, от модели качества, определенной в *ISO/IEC 9126:1991*.
7. Опишите модель качества в использовании по *ISO/IEC 9126–1:2001*.
8. Опишите метод оценки качества программных средств, определенный в *ISO/IEC 14598–1:1999*.

Раздел 12

1. Опишите организацию серии стандартов *SQuaRE*.
2. Опишите состав и назначение группы стандартов *ISO/IEC 2500n*.
3. Чем понятие меры качества ПС отличается от понятия метрики качества ПС?
4. Что такое метод измерения качества ПС?
5. Что такое функция измерения качества ПС?
6. Чем функция измерения отличается по назначению от метода измерения?
7. Определите общую эталонную модель серии *SquaRE*, определенную в *ISO/IEC 25000:2014*.
8. Опишите виды качества в жизненном цикле программных средств и систем по *ISO/IEC 25000:2014*.
9. Опишите зависимость и влияние видов качества программных средств и систем в их жизненном цикле по *ISO/IEC 25000:2014*.
10. Перечислите стандарты *SQuaRE*, входящие в группу управления качеством, опишите их назначение.
11. Кратко опишите содержание *ISO/IEC 25001:2014*.
12. Какие разделы должен содержать шаблон плана проекта оценки качества по *ISO/IEC 25001:2014*?
13. Перечислите стандарты *SQuaRE*, входящие в группу модели качества, опишите их назначение.
14. Перечислите стандарты *SQuaRE*, входящие в группу измерения качества, опишите их назначение.
15. Опишите иерархическую структуру моделей качества по стандарту *ISO/IEC 25010:2011*.

16. Какие виды моделей качества определены в стандарте *ISO/IEC 25010:2011*?
17. Назовите характеристики и подхарактеристики модели качества в использовании по стандарту *ISO/IEC 25010:2011*.
18. Назовите характеристики и подхарактеристики модели качества продукта по стандарту *ISO/IEC 25010:2011*.
19. Опишите структуру модели качества данных по стандарту *ISO/IEC 25012:2008*.
20. Опишите характеристики качества данных с собственной точки зрения по стандарту *ISO/IEC 25012:2008*.
21. Опишите характеристики качества данных с собственной и системно-зависимой точек зрения по стандарту *ISO/IEC 25012:2008*.
22. Опишите характеристики качества данных с системно-зависимой точки зрения по стандарту *ISO/IEC 25012:2008*.
23. Опишите структуру и взаимосвязь группы стандартов измерения качества.
24. Опишите содержание стандарта *ISO/IEC 25030:2007*.
25. Опишите иерархию требований, основанную на результатах проектирования архитектуры системы по *ISO/IEC 25030:2007*.
26. Опишите связь видов качества программных средств с моделью системы по *ISO/IEC 25030:2007*.
27. Определите классификацию требований к системе по *ISO/IEC 25030:2007*.
28. Какие стандарты входят в состав группы оценки качества?
29. Нарисуйте обобщенную эталонную модель процесса оценки качества программного продукта, определенную в стандарте *ISO/IEC 25040:2011*.
30. Опишите работы и задачи процесса оценки качества программного продукта по стандарту *ISO/IEC 25040:2011*.
31. Опишите входные и выходные результаты каждой работы процесса оценки качества программного продукта по стандарту *ISO/IEC 25040:2011*.
32. Перечислите состав стандартов группы расширения *ISO/IEC 2505n – ISO/IEC 2509n*.

ЧАСТЬ IV

МЕТРОЛОГИЯ ПРОГРАММНЫХ СРЕДСТВ И СИСТЕМ

Библиотека
ЮНИВЕРСИТЕТА

13. ОБЩИЕ СВЕДЕНИЯ О МЕТРОЛОГИИ

13.1. Основные понятия и определения в области метрологии

Основные понятия в области метрологии и квалиметрии определены в следующих документах:

- руководство *ISO/IEC GUIDE 99:2007(E/R)*. *Международный словарь по метрологии. Основные и общие понятия и соответствующие термины (VIM)* [63];

- рекомендации по межгосударственной стандартизации *PMГ 29–2013*. *Государственная система обеспечения единства измерений. Метрология. Основные термины и определения* [12];

- стандарт *ГОСТ 15467–79*. *Управление качеством продукции. Основные понятия. Термины и определения* [6].

Руководство *ISO/IEC GUIDE 99:2007(E/R)* является двуязычным (опубликовано на английском и русском языках, что отмечено аббревиатурой *E/R* в его обозначении).

Рекомендации *PMГ 29–2013* приняты Межгосударственным советом по стандартизации, метрологии и сертификации и имеют ранг межгосударственного нормативного документа стран СНГ. Одной из задач *PMГ 29–2013* является гармонизация с международной терминологией в области метрологии, что направлено на обеспечение единого подхода к оценке качества результатов измерений.

В *ISO/IEC GUIDE 99:2007(E/R)* и *PMГ 29–2013* приняты следующие **основные термины и определения**.

Величина (quantity) – свойство материального объекта или явления, общее в качественном отношении для многих объектов или явлений, но в количественном отношении индивидуальное для каждого из них.

Воспроизводимость измерений (measurement reproducibility, reproducibility) – прецизионность измерений в условиях воспроизводимости измерений.

Единица измерения величины (measurement unit, unit of measurement, unit) величина фиксированного размера, которой присвоено числовое значение, равное 1, определяемая и принимаемая по соглашению для количественного выражения однородных с ней величин.

Метод измерений (measurement method, method of measurement) – прием или совокупность приемов сравнения измеряемой величины с ее единицей или соотнесения со шкалой в соответствии с реализованным принципом измерений.

Методика (выполнения) измерений (measurement procedure) – установленная логическая последовательность операций и правил при измерении, выполнение которых обеспечивает получение результатов измерений в соответствии с принятым методом измерений. Обычно методика измерений регламентируется каким-либо нормативным документом.

Метрология (metrology) – наука об измерениях, методах и средствах обеспечения их единства и способах достижения требуемой точности.

Модель измерений, уравнение измерений (measurement model, model of measurement, model) – уравнение связи между величинами в конкретной измерительной задаче.

Опорное значение величины (reference quantity value, reference value) – значение величины, которое используют в качестве основы для сопоставления со значениями величин того же рода.

Повторяемость измерений (measurement repeatability, repeatability) – прецизионность измерений в условиях повторяемости измерений.

Правильность измерений (measurement trueness, trueness of measurement, trueness) – близость среднего арифметического бесконечно большого числа повторно измеренных значений величины к опорному значению величины.

Прецизионность измерений (measurement precision, precision) – близость между показаниями или измеренными значениями величины, полученными при повторных измерениях для одного и того же или аналогичных объектов при заданных условиях. Понятие прецизионность измерений используется для определения понятий повторяемости измерений, промежуточной прецизионности измерений и воспроизводимости измерений.

Промежуточная прецизионность измерений (intermediate measurement precision, intermediate precision) – прецизионность измерений в фиксированных условиях промежуточной прецизионности измерений.

Точность измерений, точность результата измерения (measurement accuracy, accuracy of measurement, accuracy) – близость измеренного значения к истинному значению измеряемой величины. Понятие точности измерений описывает качество измерений в целом, объединяя понятия правильности и прецизионности измерений.

Условия воспроизводимости измерений (reproducibility condition of measurement, reproducibility condition) – один из наборов условий измерений, включающий разные местоположения, разные средства измерений, участие разных операторов и выполнение повторных измерений на одном и том же или аналогичных объектах.

Условия повторяемости измерений (repeatability condition of measurement, repeatability condition) – один из наборов условий измерений, включающий применение одной и той же методики измерений, того же средства измерений, участие тех же операторов, те же рабочие условия, то же местоположение и выполнение повторных измерений на одном и том же или подобных объектах в течение короткого промежутка времени.

Условия промежуточной прецизионности измерений (intermediate precision condition of measurement, intermediate precision condition) – один из наборов условий измерений, включающий применение одной и той же методики измерений, то же местоположение и выполнение повторных измерений на одном и том же или аналогичных объектах в течение длительного периода времени, а также может включать другие условия, которые могут изменяться.

Погрешность результата измерения (measurement error, error of measurement, error) – разность между измеренным значением величины и опорным значением величины.

Функция измерений (measurement function) – зависимость величин модели измерений, используемая для получения измеренного значения выходной величины по известным значениям входных величин.

Шкала значений величины, шкала измерений (quantity-value scale, measurement scale) – упорядоченная совокупность значений величины, служащая исходной основой для измерений данной величины.

В ГОСТ 15467–79 приняты следующие **основные термины и определения**.

Базовое значение показателя качества продукции – значение показателя качества продукции, принятое за основу при сравнительной оценке ее качества.

Единичный показатель качества продукции – показатель качества продукции, характеризующий одно из ее свойств.

Интегральный показатель качества продукции – показатель качества продукции, являющийся отношением суммарного полезного эффекта от эксплуатации или потребления продукции к суммарным затратам на ее создание и эксплуатацию или потребление.

Комплексный показатель качества продукции – показатель качества продукции, характеризующий несколько ее свойств. В современных стандартах в области оценки качества программных средств комплексные показатели принято называть подхарактеристиками качества.

Определяющий показатель качества продукции – показатель качества продукции, по которому принимают решение оценивать ее качество.

Относительное значение показателя качества продукции – отношение значения показателя качества оцениваемой продукции к базовому значению этого показателя.

Предельное значение показателя качества продукции – наибольшее или наименьшее регламентированное значение показателя качества продукции

Регламентированное значение показателя качества продукции – значение показателя качества продукции, установленное нормативной документацией.

13.2. Типы шкал измерения атрибутов программных средств и систем

В серии стандартов *ISO/IEC 9126-2-4:2003-2004* дано следующее определение шкалы: *шкала* – набор значений с определенными свойствами [60 – 62].

В стандарте *ISO/IEC 15939:2007* шкала определяется следующим образом: *шкала* – упорядоченный непрерывный или дискретный набор значений или набор категорий, на которых отображается атрибут [35]. Данное определение шкалы используется в стандартах серии *SQuaRE*.

Типы шкал, используемых при измерении программных средств и систем, определены в стандартах *ISO/IEC 9126-2-4:2003-2004*, *ISO/IEC 25000:2014*, *ISO/IEC 15939:2007* [38, 35, 60-62]. В стандарте *ISO/IEC 25021:2012* приведены примеры и пояснения к использованию типов шкал [44].

Тип шкалы зависит от природы связи между значениями на шкале. При оценке качества программных средств и систем обычно используются четыре типа шкал: номинальная, порядковая (упорядоченная), интервальная и шкала отношений.

Номинальный (nominal) тип шкалы соответствует набору категорий. Целью мер (метрик в *ISO/IEC 9126-2-4:2003-2004*) в номинальном типе шкалы является классификация измеренных атрибутов. Даже если для классификации используются числа, упорядочение не предполагается. Числа, связанные с мерами в номинальной шкале измерений, определяют только категорию измеренного атрибута. Поэтому упорядоченность, минимум, максимум, медиана, арифметическое значение, процентное отношение и т. д., полученные на основе этих чисел, не имеют никакого практического смысла. Они соответствуют недопустимым математическим операциям.

Пример 1. Идентификация футболистов номерами.

Пример 2. Классификация дефектов программного средства или системы по их типам без упорядочения типов между собой.

Пример 3. Классификация структуры программы: 1 – линейная; 2 – разветвляющаяся; 3 – циклическая.

Пример 4. Оценивание реализованности некоторой функции по категориям «да», «нет».

Порядковый (ordinal) тип шкалы соответствует упорядоченному набору значений шкалы. Целью мер (метрик в *ISO/IEC 9126-2-4:2003-2004*) в упорядоченном типе шкалы является назначение порядка (ранга) измеренным атрибутам. Порядковый тип шкалы полезен для прибавления к номинальной шкале информации об упорядочении классов или категорий. Обычно порядковая шкала содержит небольшое количество значений.

Минимум, максимум и медиана для мер в порядковой шкале имеют практический смысл. Арифметическое значение, процентное отношение и т. д. практического смысла не имеют.

Порядковый тип шкалы называется также ранговым типом.

Пример 1. Упорядочение отказа программного продукта или системы по серьезности (1 – несущественный, 2 – пограничный, 3 – критический, 4 – катастрофический).

Пример 2. Объем программы: 1 – до 500 Кб; 2 – 501–1000 Кб; 3 – более 1000 Кб.

Пример 3. Оценивание некоторого свойства программного продукта или системы по упорядоченным категориям «неудовлетворительно», «удовлетворительно», «хорошо», «отлично».

Интервальный (interval) тип шкалы соответствует упорядоченному набору равноудаленных значений шкалы. Данный тип шкалы обычно содержит достаточно большое число количественных значений. Значение нуля, соответствующее полному отсутствию атрибута, по физической природе измеряемого атрибута невозможно.

Целью мер (метрик в *ISO/IEC 9126-2-4:2003-2004*) в интервальном типе шкалы является измерение разницы между мерами. Среднее значение, минимум, максимум и т. п. для мер в интервальной шкале имеют практический смысл. Если вычисляется отношение мер, измеренных в данном типе шкалы, то это не имеет практического смысла.

Пример 1. Значение температуры в градусах Цельсия может быть прибавлено к другому значению температуры или вычтено из него. Но данные значения не могут быть перемножены или разделены (для значений 25 и 10 °C значима только разница между ними в 15 °C).

Пример 2. Цикломатическая сложность программного средства имеет минимальное значение, равное единице. Каждое последующее значение на шкале соответствует одному дополнительному пути в графе программного средства.

Тип шкалы отношений (ratio). Тип шкалы отношений подобен интервальной шкале. Данный тип шкалы обычно также содержит достаточно большое число равноудаленных количественных значений. Отличие заключается в том, что шкала отношений содержит значение нуля, представляющее полное отсутствие атрибута. Для мер в данном типе шкалы практическое значение имеют как разница между двумя мерами, так и отношение двух мер. Среднее значение, минимум, максимум для мер в данном типе шкалы также имеет смысл.

Пример. Размер буфера; количество обнаруженных ошибок; время, затраченное на изменения.

Номинальный и порядковый типы шкал используются для оценки качественных атрибутов программного средства или системы, которые нельзя измерить количественно, и для ранжирования значений атрибутов. Обычно

данные типы шкал применяются при использовании субъективных (экспертных) методов оценки атрибутов программного средства или системы.

Интервальный тип шкалы и тип шкалы отношений используются для оценки количественных атрибутов программного средства или системы. Эти типы шкал применяются при объективных методах оценки атрибутов программного средства или системы, например, использующих инструментальные средства и/или некоторые математические формулы.

Библиотека БГУИР

14. МЕТРИКИ КАЧЕСТВА ПРОГРАММНЫХ СРЕДСТВ ПО СТАНДАРТАМ ISO/IEC TR 9126-2-4:2003-2004

14.1. Свойства и критерии обоснованности метрик

В серии стандартов *ISO/IEC 9126-1-4:2001-2004* дано следующее определение метрики: *метрика (metric)* – это определенный метод и шкала измерения. Метрики классифицируются на внутренние, внешние или метрики качества в использовании; прямые или косвенные (см. разд. 9, подразд. 11.1).

В модели внешнего и внутреннего качества, регламентированной стандартом *ISO/IEC 9126-1:2001*, метрики находятся на третьем уровне иерархии и определяют значения подхарактеристик качества программных средств. В модели качества в использовании метрики находятся на втором уровне иерархии и непосредственно определяют значения характеристик качества программных средств и систем (см. подразд. 11.3, 11.4). Применение конкретного вида метрик определяется стадией жизненного цикла программного средства и системы (см. подразд. 11.2).

В настоящее время стандарт *ISO/IEC 9126-1:2001* заменен стандартом *ISO/IEC 25010:2011* (см. пп. 12.3.1 – 12.3.3).

Вторая, третья и четвертая части *ISO/IEC TR 9126-2-4:2003-2004* являются действующими [60 – 62]. В настоящее время на их основе разрабатываются стандарты *ISO/IEC 25022*, *ISO/IEC 25023* серии *SQuaRE* (см. подразд. 12.4).

Стандарты *ISO/IEC TR 9126-2-4:2003-2004* посвящены детальному рассмотрению соответственно внешних и внутренних метрик качества программных средств и метрик качества в использовании.

В *приложении А ISO/IEC TR 9126-2-4:2003-2004* определены следующие **желательные свойства метрик**:

1) *надежность (reliability)* – надежность связана со случайной ошибкой; метрика свободна от случайной ошибки, если случайные изменения не влияют на результаты метрики;

2) *повторяемость (repeatability)* – повторное использование метрики для того же продукта теми же специалистами по оценке, используя ту же спецификацию оценки (включая ту же окружающую среду), тот же тип пользователей и окружения, должно привести к тем же результатам с соответствующими допусками; соответствующие допуски должны учитывать такие компоненты, как усталость и результат накопленных познаний;

3) *воспроизводимость (reproducibility)* – применение метрики для того же продукта различными специалистами по оценке, используя ту же спецификацию оценки (включая ту же окружающую среду), тот же тип пользователей и окружения, должно привести к тем же результатам с соответствующими допусками;

4) *доступность (availability)* метрика должна четко указывать условия (например, наличие определенных атрибутов), которые ограничивают ее употребление;

5) *показательность (indicativeness)* – это способность метрики идентифицировать части или элементы программы, которые должны быть улучшены, на основании сравнения измеренных и ожидаемых результатов;

6) *корректность (correctness)* – метрика должна обладать следующими свойствами:

- *объективность (objectivity)* – результаты метрики и ее входные данные должны быть основаны на фактах и не подвластны чувствам или мнениям специалистов по оценке или тестированию (исключая метрики удовлетворенности или привлекательности, с помощью которых измеряются чувства и мнения пользователя);

- *беспристрастность (impartiality)* – измерение не должно быть направлено на получение какого-либо специфического результата;

- *адекватность точности (sufficient precision)* – точность определяется при проектировании метрики и особенно при выборе описаний фактов, используемых как основа для метрики; разработчик метрики должен описать точность и чувствительность метрики;

- *значимость (meaningfulness)* – измерение должно давать значащие результаты, касающиеся поведения программы или характеристик качества.

Метрика должна также быть *эффективной по отношению к стоимости*. Это значит, что более дорогие метрики должны обеспечивать лучшие результаты оценки.

Разработчик метрики должен доказать ее обоснованность. Метрика должна удовлетворять хотя бы одному из следующих **критериев обоснованности метрики**:

1) *корреляция (correlation)* – изменение в значениях характеристик качества (оперативно определенных по результатам измерения основных метрик), обусловленное изменением в значениях метрики, должно определяться линейной зависимостью;

2) *трассировка (tracking)* – если метрика M непосредственно связана с величиной характеристики качества Q (оперативно определенной по результатам измерения основных метрик), то изменение величины Q ($T1$), имеющейся в момент времени $T1$, до величины Q ($T2$), полученной в момент времени $T2$, должно сопровождаться изменением значения метрики от M ($T1$)

до $M(T2)$ в том же направлении (например, если увеличивается Q , то M тоже увеличивается);

3) *непротиворечивость (consistency)* – если значения характеристик качества (оперативно полученные по результатам измерения основных метрик) $Q1, Q2, \dots, Qn$, связанные с продуктами или процессами $1, 2, \dots, n$, определяются соотношением $Q1 > Q2 > \dots > Qn$, то соответствующие значения метрики должны удовлетворять соотношению $M1 > M2 > \dots > Mn$.

4) *предсказуемость (predictability)* – если метрика используется в момент времени $T1$ для прогноза значения (оперативно полученного по результатам измерения основных метрик) характеристики качества Q в момент времени $T2$, то ошибка прогнозирования, определяемая выражением

$$(\text{прогнозное } Q(T2) - \text{фактическое } Q(T2)) / (\text{фактическое } Q(T2)),$$

должна попадать в допустимый диапазон ошибок прогнозирования;

5) *селективность (discriminative)* – метрика должна быть способной различать высокое и низкое качество программного средства.

В стандартах *ISO/IEC TR 9126-2-4:2003-2004* для каждой подхарактеристики внешнего и внутреннего качества и характеристики качества в использовании приведены таблицы, в которых даны примеры метрик качества.

Таблицы имеют следующую структуру:

- 1) название метрики;
- 2) назначение метрики (изложено в виде вопроса, на который отвечает применение метрики);
- 3) метод применения;
- 4) способ измерения, формула, исходные и вычисляемые данные;
- 5) интерпретация измеренного значения (диапазон и предпочтительные значения);
- 6) тип шкалы, используемой при измерении метрики (номинальная, порядковая, интервальная, относительная или абсолютная);
- 7) тип измеренного значения; используются следующие **типы измеренных значений**:
 - *тип размера* (например, функциональный размер, размер исходного текста);
 - *тип времени* (например, затраченное время, необходимое пользователю время);
 - *тип количества* (например, количество изменений, количество отказов);
- 8) источники входных данных для измерения;
- 9) ссылка на *ISO/IEC 12207:1995* (процессы жизненного цикла программных средств, при выполнении которых применима метрика);
- 10) целевая аудитория.

