

## **8. СТАНДАРТИЗАЦИЯ, ОЦЕНКА КАЧЕСТВА И СЕРТИФИКАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

### **8.1 Стандартизация программного обеспечения**

#### **8.1.1 Комплекс межгосударственных стандартов «Единая система программной документации»**

Разработка разнообразного программного обеспечения (ПО) потребовала разработки стандартов на ПО. Стандарты на ПО позволяют унифицировать процессы разработки, эксплуатации и сопровождения ПО, что способствует повышению качества и конкурентоспособности ПО на внутреннем и внешнем рынках.

Одним из комплексов межгосударственных стандартов для разработки ПО является Единая система программной документации (ЕСПД).

Единая система программной документации - комплекс государственных стандартов, устанавливающих взаимоувязанные правила разработки, оформления и обращения программ и программной документации.

В стандартах ЕСПД устанавливают требования, регламентирующие разработку, сопровождение, изготовление и эксплуатацию программ, что обеспечивает возможность:

- унификации программных изделий для взаимного обмена программами и применения ранее разработанных программ в новых разработках;
- снижения трудоемкости и повышения эффективности разработки, сопровождения, изготовления и эксплуатации программных изделий;
- автоматизации изготовления и хранения программной документации.

Сопровождение программы включает анализ функционирования, развитие и совершенствование программы, а также внесение изменений в нее с целью устранения ошибок.

Правила и положения, установленные в стандартах ЕСПД, распространяются на программы и программную документацию для вычислительных машин, комплексов и систем независимо от их назначения и области применения.

В состав ЕСПД входят:

- основополагающие и организационно-методические стандарты;
- стандарты, определяющие формы и содержание программных документов, применяемых при обработке данных;

- стандарты, обеспечивающие автоматизацию разработки программных документов.

Разработка организационно-методической документации, определяющей и регламентирующей деятельность организаций по разработке, сопровождению и эксплуатации программ, должна проводиться на основе стандартов ЕСПД.

### **8.1.2 Стандарт ISO/IEC 12207 – базовый стандарт в области жизненного цикла программных средств и систем**

Одним из базовых понятий в области информационных технологий является понятие жизненного цикла программных средств (ПС) и систем.

Под **жизненным циклом** (ЖЦ) подразумевается совокупность процессов, работ и задач, включающая в себя разработку, эксплуатацию и сопровождение ПС, начиная с анализа его концепции или потребности в заказе до прекращения его использования.

С понятием ЖЦ ПС или системы тесно связано понятие модели жизненного цикла. **Модель жизненного цикла** – это совокупность процессов, работ и задач ЖЦ, отражающая их взаимосвязь и последовательность выполнения.

Стандартизация ЖЦ позволяет упорядочить вопросы создания, сопровождения и управления ПС. Это ведет к повышению качества процессов ЖЦ и в конечном итоге к повышению качества самих программных средств.

На уровне международных стандартов жизненный цикл сложных программных средств наиболее полно отражен в международном стандарте **ISO/IEC 12207:1995 – Информационная технология – Процессы жизненного цикла программных средств**.

В Республике Беларусь этот стандарт введен в действие в 2004 г. под обозначением **СТБ ИСО/МЭК 12207–200**.

Стандарт **ИСО /МЭК 12207** охватывает процессы ЖЦ системы только в части ее программных средств и не определяет процессы ЖЦ для ее остальных компонентов.

В соответствии с данным стандартом ЖЦ ПС и систем имеет трехуровневую иерархическую структуру (

рис. 8.1.1). Основу жизненного цикла составляет набор **процессов**. Каждый процесс разделен на набор **работ**. Каждая работа

разделена на набор *задач*. Общее число процессов в ЖЦ ПС равно 17, работ – 74, задач – 232.

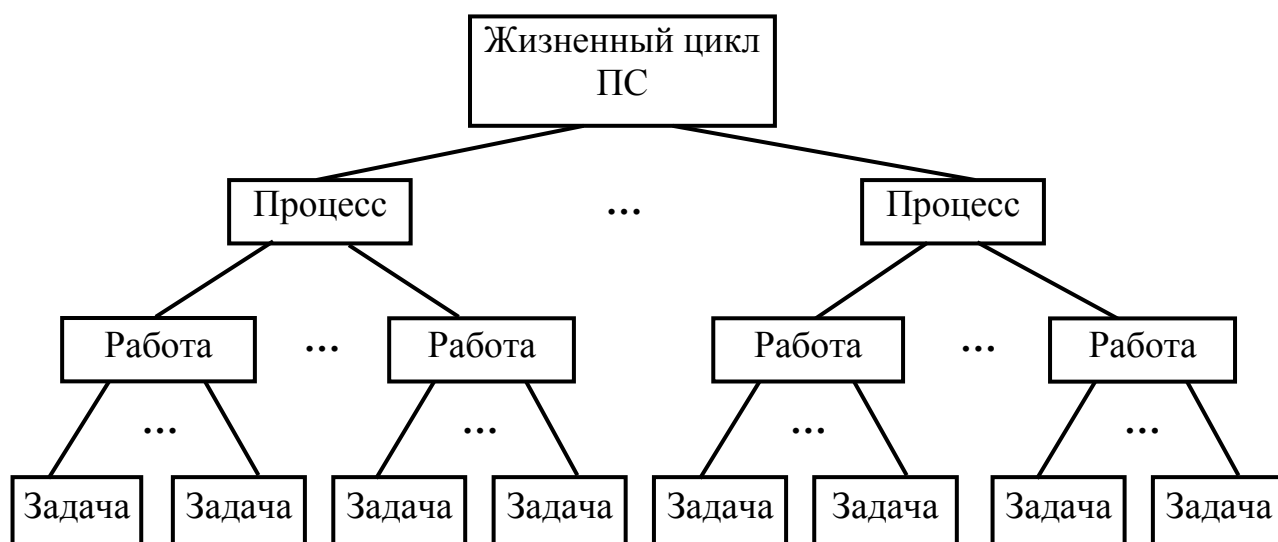


Рис. 8.1.1 Общая структура жизненного цикла программных средств

Процессы ЖЦ ПС делятся на *три группы*

- основные;
- вспомогательные;
- организационные.

На рис. 8.1.2 изображен состав процессов ЖЦ ПС.



Рис. 8.1.2. Процессы жизненного цикла программных средств

**Основные процессы жизненного цикла** – это процессы, которые реализуются под управлением основных сторон, участвующих в ЖЦ ПС. Основными сторонами являются заказчик, поставщик, разработчик, оператор и персонал сопровождения программных продуктов.

**Вспомогательные процессы жизненного цикла** – это процессы, являющиеся целенаправленными составными частями других

процессов. Их основное назначение – обеспечить успешную реализацию и качество выполнения программного проекта.

Для управления качеством программных средств в ходе жизненного цикла служат процессы обеспечения качества, верификации, аттестации, совместного анализа и аудита.

**Организационные процессы жизненного цикла** – это процессы, предназначенные для создания и совершенствования организационных структур, охватывающих процессы ЖЦ и соответствующий персонал.

### **8.1.3 Дополнения к базовому стандарту *ISO/IEC 12207***

В 2002 г. введено в действие Дополнение к базовому стандарту *ISO/IEC 12207:1995* под обозначением ***ISO/IEC 12207:1995 – Информационная технология – Процессы жизненного цикла программных средств / Amd.1: 2002***. Данное Дополнение расширяет базовый стандарт и обеспечивает согласование его положений с положениями других стандартов, связанных с жизненным циклом программных средств. В первую очередь это касается стандартов, относящихся к оценке процессов и продуктов ЖЦ ПС (*ISO/IEC 15504–1–8, ISO/IEC 14598–1–6*), а также стандартов, посвященных технологиям, ориентированным на человека (например, *ISO 13407:1999*).

В 2004 г. введено в действие второе Дополнение к базовому стандарту *ISO/IEC 12207:1995* под обозначением ***ISO/IEC 12207:1995 – Информационная технология – Процессы жизненного цикла программных средств / Amd.2:2004***. В данном Дополнении отредактированы некоторые положения Дополнения *ISO/IEC 12207:1995/Amd.1:2002* и введены дальнейшие расширения базового стандарта, в частности добавлен ряд новых процессов ЖЦ ПС (рис.8.1.3).