Для обеспечения возможности совместного использования различных метрик (независимо от их физического смысла, единиц измерения и диапазонов значений) при количественной оценке качества программных продуктов метрики в стандартах *ISO/IEC TR 9126–2–4:2003–2004* по возможности представляются в относительных единицах в виде

$$X = A/B \quad (14.1)$$

или

$$X = 1 - A/B, \quad (14.2)$$

где X – значение метрики; A – абсолютное (измеренное) значение некоторого свойства (атрибута) оцениваемого продукта, системы или документации; B – базовое значение соответствующего свойства.

Из двух вышеназванных формул для конкретной метрики выбирается та, которая соответствует *критериям трассировки и непротиворечивости*: с увеличением относительного значения метрики значение подхарактеристики и характеристики качества должно увеличиваться.

Вычисление метрик по формуле (14.1) или (14.2) позволяет привести их относительные значения в диапазон

$$0 \leq X \leq 1, \quad (14.3)$$

что упрощает их совместное использование при интегральной оценке качества программных средств.

В подразд. 14.2 – 14.4 приведены примеры метрик (по одной на каждую подхарактеристику или характеристику качества) из рекомендуемых в стандартах *ISO/IEC TR 9126–2–4:2003–2004* [60 – 62].

Следует отметить, что не все метрики, приведенные в стандартах *ISO/IEC TR 9126–2–4:2003–2004*, удовлетворяют вышеприведенным свойствам, критериям, оцениваются с помощью выражений (14.1), (14.2) или попадают в диапазон (14.3).

14.2. Внутренние метрики качества программных средств

Внутренние метрики функциональности предназначены для предсказания того, удовлетворяет ли разрабатываемый программный продукт требованиям к функциональности и предполагаемым потребностям пользователя.

Внутренние метрики надежности используются во время разработки программного продукта для предсказания того, удовлетворяет ли программный продукт заявленным потребностям в надежности.

Внутренние метрики практичности используются во время разработки программного продукта для предсказания степени, в которой программный продукт может быть понят, изучен, управляем, привлекателен и соответствует договоренностям и руководствам по практичности.

Внутренние метрики эффективности используются во время разработки программного продукта для предсказания эффективности поведения программного продукта во время тестирования или эксплуатации.

Внутренние метрики сопровождаемости используются для предсказания уровня усилий, необходимых для модификации программного продукта.

Внутренние метрики мобильности используются для предсказания воздействия программного продукта на поведение исполнителя или системы при проведении работ по переносу.

В табл. 14.1 приведены примеры внутренних метрик качества ПС, содержащихся в стандарте *ISO/IEC TR 9126-3:2003* [61]. Во втором столбце таблицы по каждой подхарактеристике приведено название одной метрики, уникальная формула или номер формулы (14.1) или (14.2) из подразд. 14.1 для оценки данной метрики. Исходные данные в третьем столбце – это данные, используемые в соответствующей формуле для вычисления значения представленной метрики.

Следует обратить внимание, что приведенные в данной таблице метрики *эффективности* измеряются не в относительных единицах, а в абсолютных. Это затрудняет их использование при интегральной оценке качества программных продуктов. Кроме того, метрика *эффективности* «Использование памяти» не удовлетворяет таким критериям обоснованности метрик, как корреляция, трассировка и непротиворечивость.

14.3. Внешние метрики качества программных средств

Внешние метрики функциональности должны измерять свойства (атрибуты) функционального поведения системы, содержащей ПС.

Внешние метрики надежности должны измерять свойства, связанные с поведением системы, содержащей ПС, во время тестирования, чтобы показать степень надежности ПС в системе в процессе эксплуатации.

Внешние метрики практичности показывают, в какой мере программное средство может быть понято, изучено, управляемо, привлекательно и соответствует договоренностям и руководствам по практичности.

Внутренние метрики качества программных средств

Название подхарактеристики	Название метрики, формула или номер формулы для ее оценки	Исходные данные для вычисления метрики по соответствующей формуле
Функциональность		
Пригодность	Полнота функциональной реализации (14.2)	<i>A</i> – число нереализованных функций, обнаруженных при оценке; <i>B</i> – число функций, описанных в спецификации требований
Правильность	Точность (14.1)	<i>A</i> – количество элементов данных, реализованных с заданными уровнями точности, подтвержденное при оценке; <i>B</i> – количество элементов данных, для которых в спецификации заданы уровни точности
Способность к взаимодействию	Соответствие интерфейсов (протоколов) (14.1)	<i>A</i> – количество интерфейсных протоколов, реализующих заданный в спецификации формат, подтвержденных при проверке; <i>B</i> – количество интерфейсных протоколов, которые должны быть реализованы в соответствии со спецификацией
Защищенность	Предотвращение разрушения данных (14.1)	<i>A</i> – количество реализованных случаев предотвращения разрушения данных из заданных в спецификации, подтвержденное при проверке; <i>B</i> – количество случаев обработки (доступа), которые определены в спецификации как способные разрушить данные
Соответствие функциональности	Соответствие функциональности (14.1)	<i>A</i> – количество корректно реализованных элементов, связанных с соответствием функциональности, подтвержденное при оценке; <i>B</i> – общее количество элементов соответствия

Название подхарактеристики	Название метрики, формула или номер формулы для ее оценки	Исходные данные для вычисления метрики по соответствующей формуле
Надежность		
Завершенность	Полнота тестирования (14.1)	A – количество тестовых комбинаций, спроектированных в плане тестирования и подтвержденных при проверке; B – количество требуемых тестовых комбинаций
Устойчивость к ошибке	Предотвращение некорректных действий (14.1)	A – количество функций, реализованных с предотвращением некорректных действий; B – количество типичных некорректных действий, которое должно быть учтено
Восстанавливаемость	Способность к восстановлению (14.1)	A – количество реализованных требований к восстановлению, подтвержденные при проверке; B – общее количество требований к восстановлению, определенных в спецификации
Соответствие надежности	Соответствие надежности (14.1)	A – количество корректно реализованных элементов, связанных с соответствием надежности, подтвержденное при оценке; B – общее количество элементов соответствия
Практичность		
Понятность	Способность к демонстрации (14.1)	A – количество демонстрируемых функций, подтвержденные при проверке; B – общее количество функций, которые должны обладать способностью к демонстрации
Обучаемость	Полнота документации пользователя и/или возможности электронной справки help (14.1)	A – количество описанных функций; B – общее количество предоставляемых функций
Простота использования	Отменяемость действий пользователя (14.1)	A – количество реализованных функций, которые могут быть отменены пользователем с восстановлением предыдущих данных; B – общее количество функций

Название подхарактеристики	Название метрики, формула или номер формулы для ее оценки	Исходные данные для вычисления метрики по соответствующей формуле
Привлекательность	Настраиваемость вида интерфейса пользователя (14.1)	A – количество типов элементов интерфейса, которые могут быть настроены; B – общее количество типов элементов интерфейса
Соответствие практичности	Соответствие практичности (14.1)	A – количество корректно реализованных элементов, связанных с соответствием практичности, подтвержденное при оценке; B – общее количество элементов соответствия
Эффективность		
Поведение во времени	Пропускная способность $X = A$	A – число задач в единицу времени, подтвержденное при проверке
Использование ресурсов	Использование памяти $X = A$	A – размер памяти в байтах (вычисленный или моделированный)
Соответствие эффективности	Соответствие эффективности (14.1)	A – количество корректно реализованных элементов, связанных с соответствием эффективности, подтвержденное при проверке; B – общее количество элементов соответствия
Сопровождаемость		
Анализируемость	Готовность диагностических функций (14.1)	A – количество реализованных диагностических функций из заданных в спецификации, подтвержденное при проверке; B – требуемое количество диагностических функций
Изменяемость	Регистрируемость изменений (14.1)	A – количество изменений в функциях/модулях, отраженных в комментариях, подтвержденное при проверке; B – общее количество изменений в функциях/модулях относительно оригинального кода
Стабильность	Влияние изменений (14.2)	A – количество обнаруженных вредных влияний после модификаций; B – количество сделанных модификаций

Название подхарактеристики	Название метрики, формула или номер формулы для ее оценки	Исходные данные для вычисления метрики по соответствующей формуле
Тестируемость	Полнота встроенных функций тестирования (14.1)	A – количество реализованных встроенных функций тестирования из заданных в спецификации, подтвержденное при проверке; B – требуемое количество встроенных функций тестирования
Соответствие сопровождаемости	Соответствие сопровождаемости (14.1)	A – количество корректно реализованных элементов, связанных с соответствием сопровождаемости, подтвержденное при оценке; B – общее количество элементов соответствия
Мобильность		
Адаптируемость	Адаптируемость структур данных (14.1)	A – количество структур данных, работоспособность которых не нарушена после адаптации, подтвержденное при проверке; B – общее количество структур данных, которые должны обладать способностью к адаптации
Настраиваемость	Объем работ по установке (14.1)	A – количество автоматических шагов установки (инсталляции), подтвержденное при проверке; B – требуемое количество шагов инсталляции
Совместимость	Доступная совместимость (14.1)	A – количество объектов, с которыми продукт может сосуществовать, из заданных в спецификации; B – количество объектов в окружающей среде, с которыми продукт должен сосуществовать
Взаимозаменяемость	Преемственность данных (14.1)	A – количество элементов данных ПС, которые продолжают использоваться после замещения (из заданных в спецификации), подтвержденное при проверке; B – количество элементов старых данных, которые должны использоваться из старого ПС
Соответствие мобильности	Соответствие мобильности (14.1)	A – количество корректно реализованных элементов, связанных с соответствием мобильности, подтвержденное при проверке; B – общее количество элементов соответствия

Внешние метрики эффективности должны измерять такие атрибуты, как характер изменения затрат времени и использования ресурсов компьютерной системы, включающей ПС, во время тестирования или эксплуатации.

Внешние метрики сопровождаемости измеряют такие атрибуты, как поведение персонала сопровождения, пользователя или системы, включающей ПС, при модификации ПС во время тестирования или сопровождения.

Внешние метрики мобильности измеряют такие атрибуты, как поведение оператора или системы при проведении работ по переносу.

В табл. 14.2 приведены примеры внешних метрик качества программных средств из стандарта *ISO/IEC TR 9126-2:2003* [60]. Во втором столбце таблицы по каждой подхарактеристике приведено название одной метрики, уникальная формула или номер формулы (14.1) или (14.2) из подразд. 14.1 для оценки этой метрики. Исходные данные в третьем столбце – это данные, используемые в соответствующей формуле для вычисления значения представленной метрики.

Не все метрики, приведенные в табл. 14.2, удовлетворяют таким критериям обоснованности метрик, как корреляция, трассировка и непротиворечивость. Это касается, например, метрики *правильности* «Точность», метрики *завершенности* «Плотность ошибок», метрики *поведения во времени* «Время отклика», метрики *стабильности* «Возникновение отказа после изменения», метрики *совместимости* «Доступная совместимость». Кроме того, данные метрики не удовлетворяют диапазону (14.3) (см. подразд. 14.1). Это затрудняет их использование при интегральной оценке качества программных продуктов.

14.4. Метрики качества программных средств в использовании

Метрики результативности оценивают, достигают ли задачи, выполняемые пользователем, заданных целей с точностью и полнотой в заданном контексте использования.

Метрики продуктивности оценивают ресурсы, которые затрачивают пользователи в соответствии с достигнутой результативностью в заданном контексте использования.

Метрики безопасности оценивают уровень риска причинения вреда людям, бизнесу, программному обеспечению, имуществу или окружающей среде в заданном контексте использования.

Метрики удовлетворенности оценивают отношение пользователя к использованию продукта в заданном контексте использования.

Внешние метрики качества программных средств

Название подхарактеристики	Название метрики, формула или номер формулы для ее оценки	Исходные данные для вычисления метрики по соответствующей формуле
Функциональность		
Пригодность	Полнота функциональной реализации (14.2)	A – количество отсутствующих функций, обнаруженных при проверке; B – количество функций, описанных в спецификации требований
Правильность	Точность $X=A/T$	A – количество результатов, подсчитанное пользователями, с уровнем точности, отличающимся от требуемого; T – продолжительность работы
Способность к взаимодействию	Способность к обмену данными (основанная на успешных попытках пользователя) (14.2)	A – количество случаев, в которых пользователю не удалось обменяться данными с другими ПС или системами; B – количество случаев, в которых пользователь пытался обмениваться данными
Защищенность	Предотвращение разрушения данных (14.2)	A – количество произошедших случаев разрушения важных данных; B – количество тестовых случаев, направленных на разрушение данных
Соответствие функциональности	Соответствие функциональности (14.2)	A – количество заданных элементов соответствия функциональности, которые не были выполнены при тестировании; B – общее количество заданных элементов соответствия функциональности
Надежность		
Завершенность	Плотность ошибок $X=A/Z$	A – количество ошибок, обнаруженных в течение определенного испытательного срока; Z – размер продукта
Устойчивость к ошибке	Предотвращение некорректных действий (14.1)	A – количество предотвращенных критических и серьезных отказов; B – количество выполненных при тестировании тестовых случаев, направленных на проверку типовых некорректных действий, которые могут привести к отказу

Название подхарактеристики	Название метрики, формула или номер формулы для ее оценки	Исходные данные для вычисления метрики по соответствующей формуле
Восстанавливаемость	Способность к восстановлению (14.1)	<i>A</i> – количество случаев успешного восстановления; <i>B</i> – количество случаев восстановления, протестированных согласно требованиям
Соответствие надежности	Соответствие надежности (14.2)	<i>A</i> – количество заданных элементов соответствия надежности, не подтвержденных при тестировании; <i>B</i> – общее количество заданных элементов соответствия надежности
Практичность		
Понятность	Полнота описания (14.1)	<i>A</i> – количество функций (или классов функций), понятных после прочтения документации на программный продукт; <i>B</i> – общее количество функций (или классов функций), реализуемых программным продуктом
Обучаемость	Эффективность документации пользователя и/или справочной системы (help) (14.1)	<i>A</i> – количество задач, успешно выполненных после получения оперативной справки и/или чтения документации; <i>B</i> – общее количество протестированных задач
Простота использования	Исправление ошибок при использовании (14.1)	<i>A</i> – число экранов или форм, где входные данные были успешно модифицированы или изменены (восстановлены) перед очередной обработкой; <i>B</i> – число экранов или форм, где пользователь пытался модифицировать или изменить (восстановить) входные данные в течение испытательного срока использования
Привлекательность	Изменяемость вида интерфейса (14.1)	<i>A</i> – количество элементов интерфейса, измененных внешне для удовлетворения пользователя; <i>B</i> – количество элементов интерфейса, которые пользователь хочет изменить

Название подхарактеристики	Название метрики, формула или номер формулы для ее оценки	Исходные данные для вычисления метрики по соответствующей формуле
Соответствие практичности	Соответствие практичности (14.2)	A – количество заданных элементов соответствия практичности, не подтвержденных при тестировании; B – общее количество заданных элементов соответствия практичности
Эффективность		
Поведение во времени	Время отклика $X=A-B$	A – момент времени получения результата; B – момент времени завершения ввода команды
Использование ресурсов	Использование устройств ввода/вывода (14.1)	A – время занятости устройств ввода/вывода; B – заданное время, предназначенное для использования устройств ввода/вывода
Соответствие эффективности	Соответствие эффективности (14.2)	A – количество заданных элементов соответствия эффективности, не подтвержденных при тестировании; B – общее количество заданных элементов соответствия эффективности
Сопровождаемость		
Анализируемость	Поддержка диагностическими функциями (14.1)	A – количество отказов, при которых персонал сопровождения с помощью диагностических функций может диагностировать причину; B – общее число зарегистрированных отказов
Изменяемость	Возможность управления изменением ПС (14.1)	A – количество фактически записанных данных регистрации изменений; B – запланированное количество данных регистрации изменений, достаточное для отслеживания изменений ПС
Стабильность	Возникновение отказа после изменения $X=A/N$	A – количество отказов, возникших в течение заданного периода после устранения отказа; N – количество отказов, устраненных путем изменения ПС

Название подхарактеристики	Название метрики, формула или номер формулы для ее оценки	Исходные данные для вычисления метрики по соответствующей формуле
Тестируемость	Доступность встроенных функций тестирования (14.1)	A – количество случаев, в которых персонал сопровождения может использовать встроенные функции тестирования; B – количество подходящих случаев, в которых можно было бы использовать встроенные тесты
Соответствие сопровождаемости	Соответствие сопровождаемости (14.2)	A – количество заданных элементов соответствия сопровождаемости, которые не были выполнены при тестировании; B – общее количество заданных элементов соответствия сопровождаемости
Мобильность		
Адаптируемость	Адаптируемость структур данных (14.1)	A – количество работоспособных данных, которые не требуют сопровождения при адаптации; B – ожидаемое число работоспособных данных в окружающей среде, к которой ПС адаптировано
Настраиваемость	Простота установки (14.1)	A – число успешных случаев приспособления пользователем операции инсталляции к среде эксплуатации; B – общее число попыток пользователя приспособить операцию инсталляции к среде окружения
Совместимость	Доступная совместимость $X=A/T$	A – число любых ограничений или непредусмотренных отказов, с которыми пользователь сталкивается во время одновременной работы с другими ПС; T – продолжительность одновременной работы с другими ПС
Взаимозаменяемость	Преемственность данных (14.1)	A – число данных замещаемого ПС, которые могут продолжать использоваться после его замещения; B – число данных замещаемого ПС, которые по плану должны продолжать использоваться после его замещения

Название подхарактеристики	Название метрики, формула или номер формулы для ее оценки	Исходные данные для вычисления метрики по соответствующей формуле
Соответствие мобильности	Соответствие мобильности (14.2)	A – количество заданных элементов соответствия мобильности, которые не были выполнены при тестировании; B – общее количество заданных элементов соответствия мобильности

В табл. 14.3 приведены примеры метрик качества в использовании из стандарта *ISO/IEC TR 9126-4:2004* [62]. Во втором столбце таблицы по каждой характеристике приведено название одной метрики, уникальная формула или номер формулы (14.1) или (14.2) из подразд. 14.1 для оценки данной метрики. Исходные данные в третьем столбце – это данные, используемые в соответствующей формуле для вычисления значения представленной метрики.

Таблица 14.3

Метрики качества ПС в использовании

Название характеристики	Название метрики, формула или номер формулы для ее оценки	Исходные данные для вычисления метрики по соответствующей формуле
Результативность	Завершение задачи (14.1)	A – количество завершенных задач; B – общее количество попыток выполнения задач
Продуктивность	Коэффициент продуктивности (14.1)	A – продуктивное время; $A = B - (B1 + B2 + B3)$, где B – продолжительность выполнения задачи; $B1$ – продолжительность помощи; $B2$ – продолжительность обработки ошибок; $B3$ – продолжительность поиска
Безопасность	Экономический ущерб (14.2)	A – число случаев экономического ущерба; B – общее число случаев использования
Удовлетворенность	Использование по собственному усмотрению (14.1)	A – количество случаев использования заданных в спецификации функций программного средства/приложений/систем; B – количество случаев их запланированного использования

15. МЕТРОЛОГИЯ КАЧЕСТВА СИСТЕМ И ПРОГРАММНЫХ СРЕДСТВ В СТАНДАРТАХ СЕРИИ SQUARE

15.1. Эталонная модель измерений по стандарту ISO/IEC 25020:2007

Измерению качества (квалиметрии) систем и программных средств в стандартах серии *SQuaRE* посвящена группа стандартов *ISO/IEC 2502n* (см. подразд. 12.4). В настоящее время в данной группе приняты два стандарта: *ISO/IEC 25020:2007* и *ISO/IEC 25021:2012*.

Стандарты *ISO/IEC 25022*, *ISO/IEC 25023*, *ISO/IEC 25024* находятся в стадии разработки. Данные стандарты посвящены соответственно мерам качества в использовании, мерам внутреннего и внешнего качества и мерам качества данных. Основой для разработки *ISO/IEC 25022* и *ISO/IEC 25023* служат стандарты *ISO/IEC 9126-2-4:2003-2004*, в которых приведены рекомендуемые наборы внешних, внутренних метрик и метрик качества в использовании программных средств. Основные положения стандартов *ISO/IEC 9126-2-4:2003-2004* рассмотрены в разд. 14.

Следует отметить, что понятие *мера качества*, используемое в стандартах серии *SQuaRE*, эквивалентно понятию *метрика качества* в стандартах *ISO/IEC 9126-1-4:2001-2004*. Термин «мера» введен в стандарты *SQuaRE* для обеспечения совместимости со стандартом *ISO/IEC 15939:2007* [35].

В стандарте *ISO/IEC 25020:2007. Программная инженерия. Требования к качеству и оценка программных продуктов (SQuaRE). Эталонная модель измерений* [43] регламентирована эталонная модель измерений качества программного продукта.

Эталонная модель измерений качества программного продукта (software product quality measurement reference model SPQM-RM) описывает связь модели качества, определенных в ней характеристик и подхарактеристик качества и атрибутов (свойств) программного продукта с соответствующими мерами качества программных средств, функциями измерений, элементами меры качества (ЭМК).