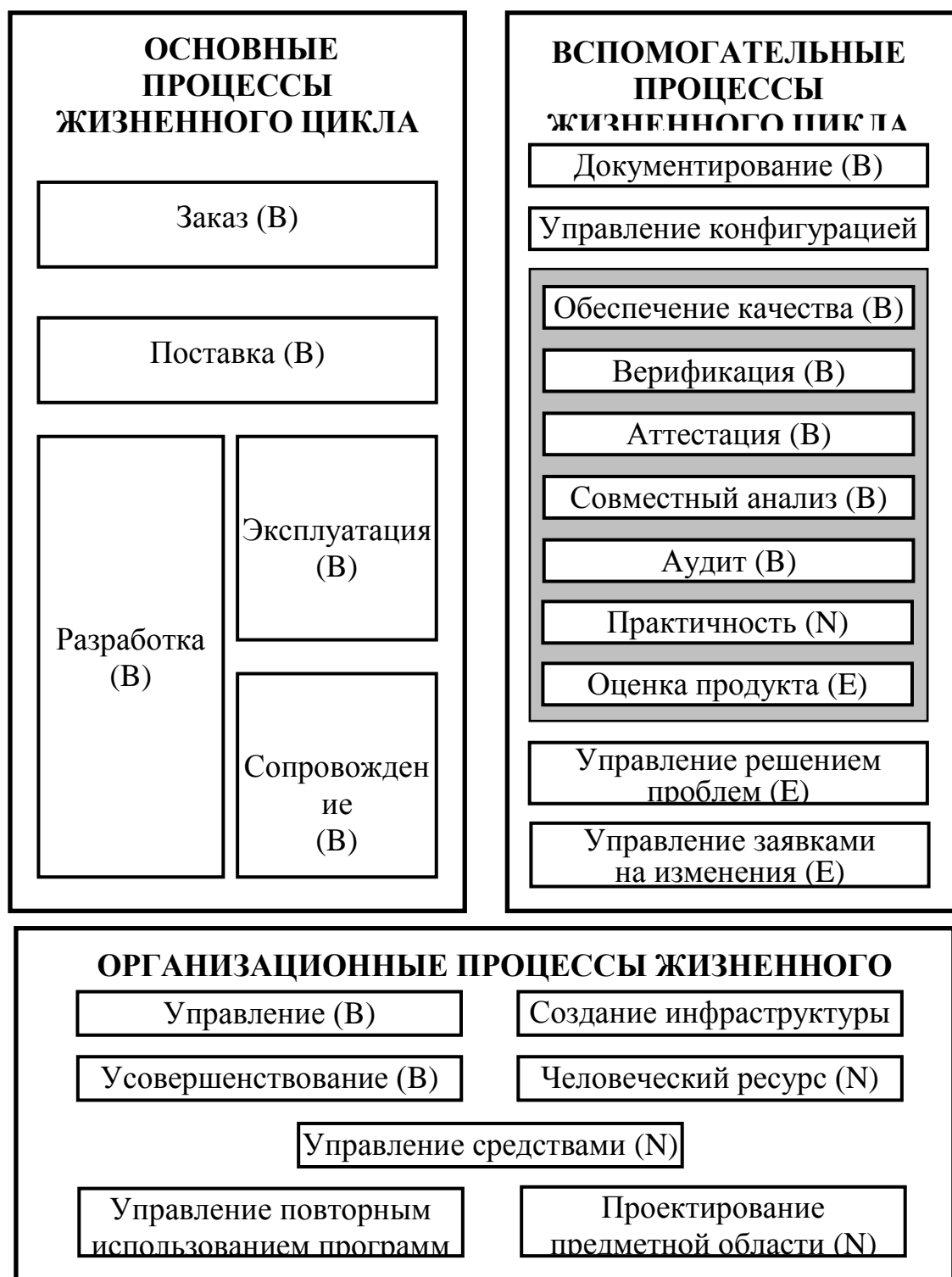


Рис. 8.1.3 Процессы жизненного цикла ПС с учетом положений ISO/IEC 12207:1995/Amd.1:2002 и ISO/IEC 12207:1995/Amd.2:2004

В 1998 г. разработан международный стандарт *ISO/IEC TR 15271:1998 – Информационная технология – Руководство по применению ISO/IEC 12207 (Процессы жизненного цикла программных средств)*. Данный стандарт дополняет стандарт

*ИСО/МЭК 12207*. В нем приведены рекомендации по практическому применению стандарта *ИСО/МЭК 12207* при реализации конкретных проектов создания программных средств.

#### **8.1.4 Применение стандарта ИСО/МЭК 12207 к конкретному программному проекту.**

Стандарт *ИСО/МЭК 12207* устанавливает **архитектуру верхнего уровня жизненного цикла программных средств** от замысла до снятия с эксплуатации (утилизации). Архитектура состоит из процессов и связей между ними. Процессы основаны на *двух принципах: модульности и ответственности*.

Процессы являются *модульными*, поскольку для них характерны:

- высокая связность (все части процесса строго взаимосвязаны);
- слабое сцепление (число интерфейсов между процессами минимально).

Принцип *ответственности* означает, что каждый процесс рассмотрен с точки зрения ответственности (обязанностей) стороны, его выполняющей.

Набор процессов, приведенный в стандарте *ИСО/МЭК 12207*, используется для больших или сложных программных проектов. Однако стандарт *ИСО/МЭК 12207* после соответствующей адаптации может быть применен к программному проекту любого типа, меньшего размера и сложности. Данный стандарт может использоваться для ПС, являющихся как самостоятельными объектами, так и частями системы. Если ПС является частью системы, его выделяют из системы, разрабатывают и включают в систему.

**Соответствие с ИСО/МЭК 12207** может быть достигнуто следующими способами:

- заказчик определяет условия договоров, набор процессов, работ и задач из *ИСО/МЭК 12207*, которым должны подчиняться поставщики;
- в программном проекте выполняется адаптация процессов, работ и задач ЖЦ в соответствии с критериями договора.

Таким образом, пользователь стандарта *ИСО/МЭК 12207* должен выбрать, практически применить и скомпоновать процессы, работы и задачи ЖЦ соответственно целям и задачам организации и проекта. При практическом применении *ИСО/МЭК 12207* должны быть сохранены его архитектура, сущность и целостность, например, за счет включения элементов стандарта с пометкой «не применяется» с объяснением причины такого решения.

### **Внедрение ИСО/МЭК 12207**

Возможны различные *способы внедрения ИСО/МЭК 12207*, в том числе:

- применение в конкретном проекте при определении процессов, работ и задач, связанных с ПС;
- использование в качестве основы для усовершенствования программных процессов в организации;
- использование в качестве компонента в процессе моделирования общего ЖЦ системы.

Можно выделить следующие *шаги стратегии практического применения* стандарта *ИСО/МЭК 12207*:

- план внедрения;
- практическое применение стандарта *ИСО/МЭК 12207*;
- проведение сопровождения пилотных проектов;
- формализация метода внедрения;
- утверждение метода внедрения.

Внедрение *ИСО/МЭК 12207* должно рассматриваться и планироваться как конкретный проект. При *планировании внедрения* должны быть определены:

- область применения проекта по внедрению (единый проект внутри организации или часть двустороннего договора; выделение одного или нескольких ключевых процессов, от которых ожидают выгоду для организации; практическое применение стандарта *ИСО/МЭК 12207* через ряд проектов с поэтапным его введением; практическое применение стандарта *ИСО/МЭК 12207* во всех проектах и во всех подразделениях организации);
- цели проекта и их соответствие целям организации;
- роли и обязанности группы персонала проекта с установлением единоличной ответственности за каждый процесс;
- ресурсы (время, деньги, персонал, оборудование), необходимые для внедрения стандарта *ИСО/МЭК 12207*;
- план управления проектом по внедрению стандарта *ИСО/МЭК 12207* (план внедрения).

*Практическое применение стандарта ИСО/МЭК 12207* должно базироваться на использовании процесса адаптации, описанного в его *Приложениях А и В*, а также учитывать рекомендации стандарта *ИСО/МЭК ТО 15271* и особенности конкретной ситуации. При выполнении работ процесса адаптации должна быть выполнена привязка установленных процессов, работ и задач (с учетом их взаимосвязей) к выбранной модели (или моделям) ЖЦ ПС. Результаты адаптации и обоснование метода внедрения должны быть документированы и включены в план внедрения *ИСО/МЭК 12207*.



Для уменьшения риска внедрения стандарта *ИСО/МЭК 12207* в организации рекомендуется выполнить один или несколько **пилотных проектов**, в которых осваиваются и сопровождаются адаптированные процессы ЖЦ и определяются критические факторы достижения успеха.

**Формализация метода внедрения** включает в себя введение нового процесса в ряде проектов или в целом в организации. После **утверждения метода внедрения ИСО/МЭК 12207** выполняется последовательная автоматизированная реализация процесса внедрения в проекте или в организации.

### **Типовая модель жизненного цикла системы**

**Типовая модель жизненного цикла системы** начинается с концепции потребности в системе, охватывает разработку, эксплуатацию и сопровождение системы и заканчивается снятием системы с эксплуатации (утилизацией).

Модель ЖЦ обычно разделяют на *периоды реализации*, например, стадии, этапы, фазы. Каждый период включает основные выполняемые в нем работы и задачи. При их завершении может потребоваться разрешение на переход к следующему периоду. Например, типовую модель ЖЦ системы часто разделяют на следующие *стадии (этапы)*:

- 1) определение потребностей;
- 2) исследование и описание основных концепций;
- 3) демонстрация и аттестация основных концепций;
- 4) проектирование и разработка;
- 5) создание и производство;
- 6) распространение и продажа;
- 7) эксплуатация;
- 8) сопровождение и поддержка;
- 9) снятие с эксплуатации (утилизация).

Данные этапы должны быть в дальнейшем адаптированы к конкретной модели ЖЦ системы.

**Типовая модель жизненного цикла программного средства** состоит из ряда работ. Данная модель начинается с формулировки замысла или концепции программного средства, продолжается работами по применению методов системной и программной инженерии, работами по эксплуатации, сопровождению и поддержке и заканчивается снятием ПС с эксплуатации. В стандарте *ИСО/МЭК 12207* все работы объединены в основные, вспомогательные и организационные процессы, из которых формируют модель ЖЦ ПС (рис.8.1.4).

**На этапе определения потребностей** определяют потребности в новой или усовершенствованной системе и формулируют их с учетом стоимости, критичности, реализуемости планируемой системы.