На рис. 15.1 показаны связи между моделью качества и конструированием мер качества из элементов мер качества. Из данного рисунка видно, что меры качества программных средств формируются с помощью применения соответствующих функций измерения к элементам мер качества.

В отдельных случаях элемент меры качества может непосредственно использоваться как мера качества программных средств или систем. При этом в качестве функции измерения используется идентифицирующая функция вида $F = \text{ЭМК}$.

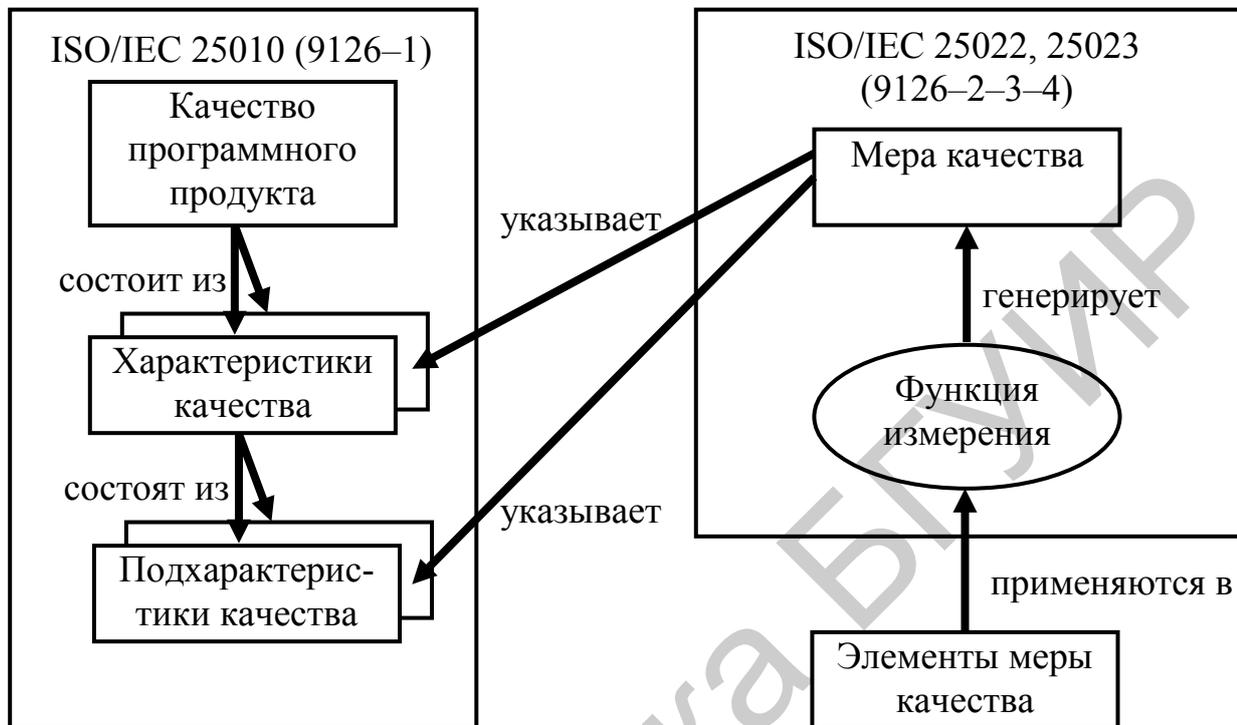


Рис. 15.1. Эталонная модель измерения качества программного продукта по стандарту ISO/IEC 25020:2007

15.2. Концепция элементов мер качества по стандарту ISO/IEC 25021:2012

Целью стандарта *ISO/IEC 25021:2012. Системная и программная инженерия. Требования к качеству и оценка систем и программных средств (SQuaRE). Элементы мер качества* [44] является определение начального набора элементов мер качества для использования в течение жизненного цикла продукта. Стандарт *ISO/IEC 25021:2012* предоставляет пользователям возможность выбора и определения подходящих ЭМК для определения внутренних и внешних мер, мер данных и мер качества в использовании. Эти меры в дальнейшем могут быть использованы для спецификации требований к качеству и оценки продуктов.

Напомним, что *качество продукта* в общем случае включает качество системы, качество программного продукта, качество данных и качество системной услуги (см. разд. 12).

В стандарте *ISO/IEC 25021:2012* приведены правила для проектирования элементов мер качества и верификации существующих ЭМК. Содержимое стандарта устанавливает связь между сериями стандартов *ISO/IEC 9126-1-4:2001-2004* [59 – 62] и *SQuaRE*. Данную связь отражает рис. 15.2.

Приведенный в стандарте *ISO/IEC 25021:2012* набор ЭМК использован при проектировании мер качества, определенных в стандартах *ISO/IEC 9126-2-4:2003-2004*. Каждая из мер качества данных стандартов состоит как минимум из двух ЭМК.

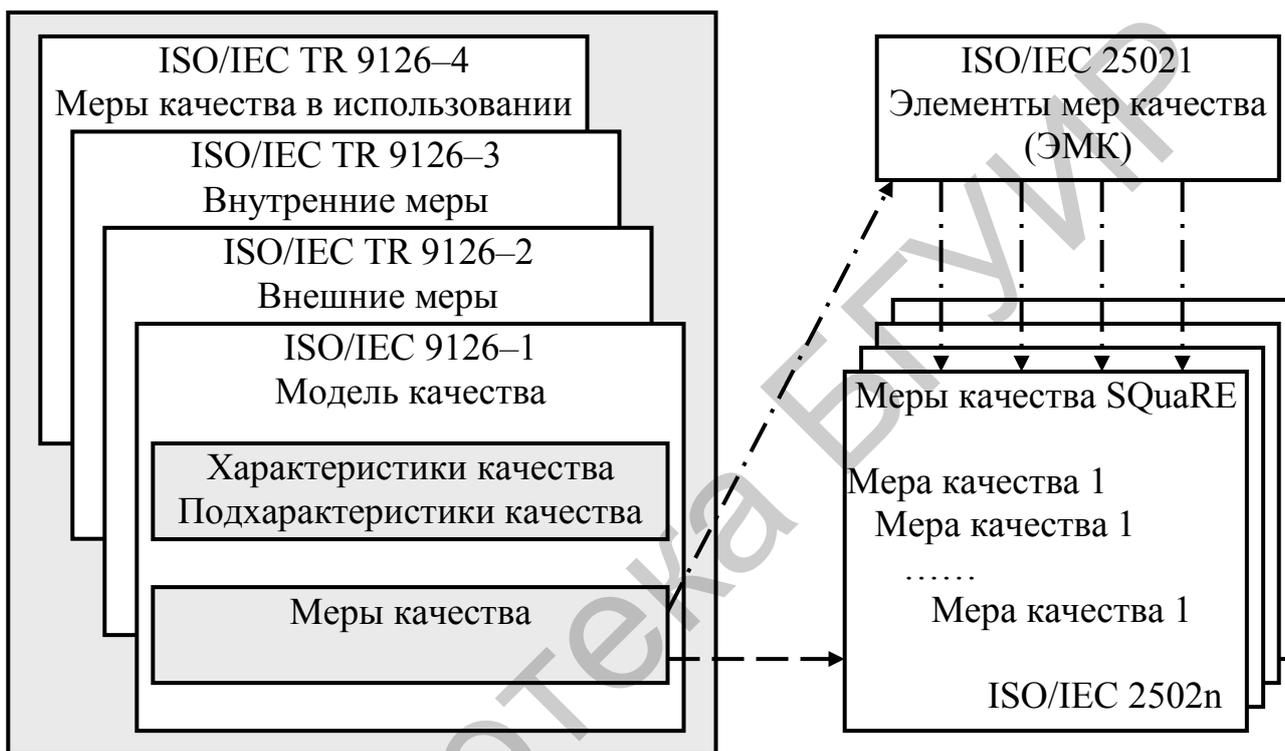


Рис. 15.2. Организация связи между сериями стандартов 9126 и SQuaRE с помощью стандарта ISO/IEC 25021

Основными целями определения и использования ЭМК являются:

- обеспечение руководства организацией разработки и реализации своих собственных ЭМК;
- содействие совместному использованию специфических ЭМК для измерения и использования свойств продукта, которые важны для различных характеристик и подхарактеристик качества продукта;
- помощь в идентификации уникального набора ЭМК, требуемого для создания всех мер качества для конкретного набора характеристик или подхарактеристик продукта.

Для определения значений элементов мер качества, используемых в эталонной модели измерения качества продукта (см. рис. 15.1), применяется метод измерения. Связь между свойством качества, методом измерения и ЭМК представлена на рис. 15.3.

Свойство качества – это свойство целевого объекта (например программного продукта), которое связывается с ЭМК и которое может быть количественно оценено с помощью метода измерения. Свойство качества при необходимости может быть связано с несколькими различными подсвойствами. Эта связь может быть выражена в виде схемы или формулы.

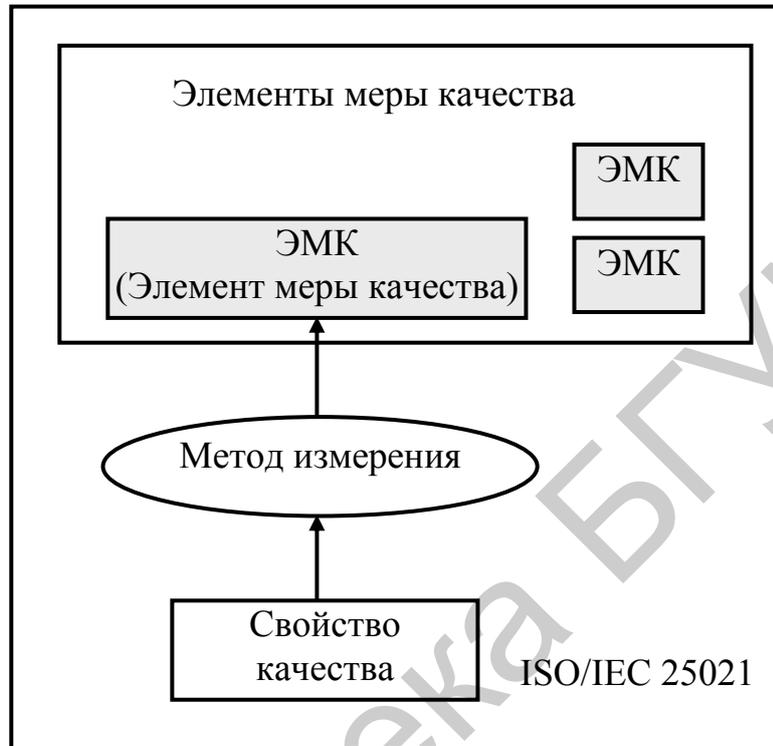


Рис. 15.3. Связь между свойством качества, методом измерения и элементом меры качества

Для количественной оценки свойства качества пользователь метода измерения должен идентифицировать и собрать соответствующие данные. Эти данные могут извлекаться из различных источников (документации, кодов и т. п.).

Метод измерения представляет собой описанную с общих позиций последовательность действий, используемую для измерения. Для одних и тех же свойств качества могут использоваться различные методы измерения, определяющие в конечном итоге соответствующие правила численной оценки свойств.

Элемент меры качества – это мера, описанная в терминах оцениваемого свойства качества и соответствующего метода измерения.

На рис. 15.4 приведен пример связи между свойством качества, методом измерения, ЭМК и мерой качества (МК) для характеристики *Надежность*.

В стандарте *ISO/IEC 25021:2012* предложено **табличное представление метода измерения**, используемого для определения и количественной оценки ЭМК.

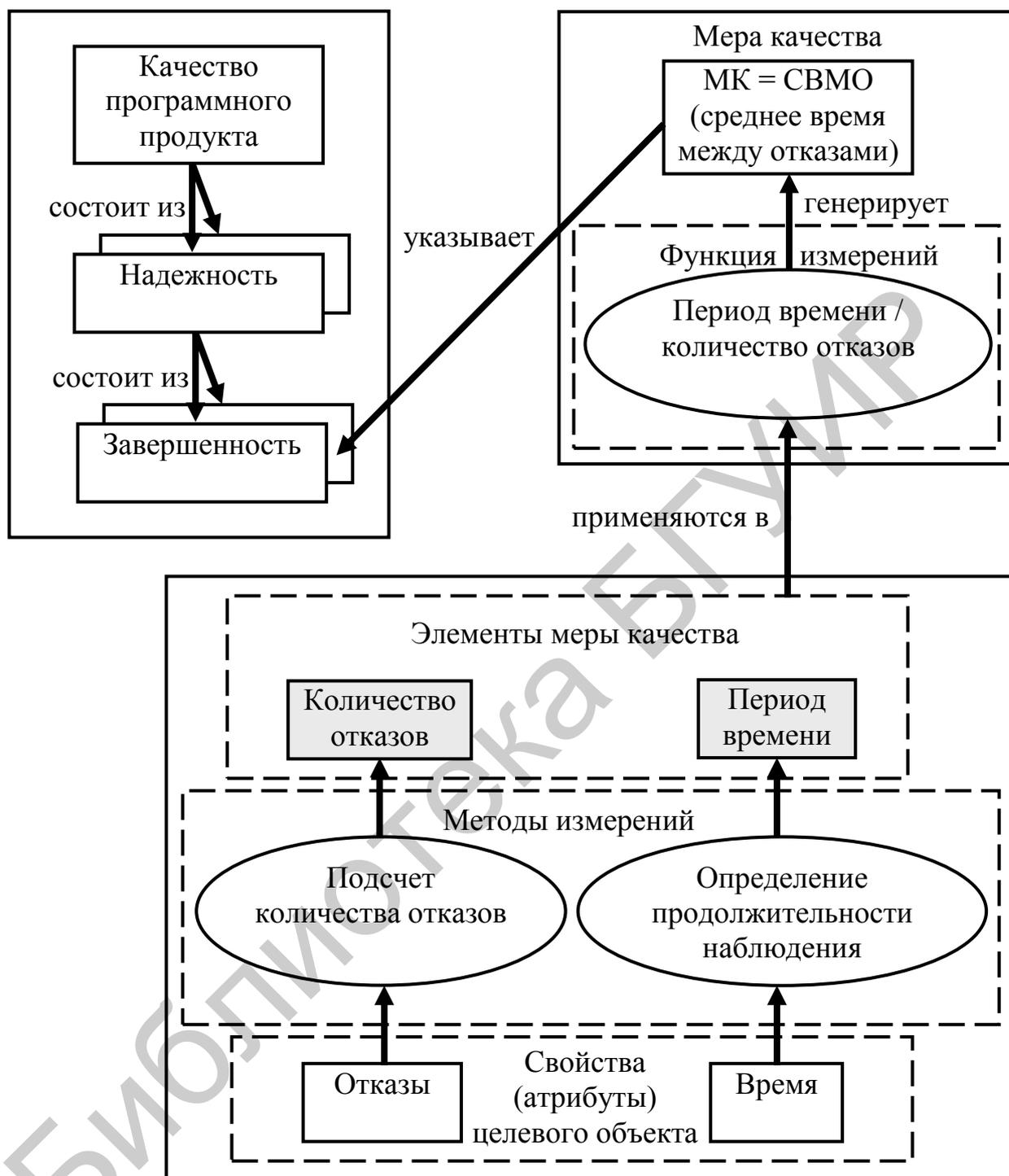


Рис. 15.4. Пример эталонной модели измерения характеристики *Надежность* программного продукта

Табличный формат метода измерения должен содержать следующую информацию:

1. Название ЭМК – уникальное имя и при необходимости серийный номер ЭМК.

2. Целевой объект – объект, который должен быть охарактеризован измерением его свойства.

3. Цели и свойство качества:

- идентификация свойства качества, обычно связанная с именем ЭМК;
- определение целей, описывающее, что намереваются узнать с помощью данного ЭМК и какая необходимая информация должна быть представлена данным ЭМК.

4. Соответствующие меры качества – конкретные меры качества, которые используют данный ЭМК.

5. Метод измерения – определяет, как собирать и как преобразовывать значение оцениваемого количественно свойства с помощью математической функции. Контекст ЭМК, процесс жизненного цикла программных средств, принудительные условия измерений и численные правила являются частями метода измерения.

6. Список подсвойств, связанных со свойством качества (при необходимости) – подсвойства, связь которых со свойством должна быть выражена в виде схемы или формулы.

7. Определение каждого подсвойства из списка подсвойств (при необходимости).

8. Вход для ЭМК – достаточно подробное описание количественной информации, используемой для измерения ЭМК, с идентификацией ее источников. Источниками могут являться, например, документированные рабочие продукты, поведение системы или программного средства, а также поведение пользователей, операторов, разработчиков, тестировщиков, персонала сопровождения.

9. Единица измерений для ЭМК – единица измерений и при необходимости соответствующая формула.

10. Количественные правила – правила численной оценки, описанные с практической точки зрения (обычно в форме текста) или с теоретической точки зрения (обычно в виде математического выражения).

11. Тип шкалы – шкала может быть номинальной, порядковой, интервальной или шкалой отношений (см. подразд. 13.2).

12. Контекст ЭМК – информация о планируемом использовании результата измерения, содержащая типичные примеры характеристик, подхарактеристик или мер качества, для оценки которых планируется применять результаты измерения ЭМК.

13. Процесс (процессы) жизненного цикла программных средств – типичные процессы жизненного цикла, связанные с целевым объектом, позволяющим получить измеренное значение ЭМК.

14. Принудительные условия измерений (при необходимости) – любые обязательные условия, связанные с методом измерений.

Вышеприведенные элементы табличного формата метода измерения разделяются на *четыре группы*. Элемент (1) предназначен для идентификации ЭМК. Элементы (2) – (4) определяют, что собой представляет данный ЭМК. Элементы (5) – (11) описывают, как измерять ЭМК. Элементы (12) – (14) предназначены для управления применением ЭМК.

В табл. 15.1 приведен пример заполнения формата представления метода измерения для дефектов (faults) в коде.

15.3. Меры качества данных по стандарту ISO/IEC 25012:2008

В пп. 12.3.4 – 12.3.7 рассмотрены модель и характеристики качества данных, регламентированные стандартом *ISO/IEC 25012:2008*. В данном стандарте характеристики качества данных рассматриваются с двух точек зрения – собственное качество данных и системно-зависимое качество данных.

Для всех характеристик качества в *ISO/IEC 25012:2008* приведены примеры мер качества данных. Как и метрики качества в стандартах *ISO/IEC 9126–2–4:2003–2004*, меры качества данных в *ISO/IEC 25012:2008* определяются с помощью функций измерения

$$A / B \quad (15.1)$$

или

$$1 - A / B. \quad (15.2)$$

Функции измерения (15.1) и (15.2) соответствуют формулам (14.1) и (14.2), приведенным в подразд. 14.1. Меры качества данных должны удовлетворять свойствам и критериям обоснованности метрик из вышеназванного подраздела. С учетом этого из функций измерения (15.1) и (15.2) выбирается та, которая соответствует *критериям трассировки и непротиворечивости*: с увеличением относительного значения меры значение характеристики качества должно увеличиваться.

В табл. 15.2 – 15.4 приведены примеры мер качества данных, содержащиеся в *ISO/IEC 25012:2008*, с собственной точки зрения, собственной и системно-зависимой точек зрения, системно-зависимой точки зрения соответственно. Во втором столбце данных таблиц по каждой характеристике приведено название меры, функция измерения (15.1) или (15.2) или уникальная функция измерения для оценки данной меры. В третьем столбце приведены элементы меры качества – исходные данные, используемые соответствующей функцией измерения для вычисления значения меры качества данных (см. подразд. 15.1, 15.2).

Метод измерения для дефектов

№ п/п	Элемент табличного формата	Описание
1	Название ЭМК	Количество дефектов в коде
2	Целевой объект	Исходный код программы
3	Цели и свойство для количественной оценки	<p>Целью является нахождение дефектов фазы кодирования, используя спецификации проектирования. Должно быть измерено количество ошибочных строк кода.</p> <p>Свойство для количественной оценки – дефект.</p> <p>Определение дефекта: 1) проявление ошибки в программном средстве; 2) некорректный шаг, процесс или определение данных в компьютерной программе [36]. Ненайденный дефект может явиться причиной отказа</p>
4	Соответствующие меры качества	<p>Меры <i>Надежности</i> программного средства, основанные на использовании плотности дефектов:</p> <ul style="list-style-type: none"> – нахождение степени обнаружения дефектов в течение фазы кодирования; – нахождение степени устранения дефектов с исправленными дефектами на фазе кодирования. <p>Подхарактеристика <i>Завершенность</i> характеристики <i>Надежность</i> программных средств</p>
5	Метод измерений	<p>Метод измерения дефектов в коде программного средства.</p> <p>Рецензирование или анализ отличий в исходном коде рецензируемой программы и идентификация откорректированных строк кода, которые содержат модифицированные строки, добавленные строки и удаленные строки кода.</p> <p><i>Примечание.</i> Исходный код программы обычно рецензируется в работах, связанных с верификацией и аттестацией, таких как рецензирование кода, тестирование модулей, причинный анализ для разрешения отказов при интеграционном тестировании</p>
6	Список под-свойств, связанных со свойством качества	Список: исполняемые операторы, ошибочные строки кода, исправленные строки кода

№ п/п	Элемент табличного формата	Описание
7	Определение каждого подсвойства	<p>Исполняемые операторы: операторы, которые могут быть отнесены к помеченным операторам, выражениям, выбирающим операторам, операторам цикла или операторам перехода.</p> <p>Неисполняемые операторы: операторы, относящиеся к объявлениям.</p> <p>Ошибочная строка кода: строка кода, которая содержит дефект (дефекты). Содержит исходный код ошибки или нет, определяется на основе спецификации.</p> <p>Правильная строка кода: строка кода без дефекта.</p> <p><i>Примечание.</i> Возможны ситуации, когда существующие строки кода правильны, а должны быть изменены спецификации. Такие строки кода не подсчитываются как ошибочные</p>
8	Вход для ЭМК	Исходный код и спецификации проектирования исходного кода
9	Единица измерения для ЭМК	Ошибочная строка кода
10	Численные правила	<p>Суммирование ошибочных строк кода.</p> <p>Правило численной оценки на основе практического анализа использует следующие действия по измерению: рецензирование или анализ отличий исходного кода рецензируемой программы и идентификация исправленных строк кода, которые содержат модифицированные строки, добавленные строки кода и удаленные строки кода</p>
11	Тип шкалы	Отношений
12	Контекст ЭМК	Этот ЭМК обычно выбирается для меры подхарактеристики <i>Завершенность</i> характеристики <i>Надежность</i> программных средств
13	Процесс жизненного цикла программных средств	Конструирование (кодирование и модульное тестирование) программных средств. Фаза кодирования
14	Принудительные условия измерения	Чтобы позволить сравнивать существующие строки кода со спецификациями проектирования, должны быть доступны спецификации проектирования

Меры качества данных с собственной точки зрения

Название характеристики	Название меры, функция измерения	Элементы меры качества
Правильность	Синтаксическая правильность поля записей (15.1)	<i>A</i> – количество записей с синтаксически правильным заданием поля; <i>B</i> – общее количество записей
Полнота	Полнота данных в файле (15.1)	<i>A</i> – количество данных в файле из требуемых в конкретном контексте использования; <i>B</i> – общее количество данных в заданном конкретном контексте предполагаемого использования
Согласованность	Согласованность файла данных (15.1)	<i>A</i> – количество согласованных данных в файле; <i>B</i> – количество данных, записанных в файл
Правдоподобие	Правдоподобие данных, используемых банком для оценки кредитного риска (15.1)	<i>A</i> – количество данных, сертифицированных посредством внутреннего аудита из полученных данных с информацией о кредитном риске; <i>B</i> – количество данных, используемых для получения информации о кредитном риске
Актуальность	Актуальность значения области данных (15.1)	<i>A</i> – количество проверок данных, в которых значения диагностируемых данных соответствуют требованиям актуальности; <i>B</i> – количество проверок значений в заданной области данных

Таблица 15.3

Меры качества данных с собственной и системно-зависимой точек зрения

Название характеристики	Название меры, функция измерения	Элементы меры качества
Доступность	<i>Собственная мера.</i> Доступность звуковых данных (15.2)	<i>A</i> – количество данных, сохраненных только в виде звука (например без текстового представления звука); <i>B</i> – количество значений данных, представляющих звук
	<i>Системно-зависимая мера.</i> Многоканальная доступность данных (15.1)	<i>A</i> – количество данных, к которым успешно получил доступ пользователь с ограниченными возможностями различного рода; <i>B</i> – количество имеющихся данных
Соответствие	<i>Собственная мера.</i> Несоответствие законам о тайне частной жизни: значения <i>A</i>	<i>A</i> – количество элементов данных, которые не соответствуют положениям законов о тайне частной жизни из-за своего содержания
	<i>Системно-зависимая мера.</i> Несоответствие законам о тайне частной жизни: архитектура <i>A</i>	<i>A</i> – количество элементов, которые не соответствуют положениям законов о тайне частной жизни из-за несостоятельности (неисправности) технической архитектуры
Конфиденциальность	<i>Собственная мера.</i> Использование шифрования (15.1)	<i>A</i> – количество зашифрованных полей базы данных; <i>B</i> – количество полей, которые должны быть зашифрованы
	<i>Системно-зависимая мера.</i> Неуязвимость (15.2)	<i>A</i> – количество успешных проникновений при тестировании на проникновение; <i>B</i> – количество попыток проникновения

Название характеристики	Название меры, функция измерения	Элементы меры качества
Эффективность	<i>Собственная мера.</i> Числа, хранимые в виде строк A	A – количество данных, хранимых в виде строк
	<i>Системно-зависимая мера.</i> Растроченный напрасно размер хранилища $\Sigma(B-A)$	A – эталонная усредненная область для эффективного хранения данных в базе данных; B – используемая область данных в любых физических файлах базы данных
Точность	<i>Собственная мера.</i> Точность значений данных (15.1)	A – количество значений данных с требуемой точностью; B – общее количество значений данных
	<i>Системно-зависимая мера.</i> Точность полей в базе данных (15.1)	A – количество полей данных в базе данных, определенных с требуемой точностью; B – общее количество полей данных в базе данных
Трассируемость	<i>Собственная мера.</i> Трассируемость значений (15.1)	A – количество данных, для которых требуемая трассируемость значений доступна; B – количество элементов данных, для которых трассируемость протестирована
	<i>Системно-зависимая мера.</i> Автоматическая трассируемость A	A – количество элементов данных, трассируемых автоматически (используя системные возможности)
Понятность	<i>Собственная мера.</i> Понятность эталонных данных благодаря существующим метаданным (15.1)	A – количество данных с существующими метаданными в файлах эталонных данных; B – количество данных в файлах эталонных данных
	<i>Системно-зависимая мера:</i> Понятность эталонных данных благодаря ссылкам на метаданные (15.1)	A – количество полей, имеющих метаданные, которые автоматически связываются с соответствующими данными; B – общее количество полей

Меры качества данных с системно-зависимой точки зрения

Название характеристики	Название меры, функция измерения	Элементы меры качества
Готовность	Готовность элементов данных (15.1)	<i>A</i> – количество элементов данных, готовых к использованию во время действий резервного копирования/восстановления; <i>B</i> – количество элементов данных процедур резервного копирования/восстановления
Мобильность	Мобильность данных (15.1)	<i>A</i> – количество данных, которые сохранили существующие атрибуты качества после миграции в другую компьютерную систему; <i>B</i> – количество мигрировавших данных
Восстанавливаемость	Восстанавливаемость (15.1)	<i>A</i> – количество элементов данных, успешно сохраненных/восстановленных во время операции резервного копирования/восстановления; <i>B</i> – количество элементов данных процедуры резервного копирования/восстановления

16. МЕТРОЛОГИЯ СЛОЖНОСТИ ПРОГРАММНЫХ СРЕДСТВ

Качество программных средств во многом зависит от их сложности. Например, чем сложнее программа, тем ниже ее надежность и сопровождаемость. Поэтому при оценке качества программ обычно оценивается и их сложность.

Сложность программных средств определяется рядом факторов. Основными из них являются сложность архитектуры и технического проекта ПС, а также сложность кодов ПС.

Сложность программных средств принято оценивать с помощью различных мер (в литературе широко используется как термин «мера сложности», так и термин «метрика сложности»).

Метрики (меры) сложности программных средств обычно подразделяются на *три основные группы* [67, 71]:

- метрики размера программных средств;
- метрики сложности потока управления программных средств;
- метрики сложности потока данных программных средств.

В *приложении Е* стандарта *ISO/IEC 9126-3:2003* метрики сложности отнесены к чисто внутренним метрикам качества программных средств [61].

В общем случае метрики сложности могут рассматриваться как элементы мер качества ПС.

16.1. Метрики размера программных средств

Метрики этой группы основаны на анализе исходных текстов программ.

Существуют различные метрики, с помощью которых может быть оценен размер программы.

К наиболее простым метрикам размера программы относятся *количество строк исходного текста программы* и *количество операторов программы*.

Из метрик размера программ широкое распространение получили *метрики Холстеда* [70].

Основу метрик Холстеда составляют *шесть базовых метрик* программы:

- η_1 – число уникальных операторов программы (словарь операторов);
- η_2 – число уникальных операндов программы (словарь операндов);
- N_1 – общее число операторов в программе;
- N_2 – общее число операндов в программе;
- f_{1j} – число вхождений j -го оператора, $j = 1, 2, \dots, \eta_1$;

– f_{2i} – число вхождений i -го операнда, $i = 1, 2, \dots, \eta_2$.

Справедливы следующие соотношения:

$$N_1 = \sum_{j=1}^{\eta_1} f_{1j} \quad , \quad N_2 = \sum_{i=1}^{\eta_2} f_{2i} \cdot$$

Базовые метрики определяются непосредственно при анализе исходных текстов программ. На основе базовых метрик Холстед предложил рассчитывать ряд производных метрик программы. Наиболее широко используются следующие метрики:

– словарь программы (общее число уникальных операторов и операндов программы):

$$\eta = \eta_1 + \eta_2; \quad (16.1)$$

– длина программы (общее количество операторов и операндов программы):

$$N = N_1 + N_2; \quad (16.2)$$

– объем программы (число логических единиц информации, необходимых для записи программы):

$$V = N \log_2 \eta. \quad (16.3)$$

Операнды программы представляют собой используемые в ней переменные и константы.

Под операторами программы Холстед подразумевает входящие в ее состав символы-ограничители (в том числе символы операций, символы скобок и символы-разделители), управляющие операторы, а также имена подпрограмм (процедур и функций). При этом парные символы (например пара из открывающей и закрывающей скобок) считаются одним оператором. Несколько служебных слов, входящих в состав одного оператора (например If...Then...Else), также считаются одним оператором.

Метки не относятся ни к операторам, ни к операндам.

При подсчете операндов и операторов Холстеда объявления и инициализацию элементов программы (например типов, констант, переменных) вместе с соответствующими символами-ограничителями принято не учитывать.

Очевидно, что совокупность операторов программы и их количество зависят от языка программирования, на котором написана программа.

В табл. 16.1 приведены основные операторы процедурно-ориентированного подмножества языка *Delphi* в интерпретации Холстеда. При подсчете количества операторов и операндов в программе, написанной на языке *Delphi*, обычно анализируется только ее раздел операторов, а также разделы операторов процедур и функций пользователя.

В табл. 16.2 приведены основные операторы языка *C* в интерпретации Холстеда.

Операторы языка Delphi в интерпретации Холстеда

Обозначение оператора	Назначение оператора
+	плюс (сложение, объединение множеств, сцепление строк)
-	минус (изменение знака, вычитание, разность множеств)
*	звездочка (умножение, пересечение множеств)
/	наклонная черта, слэш (деление)
<	меньше
>	больше
=	равно
.	точка (разделитель полей при обращении к элементам записи)
,	запятая (разделитель в перечислениях)
:	двоеточие (отделяет константы выбора в операторе Case)
;	точка с запятой (разделитель операторов программы)
()	левая и правая скобки при выделении подвыражений
[]	левая и правая квадратные скобки (выделяет индексы элементов массивов)
<=	меньше или равно
>=	больше или равно
<>	не равно
:=	операция присваивания
^	знак карата (обращение к динамической переменной)
@	коммерческое 'at' (операция взятия адреса элемента)
And	операция поразрядного логического сложения (И)
Not	операция поразрядного дополнения (НЕ)
Or	операция поразрядного логического сложения (ИЛИ)
Xor	операция поразрядного логического исключающего ИЛИ
Div	целочисленное деление
Mod	остаток от целочисленного деления
Shl	операция сдвига влево
Shr	операция сдвига вправо
In	операция проверки вхождения элемента в множество
Begin...End	составной оператор
Break	оператор безусловного выхода из цикла
Continue	оператор передачи управления на конец тела цикла
Goto <Метка>	оператор безусловного перехода
Case...Of...Else... End	оператор варианта

Обозначение оператора	Назначение оператора
If...Then...Else	оператор условного перехода
Repeat...Until	оператор цикла с постусловием
While...Do	оператор цикла с предусловием
For...To...Do	оператор цикла с параметром (с увеличением параметра)
For...Downto...Do	оператор цикла с параметром (с уменьшением параметра)
With...Do	оператор присоединения

Таблица 16.2

Операторы языка C в интерпретации Холстеда

Обозначение оператора	Назначение оператора
=	операция присваивания
+	сложение
-	вычитание (изменение знака)
*	звездочка (умножение, обращение к динамической переменной)
/	деление
%	остаток от целочисленного деления
++	инкремент
--	декремент
==	равно
!=	не равно
>	больше
<	меньше
>=	больше или равно
<=	меньше или равно
!	логическое отрицание (НЕ)
&&	логическое И
	логическое ИЛИ
~	побитовая инверсия
&	побитовое И, ссылка
	побитовое ИЛИ
^	побитовое исключающее ИЛИ
<<	побитовый сдвиг влево (в сторону старших разрядов)
>>	побитовый сдвиг вправо (в сторону младших разрядов)
+=	присваивание с суммированием
-=	присваивание с вычитанием

Обозначение оператора	Назначение оператора
*=	присваивание с умножением
/=	присваивание с делением
%=	присваивание по модулю
&=	присваивание с побитовым И
 =	присваивание с побитовым ИЛИ
^=	присваивание с побитовым исключающим ИЛИ
<<=	присваивание с побитовым сдвигом влево
>>=	присваивание с побитовым сдвигом вправо
()	левая и правая скобки при выделении подвыражений
[]	обращение к элементу массива
->	динамическое обращение к элементу структуры
.	статическое обращение к элементу структуры
,	запятая (операция и разделитель)
;	точка с запятой (разделитель операторов программы)
:	двоеточие (отделяет константы выбора в операторе switch)
?...:	условный оператор
sizeof	размер
(type)	преобразование типа
{ }	составной оператор
if...else	оператор выбора
switch()...case... default	оператор множественного выбора
do...while()	оператор цикла с постусловием
while()	оператор цикла с предусловием
for()	оператор цикла с параметром
goto <метка>	оператор безусловного перехода
continue	оператор перехода к следующему шагу цикла
break	оператор выхода из цикла

Пример 1

Расчет метрик Холстеда для программы, вычисляющей значение функции Y через разложение функции в бесконечный ряд

$$Y = X - X^3 / 3! + X^5 / 5! - X^7 / 7! + \dots$$

с точностью $Eps = 0,0001$.

Текст программы на языке *Delphi*, реализующей вычисление функции Y , приведен ниже.

```

Program Sin1;
  Const
    eps = 0.0001;
  Var
    y, x: real; n: integer; vs: real;
  Begin
    Readln (x);
    y := x; {Начальные установки}
    n := 2;
    vs := x;
    Repeat
      vs := -vs * x * x / (2 * n - 1) / (2 * n - 2); {Формирование слагаемого}
      n := n + 1;
      y := y + vs
    Until abs(vs) < eps; {Выход из цикла по выполнению условия}
    Writeln (x, y, eps)
  End.

```

Расчет базовых метрик Холстеда для данной программы приведен в табл. 16.3. По формулам (16.1) – (16.3) рассчитываются словарь, длина и объем программы.

Таблица 16.3

Расчет базовых метрик Холстеда для программы на языке Delphi,
вычисляющей значение функции $Y = \sin X$

<i>j</i>	Оператор	<i>f_{1j}</i>	<i>i</i>	Операнд	<i>f_{2i}</i>
1	;	7	1	x	6
2	:=	6	2	n	5
3	*	4	3	vs	5
4	-	3	4	y	4
5	/	2	5	eps	2
6	()	2	6	2	4
7	+	2	7	1	2
8	,	2			
9	Begin...End	1			
10	Readln ()	1			
11	Repeat...Until	1			
12	abs()	1			
13	<	1			
14	Writeln ()	1			
$\eta_1 = 14$		$N_1 = 34$	$\eta_2 = 7$		$N_2 = 28$

Словарь программы: $\eta = 14 + 7 = 21$.

Длина программы: $N = 34 + 28 = 62$.

Объем программы: $V = 62 \log_2 21 \approx 272$.

Пример 2

Расчет метрик Холстеда для программы на языке C. Программа реализует вычисление той же функции Y , что и программа из примера 1.

```
#include <stdio.h>
#include <stdlib.h>
main()
{
    const double eps = 0.0001;
    double y, x;
    int n;
    double vs;

    scanf("%f", &x);
    y = x; // Начальные установки
    n = 2;
    vs = x;
    do
    {
        vs = -vs * x * x / (2 * n - 1) / (2 * n - 2); // Формирование слагаемого
        n++;
        y += vs;
    }
    while (abs(vs) >= eps); // Выход из цикла по выполнению условия
    printf("%f %f %f\n", x, y, eps);
}
```

Расчет базовых метрик Холстеда для данной программы приведен в табл. 16.4. По формулам (16.1) – (16.3) рассчитываются словарь, длина и объем программы.

Словарь программы: $\eta = 16 + 7 = 23$.

Длина программы: $N = 38 + 25 = 63$.

Объем программы: $V = 63 \log_2 23 \approx 285$.

Расчет базовых метрик Холстеда для программы на языке C,
вычисляющей значение функции $Y = \sin X$

j	Оператор	f_{1j}	i	Операнд	f_{2i}
1	;	9	1	x	6
2	=	4	2	vs	5
3	*	4	3	n	4
4	,	4	4	y	3
5	-	3	5	eps	2
6	/	2	6	1	1
7	()	2	7	2	4
8	{...}	2			
9	scanf ()	1			
10	do...while ()	1			
11	abs()	1			
12	>=	1			
13	printf ()	1			
14	++	1			
15	+=	1			
16	&	1			
$\eta_1 = 16$		$N_1 = 38$	$\eta_2 = 7$		$N_2 = 25$

16.2. Метрики сложности потока управления программных средств

Метрики сложности потока управления программных средств принято определять на основе представления программ в виде управляющего ориентированного графа $G = (V, E)$, где V – вершины, соответствующие операторам, а E – дуги, соответствующие переходам между операторами [67, 71]. В дуге (v, u) вершина v является исходной, а u – конечной. При этом u непосредственно следует за v , а v непосредственно предшествует u . Если путь от v до u состоит более чем из одной дуги, тогда u следует за v , а v предшествует u .

Частным случаем представления ориентированного графа программы можно считать построенную в соответствии с положениями стандарта *ГОСТ 19.701–90* [7] детализированную схему алгоритма, в которой каждому блоку соответствует один оператор программы. Аналогами вершин графа являются блоки алгоритма, причем данные блоки имеют разное графическое представление в зависимости от их назначения. Дугам графа соответствуют линии передачи управления между блоками алгоритма.

Ниже рассмотрены наиболее распространенные метрики сложности потока управления программ.

Метрика Маккейба (цикломатическая сложность графа программы, цикломатическое число Маккейба) предназначена для оценки трудоемкости тестирования программы. Данная метрика определяется по формуле

$$Z(G) = e - v + 2p,$$

где e – число дуг ориентированного графа G ; v – число вершин; p – число компонентов связности графа.

Число компонентов связности графа – это количество дуг, которые необходимо добавить для преобразования графа в сильносвязный. Сильносвязным графом называется граф, любые две вершины которого взаимно достижимы. Для корректных программ, не имеющих недостижимых от начала программы участков и «висячих» точек входа и выхода, сильносвязный граф получается путем соединения дугой вершины, обозначающей конец программы, с вершиной, обозначающей начало этой программы.

Метрика Маккейба определяет минимальное количество тестовых прогонов программы, необходимых для тестирования всех ее ветвей.

Рассчитаем метрику Маккейба для программы, схема алгоритма которой приведена на рис. 16.1 (для простоты понимания ввод/вывод в данной схеме алгоритма не показан, хотя в реальных вычислениях метрики блоки ввода/вывода следует учитывать). Компонент связности графа обозначен штриховой дугой. Число дуг $e = 8$, число вершин $v = 7$, $p = 1$. Цикломатическое число Маккейба равно $Z(G) = 8 - 7 + 2 = 3$.

Следует обратить внимание, что при расчете метрики Маккейба начальную и конечную вершины алгоритма (блоки «Начало» и «Конец») необходимо учитывать.

Значение метрики Маккейба показывает, что в схеме алгоритма (см. рис. 16.1) можно выделить *три базовых независимых пути* (называемых также линейно независимыми контурами):

- 1) 1 – 2 (нет) – 4 (нет) – 5 – 7;
- 2) 1 – 2 (да) – 3 – 2 (нет) – 4 (нет) – 5 – 7;
- 3) 1 – 2 (нет) – 4 (да) – 6 – 7.

Вторым возможным вариантом совокупности базовых независимых путей является:

- 1) 1 – 2 (да) – 3 – 2 (нет) – 4 (нет) – 5 – 7;
- 2) 1 – 2 (нет) – 4 (нет) – 5 – 7;
- 3) 1 – 2 (да) – 3 – 2 (нет) – 4 (да) – 6 – 7.

Цифры в обозначении путей представляют собой номера блоков в схеме алгоритма программы.