**На этапе исследования и описания концепции** выполняется первоначальное планирование, при котором оценивают технические, стратегические и рыночно-экономические аспекты системы путем исследований, опытной разработки и оценки ее концепции. Разрабатываются ПС, реализующие аналитические или имитационные модели, необходимые при исследованиях. Выходными результатами данного этапа, передаваемыми на следующий этап, являются предварительные общие требования к системе и возможные ПС, выбранные в качестве прототипа.

**На этапе демонстрации и аттестации** уточняются характеристики системы, ее концепции и принятые решения. При этом используются методы системной инженерии. Определяются предварительные требования к ПС и его прототипам, включая тестирование и аттестацию. Характеристики системы, выбранные концепции и решения аттестуют для демонстрации того, что система пригодна для технологической разработки. Требования к системе в дальнейшем распределяются между ее компонентами, такими как технические средства, компьютеры, программные средства и персонал. Данные выходные результаты передаются на следующий этап.

**На этапе разработки** выполняется проектирование, создание, сборка, тестирование и оценка системных технических и программных средств, определение персонала подсистем и объектов сопровождения. Выходными результатами данного этапа являются система, достаточно близкая к создаваемой, необходимые документы и результаты тестирования качества системы.

На данном этапе при разработке ПС полностью применим стандарт *ИСО/МЭК 12207*. При этом возможно однократное или многократное выполнение процесса разработки. Результатами данного этапа являются исходные требования к ПС, его технический проект и тексты программ.

**На этапе изготовления** выполняются работы от постановки на производство до поставки заказчику и приемки системы. Период производства при поставке системы на рынок охватывает работы от постановки на производство до перепроектирования или снятия системы с производства.

Для ПС работа по изготовлению состоит из копирования (тиражирования) ПС и документов к нему на соответствующие носители для различных пользователей.

***На этапе распространения и продажи*** систему поставляют заказчику (пользователю) или покупателям (на рынок). *Период распространения* начинается с поставки первой работоспособной системы заказчику. *Период продажи* начинается с поставки первой партии систем на рынок и заканчивается изъятием с рынка.

***На этапе эксплуатации*** система применяется пользователями или потребителями. Данный этап заканчивается снятием системы с эксплуатации.

***На этапе сопровождения и поддержки*** выполняется модификация системы в связи с наличием в ней ошибок, дефектов, возникновением проблем или необходимостью в адаптации или усовершенствовании.

***На этапе снятия с эксплуатации*** система снимается с обслуживания, выполняется ее архивирование и обеспечение ограниченной поддержки

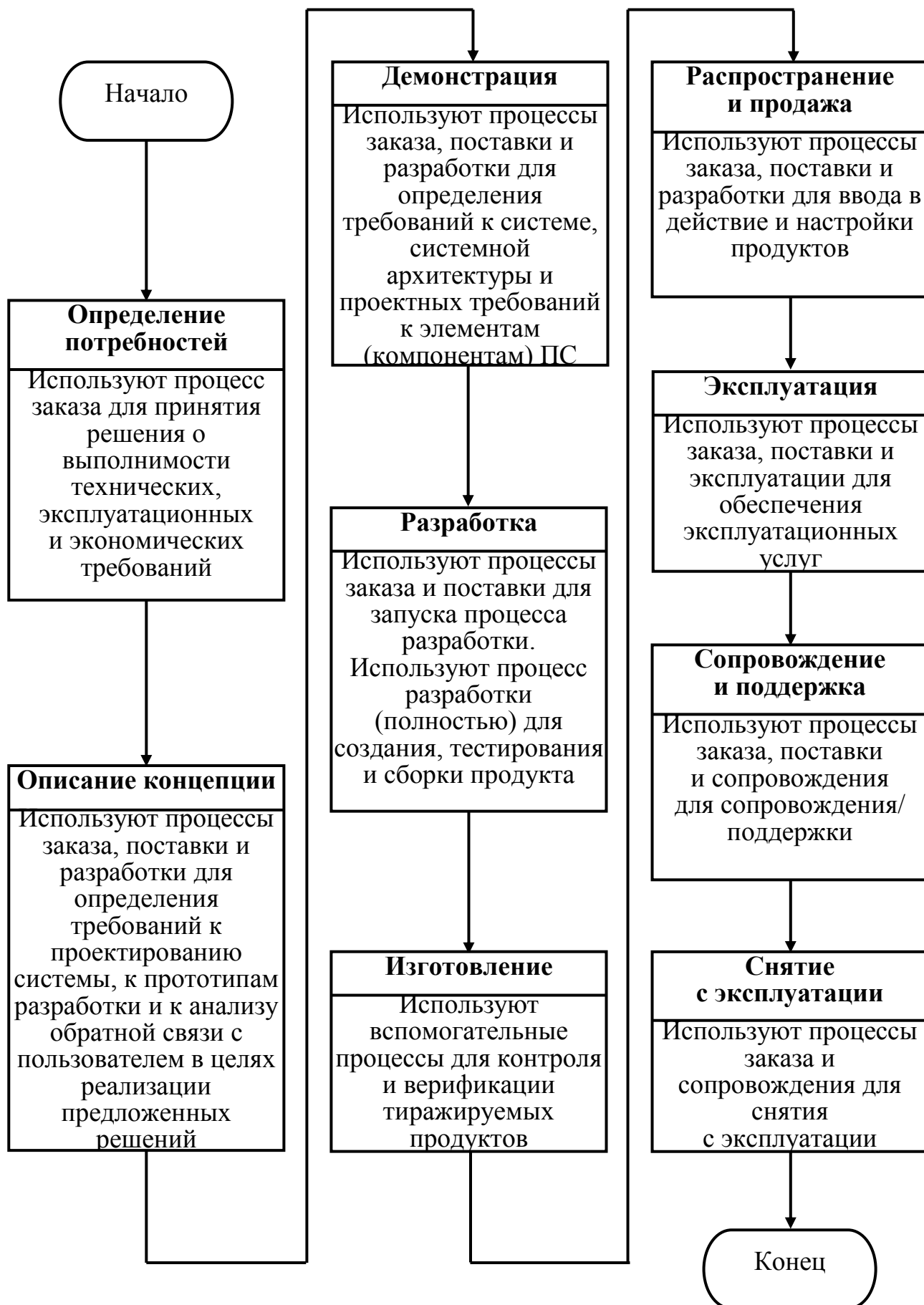


Рис. 8.1.4 Использование ИСО/МЭК 12207 для обеспечения модели жизненного цикла системы

### 8.1.5 Модели жизненного цикла

В Приложении С стандарта ИСО/МЭК ТО 15271 описаны *три фундаментальных модели ЖЦ* (их принято называть также *базовыми стратегиями разработки*):

- каскадная;
- инкрементная;
- эволюционная.

Каждая из них может использоваться самостоятельно или комбинироваться с другими для создания *гибридной* модели ЖЦ.

Три базовых стратегии могут быть реализованы различными моделями ЖЦ.

#### Каскадная модель

Каскадная фундаментальная модель ЖЦ системы реализует *принцип однократного выполнения* следующих видов деятельности:

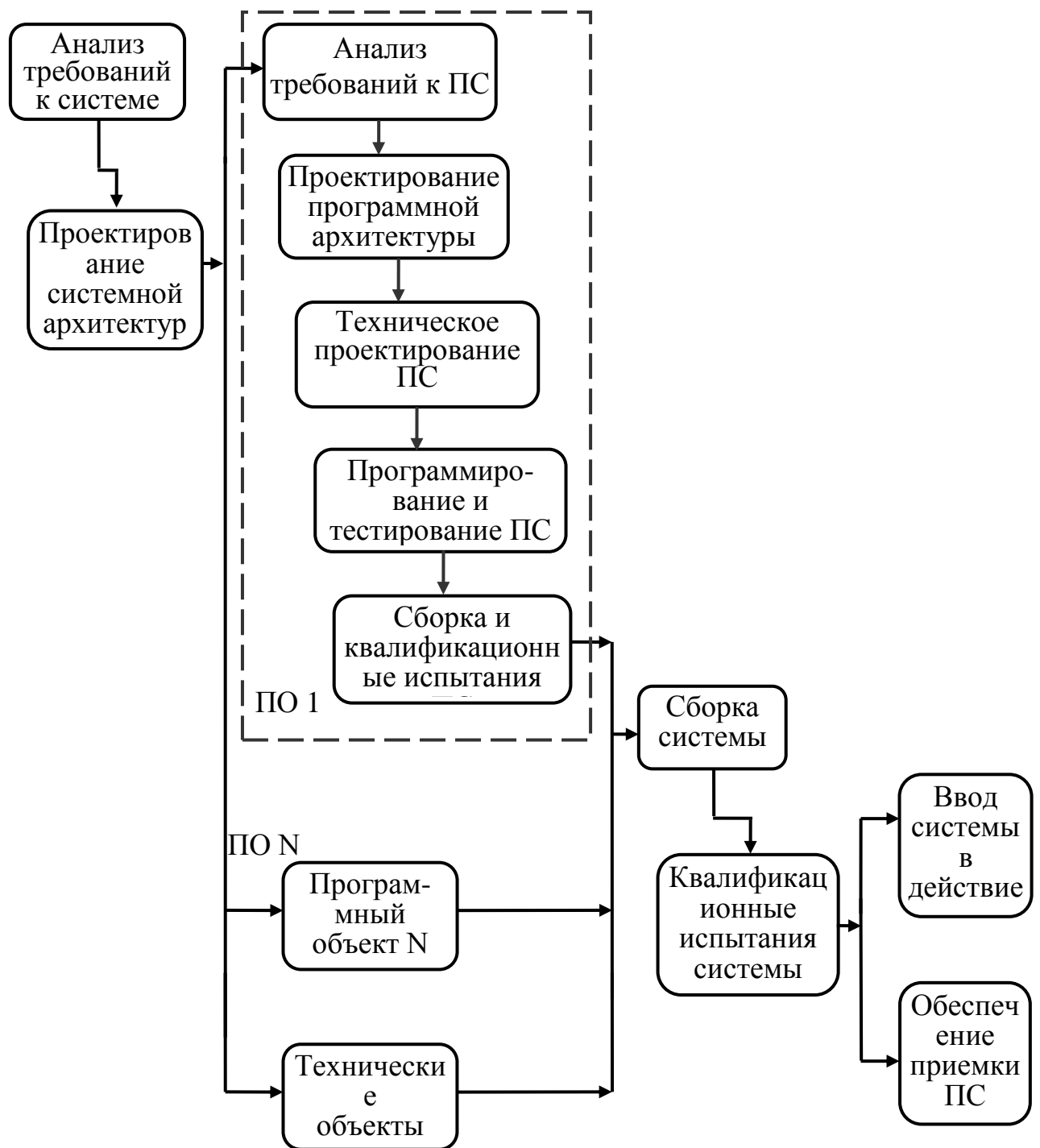
- установление потребностей пользователя;
- определение требований;
- проектирование системы;
- изготовление системы;
- испытание;
- корректировка;
- поставка или использование.

Таким образом, каскадная модель основана на *полном формулировании требований к системе в начале процесса разработки*. При применении каскадной стратегии работы и задачи процесса разработки выполняют последовательно. Они могут выполняться также частично параллельно в случаях перекрытия последовательных работ.

При одновременной разработке нескольких программных объектов работы и задачи процесса разработки могут выполняться параллельно.

Процессы сопровождения и эксплуатации реализуются после разработки. Процессы заказа, поставки, вспомогательные и организационные процессы выполняются параллельно с процессом разработки.

На 8.1.5 приведен пример каскадной модели ЖЦ системы.



ПО - программный объект  
 ПС – программные средства

Рис.8.1.5 Вариант каскадной модели

Данный вариант учитывает возможности параллельной разработки программных объектов, входящих в состав системы.

*Достоинства каскадной модели:*

- однократное представление всех характеристик системы (однократное представление требований и продукта разработки);
- необходимость единственной фазы перехода от старой системы к новой (один проход процесса разработки для перехода к новой системе).

*Использование каскадной модели затруднено в следующих случаях:*

- требования к объектам определены недостаточно четко;
- большой объем системы; это делает практически невозможным однократное выполнение работ по её созданию;
- предполагаются скорые изменения в технологии работ; это повышает риски нарушения графика проекта и разработки системы, не удовлетворяющей требованиям;
- возможны текущие изменения требований к системе; это вызывает необходимость разработки системы заново;
- ресурсы, например средства или персонал, ограничены;
- необходимость использования промежуточных продуктов разработки; последние при применении каскадной модели для использования непригодны.

С вышеперечисленными ограничениями связаны основные недостатки каскадной модели.

### **Инкрементная модель**

Инкрементная фундаментальная модель называется также ***запланированным усовершенствованием продукта***.

При ее использовании *набор требований к системе формулируется полностью в начале процесса разработки*. Затем реализуется разработка последовательности инкрементов (конструкций). Первый инкремент реализует часть требований. В последующих инкрементах *в соответствии с планом требования реализуются постепенно с небольшими добавлениями* по отношению к предыдущим инкрементам. Процесс продолжается, пока не будет закончено создание системы.

При использовании данной фундаментальной модели некоторые процессы, работы и задачи могут выполняться однократно, а другие – при создании каждого инкремента. При разработке каждого инкремента, как правило, используется каскадная модель с возможностью частичного перекрытия работ.

На 8.1.6 приведен вариант реализации инкрементной фундаментальной модели .

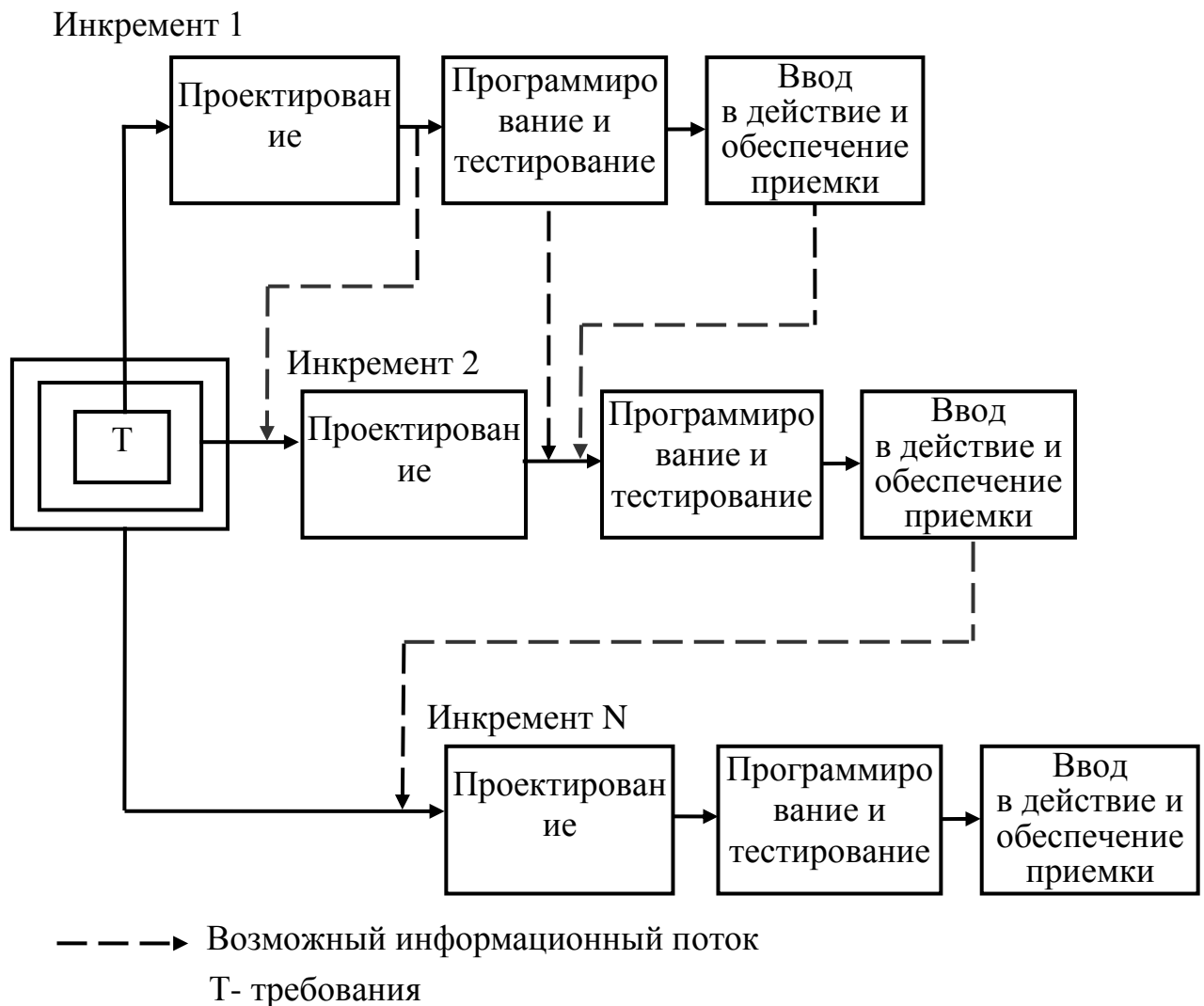


Рис. 8.1.6 Вариант инкрементной модели

*Достоинства инкрементной модели:*

- необходимость изначального формулирования характеристик системы (требований к ней);
- пригодность для использования промежуточного продукта;
- естественное разделение системы на наращиваемые компоненты (инкременты);
- возможности наращивания привлекаемого персонала и средств.

*Использование инкрементной модели затруднено в следующих случаях:*

- требования к объектам определены недостаточно точно;
- необходимо предусмотреть сразу все возможности системы (реализовать все требования к ней);
- возможны текущие изменения требований к системе;



- возможности привлечения ресурсов (средств или персонала) на длительный период ограничены;

- предполагаются скорые изменения в технологиях работ.

С вышеперечисленными ограничениями связаны основные недостатки инкрементной модели.

### **Эволюционная модель**

При использовании эволюционной фундаментальной модели ЖЦ, как и в предыдущем случае, система разрабатывается в виде последовательности конструкций (версий). Однако *требования изначально определяются только частично и уточняются в каждой последующей версии.*

При разработке каждой версии, как правило, используется каскадная модель ЖЦ с возможностью частичного перекрытия работ или задач.

Один из вариантов эволюционной модели приведен на рис. 8.1.7.

*Достоинства эволюционной модели:*

- итерационное определение возможностей системы (требований к ней);

- пригодность для использования промежуточного продукта;

- естественное разделение системы на наращиваемые компоненты;

- привлечение персонала и средств по мере необходимости;

- необходимая обратная связь с пользователем для полного понимания требований;

- упрощение надзора за изменением технологии.

*Использование эволюционной модели затруднено в следующих случаях:*

- все возможности системы (требования к ней) определены изначально;

- возможности длительного привлечения ресурсов (например средств или персонала) ограничены.