Таким образом, для тестирования совокупности базовых независимых путей исследуемой программы необходимо выполнить минимально три тестовых прогона.

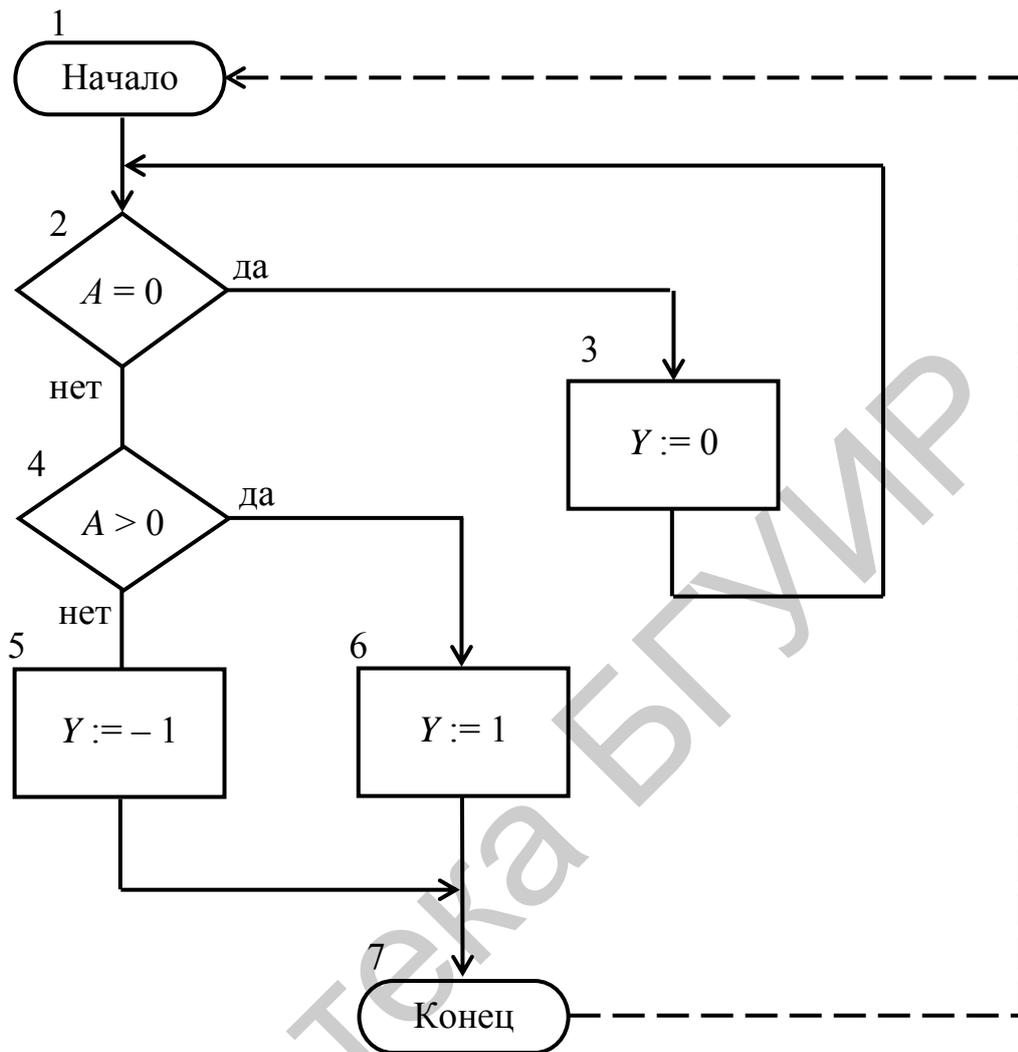


Рис. 16.1. Пример схемы алгоритма программы

Метрика Джилба определяет логическую сложность программы как насыщенность программы условными операторами IF–THEN–ELSE и операторами цикла. Обычно используются два вида метрики Джилба: **CL** – количество условных и циклических операторов, характеризующее абсолютную сложность программы; **cl** – насыщенность программы условными и циклическими операторами, характеризующая относительную сложность программы; **cl** определяется как отношение **CL** к общему количеству операторов программы (здесь под оператором подразумевается оператор конкретного языка программирования в классическом представлении, а не в интерпретации Холстеда).

Расширением метрики Джилба является **максимальный уровень вложенности условного и циклического оператора CLI**.

Использование в программе оператора выбора (например CASE в языке Delphi) с **n** разветвлениями эквивалентно применению **n – 1** оператора IF–THEN–ELSE с глубиной вложенности **n – 2**.

Например, на рис. 16.2 приведена схема алгоритма вычисления некоторой функции Y . В данной схеме используется выбор, обозначаемый символом «Решение» (ромб) с пятью разветвлениями ($n = 5$).

Эквивалентный алгоритм вычисления той же функции Y , использующий несколько операторов IF–THEN–ELSE, представлен на рис. 16.3.

На данном рисунке количество операторов IF–THEN–ELSE равно четырем ($n - 1$), максимальный уровень вложенности оператора IF–THEN–ELSE равен трем ($n - 2$).

Таким образом, для схем алгоритмов, приведенных на рис. 16.2 и 16.3, $CL = 4$, $cl = 0,36$ (количество операторов программы равно 11; блоки «Начало» и «Конец» в метрике Джилба не учитываются), $CLI = 3$.

Значения метрики Маккейба для данных алгоритмов также совпадают.

Для схемы алгоритма, представленной на рис. 16.2, $Z(G) = 13 - 10 + 2 = 5$.

Для схемы алгоритма, приведенной на рис. 16.3, $Z(G) = 16 - 13 + 2 = 5$.

Для схемы алгоритма, приведенной на рис. 16.1, значения метрики Джилба $CL = 2$, $cl = 0,4$ (количество операторов программы равно 5), $CLI = 0$.

Следует отметить, что сложность программы с помощью метрики Джилба не всегда возможно посчитать на основе схемы алгоритма, так как схема алгоритма может быть представлена укрупненно. Поэтому значения метрики Джилба в программе в общем случае следует определять на основе ее исходного текста или детализированной схемы алгоритма, каждый блок которой содержит один оператор программы (как это реализовано в алгоритмах, представленных на рис. 16.1 – 16.3).

Метрика граничных значений базируется на определении скорректированной сложности вершин графа программы [67].

Пусть $G = (V, E)$ – ориентированный граф программы с единственной начальной и единственной конечной вершинами. В этом графе число входящих в вершину дуг называется *отрицательной степенью вершины*, а число исходящих из вершины дуг – *положительной степенью вершины*. С учетом этого набор вершин графа можно разбить на две группы: вершины, у которых положительная степень меньше или равна 1; вершины, у которых положительная степень больше или равна 2. Вершины первой группы называются *принимающими вершинами*, вершины второй группы – *вершинами выбора* (или предикатными вершинами, условными вершинами, вершинами отбора).

Для оценки сложности программы с использованием метрики граничных значений граф G разбивается на максимальное число подграфов, удовлетворяющих следующим условиям: вход в подграф осуществляется через вершину выбора; каждый подграф включает вершину (нижнюю границу подграфа), в которую можно попасть из любой другой вершины подграфа.

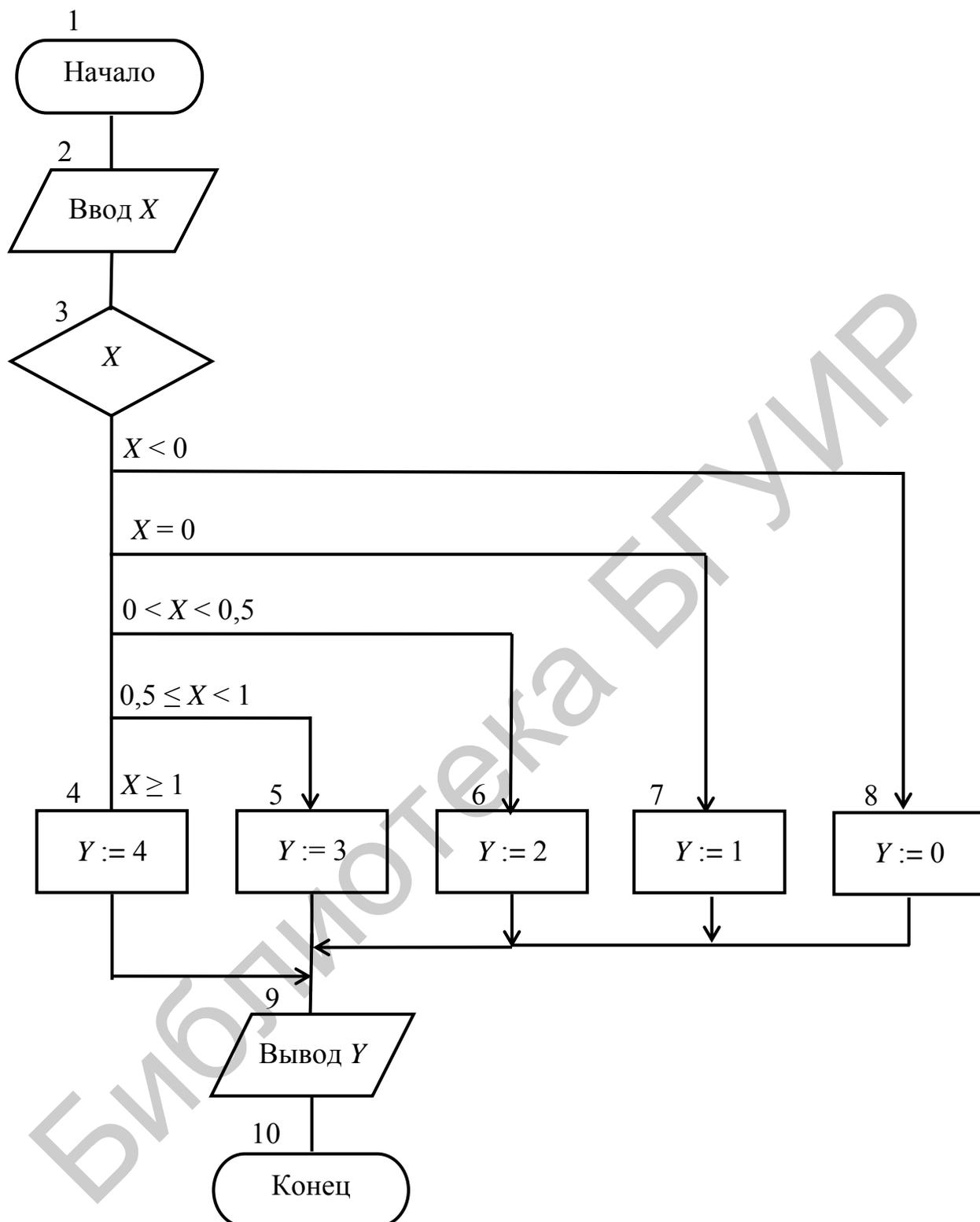


Рис. 16.2. Схема разветвляющегося алгоритма вычисления функции Y (используется символ «Решение» с многими выходами)

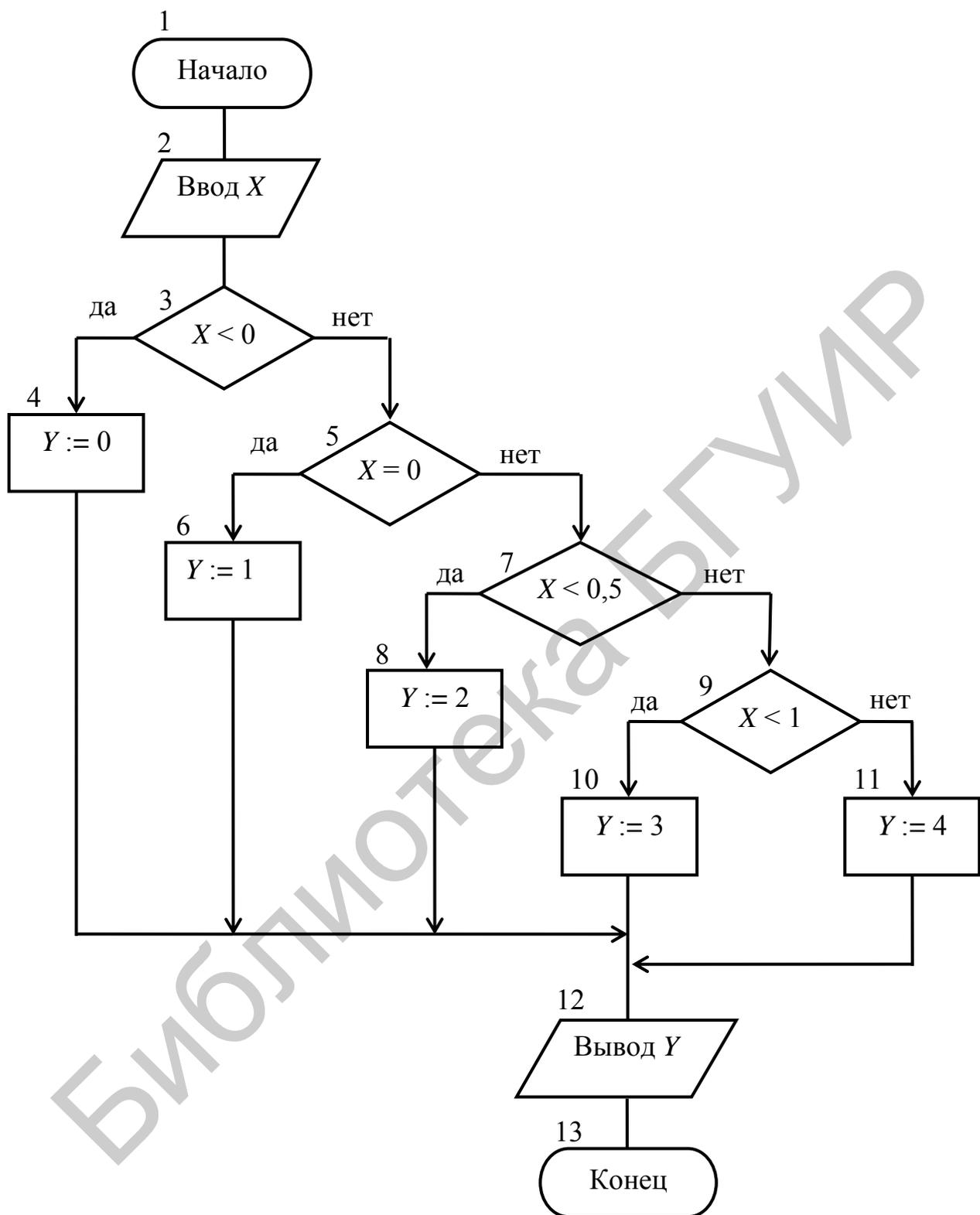


Рис. 16.3. Схема разветвляющегося алгоритма вычисления функции Y (используется символ «Решение» с двумя выходами)

Каждая вершина выбора имеет скорректированную сложность, равную числу вершин, образующих связанный с ней подграф. Каждая принимающая вершина имеет скорректированную сложность, равную 1. Конечная вершина графа имеет скорректированную сложность, равную 0.

Следует обратить внимание, что если подграф представляет собой обычное разветвление, то вершина выбора, через которую осуществляется вход в такой подграф, не является элементом подграфа. Поэтому при расчете скорректированной сложности такой вершины выбора сама эта вершина не учитывается.

Если подграф представляет собой цикл, то по одной из исходящих дуг вершины выбора осуществляется вход в тело цикла, а по другой – выход из цикла. В этом случае нижней границей подграфа является вершина перехода, находящаяся после вершины выбора на дуге выхода из цикла. После входа в подграф через вершину выбора и выполнения тела цикла для достижения нижней границы подграфа следует снова возвращаться к данной вершине выбора. Таким образом, вершина выбора, через которую осуществляется вход в циклический подграф, является элементом данного подграфа. Поэтому данную вершину выбора следует учитывать при подсчете ее скорректированной сложности.

Абсолютная граничная сложность программы S_a определяется как сумма скорректированных сложностей всех вершин графа G .

Относительная граничная сложность программы S_o определяется по формуле

$$S_o = 1 - (v - 1)/S_a ,$$

где v – общее число вершин графа программы.

В табл. 16.5 представлены свойства подграфов программы, схема алгоритма которой приведена на рис. 16.3. В табл. 16.6 приведены скорректированные сложности вершин графа данной программы. Номера вершин графа соответствуют номерам соответствующих блоков на схеме алгоритма.

Таблица 16.5

Свойства подграфов программы (к схеме алгоритма рис. 16.3)

Свойства подграфов программы	Номер вершины выбора			
	3	5	7	9
Номера вершин перехода	4, 5	6, 7	8, 9	10, 11
Номера вершин подграфа	4, 5, 6, 7, 8, 9, 10, 11	6, 7, 8, 9, 10, 11	8, 9, 10, 11	10, 11
Номер нижней границы подграфа	12	12	12	12
Скорректированная сложность вершины выбора	9	7	5	3

Таблица 16.6

Скорректированные сложности вершин графа программы
(к схеме алгоритма рис. 16.3)

Номер вершины графа программы	1	2	3	4	5	6	7	8	9	10	11	12	13	S_a
Скорректированная сложность вершины графа	1	1	9	1	7	1	5	1	3	1	1	1	0	32

В рассматриваемой схеме алгоритма все подграфы представляют собой разветвления. Поэтому вершины выбора в данные подграфы не входят и при расчете своей скорректированной сложности не учитываются (см. табл. 16.5).

Таким образом, абсолютная граничная сложность S_a программы, схема алгоритма которой приведена на рис. 16.3, равна 32. Относительная граничная сложность данной программы равна

$$S_o = 1 - (13 - 1)/32 = 0,625.$$

В табл. 16.7 представлены свойства подграфов программы, схема алгоритма которой приведена на рис. 16.1. В табл. 16.8 содержатся скорректированные сложности вершин графа данной программы.

Таблица 16.7

Свойства подграфов программы (к схеме алгоритма рис. 16.1)

Свойства подграфов программы	Номер вершины выбора	
	2	4
Номера вершин перехода	3, 4	5, 6
Номера вершин подграфа	2, 3	5, 6
Номер нижней границы подграфа	4	7
Скорректированная сложность вершины выбора	3	3

Таблица 16.8

Скорректированные сложности вершин графа программы
(к схеме алгоритма рис. 16.1)

Номер вершины графа программы	1	2	3	4	5	6	7	S_a
Скорректированная сложность вершины графа	1	3	1	3	1	1	0	10

В рассматриваемой схеме алгоритма (см. рис. 16.1) подграф с вершиной выбора 2 содержит цикл. Поэтому данная вершина выбора является элементом подграфа и участвует при подсчете своей скорректированной сложности (см. табл. 16.7).

Абсолютная граничная сложность S_a программы, схема алгоритма которой приведена на рис. 16.1, равна 10. Относительная граничная сложность данной программы равна

$$S_o = 1 - (7 - 1)/10 = 0,4.$$

В табл. 16.9 содержатся метрики сложности потока управления для программ, схемы алгоритмов которых приведены на рис. 16.1 – 16.3.

Таблица 16.9

Метрики сложности потока управления программ

Метрики сложности потока управления	Схемы алгоритмов		
	Рис. 16.1	Рис. 16.2	Рис. 16.3
Метрика Маккейба $Z(G)$	3	5	5
Абсолютная сложность программы CL по метрике Джилба	2	4	4
Относительная сложность программы cl по метрике Джилба	0,4	0,36	0,36
Максимальный уровень вложенности условного оператора CLI по метрике Джилба	0	3	3
Метрика граничных значений S_a (абсолютная граничная сложность программы)	10	14	32
Метрика граничных значений S_o (относительная граничная сложность программы)	0,4	0,357	0,625

16.3. Метрики сложности потока данных программных средств

Метрики сложности потока данных связывают сложность программы с использованием и размещением данных в этой программе. Метрики данной группы основаны на анализе исходных текстов программ.

К наиболее известным в рассматриваемой группе метрик можно отнести спен и метрику Чепина [67, 71].

Спен идентификатора – это число повторных появлений идентификатора (число появлений после его первого появления) в тексте программы. Идентификатор, встречающийся в тексте программы n раз, имеет спен, равный $n - 1$.

Величина спена связана со сложностью тестирования и отладки программы. Например, если спен идентификатора равен 10, то при трассировании программы по этому идентификатору следует ввести в текст программы 10 контрольных точек, что усложняет тестирование и отладку программы.

Метрика Чепина базируется на анализе характера использования переменных в программе.

Существуют различные варианты метрики Чепина. Ниже рассмотрен вариант (назовем данный вариант полной метрикой Чепина), в котором все множество переменных программы разбивается на четыре функциональные группы:

1. ***P*** – вводимые переменные, содержащие исходную информацию, но не модифицируемые в программе и не являющиеся управляющими переменными;
2. ***M*** – вводимые модифицируемые переменные и создаваемые внутри программы константы и переменные, не являющиеся управляющими переменными;
3. ***C*** – переменные, участвующие в управлении работой программы (управляющие переменные);
4. ***T*** – не используемые в программе («паразитные») переменные, например, вычисленные переменные, значения которых не выводятся и не участвуют в дальнейших вычислениях.