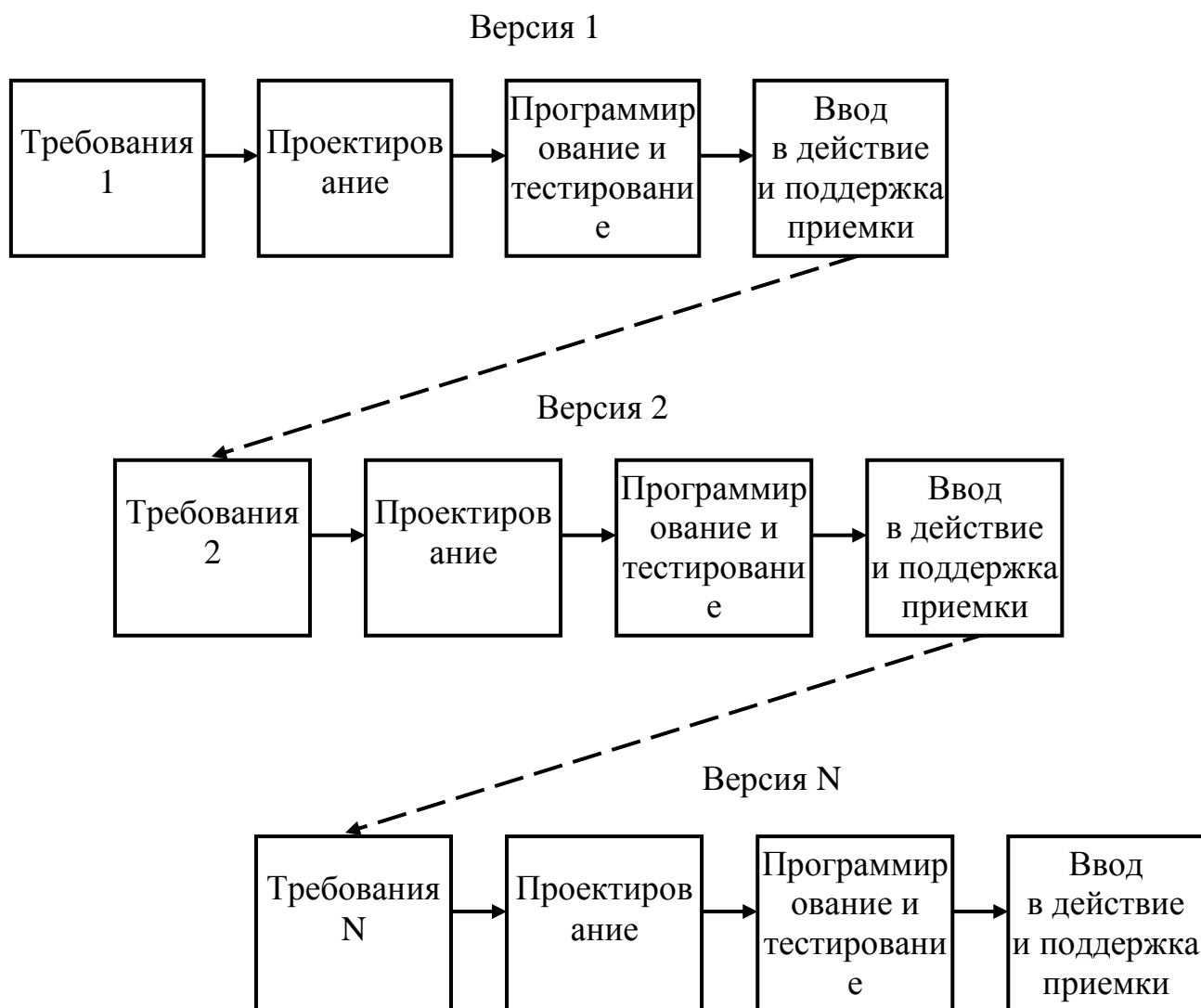


Рис.8.1.7 Вариант эволюционной модели

### 8.1.6 Стандарт ISO/IEC TR 16326

Международный стандарт **ISO/IEC TR 16326:1999** – **Программная инженерия – Руководство по применению ISO/IEC 12207 при управлении проектом** – введен в действие в 1999 г. В данных стандартах приведены уточнения и дополнения к стандарту ИСО/МЭК 12207 в части процесса управления, а также рекомендации по его использованию. Стандарт ИСО/МЭК ТО 16326 предназначен в первую очередь для администраторов проектов, отвечающих за управление реализацией основных процессов ЖЦ ПС, определенных в ИСО/МЭК 12207: заказа, поставки, разработки, эксплуатации и сопровождения.

### 8.1.7 Стандарт ISO/IEC 15910

Возрастающие масштабы применения ПС и их сложность вызывают необходимость в полной, точной и понятной документации на ПС, доступной пользователям. Часто документация разрабатывается после создания соответствующего ПС. Однако с точки зрения ее качества необходимо, чтобы она создавалась в процессе разработки ПС.

С учетом этого в 1999 г. введен в действие международный стандарт ***ISO/IEC 15910:1999 – Информационная технология – Процесс создания документации пользователя программного средства.***

Стандарт *ИСО/МЭК 15910* определяет одну из реализаций процесса документирования, описанного в *ИСО/МЭК 12207*, и предоставляет пользователю метод применения данного процесса при создании конкретного ПС. Стандарт содержит структуру комплексного плана разработки документации, однако не определяет состав требований к стилю оформления документов, устанавливая только их диапазоны.

Стандарт *ИСО/МЭК 15910* должен использоваться при создании *пользовательской документации всех видов:*

- печатной документации (руководство пользователя, краткие справочные карты);
- диалоговой (оперативной) информации;
- справочного текста (Help);
- системы диалоговой документации.

### 8.1.8 Стандарт ISO/IEC 15288

В 2002 г. введен в действие международный стандарт ***ISO/IEC 15288:2002 – Системная инженерия – Процессы жизненного цикла системы.*** Целью его разработки является определение общей структуры описания жизненного цикла систем.

В 2003 г. опубликован международный стандарт ***ISO/IEC TR 19760:2003 – Системная инженерия – Руководство по использованию ISO/IEC 15288 (Процессы жизненного цикла системы)***, содержащий рекомендации по практическому применению стандарта *ISO/IEC 15288:2002* в условиях реализации конкретных проектов создания систем.

Стандарт *ISO/IEC 15288* устанавливает всестороннюю и интегрированную структуру для управления полным жизненным циклом систем, представляющих собой совокупность аппаратных средств, программных средств, персонала (ручных операций) и

процессов. Стандарт определяет набор процессов ЖЦ и связанную с ними терминологию. Эти процессы могут применяться на любом уровне иерархии разработки систем.

Стандарт *ISO/IEC 15288* может быть использован в большинстве отраслей современной промышленности, связанных с разработкой и использованием систем (например, в аэрокосмической промышленности, телекоммуникациях, военной промышленности, информационных технологиях и т.п.).

Применение данного стандарта обеспечивает основу для целостного подхода к программному обеспечению и системному проектированию; профессиональной разработки систем; повышения качества разрабатываемых в системном проекте продуктов; существенного сокращения рисков системных проектов; использования моделей жизненных циклов, разделённых на стадии.

### Процессы жизненного цикла систем

В соответствии со стандартом *ISO/IEC 15288* жизненный цикл систем имеет трехуровневую иерархическую структуру (рис. 8.1.8).

Основу жизненного цикла составляет набор **процессов**, каждому из которых соответствует общая **цель** его выполнения. Итогом процесса являются **результаты**, связанные с успешным достижением его цели. Каждый процесс с целью его структуризации разделен на набор **работ**, необходимых для получения результатов процесса. *Общее число* процессов в ЖЦ системы равно 25, результатов – 123, работ – 208.

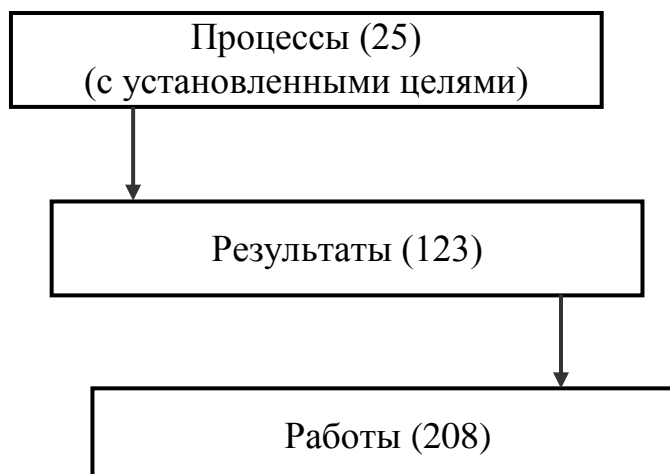


Рис.8.1.8 Иерархическая структура жизненного цикла систем в соответствии со стандартом *ISO/IEC 15288*

Представление процессов жизненного цикла посредством их результатов позволяет согласовать положения стандарта *ISO/IEC 15288* с положениями серии стандартов *ISO/IEC 15504–1–9*, широко используемых в настоящее время для оценки качества процессов ЖЦ. Структура жизненного цикла стандарта *ISO/IEC 15288* может быть использована в качестве эталонной модели ЖЦ систем при оценке процессов жизненного цикла конкретных проектов, выполняемой в соответствии с рекомендациями стандартов серии *ISO/IEC 15504*.

В соответствии со стандартом *ISO/IEC 15288* процессы жизненного цикла систем делятся на **четыре группы**:

- процессы согласования (The Agreement Processes);
- процессы предприятия (The Enterprise Processes);
- процессы проекта (The Project Processes);
- технические процессы (The Technical Processes).