Значение метрики Чепина определяется по формуле

$$Q = a_1 p + a_2 m + a_3 c + a_4 t ,$$

где a_1, a_2, a_3, a_4 – весовые коэффициенты; p, m, c, t – количество переменных в группах ***P, M, C, T*** соответственно.

Весовые коэффициенты позволяют учитывать различное влияние на сложность программы каждой функциональной группы. Наиболее часто применяются следующие значения весовых коэффициентов: $a_1 = 1, a_2 = 2, a_3 = 3, a_4 = 0,5$. С учетом данных значений формула для определения метрики Чепина принимает вид

$$Q = p + 2m + 3c + 0,5t .$$

Помимо полной метрики Чепина распространен ее вариант, при котором анализу и разбиению на четыре группы подвергаются только переменные из списка ввода/вывода программы, т. е. те переменные, которые содержатся в списке параметров операторов ввода/вывода программы. Назовем данный вариант метрикой Чепина ввода/вывода.

В табл. 16.10, 16.11 представлены метрики сложности потока данных для программы на языке *Delphi*, вычисляющей значение функции $Y = \sin X$. Исходный текст программы приведен в примере 1 (см. подразд. 16.1).

В тексте программы (см. пример 1 из подразд. 16.1) идентификаторы впервые встречаются в разделе объявлений. Поэтому значение спена i -го

идентификатора равно количеству его появлений в разделе операторов, т. е. значению f_{2i} в метриках Холстеда.

В список переменных ввода/вывода данной программы входят переменные x , y и константа eps , являющиеся параметрами операторов ввода и вывода $Readln$ и $Writeln$. Остальные переменные (n , vs) в расчете метрики Чепина ввода/вывода не участвуют.

Таблица 16.10

Спен программы

Идентификатор	x	n	vs	y	eps	Суммарный спен программы
Спен	6	5	5	4	2	22

Таблица 16.11

Метрики Чепина программы

Вариант метрики Чепина	Полная метрика Чепина				Метрика Чепина ввода/вывода			
	P	M	C	T	P	M	C	T
Группа переменных								
Переменные, относящиеся к группе	x	y, n	vs, eps	—	x	y	eps	—
Количество переменных в группе	$p = 1$	$m = 2$	$c = 2$	$t = 0$	$p = 1$	$m = 1$	$c = 1$	$t = 0$
Метрика Чепина	$Q = 1 \cdot 1 + 2 \cdot 2 + 3 \cdot 2 + 0,5 \cdot 0 = 11$				$Q = 1 \cdot 1 + 2 \cdot 1 + 3 \cdot 1 + 0,5 \cdot 0 = 6$			

ЗАКЛЮЧЕНИЕ ПО ЧАСТИ IV

Основные понятия по метрологии и квалиметрии определены в *ISO/IEC GUIDE 99:2007(E/R)*, *PMГ 29–2013*, *ГОСТ 15467–79*.

При оценке качества программных средств и систем обычно используются четыре типа шкал: номинальная, порядковая (упорядоченная), интервальная и шкала отношений.

В серии стандартов *ISO/IEC 9126* определено, что *метрика* – это метод и шкала измерения. Метрики делятся на внутренние, внешние или метрики качества в использовании; прямые или косвенные. В данной серии определены *желательные свойства метрик* (надежность, повторяемость, воспроизводимость, доступность, показательность, корректность) и *критерии обоснованности метрик* (корреляция, трассировка, непротиворечивость, предсказуемость, селективность).

Для обеспечения возможности совместного использования различных метрик при количественной оценке качества программных продуктов метрики по возможности рекомендуется представлять в относительных единицах в виде $X = A/B$ или $X = 1 - A/B$, где X – значение метрики; A – абсолютное (измеренное) значение некоторого свойства оцениваемого продукта или документации; B – базовое значение соответствующего свойства. Из двух формул для конкретной метрики выбирается та, которая соответствует *критериям трассировки и непротиворечивости*: с увеличением относительного значения метрики значение подхарактеристики и характеристики качества должно увеличиваться. Вычисление метрик по приведенным формулам позволяет привести их относительные значения в диапазон $0 \leq X \leq 1$.

Измерению качества (квалиметрии) программных средств в стандартах серии *SQuaRE* посвящена группа стандартов *ISO/IEC 2502n*. Понятие *мера качества*, используемое в серии *SQuaRE*, эквивалентно понятию метрики качества в стандартах серии *ISO/IEC 9126*.

В *ISO/IEC 25020:2007* регламентирована эталонная модель измерений качества программного продукта (*SPQM-RM*), описывающая связь модели качества, определенных в ней характеристик и подхарактеристик качества и свойств программного продукта с соответствующими мерами качества программных средств, функциями измерений, элементами меры качества.

В *ISO/IEC 25021:2012* приведены правила для проектирования и верификации элементов мер качества. Определена связь между свойством качества, методом измерения и элементом меры качества. Метод измерения применяется для определения значений элементов мер качества, используемых в эталонной модели измерения качества продукта.

Табличное представление метода измерения включает: название элемента меры качества; целевой объект; соответствующие меры качества; метод измерения; список подсвойств свойства качества; определение каждого

подсвойства; вход для элемента меры качества; единица измерений для элемента меры качества; количественные правила; тип шкалы; контекст элемента меры качества; процессы жизненного цикла программных средств; принудительные условия измерений.

Меры качества данных в *ISO/IEC 25012:2008* определяются с помощью функций измерения *A/B* или **1 – A/B** по аналогии с определением метрик качества в серии *ISO/IEC 9126*.

Метрики (меры) сложности программных средств обычно подразделяются на *три основные группы*: метрики размера программных средств; метрики сложности потока управления программных средств; метрики сложности потока данных программных средств.

К *метрикам размера программы* относятся количество строк исходного текста программы, количество операторов программы, метрики Холстеда.

К *метрикам сложности потока управления программных средств* относятся метрика Маккейба, метрика Джилба, метрика граничных значений.

К *метрикам сложности потока данных программных средств* относятся спен и метрика Чепина.

В *ISO/IEC 9126–3:2003* метрики сложности отнесены к чисто внутренним метрикам качества программных средств. В общем случае метрики сложности рассматриваются как элементы мер качества ПС.

ВОПРОСЫ ДЛЯ САМОПРОВЕРКИ ПО ЧАСТИ IV

Раздел 13

1. В каких документах определены основные понятия в области метрологии?
2. Что такое метрология?
3. Что такое метод измерений и методика измерений по *РМГ 29–2013*?
4. Дайте определение точности измерений.
5. Дайте определение правильности измерений.
6. Дайте определение прецизионности измерений.
7. Дайте определение повторяемости измерений.
8. Дайте определение условий повторяемости измерений.
9. Дайте определение воспроизводимости измерений.
10. Дайте определение условий воспроизводимости измерений.
11. Что такое функция измерений по *РМГ 29–2013*?
12. Дайте определение шкалы измерений по *РМГ 29–2013*.
13. Дайте определение шкалы измерений по *ISO/IEC 9126–2–4:2003–2004*.
14. Дайте определение шкалы измерений по *ISO/IEC 15939:2007*.
15. Назовите и охарактеризуйте типы шкал, используемые при измерении свойств качества ПС.
16. Приведите примеры свойств качества ПС, измеряемых с помощью каждого из типов шкал.

Раздел 14

1. Перечислите и опишите желательные свойства метрик качества ПС.
2. Перечислите и опишите критерии обоснованности метрик качества ПС.
3. В каком виде по возможности должны представляться метрики и почему?
4. Напишите общие формулы, используемые для представления метрик. Какая из формул выбирается для конкретной метрики?
5. В каких единицах измерения по возможности должны представляться метрики и почему?
6. По каждой характеристике качества приведите примеры внутренних метрик качества.
7. По каждой характеристике качества приведите примеры внешних метрик.
8. По каждой характеристике качества приведите примеры метрик качества в использовании.

Раздел 15

1. Опишите эталонную модель измерения качества программного продукта, определенную в *ISO/IEC 25020:2007*.
2. Сравните понятия метрики качества из *ISO/IEC 9126-1-4:2001-2004* и меры качества из стандартов серии *SQuaRE*.
3. В каком стандарте определена концепция элементов мер качества?
4. Дайте определение элемента меры качества.
5. Как связаны элементы мер качества со свойствами качества?
6. Как связаны элементы мер качества с мерами качества?
7. Перечислите основные элементы табличного формата метода измерения по *ISO/IEC 25021:2012*.
8. Приведите примеры мер качества данных с собственной точки зрения.
9. Приведите примеры мер качества данных с собственной и системно-зависимой точек зрения.
10. Приведите примеры мер качества данных с системно-зависимой точки зрения.

Раздел 16

1. На какие группы принято подразделять метрики сложности программ?
2. Что является основой для подсчета метрик размера программ?
3. Перечислите метрики, относящиеся к метрикам размера программ.
4. Назовите базовые метрики программы, являющиеся основой метрик Холстеда.
5. Что подразумевается под операторами программы в метриках Холстеда?
6. Что такое словарь операторов, словарь операндов, словарь программы в метриках Холстеда?
7. Что такое длина и объем программы в метриках Холстеда?
8. Перечислите метрики, относящиеся к метрикам сложности потока управления программ.
9. Как определяется значение метрики Маккейба?
10. Что показывает значение метрики Маккейба?
11. Как определяется значение метрики Джилба?
12. Назовите виды и расширения метрики Джилба.
13. Как определяется значение метрики граничных значений?
14. Что такое отрицательная и положительная степени вершин графа в метрике граничных значений?
15. Дайте определение принимающих вершин и вершин выбора графа в метрике граничных значений.
16. Перечислите метрики, относящиеся к метрикам сложности потока данных программ.

17. Что такое спен идентификатора и как он рассчитывается?
18. На какой информации базируется метрика Чепина?
19. Как классифицируются переменные в метрике Чепина?
20. Назовите варианты метрики Чепина.
21. Как рассчитывается значение метрики Чепина?

Библиотека БГУИР

ЧАСТЬ V

СЕРТИФИКАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Библиотека БГУИР

17. СЕРТИФИКАЦИЯ ПРОГРАММНЫХ СРЕДСТВ

17.1. Основные понятия и определения в области технического нормирования, стандартизации и оценки соответствия

5 января 2004 г. в Республике Беларусь приняты и введены в действие Закон №262-З «О техническом нормировании и стандартизации» [64] и Закон №269-З «Об оценке соответствия требованиям технических нормативных правовых актов в области технического нормирования и стандартизации».

31 декабря 2010 г. принят Закон №228-З «О внесении изменений и дополнений в некоторые законы Республики Беларусь по вопросам оценки соответствия требованиям технических нормативных правовых актов в области технического нормирования и стандартизации», содержащий новую редакцию закона №269-З [65].

Данные законы разработаны с учетом соглашений Всемирной торговой организации (ВТО) и направлены на обеспечение единой государственной политики в области технического нормирования, стандартизации и оценки соответствия.

Закон №262-З регулирует отношения, возникающие при разработке, утверждении и применении технических требований к продукции, процессам ее разработки, производства, эксплуатации (использования), хранения, перевозки, реализации и утилизации (именуемым далее процессами ее жизненного цикла) или к оказанию услуг, определяет правовые и организационные основы технического нормирования и стандартизации.

Закон №228-З определяет правовые и организационные основы оценки соответствия объектов оценки требованиям технических нормативных правовых актов в области технического нормирования и стандартизации и направлен на совершенствование механизма оценки в области подтверждения соответствия и аккредитации с учетом международных принципов и требований Соглашения по техническим барьерам в торговле ВТО.

Система технического нормирования и стандартизации – совокупность технических нормативных правовых актов в области технического нормирования и стандартизации, субъектов технического нормирования и стандартизации, а также правил и процедур функционирования системы в целом. Системой технического нормирования и стандартизации

предусмотрены следующие *виды технических нормативных правовых актов* в области технического нормирования и стандартизации:

- технические регламенты;
- технические кодексы установившейся практики;
- государственные стандарты Республики Беларусь;
- стандарты организаций;
- технические условия.

В Законе «О техническом нормировании и стандартизации» даны определения технических нормативных правовых актов [64].

Технический регламент (ТР) – это технический нормативный правовой акт, разработанный в процессе технического нормирования, устанавливающий непосредственно и/или путем ссылки на технические кодексы установившейся практики и/или государственные стандарты Республики Беларусь *обязательные для соблюдения технические требования*, связанные с безопасностью продукции и процессов ее жизненного цикла.

Технические регламенты разрабатываются в целях защиты жизни, здоровья и наследственности человека, имущества и охраны окружающей среды, а также предупреждения действий, вводящих в заблуждение потребителей продукции и услуг относительно их назначения, качества или безопасности.

Технический кодекс установившейся практики (ТКП) – это технический нормативный правовой акт, разработанный в процессе стандартизации, содержащий *основанные на результатах установившейся практики технические требования к процессам жизненного цикла* продукции или оказанию услуг.

Технические кодексы разрабатываются с целью реализации требований технических регламентов, повышения качества процессов разработки (проектирования), производства, эксплуатации (использования), хранения, перевозки, реализации и утилизации продукции или оказания услуг.

Стандарт – это технический нормативный правовой акт, разработанный в процессе стандартизации *на основе согласия большинства заинтересованных субъектов технического нормирования и стандартизации* и содержащий технические требования к продукции, процессам ее жизненного цикла или оказанию услуг.

Государственный стандарт Республики Беларусь (СТБ) – стандарт, утвержденный Государственным комитетом по стандартизации Республики Беларусь.

Государственные стандарты основываются, как правило, на современных достижениях науки, техники, международных и межгосударственных (региональных) стандартах, правилах, нормах и рекомендациях по стандартизации, прогрессивных стандартах других государств.

Общие сведения о правилах стандартизации, разработке международных, региональных и национальных стандартов приведены в части I (см. разд. 1 – 5).

Стандарт организации – стандарт, утвержденный юридическим лицом или индивидуальным предпринимателем.

Технические требования стандартов организаций распространяются только на лица, их утвердившие. Стандарты организаций не разрабатываются на продукцию, реализуемую иным юридическим или физическим лицам, или на оказываемые им услуги.

Технические условия (ТУ) – это технический нормативный правовой акт, разработанный в процессе стандартизации, утвержденный юридическим лицом или индивидуальным предпринимателем и *содержащий технические требования к конкретным типу, марке, модели, виду реализуемой ими продукции* или оказываемой услуге, включая правила приемки и методы контроля.

Технические кодексы установившейся практики, государственные стандарты, стандарты организаций и технические условия не должны противоречить требованиям технических регламентов.

В законе [64] определено, что *обязательные* требования устанавливаются в технических регламентах, принимаемых органами власти, а стандарты применяются в *добровольном* порядке.

В области оценки соответствия приняты следующие **основные термины и их определения** [65].

Аккредитация – вид оценки соответствия, результатом осуществления которого является подтверждение компетентности юридического лица Республики Беларусь или иностранного юридического лица в выполнении работ по подтверждению соответствия или проведении испытаний объектов оценки соответствия.

Аккредитованная испытательная лаборатория (центр) – юридическое лицо Республики Беларусь или иностранное юридическое лицо, аккредитованное для проведения испытаний объектов оценки соответствия в определенной области аккредитации.

Аккредитованный орган по сертификации – юридическое лицо Республики Беларусь или иностранное юридическое лицо, аккредитованное для выполнения работ по подтверждению соответствия в определенной области аккредитации.

Аттестат аккредитации – документ, удостоверяющий компетентность юридического лица Республики Беларусь или иностранного юридического лица в выполнении работ по подтверждению соответствия или проведении испытаний объектов оценки соответствия в определенной области аккредитации.

Декларация о соответствии – документ, в котором изготовитель (продавец) удостоверяет соответствие продукции требованиям технических нормативных правовых актов в области технического нормирования и стандартизации.

Декларирование соответствия – подтверждение соответствия, осуществляемое изготовителем (продавцом).

Национальная система аккредитации Республики Беларусь – совокупность уполномоченных государственных органов, органа по аккредитации, аккредитованных органов по сертификации и аккредитованных испытательных лабораторий (центров), нормативных правовых актов, в том числе технических нормативных правовых актов, в области технического нормирования и стандартизации, определяющих процедуры аккредитации и функционирование системы в целом.

Национальная система подтверждения соответствия Республики Беларусь – совокупность уполномоченных государственных органов, аккредитованных органов по сертификации и аккредитованных испытательных лабораторий (центров), нормативных правовых актов, в том числе технических нормативных правовых актов, в области технического нормирования и стандартизации, определяющих процедуры подтверждения соответствия и функционирование системы в целом.

Область аккредитации – сфера деятельности, в которой аккредитованному органу по сертификации или аккредитованной испытательной лаборатории (центру) предоставлено право на выполнение работ по подтверждению соответствия или проведение испытаний объектов оценки соответствия.

Оценка соответствия – деятельность по определению соответствия объектов оценки соответствия требованиям технических нормативных правовых актов в области технического нормирования и стандартизации.

Подтверждение соответствия – вид оценки соответствия, результатом осуществления которого является документальное удостоверение соответствия объекта оценки соответствия требованиям технических нормативных правовых актов в области технического нормирования и стандартизации.

Сертификат компетентности – документ, удостоверяющий профессиональную компетентность персонала в выполнении определенных работ (оказании определенных услуг).

Сертификат соответствия – документ, удостоверяющий соответствие объекта оценки соответствия требованиям технических нормативных правовых актов в области технического нормирования и стандартизации.

Сертификация – подтверждение соответствия, осуществляемое аккредитованным органом по сертификации.

Система управления качеством – часть общей системы управления, включающая организационную структуру, планирование, ответственность, методы, процедуры, процессы, ресурсы, необходимые для обеспечения качества продукции (работ, услуг).

Схема подтверждения соответствия – совокупность и последовательность действий, результаты которых рассматриваются в качестве доказательств соответствия объекта оценки соответствия требованиям технических нормативных правовых актов в области технического нормирования и стандартизации.

17.2. Общие сведения об оценке соответствия в Республике Беларусь

Оценка соответствия в Беларуси выполняется на основе *Закона Республики Беларусь от 31 декабря 2010 г. №228-З «О внесении изменений и дополнений в некоторые законы Республики Беларусь по вопросам оценки соответствия требованиям технических нормативных правовых актов в области технического нормирования и стандартизации»*, содержащего новую редакцию *Закона «Об оценке соответствия требованиям технических нормативных правовых актов в области технического нормирования и стандартизации»* [65].

Оценка соответствия осуществляется *в целях*:

- обеспечения защиты жизни, здоровья и наследственности человека, имущества и охраны окружающей среды;
- повышения конкурентоспособности продукции (работ, услуг);
- обеспечения энерго- и ресурсосбережения;
- создания благоприятных условий для обеспечения свободного перемещения продукции на внутреннем и внешнем рынках, а также для участия в международном экономическом, научно-техническом сотрудничестве и международной торговле.

Принципами оценки соответствия являются:

- гармонизация с международными и межгосударственными (региональными) подходами в области оценки соответствия;
- обеспечение идентичности процедур оценки соответствия отечественных и иностранных объектов оценки соответствия;
- соблюдение требований конфиденциальности сведений, полученных при выполнении работ по оценке соответствия.

Объектами оценки соответствия при подтверждении соответствия являются:

- продукция;
- процессы жизненного цикла продукции (разработка, производство, эксплуатация, хранение, перевозка, реализация и утилизация);
- выполнение работ;
- оказание услуг;
- система управления качеством;
- система управления окружающей средой;
- система управления безопасностью продукции;
- система управления охраной труда;
- профессиональная компетентность персонала в выполнении определенных работ (оказании определенных услуг);

• иные объекты, в отношении которых установлены требования технических нормативных правовых актов в области технического нормирования и стандартизации.

Субъектами оценки соответствия являются:

- уполномоченные государственные органы;
- орган по аккредитации;
- аккредитованные органы по сертификации;
- аккредитованные испытательные лаборатории (центры);
- заявители на аккредитацию;
- заявители на подтверждение соответствия.
- заявители на проведение испытаний;
- изготовители (продавцы).

Законом [65] предусмотрено *два вида оценки соответствия*: аккредитация и подтверждение соответствия (рис. 17.1).

К *документам* об оценке соответствия относятся:

- аттестат аккредитации;
- сертификат соответствия;
- декларация о соответствии;
- сертификат компетентности.



Рис. 17.1. Виды оценки соответствия

Систему подтверждения соответствия в Республике Беларусь иллюстрируют рис. 17.2 и 17.3 [81].

Национальным органом по оценке соответствия Республики Беларусь является Государственный комитет по стандартизации, метрологии и сертификации при Совете Министров Республики Беларусь (Госстандарт). К основным органам по сертификации относится Белорусский государственный институт стандартизации и сертификации (БелГИСС).