На рис. 8.1.9 изображены процессы жизненного цикла системы с учетом их распределения по группам.



Рис. 8.1.9 Процессы жизненного цикла системы по стандарту ISO/IEC 15288:2002

## 8.2 Стандартизация оценки качества программного обеспечения.

### 8.2.1 Основные понятия в области оценки качества программного обеспечения

Особое внимание уделяется управлению качеством ПС на протяжении всего их ЖЦ.

На процесс разработки и деятельность по оценке качества ПС оказывают влияние следующие *обобщенные показатели ПС*:

- 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.

Существует ряд международных, межгосударственных и национальных стандартов, посвященных вопросам управления качеством ПС, включая планирование, обеспечение и контроль качества ПС. Основным компонентом контроля качества ПС и основой для его обеспечения является оценка качества.

Рассмотрим *основные термины*, соответствующие определениям *ГОСТ 28806–90, СТБ ИСО/МЭК 9126–2003, ISO/IEC 9126–1–4:2001–2004, ISO/IEC 14598–1:1999*.

*Атрибут (attribute)*: измеримое физическое или абстрактное свойство продукта. Атрибуты могут быть внешними или внутренними.

*Внешнее качество (external quality)*: степень, в которой продукт удовлетворяет установленные и подразумеваемые потребности при использовании в заданных условиях.

*Внешняя мера (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.

*Внутреннее качество (internal quality)*: полный набор атрибутов продукта, определяющих его способность удовлетворять установленные и подразумеваемые потребности при использовании в

заданных условиях.

*Внутренняя мера (internal measure)*: собственная мера продукта, прямая или косвенная.

*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 ПП, относящаяся к его способности удовлетворять установленные и подразумеваемые потребности.

*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 используется данное ПС.

*Критерий оценки качества (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 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 установлены, но являются действительными потребностями при использовании продукта в конкретных условиях.

*Подхарактеристика качества ПС (software quality subcharacteristic)*: это характеристика качества программного средства, входящая в состав другой характеристики качества.

*Показатель качества ПС (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. Промежуточный продукт может также быть конечным продуктом.

*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 ПС (software quality characteristic)*: набор свойств программного средства, с помощью которых описывается и оценивается его качество.

## **8.2.2 Стандартизация качества программного обеспечения в Республике Беларусь**

В настоящее время в области оценки качества ПС на территории Республики Беларусь действуют следующие основные стандарты:

- стандарт СССР *ГОСТ 28806–90. Качество программных*

*средств. Термины и определения;*

- межгосударственный стандарт стран СНГ **ГОСТ 28195–99. Оценка качества программных средств. Общие положения;**

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

Стандарт **СТБ ИСО/МЭК 9126–2003** представляет собой аутентичный перевод международного стандарта **ISO/IEC 9126:1991**.

В стандарте **ГОСТ 28806–90** даются основные термины и определения, принятые в области обеспечения качества программного обеспечения.

**ГОСТ 28195–99** определяет **оценку качества** программного средства как совокупность операций, включающих выбор номенклатуры характеристик качества оцениваемого программного средства, определение значений этих характеристик и сравнение их с базовыми значениями. В соответствии с данным стандартом оценка качества должна проводиться применительно ко всем работам ЖЦ ПС при планировании характеристик качества ПС, контроле качества 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 характеристик и 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.

Под **показателем качества продукции** в **ГОСТ 28195–99** подразумевается количественная характеристика одного или нескольких свойств продукции, составляющих ее качество, рассматриваемая применительно к определенным условиям ее создания и эксплуатации или потребления.

Обеспечение и оценка качества ПС выполняются в течение жизненного цикла программных средств и неразрывно связаны с положениями стандарта **ИСО/МЭК 12207**.

## **Классификация методов определения показателей качества программного средства**

Стандарт *ГОСТ 28195–99* классифицирует методы определения показателей качества ПС следующим образом:

- *по способам получения информации о показателе качества:*
  - измерительный;
  - регистрационный;
  - органолептический;
  - расчетный;
- *по источникам получения информации о показателе качества:*
  - экспертный;
  - социологический;
  - 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).

**Органолептический метод** – это метод получения информации о свойствах и характеристиках ПС, основанный на восприятии органов чувств (зрения и слуха) человека. 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.

**Социологический метод** – это метод получения информации о свойствах и характеристиках ПС на основе обработки специальных

анкет-опросников. Так могут определяться, например, отдельные показатели удобства использования.

**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.

### **Иерархическая модель оценки качества программного средства**

Стандарты *ГОСТ 28806–90*, *ГОСТ 28195–99*, *СТБ ИСО/МЭК 9126–2003* регламентируют выполнение оценки качества ПС и систем 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* определяют первые два уровня иерархической модели качества. При этом номенклатура характеристик первого уровня является **обязательной**, а номенклатура характеристик второго уровня (подхарактеристик) – **рекомендуемой**.

Стандарт *ГОСТ 28195–99* определяет **четырёхуровневую** иерархическую модель оценки качества ПС. Номенклатура характеристик и подхарактеристик первых двух уровней является **обязательной**, а номенклатура подхарактеристик третьего и четвертого уровней – **рекомендуемой**.

Вышеназванные стандарты определяют **шесть основных характеристик качества** ПС, находящихся на верхнем уровне модели качества. Следует отметить, что характеристики верхнего уровня, регламентированные *ГОСТ 28806–90* и *СТБ ИСО/МЭК 9126–2003*, соответствуют принятым в настоящее время в мировой практике. В то же время характеристики и подхарактеристики, определенные в *ГОСТ 28195–99*, частично не соответствуют иерархической модели качества, принятой в международных стандартах.

В стандартах *ГОСТ 28806–90* и *СТБ ИСО/МЭК 9126–2003* определены следующие **основные характеристики качества** ПС (характеристики качества верхнего уровня):

1. **Функциональность (Functionality)** – совокупность свойств ПС, определяемая наличием и конкретными особенностями набора

функций, способных удовлетворять заданные или подразумеваемые потребности.

**2. Надежность (Reliability)** – совокупность свойств, характеризующая способность ПС сохранять заданный уровень пригодности в заданных условиях в течение заданного интервала времени.

**3. Удобство использования (практичность, Usability)** – совокупность свойств программного средства, характеризующая усилия, необходимые для его использования, и индивидуальную оценку результатов его использования заданным или подразумеваемым кругом пользователей.

**4. Эффективность (Efficiency)** – совокупность свойств программного средства, характеризующая те аспекты его уровня пригодности, которые связаны с характером и временем использования ресурсов, необходимых при заданных условиях функционирования.

**5. Сопровождаемость (Maintainability)** – совокупность свойств программного средства, характеризующая усилия, которые необходимы для его модификации.

**6. Мобильность (Portability)** – совокупность свойств программного средства, характеризующая приспособленность для переноса из одной среды функционирования в другие.

## **Метод оценки качества программных средств по ГОСТ 28195–99**

Стандартом *ГОСТ 28195–99* рекомендован метод интегральной оценки качества программных средств, основанный на иерархической модели качества.

Основу описываемого метода оценки качества составляет *четырёхуровневая иерархическая модель качества*. *ГОСТ 28195–99* предлагает следующую **терминологию** для показателей качества каждого уровня:

*уровень 1* - факторы качества (в терминологии, принятой в международных стандартах, соответствуют характеристикам качества);

*уровень 2* - критерии качества (в международной терминологии – подхарактеристики качества);

*уровень 3* - метрики (соответствует международной терминологии);

*уровень 4* - оценочные элементы или единичные показатели (данный уровень в международных стандартах отсутствует).

Факторы и критерии качества, определенные в стандарте *ГОСТ 28195–99*, приведены на рис. 8.2.1.

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.



Рис. 8.2.1 Факторы и критерии качества ПС по *ГОСТ 28195–99*

В качестве примера на рис. 8.2.2 модель фактора *Надежность*.

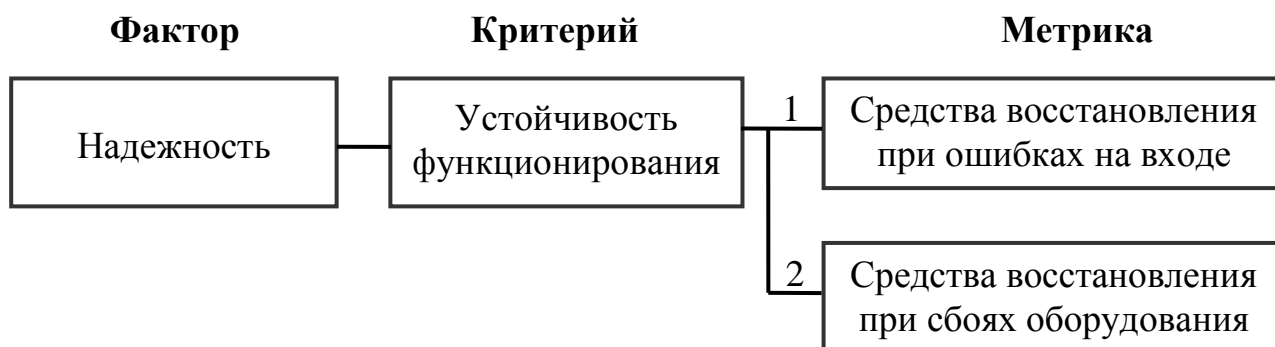


Рис. 8.2.2 Модель надежности для фазы анализа

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.