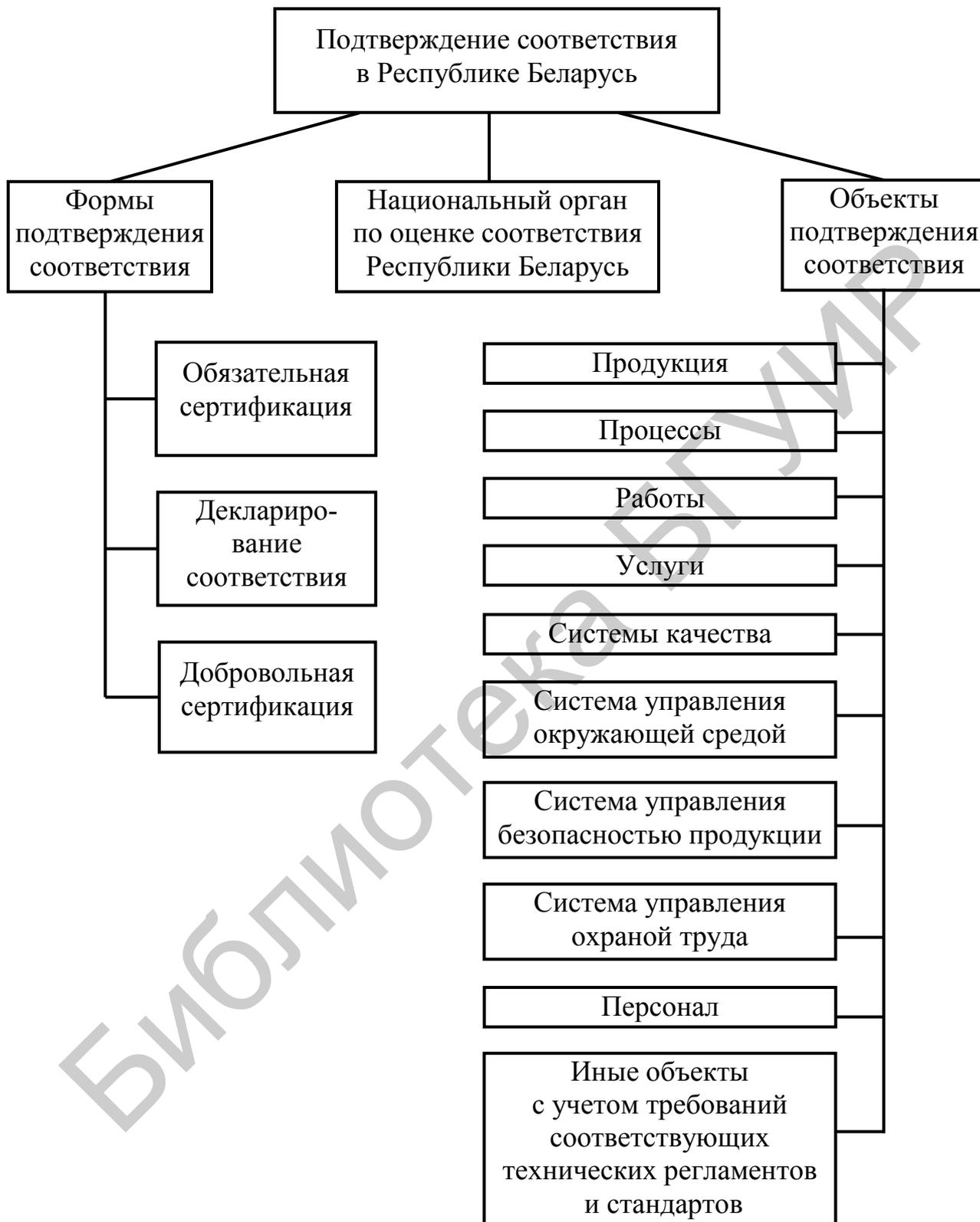


Рис. 17.2. Подтверждение соответствия в Республике Беларусь

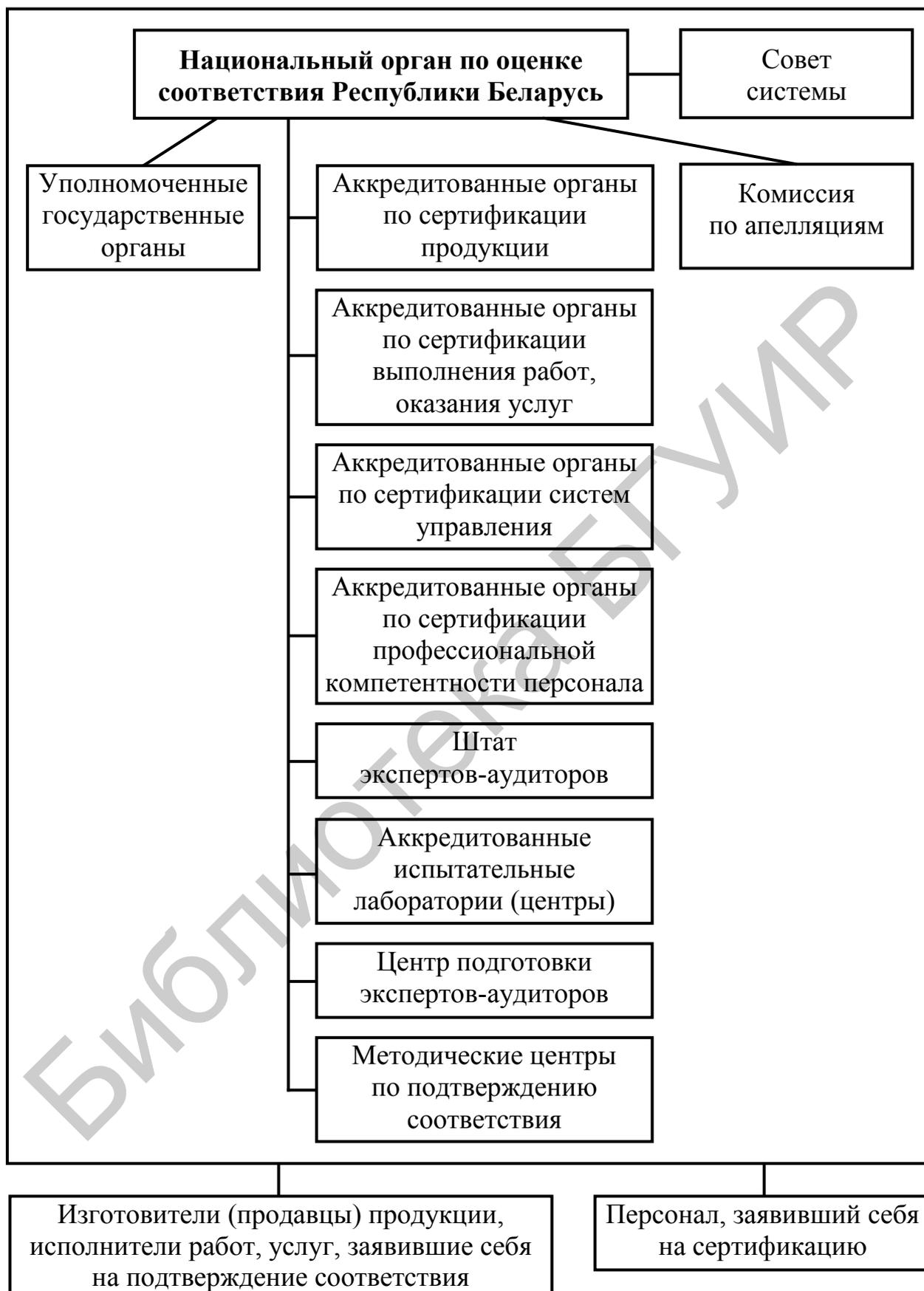


Рис. 17.3. Структура Национальной системы подтверждения соответствия Республики Беларусь

Подтверждение соответствия осуществляется *в целях*:

- удостоверения соответствия объектов оценки требованиям технических нормативных правовых актов в области технического нормирования и стандартизации;
- обеспечения защиты жизни, здоровья и наследственности человека, имущества и охраны окружающей среды;
- предупреждения действий, вводящих в заблуждение потребителей продукции (работ, услуг) относительно ее назначения, качества и безопасности.

Принципами подтверждения соответствия являются:

- открытость и доступность процедур подтверждения соответствия;
- независимость аккредитованных органов по сертификации и аккредитованных испытательных лабораторий (центров) от заявителей на подтверждение соответствия и заявителей на проведение испытаний объектов оценки соответствия;
- минимизация сроков проведения и затрат заявителей на прохождение процедур подтверждения соответствия;
- защита имущественных интересов заявителей, в том числе путем соблюдения требований конфиденциальности сведений, полученных в процессе прохождения процедур подтверждения соответствия;
- недопустимость ограничения конкуренции аккредитованных органов по сертификации и аккредитованных испытательных лабораторий (центров) при выполнении работ по подтверждению соответствия и проведении испытаний объектов оценки соответствия.

Порядок выполнения работ по подтверждению соответствия устанавливается в Национальной системе подтверждения соответствия Республики Беларусь. Положительные результаты подтверждения соответствия удостоверяются одним из следующих *документов*:

- *сертификатом соответствия* или *сертификатом компетентности*, выдаваемым аккредитованным органом по сертификации заявителю на подтверждение соответствия;
- *декларацией о соответствии*, принятой заявителем на подтверждение соответствия и зарегистрированной аккредитованным органом по сертификации.

Подтверждение соответствия может носить ***обязательный*** или ***добровольный характер***.

Обязательное подтверждение соответствия осуществляется в формах:

- обязательной сертификации;
- декларирования соответствия.

Добровольное подтверждение соответствия осуществляется только в форме добровольной сертификации.

Обязательному подтверждению соответствия подлежат объекты оценки, включенные в ***перечень*** продукции, услуг, персонала и иных объектов оценки

соответствия, подлежащих обязательному подтверждению соответствия в Республике Беларусь (Постановление государственного комитета по стандартизации Республики Беларусь от 16 декабря 2008 г. №60 с изменениями от 19 сентября 2014 г. №42). В данном перечне устанавливаются:

- виды продукции, работ, услуг и иные объекты оценки соответствия, подлежащие обязательному подтверждению соответствия;

- технические нормативные правовые акты в области технического нормирования и стандартизации, на соответствие которым проводится обязательное подтверждение соответствия;

- формы обязательного подтверждения соответствия.

Критериями при формировании перечня являются:

- потенциальная опасность продукции, работ, услуг и функционирования других объектов оценки соответствия для жизни, здоровья и наследственности человека, имущества и окружающей среды;

- предупреждение действий, вводящих в заблуждение потребителей продукции (работ, услуг) относительно ее назначения, качества и безопасности.

Обязательная сертификация осуществляется аккредитованным органом по сертификации на основе договора с заявителем. Схемы подтверждения соответствия при обязательной сертификации зависят от видов объектов оценки. Данные схемы должны устанавливаться соответствующим техническим регламентом. Если схемой установлена необходимость проведения испытаний продукции, то они проводятся аккредитованной испытательной лабораторией (центром) на основе договора с заявителем.

Декларирование соответствия осуществляется заявителем на подтверждение соответствия одним из следующих способов:

- путем принятия декларации о соответствии на основании собственных доказательств;

- путем принятия декларации о соответствии на основании собственных доказательств и доказательств, полученных с участием аккредитованного органа по сертификации или аккредитованной испытательной лаборатории.

Схемы подтверждения соответствия при декларировании соответствия различных видов продукции должны устанавливаться соответствующим техническим регламентом. Если схемой установлена необходимость проведения испытаний продукции, то они проводятся аккредитованной испытательной лабораторией (центром) на основе договора с заявителем.

Добровольная сертификация осуществляется аккредитованным органом по сертификации по инициативе заявителя на подтверждение соответствия на основе договора. При добровольной сертификации заявитель самостоятельно выбирает технические нормативные правовые акты, на соответствие которым осуществляется сертификация, и определяет номенклатуру контролируемых показателей. В номенклатуру показателей обязательно должны включаться показатели безопасности, если они установлены в соответствующих технических нормативных правовых актах.

Формы и процедуры обязательного подтверждения соответствия, а также процедуры добровольной сертификации устанавливаются в документах Национальной системы подтверждения соответствия Республики Беларусь (НСПС РБ).

Примеры *основополагающих документов НСПС РБ* [80]:

1. *ТКП 5.1.01–2012 (03220). Национальная система подтверждения соответствия Республики Беларусь. Основные положения.* Данный ТКП устанавливает основные положения организации работ по подтверждению соответствия, структуру НСПС РБ, функции ее органов и является основополагающим в комплексе технических кодексов и стандартов, обеспечивающих функционирование НСПС РБ. Данный ТКП обязателен для всех субъектов оценки соответствия, участвующих в работах по подтверждению соответствия объектов оценки соответствия.

2. *ТКП 5.1.02–2012 (03220). Национальная система подтверждения соответствия Республики Беларусь. Сертификация продукции. Основные положения.* Данный ТКП устанавливает основные положения, регламентирующие процедуры сертификации продукции в НСПС РБ. На основе ТКП при необходимости разрабатываются и применяются совместно с ним технические кодексы, устанавливающие процедуры сертификации групп однородной продукции, учитывающие особенности ее производства, испытаний, поставок и эксплуатации. Положения данного ТКП применяются в случае, если процедуры сертификации продукции не установлены в техническом регламенте. ТКП обязателен для всех субъектов оценки соответствия, участвующих в сертификации продукции.

3. *ТКП 5.1.03–2012 (03220). Национальная система подтверждения соответствия Республики Беларусь. Декларирование соответствия продукции. Основные положения.* ТКП устанавливает основные положения, регламентирующие проведение декларирования соответствия продукции в НСПС РБ. ТКП обязателен для субъектов оценки соответствия, участвующих в декларировании соответствия продукции.

4. *ТКП 5.1.04–2012 (03220). Национальная система подтверждения соответствия Республики Беларусь. Сертификация выполнения работ, оказания услуг. Основные положения.* Данный ТКП устанавливает основные положения, регламентирующие сертификацию выполнения работ, оказания услуг в НСПС РБ. На основе этого ТКП при необходимости могут разрабатываться и применяться совместно с ним технические кодексы, устанавливающие процедуры сертификации определенного вида работ, услуг, учитывающие особенности процесса их выполнения или оказания. ТКП обязателен для всех субъектов оценки соответствия, участвующих в сертификации работ и услуг.

5. *ТКП 5.1.05–2012 (03220). Национальная система подтверждения соответствия Республики Беларусь. Сертификация систем управления. Основные положения.* Данный ТКП устанавливает общие требования к процедуре проведения работ по сертификации систем управления (систем

менеджмента качества, систем управления охраной труда, систем управления окружающей средой, систем менеджмента информационной безопасности, систем обеспечения качества на основе принципов надлежащей производственной практики (GMP) и др.) на соответствие требованиям технических нормативных правовых актов в области технического нормирования и стандартизации на системы управления в НСПС РБ. ТКП обязателен для всех субъектов оценки соответствия, участвующих в сертификации систем управления.

6. *ТКП 5.1.06–2012 (03220). Национальная система подтверждения соответствия Республики Беларусь. Сертификация профессиональной компетентности персонала. Основные положения.* Данный ТКП устанавливает общие требования к процедурам проведения сертификации профессиональной компетентности персонала, осуществляющего конкретные виды работ (услуг) в определенной области деятельности, в НСПС РБ. ТКП обязателен для субъектов оценки соответствия, участвующих в сертификации профессиональной компетентности персонала.

7. *ТКП 5.1.07–2007 (03220). Национальная система подтверждения соответствия Республики Беларусь. Экологическая сертификация. Порядок экологической сертификации услуг в области охраны окружающей среды.* Данный ТКП устанавливает требования к порядку экологической сертификации услуг в области охраны окружающей среды в рамках НСПС РБ. ТКП применяется при проведении добровольной экологической сертификации услуг в области охраны окружающей среды.

9. *ТКП 5.1.15–2008 (03220). Национальная система подтверждения соответствия Республики Беларусь. Экологическая сертификация. Основные положения.* Данный ТКП устанавливает основные положения экологической сертификации в НСПС РБ и является основополагающим в комплексе ТНПА, обеспечивающих организацию и проведение работ по экологической сертификации. ТКП учитывает требования *СТБ ИСО 14001–2005, СТБ ИСО 14004–2005, СТБ ИСО 14031–2003, СТБ ИСО 19011–2003, ИСО/МЭК Руководства 2:1996, ИСО/МЭК Руководства 60:1994, ИСО/МЭК Руководства 65:1996, ИСО/МЭК Руководства 66:1999.*

17.3. Организация сертификации программных средств

К программным средствам может применяться такая форма подтверждения соответствия, как сертификация.

Отличием процесса сертификации программных средств от сертификации других видов продукции является высокая сложность. Связано это в первую очередь с невозможностью провести исчерпывающее

тестирование сертифицируемых ПС, имеющих, как правило, достаточно большой объем.

Существует два вида сертификации программных средств – обязательная и добровольная. Результатом положительного прохождения сертификации является выдача **сертификата соответствия**.

Обязательной сертификации подвергаются программные средства, выполняющие особо ответственные функции, в которых недостаточное качество и ошибки представляют потенциальную опасность для жизни, здоровья и наследственности человека, имущества и окружающей среды.

Добровольная сертификация применяется для удостоверения качества программных средств с целью повышения их конкурентоспособности и создания благоприятных условий для обеспечения свободного перемещения продукции на внутреннем и внешнем рынках.

Методология принятия решений о допустимости выдачи сертификата на программное средство основывается на оценке степени его соответствия действующим и/или специально разработанным документам:

- соответствующим международным, межгосударственным и национальным стандартам;
- стандартам на сопровождающую документацию;
- нормативным и эксплуатационным документам на конкретное программное средство: техническим условиям, техническим описаниям, спецификациям требований и другим регламентирующим документам.

Процесс сертификации программных средств включает [68]:

- 1) анализ и выбор разработчиком или заказчиком компетентного в данной области органа по сертификации;
- 2) подачу заявителем заявки на сертификацию в орган сертификации;
- 3) принятие органом сертификации решения по заявке, выбор испытательной лаборатории и схемы сертификации;
- 4) обзор и идентификацию версий ПС, подлежащих испытаниям;
- 5) сертификационные испытания ПС испытательной лабораторией;
- 6) анализ полученных результатов и принятие решения лабораторией или органом сертификации о возможности выдачи заявителю сертификата соответствия;
- 7) выдачу органом сертификации заявителю сертификата на сертифицированную версию программного средства;
- 8) осуществление инспекционного контроля органом сертификации за сертифицированным программным средством;
- 9) проведение заявителем корректирующих мероприятий при нарушении соответствия программного средства установленным требованиям;
- 10) регистрацию и публикацию информации о результатах сертификации программного средства.

Международными стандартами определены состав и содержание документов, поддерживающих организацию сертификации программных средств.

В состав документов заявителя входят:

- заявка на проведение сертификации;
- проект договора на сертификационные испытания;
- программное средство;
- комплект технической документации, включая техническое задание или спецификацию требований и эксплуатационную документацию на программное средство и его компоненты.

В состав документов органа сертификации входят:

- регистрационная карта сертифицируемого объекта;
- заключение по результатам рассмотрения заявки на сертификацию;
- задание на проведение сертификации и требования к ней;
- план сертификационных испытаний;
- заключение по результатам сертификационных испытаний;
- оформленный сертификат соответствия.

В состав документов испытательной лаборатории входят:

- характеристики объекта испытаний;
- комплект технической документации;
- действующие международные, государственные и ведомственные стандарты на разработку и испытания программных средств и на техническую документацию;
- программа сертификационных испытаний по всем требованиям технического задания и положениям документации;
- методика сертификационных испытаний по каждому разделу требований технического задания и документации;
- инструментальные средства и методы испытаний;
- регистрационная карта сертификационных испытаний;
- протоколы сертификационных испытаний;
- отчет о проведенных испытаниях и предложение о выдаче сертификата.

Базовыми стандартами, используемыми испытательной лабораторией в процессе оценки качества программного средства и его соответствия требованиям к качеству, являются стандарты *СТБ ИСО/МЭК 9126–2003*, *ISO/IEC 25010:2011*, *ISO/IEC 25040:2011*, *ISO/IEC 25041:2012* и др. [49, 50]. Модели качества и общие процессы оценки, регламентированные в данных стандартах, рассмотрены в разд. 10 и 12.

ЗАКЛЮЧЕНИЕ ПО ЧАСТИ V

В настоящее время в области технического нормирования, стандартизации и оценки соответствия в Республике Беларусь действуют Закон №262-З от 5.01.2004 г. и Закон №228-З от 31.12.2010 г. В Законе №262-З определены следующие *виды технических нормативных правовых актов* в области технического нормирования и стандартизации: технические регламенты, технические кодексы установившейся практики, государственные стандарты Республики Беларусь, технические условия.

Законом №228-З предусмотрено два вида оценки соответствия: аккредитация и подтверждение соответствия.

Подтверждение соответствия может носить обязательный или добровольный характер. *Обязательное подтверждение соответствия* осуществляется в формах обязательной сертификации или декларирования соответствия. *Добровольное подтверждение соответствия* осуществляется в форме добровольной сертификации. *Форма, правила и процедуры* обязательного подтверждения соответствия, *правила и процедуры* добровольной сертификации устанавливаются в документах Национальной системы подтверждения соответствия Республики Беларусь (НСПС РБ).

К программным средствам (ПС) может применяться такая форма подтверждения соответствия, как сертификация. Существует два вида *сертификации ПС* – обязательная и добровольная. *Обязательной сертификации* подвергаются ПС, выполняющие ответственные функции, в которых недостаточное качество и ошибки представляют потенциальную опасность для жизни и здоровья человека, имущества и окружающей среды. *Добровольная сертификация* применяется для удостоверения качества ПС с целью повышения их конкурентоспособности на внутреннем и внешнем рынках. Результатом положительного прохождения сертификации является выдача *сертификата соответствия*.

Процесс сертификации ПС включает: анализ и выбор разработчиком или заказчиком компетентного в данной области органа по сертификации; подачу заявителем заявки на сертификацию в орган сертификации; принятие органом сертификации решения по заявке, выбор испытательной лаборатории и схемы сертификации; идентификацию версий ПС, подлежащих испытаниям; сертификационные испытания ПС испытательной лабораторией; анализ полученных результатов и принятие решения лабораторией или органом сертификации о возможности выдачи заявителю сертификата соответствия; выдачу органом сертификации заявителю сертификата на сертифицированную версию ПС; осуществление инспекционного контроля органом сертификации за сертифицированным ПС; проведение заявителем корректирующих мероприятий при нарушении соответствия ПС установленным требованиям; регистрацию и публикацию информации о результатах сертификации ПС.

ВОПРОСЫ ДЛЯ САМОПРОВЕРКИ ПО ЧАСТИ V

Раздел 17

1. Назовите законы в области технического нормирования, стандартизации и оценки соответствия, действующие в Республике Беларусь.
2. Перечислите и дайте определения технических нормативных правовых актов в области технического нормирования и стандартизации, существующих на территории Республики Беларусь, в соответствии с действующим законодательством.
3. Что такое оценка соответствия?
4. Назовите виды оценки соответствия.
5. Назовите цели оценки соответствия.
6. Перечислите принципы оценки соответствия.
7. Что может являться объектами оценки соответствия?
8. Перечислите субъекты оценки соответствия.
9. Перечислите виды документов об оценке соответствия.
10. Перечислите принципы подтверждения соответствия.
11. Какой характер может носить подтверждение соответствия в Республике Беларусь?
12. Назовите формы подтверждения соответствия в Республике Беларусь и охарактеризуйте каждую из них.
13. Что является критерием при формировании перечня продукции, услуг, персонала и других объектов оценки соответствия, подлежащих обязательному подтверждению соответствия в Республике Беларусь?
14. Какие документы относятся к основополагающим документам Национальной системы подтверждения соответствия Республики Беларусь? Приведите примеры.
15. Назовите форму подтверждения соответствия, которая может применяться к программным средствам. Перечислите ее виды.
16. Перечислите документы, на степень соответствия которым может оцениваться программное средство при его сертификации.
17. Опишите процесс сертификации программных средств.
18. Перечислите документы заявителя, необходимые для организации сертификации программных средств.
19. Перечислите документы органа сертификации, поддерживающие организацию сертификации программных средств.
20. Перечислите документы испытательной лаборатории, поддерживающие организацию сертификации программных средств.
21. Назовите базовые стандарты, используемые испытательной лабораторией в качестве основы процесса оценки качества программного средства и его соответствия требованиям к качеству.

ЛИТЕРАТУРА

1. ГОСТ 1.2–2009. Межгосударственная система стандартизации. Стандарты межгосударственные, правила и рекомендации по межгосударственной стандартизации. Правила разработки, принятия, применения, обновления и отмены. – Введ. 2010–05–01. – М. : Стандартиформ, 2010.
2. ГОСТ 1.3–2008. Межгосударственная система стандартизации. Правила и методы принятия международных и региональных стандартов в качестве межгосударственных стандартов. – Введ. 2010–05–01. – М. : Стандартиформ, 2010.
3. ГОСТ 1.5–2001. Межгосударственная система стандартизации. Стандарты межгосударственные, правила и рекомендации по межгосударственной стандартизации. Общие требования к построению, изложению, оформлению, содержанию и обозначению. – Введ. 2002–09–01. – М. : Стандартиформ, 2001.
4. ГОСТ ISO 9001–2011. Системы менеджмента качества. Требования. – Введ. 2013–01–01. – М. : Стандартиформ, 2012.
5. ГОСТ ИСО/МЭК 12207–2002. Информационная технология. Процессы жизненного цикла программных средств. – Введ. 2008–02–01. – М. : Стандартиформ, 2003.
6. ГОСТ 15467–79. Управление качеством продукции. Основные понятия. Термины и определения. – Введ. – 1979–07–01. – Минск : Госстандарт, 2008.
7. ГОСТ 19.701–90. Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения. – Введ. 1992–01–01. – М. : Изд-во стандартов, 1991.
8. ГОСТ 28195–89. Оценка качества программных средств. Общие положения. – Введ. 1990–07–01. – М. : Изд-во стандартов, 1990.
9. ГОСТ 28195–99. Оценка качества программных средств. Общие положения. – Введ. 2000–03–01. Минск : Межгосударственный совет по стандартизации, метрологии и сертификации, 1999.
10. ГОСТ 28806–90. Качество программных средств. Термины и определения. – Введ. 1992–01–01. – Минск : Госстандарт, 2005.
11. ГОСТ Р ИСО/МЭК 12207–2010. Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств. – Введ. 2012–03–01. – М. : Стандартиформ, 2011.
12. РМГ 29-2013. Государственная система обеспечения единства измерений. Метрология. Основные термины и определения. – Введ. 2015–01–01. – М. : Стандартиформ, 2014.
13. СТБ 1500–2004. Техническое нормирование и стандартизация. Термины и определения. – Введ. 2005–07–01. – Минск : Госстандарт, 2004.

14. СТБ ИСО/МЭК 12207–2003. Информационные технологии. Процессы жизненного цикла программных средств. – Введ. 2003–03–19. – Минск : Госстандарт, 2003.
15. СТБ ИСО/МЭК 9126–2003. Информационные технологии. Оценка программной продукции. Характеристики качества и руководства по их применению. – Введ. 2003–11–01. – Минск : Госстандарт, 2003.
16. СТБ ECSS–E–ST–40C–2014. Космическая техника. Разработка продукции. Общие требования к программному обеспечению. – Введ. 2015–01–15. – Минск : Госстандарт, 2014.
17. СТБ ECSS–Q–ST–80C–2014. Космическая техника. Обеспечение качества продукции. Гарантия качества программного обеспечения. – Введ. 2014–10–01. – Минск : Госстандарт, 2014.
18. СТБ ECSS–Q–NB–80–03A–2014. Космическая техника. Обеспечение качества продукции. Надежность и безопасность программного обеспечения. – Введ. 2014–10–01. – Минск : Госстандарт, 2014.
19. СТБ ISO/IEC 25000–2009. Разработка программного обеспечения. Требования к качеству и оценка программного продукта (SQuaRE). Руководство по SQuaRE. – Введ. 2010–01–01. – Минск : Госстандарт, 2009.
20. СТБ ISO/IEC 25001–2009. Разработка программного обеспечения. Требования к качеству и оценка программного продукта (SQuaRE). Планирование и управление. – Введ. 2010–01–01. – Минск : Госстандарт, 2009.
21. ТКП 1.2–2004. Система технического нормирования и стандартизации Республики Беларусь. Правила разработки государственных стандартов. – Введ. 2005–01–01. – Минск : Госстандарт, 2004.
22. ТКП 1.7–2007. Система технического нормирования и стандартизации Республики Беларусь. Правила разработки межгосударственных стандартов. – Введ. 2007–10–01. – Минск : Госстандарт, 2007.
23. ТКП 1.9–2007. Система технического нормирования и стандартизации Республики Беларусь. Правила принятия международных, региональных и национальных стандартов других государств в качестве государственных стандартов. – Введ. 2007–09–01. – Минск : Госстандарт, 2007.
24. ECSS–E–ST–40C. Космическая инженерия. Программное обеспечение. – Введ. 2009–03–06. – Нидерланды : ESTEC, 2009.
25. ECSS–Q–ST–80C. Обеспечение космического продукта. Обеспечение программного продукта. – Введ. 2009–03–06. – Нидерланды : ESTEC, 2009.
26. ISO/IEC 12207:1995. Информационная технология. Процессы жизненного цикла программных средств. – Введ. 1995–08–01. – Женева : ISO/IEC, 1995.
27. ISO/IEC 12207:2008. Системная и программная инженерия. Процессы жизненного цикла программных средств. – Введ. 2008–02–01. – Нью-Йорк : ISO/IEC-IEEE, 2008.
28. ISO/IEC 14598–1:1999. Информационная технология. Оценка программного продукта. Ч. 1 : Общий обзор. – Введ. 1999–04–15. – Женева : ISO/IEC, 1999.

29. ISO/IEC 14598-2:2000. Программная инженерия. Оценка продукта. Ч. 2 : Планирование и управление. – Введ. 2000-02-01. – Женева : ISO/IEC, 2000.
30. ISO/IEC 14598-3:2000. Программная инженерия. Оценка продукта. Ч. 3 : Процесс для разработчиков. – Введ. 2000-02-01. – Женева : ISO/IEC, 2000.
31. ISO/IEC 14598-4:1999. Программная инженерия. Оценка продукта. Ч. 4 : Процесс для заказчиков. – Введ. 1999-10-01. – Женева : ISO/IEC, 1999.
32. ISO/IEC 14598-5:1998. Информационная технология. Оценка программного продукта. Ч. 5 : Процесс для оценщиков. – Введ. 1998-07-01. – Женева : ISO/IEC, 1998.
33. ISO/IEC 14598-6:2001. Программная инженерия. Оценка продукта. Ч. 6 : Документация модулей оценки. – Введ. 2001-06-01. – Женева : ISO/IEC, 2001.
34. ISO/IEC 15288:2008. Системная и программная инженерия. Процессы жизненного цикла системы. – Введ. 2008-02-01. – Нью-Йорк : ISO/IEC-IEEE, 2008.
35. ISO/IEC 15939:2007. Программная инженерия. Процесс измерения программных средств. – Введ. 2007-08-01. – Женева : ISO/IEC, 2007.
36. ISO/IEC/IEEE24765:2009. Системная и программная инженерия. Словарь. – Введ. 2010-12-15. – Женева : ISO/IEC, 2011.
37. ISO/IEC 25000:2005. Программная инженерия. Требования к качеству и оценка программных продуктов (SQuaRE). Руководство по SQuaRE. – Введ. 2005-08-01. – Женева : ISO/IEC, 2005.
38. ISO/IEC 25000:2014. Системная и программная инженерия. Требования к качеству и оценка систем и программных средств (SQuaRE). Руководство по SQuaRE. – Введ. 2014-03-15. – Женева : ISO/IEC, 2014.
39. ISO/IEC 25001:2007. Программная инженерия. Требования к качеству и оценка программных продуктов (SQuaRE). Планирование и управление. – Введ. 2007-02-01. – Женева : ISO/IEC, 2007.
40. ISO/IEC 25001:2014. Системная и программная инженерия. Требования к качеству и оценка систем и программных средств (SQuaRE). Планирование и управление. – Введ. 2014-03-15. – Женева : ISO/IEC, 2014.
41. ISO/IEC 25010:2011. Системная и программная инженерия. Требования к качеству и оценка систем и программных средств (SQuaRE). Модели качества систем и программных средств. – Введ. 2011-03-01. – Женева : ISO/IEC, 2011.
42. ISO/IEC 25012:2008. Программная инженерия. Требования к качеству и оценка программных продуктов (SQuaRE). Модель качества данных. – Введ. 2008-12-15. – Женева : ISO/IEC, 2008.
43. ISO/IEC 25020:2007. Программная инженерия. Требования к качеству и оценка программных продуктов (SQuaRE). Эталонная модель измерений. – Введ. 2007-05-15. – Женева : ISO/IEC, 2007.

44. ISO/IEC 25021:2012. Системная и программная инженерия. Требования к качеству и оценка систем и программных средств (SQuaRE). Элементы мер качества. – Введ. 2012–11–01. – Женева : ISO/IEC, 2012.

45. ISO/IEC 25022. Системная и программная инженерия. Требования к качеству и оценка систем и программных средств (SQuaRE). Измерение качества в использовании.

46. ISO/IEC 25023. Системная и программная инженерия. Требования к качеству и оценка систем и программных средств (SQuaRE). Измерение качества систем и программных продуктов.

47. ISO/IEC 25024:2015. Системная и программная инженерия. Требования к качеству и оценка систем и программных средств (SQuaRE). Измерение качества данных. – Введ. 2015–10–15. – Женева : ISO/IEC, 2015.

48. ISO/IEC 25030:2007. Программная инженерия. Требования к качеству и оценка программных продуктов (SQuaRE). Требования к качеству. Введ. 2007–06–01. – Женева : ISO/IEC, 2007.

49. ISO/IEC 25040:2011. Системная и программная инженерия. Требования к качеству и оценка систем и программных средств (SQuaRE). Процесс оценки. – Введ. 2011–03–01. – Женева : ISO/IEC, 2011.

50. ISO/IEC 25041:2012. Системная и программная инженерия. Требования к качеству и оценка систем и программных средств (SQuaRE). Руководство по оценке для разработчиков, приобретателей и независимых оценщиков. – Введ. 2012–10–15. – Женева : ISO/IEC, 2012.

51. ISO/IEC 25045:2010. Системная и программная инженерия. Требования к качеству и оценка систем и программных средств (SQuaRE). Модуль оценки для восстанавливаемости. – Введ. 2010–09–01. – Женева : ISO/IEC, 2010.

52. ISO/IEC 25051:2014. Программная инженерия. Требования к качеству и оценка систем и программных средств (SQuaRE). Требования к качеству готовых к использованию программных продуктов (RUSP) и инструкции по тестированию. – Введ. 2014–02–15. – Женева : ISO/IEC, 2014.

53. ISO/IEC TR 25060:2010. Системная и программная инженерия. Требования к качеству и оценка систем и программных продуктов (SQuaRE). Общий промышленный формат (CIF) для практичности. Общая информация, связанная с практичностью. – Введ. 2010–07–15. – Женева : ISO/IEC, 2015.

54. ISO/IEC 25062:2006. Программная инженерия. Требования к качеству и оценка программных продуктов (SQuaRE). Общий промышленный формат (CIF) для отчетов по тестированию практичности. – Введ. 2006–04–01. – Женева : ISO/IEC, 2006.

55. ISO/IEC 25063:2014. Системная и программная инженерия. Требования к качеству и оценка систем и программных продуктов (SQuaRE). Общий промышленный формат (CIF) для практичности. Описание контекста использования. – Введ. 2014–03–15. – Женева : ISO/IEC, 2014.

56. ISO/IEC 25064:2013. Системная и программная инженерия. Требования к качеству и оценка программных продуктов (SQuaRE). Общий

промышленный формат (CIF) для практичности. Отчет о потребностях пользователя. – Введ. 2013–09–01. – Женева : ISO/IEC, 2013.

57. ISO/IEC 25066. Системная и программная инженерия. Требования к качеству и оценка систем и программных средств (SQuaRE). Общий промышленный формат (CIF) для практичности. Отчет об оценке.

58. ISO/IEC 9126:1991. Информационная технология. Оценка программного продукта. Характеристики качества и руководства по их применению. – Введ. 1991–12–01. – Женева : ISO/IEC, 1991.

59. ISO/IEC 9126–1:2001. Программная инженерия. Качество продукта. Ч. 1 : Модель качества. – Введ. 2001–06–15. – Женева : ISO/IEC, 2001.

60. ISO/IEC TR 9126–2:2003. Программная инженерия. Качество продукта – Ч. 2 : Внешние метрики. – Введ. 2003–07–01. – Женева : ISO/IEC, 2003.

61. ISO/IEC TR 9126–3:2003. Программная инженерия. Качество продукта. Ч. 3 : Внутренние метрики. – Введ. 2003–07–01. – Женева : ISO/IEC, 2003.

62. ISO/IEC TR 9126–4:2004. Программная инженерия. Качество продукта – Ч. 4 : Метрики качества в использовании. – Введ. 2004–04–01. – Женева : ISO/IEC, 2004.

63. ISO/IEC GUIDE 99:2007(E/R). Международный словарь по метрологии. Основные и общие понятия и соответствующие термины (VIM). – Введ. 2007–12–15. – Женева : ISO/IEC, 2007.

64. Закон Республики Беларусь от 5 января 2004 г. №262-З «О техническом нормировании и стандартизации» [Электронный ресурс]. – 2015. – Режим доступа : <http://www.pravo.by/main.aspx?guid=3871&p0=H10400262&p2> .

65. Закон Республики Беларусь от 31 декабря 2010 г. №228-З «О внесении изменений и дополнений в некоторые законы Республики Беларусь по вопросам оценки соответствия требованиям технических нормативных правовых актов в области технического нормирования и стандартизации» [Электронный ресурс]. – 2015. – Режим доступа : http://st.vstu.by/files/9613/6674/6539/_31_12_2010_228-___1.pdf .

66. Бахтизин, В. В. Стандартизация и сертификация программного обеспечения : учеб. пособие / В. В. Бахтизин, Л. А. Глухова. – Минск : БГУИР, 2006.

67. Изосимов, А. В. Метрическая оценка качества программ / А. В. Изосимов, А. Л. Рыжко. – М. : МАИ, 1989.

68. Липаев, В. В. Обеспечение качества программных средств. Методы и стандарты / В. В. Липаев. – М. : СИНТЕГ, 2001.

69. Фатрелл, Р. Управление программными проектами: достижение оптимального качества при минимуме затрат / Р. Фатрелл, Д. Шафер, Л. Шафер. – М. : Вильямс, 2003.

70. Холстед, М. Х. Начала науки о программах / М. Х. Холстед. – М. : Финансы и статистика, 1981.

71. Черников, Б. В. Управление качеством программного обеспечения : учебник / Б. В. Черников. – М. : ИД «Форум»: ИНФРА-М, 2012.

72. About ETSI [Электронный ресурс]. – 2015. – Режим доступа : <http://www.etsi.org/about> .

73. About us [Электронный ресурс]. – 2015. – Режим доступа : <http://www.cencenelec.eu/aboutus/Pages/default.aspx> .

74. CEN Community [Электронный ресурс]. – 2015. – Режим доступа : <http://standards.cen.eu> .

75. Developing ISO standards [Электронный ресурс]. – 2015. – Режим доступа : http://www.iso.org/iso/home/standards_development/resources-for-technical-work/support-for-developing-standards.htm .

76. ISO deliverables [Электронный ресурс]. – 2015. – Режим доступа : [http://www.iso.org/iso/home/standards_development/deliverables-all.htm?="](http://www.iso.org/iso/home/standards_development/deliverables-all.htm?=) .

77. ISO/IEC JTC001 «Информационные технологии» [Электронный ресурс]. – 2015. – Режим доступа : <http://isotc.iso.org/livelink/livelink/open/jtc1> .

78. Who develops ISO standards? [Электронный ресурс]. – 2008. – Режим доступа : http://www.iso.org/iso/home/standards_development/who-develops-iso-standards.htm .

79. Who we are [Электронный ресурс]. – 2015. – Режим доступа : <http://www.cen.eu> .

80. Национальная система подтверждения соответствия. основополагающие документы Национальной системы подтверждения соответствия Республики Беларусь [Электронный ресурс]. – 2015. – Режим доступа : http://www.gosstandart.gov.by/ru-RU/National-s_osov-pol-doc.php .

81. Национальная система подтверждения соответствия. Структура национальной системы подтверждения соответствия Республики Беларусь [Электронный ресурс]. – 2015. – Режим доступа : <http://www.gosstandart.gov.by/ru-RU/National-sys-podtverzhd-sootv.php> .

82. Средства моделирования (CASE) и поддержки всех стадий разработки ПО [Электронный ресурс]. – 2015. – Режим доступа : <http://www.interface.ru/home.asp?artId=99> .

Учебное издание

**Бахтизин Вячеслав Вениаминович
Глухова Лилия Александровна**

**МЕТРОЛОГИЯ, СТАНДАРТИЗАЦИЯ
И СЕРТИФИКАЦИЯ
В ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЯХ**

УЧЕБНОЕ ПОСОБИЕ

В двух частях
Часть 2

Редактор *Е. И. Герман*
Корректор *Е. Н. Батурчик*
Компьютерная правка, оригинал-макет *А. В. Бас*

Подписано в печать 12.10.2016. Формат 60x84 1/16. Бумага офсетная. Гарнитура «Таймс».
Отпечатано на ризографе. Усл. печ. л. 12,09. Уч.-изд. л. 12,7. Тираж 300 экз. Заказ 313.

Издатель и полиграфическое исполнение: учреждение образования
«Белорусский государственный университет информатики и радиоэлектроники».
Свидетельство о государственной регистрации издателя, изготовителя,
распространителя печатных изданий №1/238 от 24.03.2014,
№2/113 от 07.04.2014, №3/615 от 07.04.2014.
ЛП №02330/264 от 14.04.2014.
220013, Минск, П. Бровки, 6