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

*СТБ ИСО/МЭК 9126–2003* определяет метод оценки качества ПС, основанный на *трехуровневой иерархической модели качества*. На первом уровне модели находятся шесть характеристик качества. Второй уровень составляют подхарактеристики и третий – метрики качества.

Модель процесса оценки, положенная в основу рассматриваемого метода, приведена на рис 8.2.3. Данная модель отражает основные стадии и этапы, требуемые для оценки качества ПС.



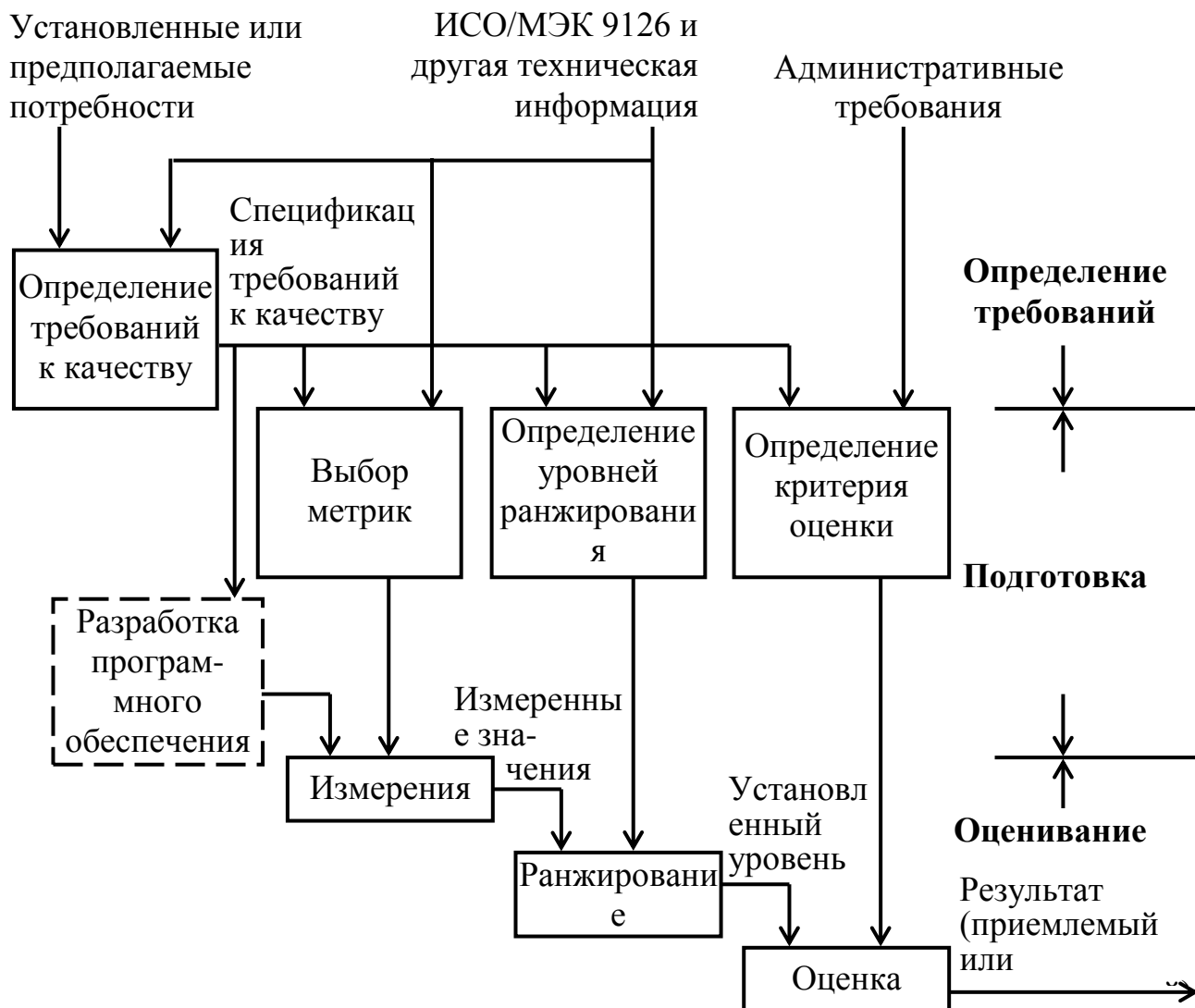


Рис. 8.2.3. Модель процесса оценки по СТБ ИСО/МЭК 9126–2003

Процесс оценки состоит из *трех стадий*: определение требований к качеству ПС, 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 ПС и должны быть определены до начала разработки. Так как ПС разделяется на компоненты, то требования для ПС 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 ПС определяется уровнем входящих в них подхарактеристик, а значения подхарактеристик в свою очередь определяются значениями входящих в них метрик.

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 ПС. Каждый количественный признак и каждое количественно оцениваемое взаимодействие ПС 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, потому что метрики пользователя являются решающими.

#### *Этап 2.2. Определение уровней ранжирования*

Для измерения количественных признаков ПС используются метрики качества. Измеренные значения отображаются на некоторой шкале. Данные значения не показывают уровень удовлетворения требований к качеству ПС. Для этой цели шкалы метрик должны быть разделены на диапазоны, соответствующие различным степеням удовлетворения требований (рис.8.2.4).

#### *Этап 2.3. Определение критерия оценки*

Для определения общего качества ПС должна быть учтена вся совокупность результатов оценивания различных метрик. Оценщик должен подготовить для этого процедуры, используя, например, таблицы решений или средние взвешенные значения. Обычно при этом учитываются и другие аспекты, такие как время и стоимость, которые являются косвенными факторами качества ПС.

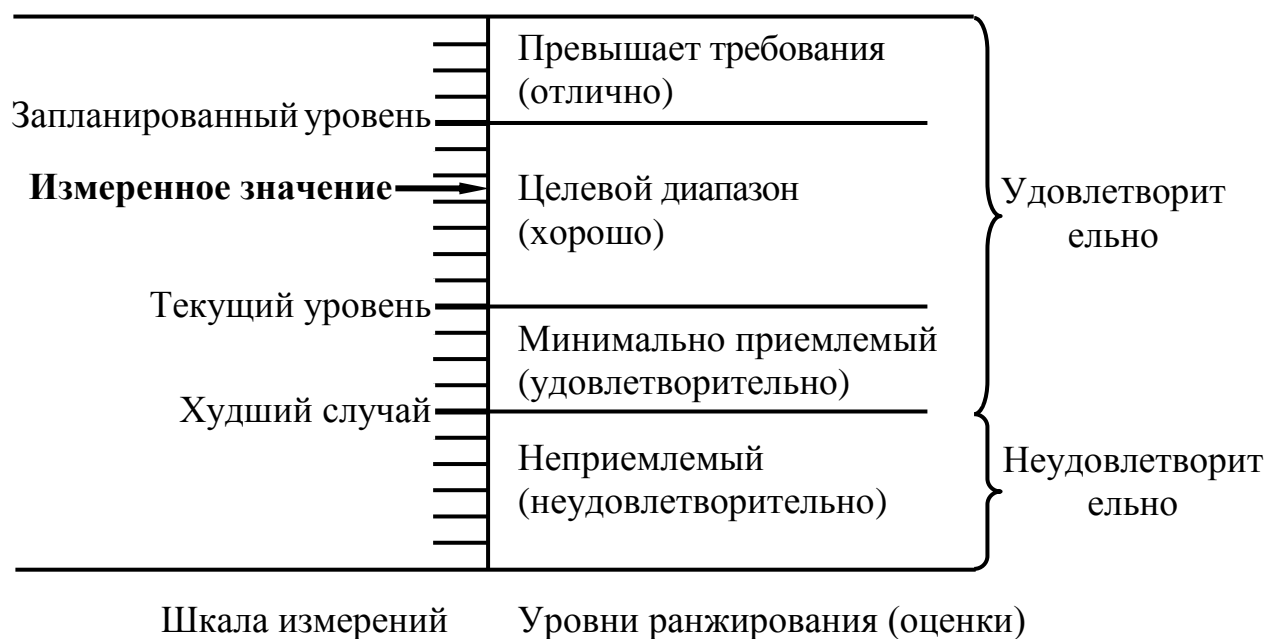


Рис.8.2.4 Варианты ранжирования измеренных значений метрик по ISO/IEC 14598–1:1999

### ***Стадия 3. Процедура оценивания***

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

#### ***Этап 3.1. Измерение***

Для измерения выбранные метрики применяются к ПС. Результатом являются значения в масштабах метрик.

#### ***Этап 3.2. Ранжирование***

На этапе ранжирования устанавливается уровень ранжирования для измеренного значения (см. рис. 8.2.4).

#### ***Этап 3.3. Оценка***

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

К *недостаткам* данного метода оценки качества следует отнести отсутствие рекомендуемых вариантов метрик и представление метода лишь в общем виде (в виде модели). Это затрудняет его конкретное использование.

*ГОСТ 28195–99* определяет **оценку качества** программного средства как совокупность операций, включающих выбор номенклатуры характеристик качества оцениваемого программного средства, определение значений этих характеристик и сравнение их с базовыми значениями. В соответствии с данным стандартом оценка качества должна проводиться применительно ко всем работам ЖЦ ПС

при планировании характеристик качества ПС, контроле качества в процессе разработки, проверке эффективности модификации ПС в процессе сопровождения.

### 8.2.3 Стандартизация качества программных средств за рубежом

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*. В Беларуси применяется стандарт *СТБ ИСО/МЭК 9126–2003*, являющийся аутентичным переводом вышеназванного стандарта.

В настоящее время стандарт *ISO/IEC 9126:1991* заменен на две взаимосвязанные серии стандартов: *ISO/IEC 9126–1–4* и *ISO/IEC 14598–1–6*.

Стандарт *ISO/IEC 9126–1–4* регламентирует иерархическую модель качества программных средств. На верхнем уровне модели находятся характеристики. Характеристики разделяются на подхарактеристики. Подхарактеристики определяются метриками. Метрики измеряют атрибуты (свойства) ПС.

Данный стандарт состоит из четырех частей под общим названием *Программная инженерия – Качество продукта*.

В стандарте регламентированы следующие *виды метрик*:

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

**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.

**Внутренние метрики** – это метрики, измеряющие собственные свойства ПС. Они измеряются в процессе разработки ПС 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.

**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. Очевидно, что 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.

Стандарт *ISO/IEC 14598-1-6* определяет процессы оценки качества программного продукта, содержит руководство и требования к оценке. Стандарт может применяться при разработке, приобретении и независимой оценке программного средства.

В течение жизненного цикла программного средства его качество изменяется. Требуемое качество, определенное в начале ЖЦ, отличается от фактического качества поставленного продукта. 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 стандарт *ISO/IEC 9126-1:2001* определяет следующие **виды качества программных средств**:

- *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 ПП, удовлетворяющего потребностям пользователя, обычно требует итеративного подхода к разработке программного средства с постоянной обратной связью с потенциальным пользователем;

- *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;

- *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 ПП; требования к внутреннему качеству могут использоваться как цель при аттестации продукта на различных стадиях разработки; они могут использоваться для определения стратегий разработки и критериев оценки и верификации в течение разработки; требования к внутреннему качеству должны определяться количественно, используя внутренние метрики;

- *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 ПП могут улучшаться при реализации кода, проверке или тестировании, но фундаментальная основа качества программного продукта, представленная внутренним качеством, остается неизменной до повторного проектирования;

- *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;

- *внешнее качество* – совокупность характеристик программного продукта с внешней точки зрения; это качество, измеряемое и оцениваемое на основе внешних метрик при выполнении ПП во время тестирования (испытаний) в моделируемой среде с моделируемыми данными или во время эксплуатации;

- *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;

- *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 ПП; пользователь оценивает только те атрибуты программного продукта, которые он применяет в своих задачах.

### **Метрики качества программных средств**

Стандарт *ISO/IEC 9126–1:2001* классифицирует метрики качества ПС на внутренние, внешние и метрики качества в использовании.

В *Приложении А* данных частей стандарта определены следующие *желательные свойства метрик*:

1) *надежность*; надежность связана со случайной ошибкой; метрика свободна от случайной ошибки, если случайные изменения не влияют на результаты метрики;

2) *повторяемость*; 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) *однотипность*; 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) *применимость*; метрика должна четко указывать условия (например, наличие определенных атрибутов), которые ограничивают её употребление;

5) *показательность*; это способность метрики идентифицировать части или элементы программы, которые должны быть улучшены, на основании сравнения измеренных и ожидаемых результатов;

6) *корректность*; метрика должна обладать следующими свойствами:

- *объективность*; результаты метрики и её 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*; измерение не должно быть направлено на получение какого-либо специфического результата;

- *адекватность точности*; точность определяется при проектировании метрики и особенно при выборе описаний фактов, используемых как основа для метрики; разработчик метрики должен описать точность и чувствительность метрики;

7) *значимость*; 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*; изменение в значениях характеристик качества (оперативно определенных по результатам измерения основных метрик), обусловленное изменением в значениях метрики, должно определяться линейной зависимостью;

2) *трассировка*; если метрика М непосредственно связана с величиной характеристики качества Q (оперативно определенной по результатам измерения основных метрик), то изменение величины Q (Т1), имеющейся в момент времени Т1, к величине Q (Т2), полученной в момент времени Т2, должно сопровождаться изменением значения метрики от М (Т1) до М (Т2) в том же направлении (например, если увеличивается Q, то М тоже увеличивается);

3) *непротиворечивость*; если значения характеристик качества (оперативно полученные по результатам измерения основных метрик) Q1, Q2,..., Qn, связанные с продуктами или процессами 1, 2..., n, определяются соотношением Q1> Q2> ... > Qn, то соответствующие значения метрики должны удовлетворять соотношению M1> M2> ... > Mn.

4) *предсказуемость*; если метрика используется в момент времени Т1 для прогноза значения (оперативно полученного по результатам измерения основных метрик) характеристики качества Q в момент времени Т2, то ошибка прогнозирования, определяемая выражением

$$\frac{\text{прогнозное } Q(T2) - \text{фактическое } Q(T2)}{\text{фактическое } Q(T2)},$$

должна попадать в допустимый диапазон ошибок прогнозирования;

5) *селективность*; метрика должна быть способной различать высокое и низкое качество программного средства.

### **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 ПП заявленным потребностям в надежности.



***Внутренние метрики практичности*** используются во время разработки программного продукта для предсказания степени, в которой ПП 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 ПП 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.

### **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.

***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.

## **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.

## **Метод оценки качества программных средств по ISO/IEC 14598–1:1999**

Стандарт *ISO/IEC 14598–1:1999* регламентирует метод оценки качества программных средств, который основан на иерархической модели качества, определенной в *ISO/IEC 9126–1:2001*.

На рис. 8.2.5 приведена схема процесса оценки качества ПС. 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 продуктов разработки (сопровождения).

### **Стадия 1. Установка требований к оценке**

Данная стадия состоит из трех этапов.

#### **Этап 1.1. Установка цели оценки**

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



Рис. 8.2.5 Процесс оценки качества по ISO/IEC 14598–1:1999

*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 *ISO/IEC 12207*.

*В процессе заказа ПП* заказчик должен установить требования к внешнему качеству ПП, определить требования к поставщику и оценить качество разработанного ПП по этим требованиям перед его приемкой. Оценка качества ПС заказчиком детально рассмотрена в стандарте *ISO/IEC 14598–4:1999*.

При покупке готового программного продукта оценка может использоваться для сравнения альтернативных ПП и гарантии, что выбранный ПП удовлетворяет требованиям по качеству. Оценка качества программного средства оценщиком детально рассмотрена в стандарте *ISO/IEC 14598–5:1998*.

*В процессе поставки* поставщик может использовать результаты оценки ПП, чтобы гарантировать, что ПП удовлетворяет требуемым критериям качества как в сравнении с другими ПП, так и установленным заказчиком.

*В процессе разработки* на основании требований к внешнему качеству ПП, установленных заказчиком, разработчик для каждого компонента ПП должен определить спецификацию требований к внешнему качеству в терминах характеристик и подхарактеристик.

На базе спецификации требований к внешнему качеству разработчик должен определить требования к внутреннему качеству ПП. Эти требования должны использоваться на начальных этапах процесса разработки (когда еще нет исполнимых кодов ПП или его компонентов) для проверки качества промежуточных продуктов с целью прогноза и предварительной оценки качества конечного ПП. На последующих этапах процесса разработки (когда уже имеются исполнимые коды модулей, компонентов или всего ПП) должны использоваться требования к внешнему качеству. На их основе должна выполняться оценка внешнего качества промежуточных продуктов и конечного ПП (с целью прогноза итогового внешнего качества в среде эксплуатации). С этой целью обычно используется моделируемая среда с моделируемыми данными. Оценка качества ПС разработчиком рассмотрена в стандарте *ISO/IEC 14598–3:2000*.

*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 ЖЦ ПС. Например, в процессе разработки интерес представляет оценка промежуточного продукта. На ранних этапах процесса разработки это может быть спецификация требований, архитектура ПП, технический проект ПП, 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 ПП 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 ПС в окружающей среде с целью подтверждения соответствия требованиям к внешнему качеству и прогноза качества ПП 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 общая модель качества из стандарта *ISO/IEC 9126-1:2001*. 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 экономность их использования. Примеры метрик качества приведены в стандартах *ISO/IEC TR 9126-2, -3, -4*.

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 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 (см. рис.).

#### *Этап 2.3. Установка критериев для оценки*

Чтобы оценить качество продукта, необходимо некоторым образом объединить результаты оценки его различных характеристик. С этой целью должна быть разработана процедура, включающая отдельные критерии для различных характеристик качества. Каждая из характеристик может быть определена в терминах единственной подхарактеристики или средневзвешенной комбинации подхарактеристик. Процедура обычно включает и другие аспекты, такие как время и стоимость, которые вносят вклад в оценку качества программного продукта в конкретной среде окружения.

### ***Стадия 3. Проектирование оценки***

Данная стадия состоит из одного этапа.

#### *Этап 3.1. Разработка 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*. План оценки должен быть согласован с планом количественной оценки качества, регламентированным в *ISO/IEC 14598–2:2000*.

### ***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.

#### *Этап 4.3. Оценка результатов*

Оценка результатов – заключительный этап процесса оценки ПС. 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 сравнивается с другими аспектами, такими как время и стоимость.

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

### **8.3 Сертификация программных средств**

Сертификация в Республике Беларусь основана на *Законах № 262-З «О техническом нормировании и стандартизации» и № 269-З «Об оценке соответствия требованиям технических нормативных правовых актов в области технического нормирования и стандартизации»*, принятых 5 января 2004 г.

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 сертификация и добровольная сертификация. 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 программного средства 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:

- 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 *ISO/IEC 14598–1:1999* и *ISO/IEC 14598–5:1998*.